Title
Iterative hard thresholding in genome-wide association studies: Generalized linear models, prior weights, and double sparsity.

Permalink
https://escholarship.org/uc/item/1709349q

Journal
GigaScience, 9(6)

ISSN
2047-217X

Authors
Chu, Benjamin B
Keys, Kevin L
German, Christopher A
et al.

Publication Date
2020-06-01

DOI
10.1093/gigascience/giaa044

Peer reviewed
TECHNICAL NOTE

Iterative hard thresholding in genome-wide association studies: Generalized linear models, prior weights, and double sparsity

Benjamin B. Chu, Kevin L. Keys, Christopher A. German, Hua Zhou, Jin J. Zhou, Eric M. Sobel, Janet S. Sinsheimer, and Kenneth Lange

Abstract

Background: Consecutive testing of single nucleotide polymorphisms (SNPs) is usually employed to identify genetic variants associated with complex traits. Ideally one should model all covariates in unison, but most existing analysis methods for genome-wide association studies (GWAS) perform only univariate regression. Results: We extend and efficiently implement iterative hard thresholding (IHT) for multiple regression, treating all SNPs simultaneously. Our extensions accommodate generalized linear models, prior information on genetic variants, and grouping of variants. In our simulations, IHT recovers up to 30% more true predictors than SNP-by-SNP association testing and exhibits a 2–3 orders of magnitude decrease in false-positive rates compared with lasso regression. We also test IHT on the UK Biobank hypertension phenotypes and the Northern Finland Birth Cohort of 1966 cardiovascular phenotypes. We find that IHT scales to the large datasets of contemporary human genetics and recovers the plausible genetic variants identified by previous studies. Conclusions: Our real data analysis and simulation studies suggest that IHT can (i) recover highly correlated predictors, (ii) avoid over-fitting, (iii) deliver better true-positive and false-positive rates than either marginal testing or lasso regression, (iv) recover unbiased regression coefficients, (v) exploit prior information and group-sparsity, and (vi) be used with biobank-sized datasets. Although these advances are studied for genome-wide association studies inference, our extensions are pertinent to other regression problems with large numbers of predictors.

Keywords: GWAS; multiple regression; high dimensional inference; iterative hard thresholding; biobank

Received: 19 November 2019; Revised: 27 February 2020; Accepted: 14 April 2020
Iterative hard thresholding in genome-wide association studies selects non-zero parameters by minimizing the criterion of parsimony in multiple regression. The lasso [7, 8] is most investigated to model interactions. Of course, these advantages remain within the scope of marginal analysis.

Despite their numerous successes [2], marginal regression is less than ideal for GWAS. It implicitly assumes that all SNPs have independent effects. In contrast, multiple regression can capture the biology behind GWAS more realistically because traits are usually determined by multiple SNPs acting in unison. Marginal regression selects associated SNPs 1 by 1 on the basis of the P-value. With minor adjustments, the coordinate descent algorithm for the lasso carries over to MCP penalized regression [12, 13]. Model selection is achieved without severe shrinkage, and inference in GWAS improves [14]. However, in our experience its false-negative rate is considerably higher than iterative hard thresholding (IHT)’s rate [15]. A second remedy for the lasso, stability selection, weeds out false-positive results by looking for consistent predictor selection across random halves of the data [16]. However, it is known to be under-powered for GWAS compared to standard univariate selection [17].

In contrast, IHT minimizes a loss \( \ell(\beta) \) subject to the nonconvex sparsity constraint \( \| \beta \|_0 \leq k \), where \( \| \beta \|_0 \) counts the number of non-zero components of \( \beta \) [18–20]. Fig. 1 explains graphically how the \( \ell_0 \) penalty of IHT reduces the bias of the selected parameters compared to \( \ell_1 \) and MCP penalties. In the figure \( \lambda, \gamma, k \) are chosen so that the same range of \( \beta \) values are sent to zero. To its detriment, the lasso penalty shrinks all \( \beta \)'s, no matter how large their absolute values. The nonconvex MCP penalty avoids shrinkage for large \( \beta \)'s but exerts shrinkage for intermediate \( \beta \)’s. IHT, which is both nonconvex and discontinuous, avoids shrinkage altogether. For GWAS, the sparsity model-size constant \( k \) also has a simpler and more intuitive interpretation than the lasso tuning constant \( \lambda \). Finally, both false-positive and false-negative rates are well controlled. Balanced against these advantages is the loss of convexity in optimization and concomitant loss of computational efficiency. In practice, the computational barriers are surmountable and are compensated by the excellent results delivered by IHT in high-dimensional regression problems such as multiple GWAS regression.

This article has 4 interrelated goals. First, we extend IHT to generalized linear models (GLM). These models encompass most of applied statistics. Previous IHT algorithms focused on normal or logistic sparse regression scenarios. Our software can also perform sparse regression under Poisson and negative binomial response distributions and can be easily extended to other GLM distributions as needed. The key to our extension is the derivation of a nearly optimal step size \( s \) for improving the log-likelihood at each iteration. Second, we introduce doubly sparse regression to IHT. Previous authors have considered group sparsity [21]. The latter tactic limits the number of groups selected.

\[ f(\beta) = \ell(\beta) + \lambda \| \beta \|_1. \]

where \( \ell(\beta) \) is a convex loss, \( \lambda \) is a sparsity tuning constant, and \( \| \beta \|_1 = \sum \| \beta_j \|_1 \) is the \( \ell_1 \) norm of the parameters. The lasso has the virtues of preserving convexity and driving most parameter estimates to 0. Minimization can be conducted efficiently via cyclic coordinate descent [9, 10]. The magnitude of the nonzero tuning constant \( \lambda \) determines the number of predictors selected.
It is also useful to limit the number of predictors selected per group. Double sparsity strikes a compromise that encourages selection of correlated causative variants in linkage disequilibrium (LD). Notably, this technique generalizes group-IHT. Third, we demonstrate how to incorporate predetermined SNP weights in IHT. Our simple and interpretable weighting option allows users to introduce prior knowledge into sparse projection. Thus, one can favor predictors whose association to the response is supported by external evidence. Fourth, we present MendelIHT.jl: a scalable, open source, and user-friendly software for IHT in the high-performance programming language Julia [22].

**Model Development**

This section sketches our extensions of IHT.

**IHT background**

IHT was originally formulated for sparse signal reconstruction, which is framed as sparse linear least-squares regression. In classical linear regression, we are given an \( n \times p \) design matrix \( X \) and a corresponding \( n \)-component response vector \( y \). We then postulate that \( y \) has mean \( E(y) = X\beta \) and that the residual vector \( y - X\beta \) has independent Gaussian components with a common variance. The parameter (regression coefficient) vector \( \beta \) is estimated by minimizing the sum of squares \( f(\beta) = (1/2)\|y - X\beta\|^2_2 \). The solution to this problem is known as the ordinary least-squares estimator and can be written explicitly as \( \hat{\beta} = (XX^{-1})Xy \), provided the problem is overdetermined \( (n > p) \). This paradigm breaks down in the high-dimensional regime \( n \ll p \), where the parameter vector \( \beta \) is underdetermined.

In the spirit of parsimony, IHT seeks a sparse version of \( \beta \) that gives a good fit to the data. This is accomplished by minimizing \( f(\beta) \) subject to \( \|\beta\|_0 < k \) for a small value of \( k \), where \( \| \cdot \|_0 \) counts the number of nonzero entries of a vector. The optimization problem is formally:

\[
\min \; \frac{1}{2}\| y - X\beta \|^2_2 \quad \text{subject to} \quad \|\beta\|_0 \leq k. \tag{1}
\]

IHT abandons the explicit formula for \( \hat{\beta} \) because it fails to respect sparsity and involves the numerically intractable matrix inverse \((XX^{-1})^{-1}\).

IHT combines 3 core ideas. The first is steepest descent. Elementary calculus tells us that the negative gradient \(-\nabla f(\beta)\) is the direction of steepest descent of \( f(\beta) \) at \( \beta \). First-order optimization methods like IHT define the next iterate in minimization by the formula \( \beta_{n+1} = \beta_n + s_n v_n \), where \( v_n = -\nabla f(\beta_n) \) and \( s_n > 0 \) is some optimally chosen step size. In the case of linear regression \(-\nabla f(\beta) = X(y - X\beta)\). To reduce the error at each iteration, the optimal step size \( s_n \) can be selected by minimizing the second-order Taylor expansion

\[
f(\beta_n + s_n v_n) = f(\beta_n) + s_n \nabla f(\beta_n)^t v_n + \frac{s_n^2}{2} \nabla^2 f(\beta_n) v_n\]

\[
= f(\beta_n) - s_n \nabla f(\beta_n)^t v_n + \frac{s_n^2}{2} \nabla^2 f(\beta_n) v_n \]

with respect to \( s_n \). Here \( \nabla^2 f(\beta) = XX^t \) is the Hessian matrix of second partial derivatives. Because \( f(\beta) \) is quadratic, the expansion is exact. Its minimum occurs at the step size

\[
s_n = \frac{\nabla f(\beta_n)^t v_n}{\nabla^2 f(\beta_n) v_n}. \tag{2}
\]

This formula summarizes the second core idea.

The third component of IHT involves projecting the steepest descent update \( \beta_n + s_n v_n \) onto the sparsity set \( \beta = \{ \beta : \| \beta \|_0 \leq k \} \). The relevant projection operator \( P_k(\beta) \) sets all but the \( k \) largest entries of \( \beta \) in magnitude to 0. In summary, IHT solves problem (1) by updating the parameter vector \( \beta \) according to the recipe:

\[
\beta_{n+1} = P_k(\beta_n + s_n \nabla f(\beta_n))
\]

with the step size given by formula (2). An optional debiasing step can be added to improve parameter estimates. This involves replacing \( \beta_{n+1} \) by the exact minimum point \( f(\beta) \) in the subspace defined by the support \( \{ j : \beta_{n+1,j} \neq 0 \} \) of \( \beta_{n+1} \). Debiasing is efficient because it solves a low-dimensional problem. Several versions of hard-thresholding algorithms have been proposed in the signal-processing literature. The first of these, NHT [20], omits debiasing. The rest, e.g., GraHTP [23], median penalization [24], and CoSaMp [25], offer debiasing.

IHT for generalized linear models

A GLM involves responses \( y \) following a natural exponential distribution with density in the canonical form

\[
f(y \mid \theta, \phi) = \exp \left[ \frac{y \theta - b(\phi)}{a(\phi)} + c(y, \phi) \right]
\]

where \( y \) is the data, \( \theta \) is the natural parameter, \( \phi > 0 \) is the scale (dispersion), and \( a(\phi), b(\phi), c(\phi, y) \) are known functions that vary depending on the distribution [26, 27]. Simple calculations show that \( y \) has mean \( \mu = b(\theta) \) and variance \( \sigma^2 = b''(\theta)a(\phi) \); accordingly, \( \sigma^2 \) is a function of \( \mu \). Table 1 summarizes the mean domains and variances of a few common exponential families.

| Family          | Mean domain | \( \text{var}(y) \) | \( g(\phi) \) |
|-----------------|-------------|---------------------|-------------|
| Normal          | \( \mathbb{R} \) | \( \phi^2 \)         | 1           |
| Poisson         | \([0, \infty)\) | \( \mu \)            | \( e^{\phi} \) |
| Bernoulli       | \([0, 1]\) | \( \mu(1 - \mu) \) | \( e^{\phi}/(1 + e^{\phi}) \) |
| Gamma           | \([0, \infty)\) | \( \mu^2 \phi \) | \( s^{-1} \) |
| Inverse Gaussian| \([0, \infty)\) | \( \mu \phi \) | \( s^{-1/2} \) |
| Negative binomial| \([0, \infty)\) | \( \mu(\mu + 1) \) | \( e^{\phi} \) |

In GLM, \( \mu = g(\phi) \) denotes the mean, \( \sigma = \sqrt{g(\phi)} \) the linear responses, \( g \) is the inverse link function, and \( \phi \) the dispersion. Except for the negative binomial, all inverse links are canonical.

To put each predictor on an equal footing, each should be standardized to have mean 0 and variance 1. Including an additional intercept term is standard practice.

If we assemble a design matrix \( X \) by stacking the row vectors \( x_n \), then we can calculate the loglikelihood, score, and expected
Iterative hard thresholding in genome-wide association studies

The effectiveness of group sparsity in penalized regression has been demonstrated in general [30, 31] and for GWAS [32] in particular. Group IHT [21] enforces group sparsity but does not enforce within-group sparsity. In GWAS, model selection is desired within groups as well to pinpoint causal SNPs. Furthermore, a concern in GWAS is that two causative SNPs can be highly correlated with each other due to LD. When sensible group information is available, doubly sparse IHT encourages the detection of causative yet correlated SNPs while enforcing sparsity within groups. Here we discuss how to carry out a doubly sparse projection that enforces both within- and between-group sparsity.

Doubly sparse projections

The final algorithm combining doubly sparse projections, prior weight scaling, and debiasing is summarized in Algorithm 1.

Algorithm 1: Iterative hard thresholding

Input: Design matrix X, response vector y, membership vector g, weight vector \( w \), max number of groups \( J \), and overall sparsity projection \( P(\beta) \).

1. Initialize: \( \hat{\beta} = 0 \).
2. while not converged do
3. \( \beta = \beta \Rightarrow \)
4. Ascent direction with scaling: \( \hat{\beta} = \beta \odot (\beta_n + sv) \)
5. Project to sparsity: \( \hat{\beta} = P(\hat{\beta}) \odot w \) (where \( \odot \) is elementwise division)
6. while \( L(\hat{\beta}) \leq L(\beta_n) \), backtrack \( \leq 5 \) do
7. \( s = sf/2 \)
8. \( \beta_n + 1 \)\( \beta_n + 1 \)
9. end\( \beta_n + 1 \)
10. \( \beta \) = \( \beta_n + 1 \)\( \beta_n + 1 \)
11. end\( \beta_n + 1 \)
12. Output: \( \beta \) with \( J \) active groups and \( \lambda_g \) active predictors for group \( g \).

Results

Readers can reproduce our results by accessing the software, documentation, and Jupyter notebooks on our Github page: https://github.com/OpenMendel/MendelIHT.jl

Scalability of IHT

To test the scalability of our implementation, we ran IHT on \( p = 10^6 \) SNPs for sample sizes \( n = 10,000, 20,000, \ldots, 120,000 \) with 5 independent replicates per \( n \). All simulations rely on a true sparsity level of \( k = 10 \). Using a machine with 63 GB of RAM and a single 3.3-GHz Intel-E5-2670, Fig. 2 plots the IHT median CPU time per iteration, median iterations to convergence, and median memory usage under Gaussian, logistic, Poisson, and negative binomial distributions.
Figure 2: (a, d) Median Time per iteration scales linearly with data size. Speed is measured for compressed genotype files. On uncompressed data, all responses are roughly 10 times faster. (b, e) Median memory usage scales as $\sim 2^{np}$ bits. Note memory for each response are usages in addition to loading the genotype matrix. Uncompressed data require 32 times more memory. (c) Debiasing reduces median iterations until convergence for all but negative binomial (Neg Bin) regression. Benchmarks were carried out on compressed data with 10^6 SNPs and sample sizes ranging from 10,000 to 120,000. Hence, the largest matrix here requires 30 GB and can still fit into personal computer memories.

The formation of the vector $\mu$ of predicted values requires only a limited number of nonzero regression coefficients. Consequently, the computational complexity of this phase of IHT is relatively light. In contrast, calculation of the Fisher score (gradient) and information (expected negative Hessian) depends on the entire genotype matrix $X$. Fortunately, each of the $np$ entries of $X$ can be compressed to 2 bits. Fig. 2b and e show that IHT memory demands beyond storing $X$ never exceeded a few gigabytes. Fig. 2a and d show that IHT run time per iteration increases linearly in problem size $n$. Similarly, we expect increasing $p$ will increase run time linearly because the bottleneck of IHT is the matrix-vector multiplication step in computing the gradient, which scales as $O(np)$. Debiasing increases run time per iteration only slightly. Except for negative binomial responses, debiasing is effective in reducing the number of iterations required for convergence and hence overall run time.

Cross-validation in model selection

In actual studies, the true number of genetic predictors $k_{true}$ is unknown. This section investigates how $q$-fold cross-validation can determine the best model size on simulated data. Under normal, logistic, Poisson, and negative binomial models, we considered 50 different combinations of $X$, $y$, and $\beta_{true}$ with $k_{true} = 10$, $n = 5,000$ samples, and $p = 50,000$ SNPs fixed in all replicates. Here, $k_{true}$ is chosen so that it is closer to our NFBC and UK Biobank results. On these datasets we conducted 5-fold cross-validation across 20 model sizes $k$ ranging from 1 to 20. Fig. 3 plots deviance residuals on the holdout dataset for each of the 4 GLM responses (mean squared error in the case of normal responses) and the best estimate $\hat{k}$ of $k_{true}$.

Fig. 3 shows that $k_{true}$ can be effectively recovered by cross-validation. In general, prediction error starts off high where the proposed sparsity level $k$ severely underestimates $k_{true}$ and plateaus when $k_{true}$ is reached (Fig. 3a–d). Furthermore, the estimated sparsity $\hat{k}$ for each run is narrowly centered around $k_{true}$ (Fig. 3e and f). In fact, $|\hat{k} - k_{true}| \leq 4$ always holds. When $\hat{k}$ exceeds $k_{true}$, the estimated regression coefficients for the false predictors tend to be very small. In other words, IHT is robust to overfitting, in contrast to lasso penalized regression. We see qualitatively similar results when $k_{true}$ is large. This proved to be the case in our previous article [15] for Gaussian models with $k_{true} \in \{100, 200, 300\}$.

Comparing IHT to lasso and marginal tests in model selection

Comparison of the true-positive and false-positive rates of IHT and its main competitors is revealing. For lasso regression we use the glmnet implementation of cyclic coordinate descent [9, 10, 37] (v2.0-16 implemented in R 3.5.2); for marginal testing we use the beta version of MendelGWAS [38]. As explained later, Poisson regression is supplemented by zero-inflated Poisson regression implemented under the pscl [39] (v1.5.2) package of...
Iterative hard thresholding in genome-wide association studies

Figure 3: Five-fold cross-validation results are capable of identifying the true model size $k_{true}$. (a–d) Deviance residuals of the testing set are minimized when the estimated model size $\hat{k} = k_{true}$. Each line represents 1 simulation. (e–h) $\hat{k}$ is narrowly spread around $k_{true} = 10$.

Table 2: IHT achieves the best balance of false-positive and true-positive results compared to lasso and marginal (single-SNP) regression

| Test | Normal | Logistic | Poisson | Neg Bin |
|------|--------|----------|---------|---------|
| IHT  | TP 8.84 | 6.28      | 7.20    | 8.88    |
|      | FP 0.02 | 0.10      | 1.28    | 0.22    |
| Lasso| TP 9.52 | 8.16      | 9.28    | NA      |
|      | FP 31.26| 45.76     | 102.24  | NA      |
| Marginal| TP 7.18 | 5.76      | 9.04(5.94) | 5.98 |
|      | FP 0.06 | 0.02      | 1,527.90(0.00) | 0.00 |

Average values over 50 replicates. In each replicate there are $k = 10$ causal SNPs. Best model sizes for IHT and lasso were chosen by cross-validation. FP: false positive; TP: true positive; NA: not applicable because glmnet does not support negative binomial regression. Parentheses indicate zero-inflated Poisson regression.

R. Unfortunately, glmnet does not accommodate negative binomial regression. Because both glmnet and psc1 operate on floating point numbers, we limit our comparisons to small problems with 1,000 subjects, 10,000 SNPs, 50 replicates, and $k = 10$ causal SNPs. IHT performs model selection by 3-fold cross-validation across model sizes ranging from 1 to 50. This range is generous enough to cover the models selected by lasso regression. We adjust for multiple testing in the marginal case by applying a $P$-value cut-off of $5 \times 10^{-6}$.

Table 2 demonstrates that IHT achieves the best balance between maximizing true-positive results and minimizing false-positive results. IHT finds more true-positive results than marginal testing and almost as many as lasso IHT. IHT also finds far fewer false-positive results than lasso regression. Poisson regression is exceptional in yielding an excessive number of false-positive results in marginal testing. A similar but less extreme trend is observed for lasso regression. The marginal false-positive rate is reduced by switching to zero-inflated Poisson regression. This alternative model is capable of handling overdispersion due to an excess of 0 values. Interestingly, IHT rescues the Poisson model by accurately capturing the simultaneous impact of multiple predictors.

Reconstruction quality for GWAS data

Table 3 demonstrates that IHT estimates show little bias compared to estimates from lasso and marginal regressions. These trends hold with or without debiasing as described earlier. The proportion of variance explained is approximately the same in both scenarios. The displayed values are the average estimated $\beta$’s, computed among the SNPs actually found. As expected, lasso estimates show severe shrinkage compared to IHT. Estimates from marginal tests are severely overestimated because each SNP is asked to explain more trait variance than it should. As the magnitude of $\beta_{true}$ decreases, IHT estimates show an upward absolute bias, consistent with the winner’s curse phenomenon. When sample sizes are small, small effect sizes make most predictors imperceptible amid the random noise. The win-

| Test       | Normal | Logistic | Poisson | Neg Bin |
|------------|--------|----------|---------|---------|
| $\beta_{true}$ | $\beta_{IHT}$ | $\beta_{lasso}$ | $\beta_{logistic}$ | $\beta_{Poisson}$ | $\beta_{NegBin}$ |
| 0.5        | 0.501(0.015) | 0.508(0.039) | 0.492(0.039) | 0.567(0.670) |
| 0.25       | 0.249(0.013) | 0.256(0.038) | 0.247(0.012) | 0.249(0.012) |
| 0.10       | 0.097(0.014) | 0.125(0.016) | 0.100(0.014) | 0.010(0.012) |
| 0.05       | 0.063(0.007) | 0.108(0.006) | 0.057(0.008) | 0.060(0.008) |
| $\beta_{true}$ | $\beta_{marginal}$ | $\beta_{logistic}$ | $\beta_{Poisson}$ | $\beta_{NegBin}$ |
| 0.5        | 0.451(0.015) | 0.366(0.058) | 0.458(0.037) | NA |
| 0.25       | 0.199(0.013) | 0.137(0.032) | 0.208(0.015) | NA |
| 0.10       | 0.046(0.014) | 0.022(0.016) | 0.058(0.016) | NA |
| 0.05       | 0.012(0.008) | 0.008(0.003) | 0.012(0.009) | NA |

Displayed coefficients are average fitted values $\pm$1 standard error for the discovered predictors over 100 replicates. $\ast = $ zero true-positive results observed on average. NA = not available because glmnet does not support negative binomial lasso regression. There are $k = 10$ true SNPs.
ner’s curse operates in this regime and cannot be eliminated by IHT. Lasso’s strong shrinkage overshadows the bias of the winner’s curse and yields estimates smaller than true values.

The results displayed in Table 3 reflect \( n = 50,000 \) subjects, \( p = 100,000 \) SNPs, 100 replicates, and a sparsity level \( k \) fixed at its true value \( k_{true} = 10 \). The \( \lambda \) value for lasso is chosen by cross-validation. To avoid datasets with monomorphic SNPs, the minimum MAF is set at 0.05. For linear, logistic, and Poisson regressions in marginal tests, we first screen for potential SNPs via a score test. Only top SNPs are used in the more rigorous and computationally intensive likelihood ratio tests, which gives the \( \beta \) estimates. This procedure is described in Zhou et al. [38].

We ran likelihood ratio tests for all SNPs in the negative binomial model because the screening procedure is not yet implemented. However, the inflation in parameter estimates is present throughout all marginal tests.

### Correlated covariates and doubly sparse projections

Next we study how well IHT works on correlated data and whether doubly sparse projection can enhance model selection. Table 4 shows that, in the presence of extensive LD, IHT performs reasonably well even without grouping information. When grouping information is available, group IHT enhances model selection. The results displayed in Table 4 reflect \( n = 1,000 \) samples, \( p = 10,000 \) SNPs, and 100 replicates. Each SNP belongs to 1 of 50 disjoint groups containing 20 SNPs each; samples, \( \text{Neg Bin} 11.0 \pm 2.1 \), \( \text{Poisson} 8.0 \pm 1.9 \), \( \text{Logistic} 3.8 \pm 2.2 \). We ran 100 independent simulation studies under this setup, where the large, medium, small, and non-coding groups are each allowed 5, 3, 2, and 2 active SNPs. The results are displayed in Table 5. We find that even in this worst-case scenario where group information is completely lacking, grouped IHT does no worse than ungrouped IHT.

### Introduction of prior weights

This section considers how scaling by prior weights helps in model selection. Table 6 compares weighted IHT reconstructions with unweighted reconstructions where all weights \( w_i = 1 \). The weighted version of IHT consistently finds \( \sim 10\% \) more true predictors than the unweighted version. Here we simulated 50 replicates involving 1,000 subjects, 10,000 uncorrelated variants, and \( k = 1 \) true predictors for each GLM. For the sake of simplicity, we defined a prior weight \( w_i = 2 \) for 110% all variants, including the 10 true predictors. For the remaining SNPs the prior weight is \( w_i = 1 \). These choices reflect a scenario where 10% of all genotyped variants fall in protein-coding regions, including the 10 true predictors, and where such variants are twice as likely to influence a trait as those falling in non-coding regions.

### Hypertension GWAS in the UK Biobank

In this section we test IHT on the second release of UK Biobank [41] data. This dataset contains \( \sim 500,000 \) samples and \( \sim 800,000 \) SNPs without imputation. Phenotypes are systolic blood pressure (SBP) and diastolic blood pressure (DBP), averaged over 4 or fewer readings. To adjust for ancestry and relatedness, we included the following nongenetic covariates: sex, hospital center, age, age², BMI, and the top 10 principal components com-

---

**Table 4: Doubly sparse IHT enhances model selection on simulated data**

| Model      | Ungrouped IHT | Grouped IHT |
|------------|---------------|-------------|
| Normal     | 11.1 ± 1.9    | 3.9 ± 1.9   |
| Logistic   | 3.8 ± 1.6     | 11.2 ± 1.6  |
| Poisson    | 11.5 ± 2.2    | 3.5 ± 2.2   |
| Neg Bin    | 11.0 ± 2.1    | 4.0 ± 2.1   |

**Table 5: Doubly sparse IHT is comparable to regular IHT on NFBC dataset using arbitrary groups**

| Model      | Ungrouped IHT | Grouped IHT |
|------------|---------------|-------------|
| Normal     | 17.0 ± 1.2    | 2.0 ± 1.2   |
| Logistic   | 15.7 ± 1.5    | 3.3 ± 1.5   |
| Poisson    | 17.1 ± 1.3    | 1.9 ± 1.3   |
| Neg Bin    | 17.2 ± 1.5    | 1.8 ± 1.5   |

**Table 6: Weighted IHT enhances model selection**

| Model      | Unweighted IHT | Weighted IHT |
|------------|----------------|--------------|
| Normal     | 9.2 ± 0.4      | 9.4 ± 0.5    |
| Logistic   | 7.3 ± 0.6      | 8.0 ± 0.6    |
| Poisson    | 8.0 ± 0.6      | 8.3 ± 0.6    |
| Neg Bin    | 9.2 ± 0.5      | 9.4 ± 0.5    |
Table 7: UK Biobank GWAS results generated by running IHT on Stage 2 Hypertension (S2 Hyp) under a logistic model

| SNP ID          | Chromosome | Position (bp) | \( \beta \) | Known? |
|-----------------|------------|---------------|-------------|--------|
| rs17367504      | 1          | 11,862,778    | 0.046       | [45, 46]|
| rs757110        | 2          | 17,418,477    | 0.025       |        |
| rs1898841       | 2          | 165,070,207   | 0.022       | [46]   |
| rs1347264       | 2          | 164,999,883   | 0.020       | [46]   |
| rs16998073      | 4          | 81,184,341    | 0.048       | [45, 46]|
| rs1173771       | 4          | 32,815,028    | 0.046       | [46]   |
| rs13107325      | 6          | 103,188,709   | 0.030       | [45, 46]|
| rs72742749      | 5          | 32,834,974    | 0.029       |        |
| rs11241955      | 5          | 127,626,884   | 0.028       |        |
| rs2072495       | 5          | 158,296,996   | 0.027       |        |
| rs805293        | 6          | 31,688,518    | 0.029       | [46]   |
| rs2392929       | 7          | 106,414,069   | 0.039       | [45, 46]|
| rs73203495      | 8          | 11,580,334    | 0.031       |        |
| rs12258967      | 10         | 18,727,959    | 0.039       | [45, 46]|
| rs11191580      | 10         | 104,906,211   | 0.039       |        |
| rs2274224       | 10         | 96,039,597    | 0.036       | [46]   |
| rs1530440       | 10         | 63,524,591    | 0.028       | [45, 46]|
| rs10895001      | 11         | 100,533,021   | 0.043       | [46]   |
| rs2253579       | 11         | 47,440,758    | 0.035       | [46]   |
| rs2923089       | 11         | 10,357,572    | 0.029       |        |
| rs762551        | 11         | 75,041,917    | 0.027       | [46]   |
| rs4548577       | 11         | 46,998,512    | 0.026       |        |
| rs2681492       | 12         | 90,013,089    | 0.030       | [45, 46]|
| rs10849937      | 12         | 111,792,427   | 0.030       | [46]   |
| rs35085068      | 14         | 23,409,909    | 0.027       |        |
| rs12091664      | 15         | 98,338,524    | 0.027       |        |
| rs7497304       | 15         | 91,429,176    | 0.021       | [45, 46]|
| rs2677738       | 15         | 91,441,673    | 0.021       | [46]   |
| rs3744760       | 17         | 43,195,981    | 0.043       | [46]   |
| rs292445        | 18         | 55,897,720    | 0.026       |        |
| rs762551        | 19         | 75,041,917    | 0.027       | [46]   |
| rs34328549      | 19         | 7,253,184     | 0.035       | [46]   |
| rs16982520      | 20         | 57,758,720    | 0.030       | [45, 46]|

\( \hat{\beta} \) is the estimated effect size.

Cardiovascular GWAS in NFBC1966

We also tested IHT on data from the 1966 Northern Finland Birth Cohort (NFBC1966) [40]. Although this dataset is relatively modest with 5,402 participants and 364,590 SNPs, it has 2 virtues. First, it has been analyzed multiple times [15, 40, 47], so comparison with earlier analysis is easy. Second, due to a population bottleneck [48], the participants’ chromosomes exhibit more extensive LD than is typically found in less isolated populations. Multiple-regression methods, including the lasso, have been criticized for their inability to deal with the dependence among predictors induced by LD. Therefore, this dataset provides an interesting test case.

High-density lipoprotein phenotype as a normal model

Using IHT we find previously associated SNPs as well as a few new potential associations. We model the high-density lipoprotein (HDL) phenotype as normally distributed and find a best model size \( k = 9 \) based on 5-fold cross-validation across model sizes \( k = \{1, 2, \ldots, 20\} \). Without debiasing, the analysis was completed in 2 hours and 4 minutes with 30 CPU cores on a single machine. Table 8 displays the recovered predictors. SNP rs1800961 was replaced by rs7499892 with similar effect size if we add the debiasing step in obtaining the final model.

Importantly, IHT is able to simultaneously recover effects for SNPs (1) rs9261224, (2) rs6917603, and (3) rs6917603 with pairwise correlations of \( r_{13} = 0.984, r_{23} = 0.62 \). This result is achieved without grouping of SNPs, which can further increase association power. Compared with earlier analyses of these data, we find 3 SNPs that were not listed in our previous IHT article [15], presumably due to slight algorithmic modifications. The authors of NFBC [40] found 5 SNPs associated with HDL under SNP-by-SNP testing. We did not find SNPs rs2167079 and rs255049. To date, rs255049 was replicated [47]. SNP rs1800961 was reported to be associated with elevated SBP/DBP [45] or that exhibit genome-wide significance when the same data are analyzed as an ordinal trait [46]. Ordinal univariate GWAS treats the different stages of hypertension as ordered categories. Ordinal GWAS has higher power than logistic or multinomial GWAS [46]. The known SNPs displayed in Table 7 tend to have larger absolute effect sizes (mean \( = 0.033 \)) than the unknown SNPs (mean \( = 0.027 \)). Finally, IHT is able to recover 2 pairs of highly correlated SNPs: (rs1347264, rs1898841) and (rs7497304, rs2677738) with pairwise correlations of \( r_{13} = 0.59 \) and \( r_{34} = 0.49 \).

Discussion

Multiple-regression methods like IHT provide a principled way of model fitting and variable selection. With increasing computing power and better software, multiple-regression methods are likely to prevail over univariate methods. This article introduces a scalable implementation of IHT for GLMs. Because lasso regression can handle group and prior weights, we have also extended IHT to incorporate such prior knowledge. When this prior knowledge is available, enhanced IHT outperforms standard IHT. Given its sharper parameter estimates and more robust model selection, IHT is clearly superior to lasso selection or marginal association testing in GWAS.
Our real data analyses and simulation studies suggest that IHT can (i) recover highly correlated SNPs, (ii) avoid overfitting, (iii) deliver better true-positive and false-positive rates than either marginal testing or lasso regression, (iv) recover unbiased regression coefficients, and (v) exploit prior information and group sparsity. Our Julia implementation of IHT exploits parallel computing strategies that scale to biobank-level data. In our opinion, the time is ripe for the genomics community to embrace multiple-regression models as a supplement to and possibly a replacement for marginal analysis.

Although we focused our attention on GWAS, the potential applications of IHT reach far beyond gene mapping. Our IHT implementation accepts arbitrary numeric data and is suitable for a variety of applied statistics problems. Genomics and the broader field of bioinformatics are blessed with rich, ultra-high-dimensional data. IHT is designed to solve such problems. By extending IHT to the realm of GLMs, it becomes possible to fit regression models with more exotic distributions than the Gaussian distributions implicit in ordinary linear regression. In our view IHT will eventually join and probably supplant lasso regression as the method of choice in GWAS and other high-dimensional regression settings.

**Methods**

**Data simulation**

Our simulations mimic scenarios for a range of rare and common SNPs with or without LD. Unless otherwise stated, we designate 10 SNPs to be causal with effect sizes of 0.1, 0.2, ..., 1.0.

To generate independent SNP genotypes, we first sample a MAF $\rho_j \sim \text{Uniform}(0, 0.5)$ for each SNP $j$. To construct the genotype of person $i$ at SNP $j$, we then sample from a binomial distribution with success probability $\rho_j$ and 2 trials. The vector of genotypes (minor-allele counts) for person $i$ form row $x_t$ of the design matrix $X$. To generate SNP genotypes with LD, we divide all SNPs into blocks of length 20. Within each block, we first sample $x_t \sim \text{Bernoulli}(0.5)$. Then we form a single haplotype block of length 20 by the following Markov chain procedure:

$$x_{i+1} = \begin{cases} x_i & \text{with probability } p \\ 1 - x_i & \text{with probability } 1 - p \end{cases}$$

with default $p = 0.75$. For each block we form a pool of 20 haplotypes using this procedure, ensuring that each of the 40 alleles (2 at each SNP) are represented at least once. For each person, the genotype vector in a block is formed by sampling 2 haplotypes with replacement from the pool and summing the number of minor alleles at each SNP.

Depending on the simulation, the number of subjects ranges from 1,000 to 120,000, and the number of independent SNPs ranges from 10,000 to 1,000,000. We simulate data under 4 GLM distributions: normal (Gaussian), Bernoulli, Poisson, and negative binomial. We generate component $y_i$ of the response vector $y$ by sampling from the corresponding distribution with mean $\mu_i = g(x_t \beta)$, where $g$ is the inverse link function. For normal models we assume unit variance, and for negative binomial models we assume 10 required failures. To avoid overflows, we clamp the mean $g(x_t \beta)$ to stay within $[-20, 20]$. (See Ad Hoc Tactics for a detailed explanation). We apply the canonical link for each distribution, except for the negative binomial, where we apply the log link.

**Table 8:** NFBC GWAS results generated by running IHT on high-density lipoprotein (HDL) phenotype as a normal response

| SNP ID   | Chromosome | Position (bp) | $\hat{\beta}$ | Known? |
|----------|------------|---------------|--------------|--------|
| rs6917603 | 6          | 30,125,050    | 0.17         | [15, 45] |
| rs9261256 | 6          | 30,129,920    | −0.07        | [15]   |
| rs9261224 | 6          | 30,121,866    | −0.03        |        |
| rs7120118 | 11         | 47,242,866    | −0.03        | [15, 40, 45] |
| rs1532085 | 15         | 56,470,658    | −0.04        | [15, 40, 45] |
| rs3764261 | 16         | 55,550,825    | −0.05        | [15, 40, 45] |
| rs3852700 | 16         | 65,829,359    | −0.03        |        |
| rs1800961 | 20         | 42,475,778    | 0.03         | [45]   |

$\hat{\beta}$ is the estimated effect size.

**Figure 4:** Manhattan plot comparing a logistic (univariate) GWAS vs logistic IHT on UK Biobank data. Colored dots are log$_{10}$ P-values from a logistic GWAS, and the circled dots are SNPs recovered by IHT.
Real data’s quality control procedures

UK Biobank
Following the UK Biobank’s own quality control procedures, we first filtered all samples for sex discordance and high heterozygosity/missingness. Second, we included only participants of European ancestry and excluded first- and second-degree relatives on the basis of empiric kinship coefficients. Third, we also exclude imputed participants who had taken hypertension-related medications at baseline. Finally, we only included participants with \( ≥98\% \) genotyping success rate over all chromosomes and SNPs with \( ≥99\% \) genotyping success rate over all included individuals. Calculation of kinship coefficients and filtering were carried out via the OpenMendel module SnpArrays [50]. Remaining missing genotypes were imputed using modal genotypes at each SNP. After these quality control procedures, our UK Biobank data are the same data that were used by German et al. [46].

Northern Finland Birth Cohort
We imputed missing genotypes with Mendel [51]. Following Keys et al. [15], we excluded participants with missing phenotypes, fasting participants, and participants receiving diabetes medication. We conducted quality control measures using the OpenMendel module SnpArrays [50]. On the basis of these measures, we excluded SNPs with MAF ≤ 0.01 and Hardy-Weinberg equilibrium \( P \)-values ≤ 10\(^{-5}\). Concerning non-genetic predictors, we included sex (the sexOCPC factor defined in Sabatti et al. [40]) as well as the first 2 principal components of the genotype matrix computed via PLINK 2.0 alpha [52]. To put predictors, genetic and non-genetic, on an equal footing, we standardized all predictors to have mean zero and unit variance.

Linear algebra with compressed genotype files
The genotype count matrix stores minor-allele counts. The PLINK genotype compression protocol [52] compactly stores the corresponding 0’s, 1’s, and 2’s in 2 bits per SNP, achieving a compression ratio of 32:1 compared with storage as floating point numbers. For a sparsity level \( k \) model, we use OpenBLAS (a highly optimized linear algebra library) to compute predicted values. This requires transforming the \( k \) pertinent columns of \( X \) into a floating point matrix \( X_k \) and multiplying it by the corresponding entries \( \beta_k \) of \( \beta \). The inverse link is then applied to \( X_k \beta_k \) to give the mean vector \( \mu = g(X_k \beta_k) \). In computing the GLM gradient (formula 3), formation of the vector \( W_i(y - \mu) \) involves no matrix multiplications. Computation of the gradient \( X'W_i(y - \mu) \) is more complicated because the full matrix \( X \) can no longer be avoided. Fortunately, the OpenMendel module SnpArrays [50] can be invoked to perform compressed matrix times vector multiplication. Calculation of the step length of IHT requires computation of the quadratic form \( \nabla L \). This computation requires a single compressed matrix times vector multiplication. Finally, good statistical practice calls for standardizing covariates. To standardize the genotype counts for SNP \( j \), we estimate its MAF \( p_j \) and then substitute the ratio \( (x_{ij} - 2p_j) / (2p_j(1 - p_j))^{1/2} \) for the genotype count \( x_{ij} \) for person \( i \) at SNP \( j \). This procedure is predicated on a binomial distribution for the count \( x_{ij} \). Our previous article [15] shows how to accommodate standardization in the matrix operations of IHT without actually forming or storing the standardized matrix.

OpenBLAS has advantages in parallelization, but it requires floating point arrays. Once the genotype matrix \( X \) exceeds the memory available in RAM, expensive data swapping between RAM and disk memory sets in. This dramatically slows matrix multiplication. SnpArrays is less vulnerable to this hazard owing to compression. Once compressed data exceed RAM, SnpArrays also succumbs to the swapping problem. Current laptop and desktop computers seldom have >32 GB of RAM, so we may wish to resort to cluster or cloud computing when input files exceed 32 GB.

Computations involving non-genetic covariates
Non-genetic covariates are stored as double or single precision floating point entries in an \( n \times r \) design matrix \( Z \). To accommodate an intercept, the first column should be a vector of 1’s. Let \( y \) denote the vector of regression coefficients corresponding to \( Z \). The full design matrix is the block matrix \( (XZ) \). Matrix multiplications involving \( (XZ) \) should be carried out via

\[
(XZ) \left( \begin{array}{c} \hat{\beta} \\ \gamma \end{array} \right) = X\beta + Z\gamma \quad \text{and} \quad (XZ)'v = \left( \begin{array}{c} X'v \\ Z'v \end{array} \right).
\]

Adherence to these rules ensures a low memory footprint. Multiplication involving \( X \) can be conducted as previously explained. Multiplication involving \( Z \) can revert to BLAS.

Parallel computation
The OpenBLAS library accessed by Julia is inherently parallel. Beyond that we incorporate parallel processing in cross-validation. Recall that in \( q \)-fold cross-validation, we separate subjects into \( q \) disjoint subsets. We then fit a training model using \( q - 1 \) of those subsets on all desired sparsity levels and record the mean-squared prediction error on the omitted subset. Each of the \( q \) subsets serves as the testing set exactly once. Testing error is averaged across the different folds for each sparsity level \( k \). The lowest average testing error determines the recommended sparsity.

MendelIHT.jl offers 2 parallelism strategies in cross-validation. Either the \( q \) training sets are each loaded to \( q \) different CPUs where each computes and tests different sparsity levels sequentially, or each of the \( q \) training sets is cycled through sequentially and each sparsity parameter is fitted and tested in parallel. The former tactic requires enough disk space and RAM to store \( q \) different training datasets [where each typically requires \( q - 1 \)/\( q \) GB of the full data] but offers immense parallel power because one can assign different computers to handle different sparsity levels. This tactic allows one to fit biobank-scale data in less than one day, assuming enough storage space and computers are available. The latter tactic requires cycling through the training sets sequentially. Because intermediate data can be deleted, this tactic only requires enough disk space and RAM to store one copy of the training set. MendelIHT.jl uses one of Julia’s [22] standard libraries, Distributed.jl, to achieve the aforementioned parallel strategies.

Ad hoc tactics to prevent overflows
In Poisson and negative binomial regressions, the inverse link argument \( \exp(x' \beta) \) experiences numerical overflows when the inner product \( x' \beta \) is too large. In general, we avoid running Poisson regression when response means are large. In this regime a normal approximation is preferred. As a safety fea-
Convergence and backtracking

For each proposed IHT step we check whether the objective $L(\beta)$ increases. When it does not, we step-half at most 5 times to restore the ascent property. Convergence is declared when

$$\frac{||\beta_{t+1}-\beta_t||_\infty}{||\beta_t||_\infty+1} < \text{Tolerance},$$

with the default tolerance being 0.0001. The addition of 1 in the denominator of the convergence criterion guards against division by 0.

Availability of Source Code and Requirements

Project name: MendelIHT
Project home page: https://github.com/OpenMendel/MendelIHT
Operating systems: Mac OS, Linux, Windows
Programming language: Julia 1.0, 1.2
License: MIT

The code to generate simulated data, as well as their subsequent analysis, is available in our GitHub repository under the “figures” folder. Project.toml and Manifest.toml files can be used together to instantiate the same computing environment in our article. Notably, MendelIHT.jl interfaces with the OpenMendel [38] package SnpArrays.jl [50] and JuliaStats’s packages Distribution.jl [53] and GLM.jl [54].

Availability of Supporting Data and Materials

The Northern Finland Birth Cohort 1966 (NFBC1966) [40] was downloaded from dbGaP under dataset accession phl002005.v1.p1. UK Biobank data are retrieved under Project ID: 48152 and 15678. An archival snapshot of the code and other supporting data is available via the GigaScience Database, GigaDB [55].

Additional Files

Supplementary information: Supplementary data are available in GigaDB.

Abbreviations

BLAS: Basic Linear Algebra Subprograms; BMI: body mass index; bp: base pairs; CPU: central processing unit; DBP: diastolic blood pressure; GLM: generalized linear models; GWAS: genome-wide association studies; HDL: high-density lipoprotein; HTP: hard thresholding pursuit; IHT: iterative hard thresholding; LD: linkage disequilibrium; LDL: low-density lipoprotein; MAF: minor allele frequency; MCP: minimax concave penalty; Neg Bin: negative binomial; NFBC: Northern Finland Birth Cohort; NHLBI: National Heart, Lung, and Blood Institute; NIH: National Institutes of Health; NIHT: normalized iterative hard threshold algorithm; RAM: random access memory; S2 Hyp: Stage 2 hypertension; SBP: systolic blood pressure; SNP: single-nucleotide polymorphism.

Competing Interests

The authors declare that they have no competing interests.

Ethics and Consent for Publication

As described in [40], informed consent from all study participants of NFBC1966 was obtained using protocols approved by the Ethical Committee of the Northern Ostrobothnia Hospital District.

Funding

B.B.C. was supported by NIH T32-HG002536 training grant and the 2018 Google Summer of Code. K.L.K. was supported by a diversity supplement to NHLBI grant R01HL135156, the UCSF Bakar Computational Health Sciences Institute, the Gordon and Betty Moore Foundation grant GBMF3834, and the Alfred P. Sloan Foundation grant 2013-10-27 to UC Berkeley through the Moore-Sloan Data Sciences Environment initiative at the Berkeley Institute for Data Science (BIDS). E.M.S., K.L., and H.Z. were supported by grants from the National Human Genome Research Institute (HG006139) and the National Institute of General Medical Sciences (GM053275). J.S.S. was supported by grants from the National Institute of General Medical Sciences (GM053275), the National Human Genome Research Institute (HG009120), and the National Science Foundation (DMS-1264153). C.A.G. was supported by the Burroughs Wellcome Fund Inter-school Training Program in Chronic Diseases (BWF-CHIP).

Authors’ Contributions

B.B.C., K.L.K., E.M.S., J.S.S., and K.L. contributed to the design of the study, interpretation of results, and writing of the original draft manuscript. B.B.C. designed and implemented the simulations and conducted the data analyses. C.A.G., H.Z., and J.J.Z. contributed to the analysis of UK Biobank results. B.B.C. and K.L.K. developed the software. B.B.C. and K.L. developed the algorithms. E.M.S assisted in the comparisons to marginal GWAS. All authors have read, made suggestions, and ultimately approved the final manuscript.

Acknowledgments

We thank UCLA for providing the computing resources via the Hoffman2 cluster.

The NFBC1966 Study is conducted and supported by the National Heart, Lung, and Blood Institute (NHLBI) in collaboration with the Broad Institute, UCLA, University of Oulu, and the National Institute for Health and Welfare in Finland. This manuscript was not prepared in collaboration with investigators of the NFBC1966 Study and does not necessarily reflect the opinions or views of the NFBC1966 Study Investigators, Broad Institute, UCLA, University of Oulu, National Institute for Health and Welfare in Finland, and the NHLBI.

References

1. Cantor RM, Lange K, Sinhimer JS. Prioritizing GWAS results: a review of statistical methods and recommendations for their application. Am J Hum Genet 2010;86:6–22.
2. Visscher PM, Wray NR, Zhang Q, et al. 10 years of GWAS discovery: biology, function, and translation. Am J Hum Genet 2017;101:5–22.
3. Bush WS, Moore JH. Genome-wide association studies. PLoS Comput Biol 2012;8:e1002822.
4. Han B, Eskin E. Random-effects model aimed at discovering associations in meta-analysis of genome-wide association studies. Am J Hum Genet 2011;88:586–98.
5. Loh PR, Tucker G, Bulik-Sullivan BK, et al. Efficient Bayesian mixed-model analysis increases association power in large cohorts. Nat Genet 2015;47:284.
6. Rahman SK, Sathik MM, Kannan KS. Multiple linear regression models in outlier detection. Int J Res Comput Sci 2012;2(2):23.
7. Tibshirani R. Regression shrinkage and selection via the lasso. J R Stat Soc B Methodol 1996;58:267–88.
8. Vattikuti S, Lee JI, Chang CC, et al. Applying compressed sensing to genome-wide association studies. GigaScience 2014;3:10.
9. Friedman J, Hastie T, Tibshirani R. Regularization paths for generalized linear models via coordinate descent. J Stat Softw 2010;33:1.
10. Wu TT, Lange K. Coordinate descent algorithms for lasso penalized regression. Ann Appl Stat 2008;2:224–44.
11. Zhang T. Analysis of multi-stage convex relaxation for sparse regularization. J Mach Learn Res 2010;11:1081–107.
12. Breheny P, Huang J. Coordinate descent algorithms for nonconvex penalized regression, with applications to biological feature selection. Ann Appl Stat 2011;5:232–53.
13. Mazumder R, Friedman JH, Hastie T. SparseNet: Coordinate descent with nonconvex penalties. J Am Stat Assoc 2011;106:1125–38.
14. Hoffman GE, Logsdon BA, Mezey JG. PUMA: A unified framework for penalized multiple regression analysis of GWAS data. PLoS Comput Biol 2013;9:e1003101.
15. Keys KL, Chen GK, Lange K. Iterative hard thresholding for model selection in genome-wide association studies. Genet Epidemiol 2017;41:756–68.
16. Meinsmaunen N, Bühlmann P. Stability selection. J R Stat Soc B Methodol 2010;72:417–73.
17. Alexander DH, Lange K. Stability selection for genome-wide association. Genet Epidemiol 2011;35:722–8.
18. Beck A. Introduction to Nonlinear Optimization: Theory, Algorithms, and Applications with MATLAB. SIAM; 2014.
19. Beck A, Teboulle M. A linearly convergent algorithm for solving a class of nonconvex/affine feasibility problems. In: Fixed-Point Algorithms for Inverse Problems in Science and Engineering. Springer; 2011:33–48.
20. Blumensath T, Davies ME. Normalized iterative hard thresholding: Guaranteed stability and performance. IEEE J Sel Top Signal Process 2010;4:298–309.
21. Yang F, Barber RF, Jain P, et al. Selective inference for group-sparse linear models. In: Advances in Neural Information Processing Systems. 2016:2469–77.
22. Bezanson J, Edelman A, Karpinski S, et al. Julia: A fresh approach to numerical computing. SIAM Rev 2017;59:65–98.
23. Foucart S. Hard thresholding pursuit: an algorithm for compressive sensing. SIAM J Num Anal 2011;49:2543–63.
24. Yuan XT, Li P, Zhang T. Gradient hard thresholding pursuit. J Mach Learn Res 2017;18:166–1.
25. Needell D, Tropp JA. CoSaMP: Iterative signal recovery from incomplete and inaccurate samples. Appl Comput Harmon Anal 2009;26:301–21.
26. Dobson AJ, Barnett A. An Introduction to Generalized Linear Models. Chapman and Hall/CRC; 2008.
27. McCullagh P. Generalized Linear Models. Routledge; 2018.
28. Lange K. Numerical Analysis for Statisticians. Springer Science & Business Media; 2010.
29. Xu J, Chi E, Lange K. Generalized linear model regression under distance-to-set penalties. In: Advances in Neural Information Processing Systems 30. Curran Associates Inc; 2017:1385–95.
30. Meier L, Van De Geer S, Buhlmann P. The group lasso for logistic regression. J R Stat Soc B Methodol 2008;70:53–71.
31. Friedman J, Hastie T, Tibshirani R. A note on the group lasso and a sparse group lasso. arXiv 2010: 1001.0736.
32. Zhou H, Sehl ME, Sinheimer JS, et al. Association screening of common and rare genetic variants by penalized regression. Bioinformatics 2010;26:2375.
33. Zhou H, Alexander DH, Sehl ME, et al. Penalized regression for genome-wide association screening of sequence data. Pac Symp Biocomput 2011:106–117, doi:10.1142/9789814335058_0012.
34. Zeng J, De Vlaming R, Wu Y, et al. Signatures of negative selection in the genetic architecture of human complex traits. Nature Genet 2018;50(7):746.
35. Schoech AP, Jordan DM, Loh PR, et al. Quantification of frequency-dependent genetic architectures in 25 UK Biobank traits reveals action of negative selection. Nat Commun 2019;10(1):790.
36. de Lamare RC. Knowledge-aided normalized iterative hard thresholding algorithms and applications to sparse reconstruction. arXiv 2018: 1809.09281.
37. Wu TT, Chen YF, Hastie T, et al. Genome-wide association analysis by lasso penalized logistic regression. Bioinformatics 2009;25:714–21.
38. Zhou H, Sinheimer JS, Bates DM, et al. OpenMendel: a cooperative programming project for statistical genetics. Hum Genet 2020;139(1):61–71.
39. Zeileis A, Kleiber C, Jackman S. Regression models for count data in R. J Stat Softw 2008;27(8):1–25.
40. Sabatti C, Service SK, Hartikainen AL, et al. Genome-wide association analysis of metabolic traits in a birth cohort from a founder population. Nat Genet 2009;41:35.
41. Sudlow C, Gallacher J, Allen N, et al. UK BioBank: an open access resource for identifying the causes of a wide range of complex diseases of middle and old age. PLoS Med 2015;12:e1001779.
42. Abraham G, Qiu Y, Inouye M. FlashPCA: principal component analysis of Biobank-scale genotype datasets. Bioinformatics 2017;33(17):2776–8.
43. Whelton PK, Carey RM, Aronow WS, et al. 2017 ACC/AHA/ AAPA/ABC/ACP/AHS/ASH/ASPC/NMA/PCNA guideline for the prevention, detection, evaluation, and management of high blood pressure in adults: a report of the American College of Cardiology/American Heart Association Task Force on Clinical Practice Guidelines. J Am Coll Cardiol 2018;71(19):e127–e248.
44. German CA, Zhou H. MendelPlots.jl: Julia package for plotting results from GWAS. Zenodo 2020. https://doi.org/10.5281/zenodo.3665675.
45. MacArthur J, Bowler E, Cerezo M, et al. The new NHGRI-EBI Catalog of published genome-wide association studies (GWAS Catalog). Nucleic Acids Res 2016;45:D896–D901.
46. German CA, Sinsheimer JS, Klimentidis YC, et al. Ordered multinomial regression for genetic association analysis of ordinal phenotypes at Biobank scale. Genet Epidemiol 2020;44(3):248–60.
47. Gai L, Eskin E. Finding associated variants in genome-wide association studies on multiple traits. Bioinformatics 2018;34:i467–74.
48. Martin AR, Karczewski KJ, Kerminen S, et al. Haplotype sharing provides insights into fine-scale population history and disease in Finland. Am J Hum Genet 2018;102:760–75.
49. Melquist S, Craig DW, Huentelman MJ, et al. Identification of a novel risk locus for progressive supranuclear palsy by a pooled genomewide scan of 500,288 single-nucleotide polymorphisms. Am J Hum Genet 2007;80:769–78.
50. Zhou H, Papp JC, Ko S, et al. SnpArrays.jl: Julia package for compressed storage of SNP data. Zenodo 2020. https://doi.org/10.5281/zenodo.3665823.
51. Lange K, Papp JC, Sinsheimer JS, et al. Mendel: the Swiss army knife of genetic analysis programs. Bioinformatics 2013;29:1568–70.
52. Chang CC, Chow CC, Tellier LC, et al. Second-generation PLINK: rising to the challenge of larger and richer datasets. Gigascience 2015;4:7.
53. Besançon M, Anthoff D, Arslan A, et al. Distributions.jl: Definition and modeling of probability distributions in the JuliaStats Ecosystem. arXiv 2019, 1907.08611.
54. Lin D, White JM, Byrne S, et al. JuliaStats/Distributions.jl: a Julia package for probability distributions and associated functions. 2019. https://doi.org/10.5281/zenodo.3730565
55. Chu BB, Keys KL, German CA, et al. Supporting data for “Iterative hard thresholding in genome-wide association studies: Generalized linear models, prior weights, and double sparsity.” GigaScience Database 2020. http://dx.doi.org/10.5524/100722.