Application Note
lassosum2: an updated version complementing LDpred2

Florian Privé,[1,*] Bjarni J. Vilhjálmsson,[1,2] and Timothy S. H. Mak[3]

[1]National Centre for Register-Based Research, Aarhus University, Aarhus, 8210, Denmark.
[2]Bioinformatics Research Centre, Aarhus University, Aarhus, 8000, Denmark.
[3]Fano Labs, Hong Kong
[*]To whom correspondence should be addressed.

Contact: `florian.prive.21@gmail.com`

Abstract

We present lassosum2, a new version of the polygenic score method lassosum, which we re-implement in R package bigsnpr. This new version uses the exact same input data as LDpred2 and is also very fast, which means that it can be run with almost no extra coding nor computational time when already running LDpred2. It can also be more robust than LDpred2, e.g. in the case of a large GWAS sample size misspecification. Therefore, lassosum2 is complementary to LDpred2.

Introduction

Multiple studies have shown that LDpred2 and lassosum are consistently ranking best among methods for polygenic prediction (Privé *et al.* 2020; Pain *et al.* 2020; Kulm *et al.* 2021). This has motivated this work where we present a re-implementation of lassosum in R package bigsnpr (Privé *et al.* 2018), which we call lassosum2. This new version uses the exact same input data as LDpred2 and is also very fast, which means that it can be run with almost no extra coding nor computational time when already running LDpred2. Therefore, users of R package bigsnpr can now run two of the best polygenic score methods, and then can choose the best model or stack models to improve prediction (Privé *et al.* 2019; Kulm *et al.* 2021). This new version of lassosum also comes with a few improvements that leads to higher predictive performance for e.g. asthma and type 1 diabetes. The code is also substantially simpler, e.g. it does not require splitting the genome in independent LD blocks anymore, which might not always be easy (Privé 2021). Finally, we show that lassosum can be more robust than LDpred2, e.g. in the case of a large GWAS sample size misspecification. Again, this motivates our development that makes it easy to run both methods to get the best prediction possible.

New implementation

From Mak *et al.* (2017), the solution from lassosum can be obtained by iteratively updating

$$\beta_j^{(t)} = \begin{cases} \text{sign}\left(u_j^{(t)}\right)\left(\left|u_j^{(t)}\right| - \lambda\right) / \left(\widetilde{X}_j^T \widetilde{X}_j + s\right) & \text{if } \left|u_j^{(t)}\right| > \lambda\,, \\ 0 & \text{otherwise.} \end{cases}$$

where

$$u_j^{(t)} = r_j - \widetilde{X}_j^T \left(\widetilde{X}\beta^{(t-1)} - \widetilde{X}_j \beta_j^{(t-1)}\right)\,.$$

Following notations from Privé *et al.* (2020) and denoting $\widetilde{X} = \sqrt{\frac{1-s}{n-1}} C_n G S^{-1}$, when G is the genotype matrix, C_n is the centering matrix and S is the diagonal matrix of standard deviations of G, then $\widetilde{X}_j^T \widetilde{X} = (1-s) R_{j,\cdot} = (1-s) R_{\cdot,j}^T$ where R is the correlation matrix between variants. Then $\widetilde{X}_j^T \widetilde{X}_j + s = 1$ and

$$u_j^{(t)} = \widehat{\beta}_j + (1-s)\left(\beta_j^{(t-1)} - R_{\cdot,j}^T \beta^{(t-1)}\right)\,,$$

where $r_j = \widehat{\beta}_j = \dfrac{\widehat{\gamma}_j}{\sqrt{n_j\,\text{se}(\widehat{\gamma}_j)^2 + \widehat{\gamma}_j^2}}$ and $\widehat{\gamma}_j$ is the GWAS effect of variant j and n is the GWAS sample size (Mak *et al.* 2017; Privé *et al.* 2021). Then the most time-consuming part is computing $R_{\cdot,j}^T \beta^{(t-1)}$. To make this faster, instead of computing $R_{\cdot,j}^T \beta^{(t-1)}$ at each iteration (j and t), we can start with a vector with only 0s initially (for all j) since $\beta^{(0)} \equiv 0$, and then updating this vector when $\beta_j^{(t)} \neq \beta_j^{(t-1)}$ only. Note that only positions k for which $R_{k,j} \neq 0$ must be updated in this vector $R_{\cdot,j}^T \beta^{(t-1)}$.

In this new implementation of the lassosum model, the input parameters are the correlation matrix R, the GWAS summary statistics ($\widehat{\gamma}_j$, $\text{se}(\widehat{\gamma}_j)$ and n_j), and the two hyper-parameters λ and s. Therefore, except for the hyper-parameters, this is the exact same input as for LDpred2 (Privé *et al.* 2020). For s, we now

try $\{0.1, 0.2, \ldots, 1.0\}$ by default, instead of only $\{0.2, 0.5, 0.8, 1.0\}$. For λ, the default in lassosum uses a sequence of 20 values equally spaced on a log scale between 0.1 and 0.001; we now use a sequence between λ_0 and $\lambda_0/100$ by default in lassosum2, where $\lambda_0 = \max_j \left|\widehat{\beta}_j\right|$ is the minimum λ for which no variable enter the model because the L1-regularization is too strong. Note that we do not provide an "auto" version using pseudo-validation (as in Mak *et al.* (2017)) as we have not found it to be very robust (Figures 1 and S3). Also note that, as in LDpred2, we run lassosum2 genome-wide using a sparse correlation matrix which assumes that variants further away than 3 cM are not correlated, and therefore we do not require splitting the genome into independent LD blocks anymore (as in lassosum).

Results and Discussion

We use the exact same data as in Privé *et al.* (2020); we use external summary statistics for eight case-control phenotypes and predict in the UK Biobank (UKBB) unrelated European data by keeping 10,000 individuals as validation set for tuning hyper-parameters and using the remaining 352,320 individuals as test set (Bycroft *et al.* 2018). As we use the exact same split as before, results can be directly compared to results in Privé *et al.* (2020) and we run only lassosum2 as the new method. Figure 1 compares lassosum2 with lassosum and LDpred2. AUC values for lassosum2 are similar to lassosum for breast cancer (BRCA), coronary artery disease (CAD), depression (MDD), prostate cancer (PRCA), rheumatoid arthritis (RA) and type 2 diabetes (T2D), and slighly better for asthma and type 1 diabetes (T1D). Prediction performance with LDpred2 is still slightly higher than with lassosum2, except for MDD and T1D for which predictive accuracy is similar (Figure 1). In terms of computational speed, lassosum2 runs in a few minutes compared to a few hours for LDpred2.

We then design simulations where variants have different GWAS sample sizes, which is often the case when meta-analyzing GWAS from multiple cohorts without the same genome coverage. Using 16,410 HapMap3 variants from chromosome 22, we simulate quantitative phenotypes assuming an heritability of 20% and 2000 causal variants. For half of the variants, we use 100% of 300,000 individuals for GWAS, but use only 80% for one quarter of them and 60% for the remaining quarter. We then run C+T, LDpred2-inf, LDpred2(-grid), LDpred2-auto, lassosum, lassosum2, PRS-CS and SBayesR by using either the true per-variant GWAS sample size, or the total sample size. Averaged over 10 simulations, when providing true per-variant GWAS sample sizes, squared correlations (in %) between the polygenic scores and the simulated phenotypes are of 13.9 for C+T, 15.4 for PRS-CS, 17.5 for lassosum, 18.0 for lassosum2, and 17.8 for LDpred2 (Figure S2). Note that C+T does not use this sample size information, that PRS-CS can only use one global sample size, and that SBayesR always diverged (so we discarded it). For methods that use the per-variant GWAS sample sizes, predictive performance slightly decreases from 17.5 to 17.4 for lassosum, from 18.0 to 17.5 for lassosum2, but dramatically decreases for LDpred2 with new values of 11.5 for LDpred2-grid, 5.7 for LDpred2-auto and 6.8 for LDpred2-inf (Figure S2). Therefore, this extreme simulation scenario shows that LDpred2 can be sensitive to GWAS sample size misspecification, whereas lassosum (and lassosum2) seems little affected by this.

To sum up, here we propose a new version of lassosum, lassosum2, that can be run with almost no extra coding nor computational time when already running LDpred2 because it uses the same input arguments

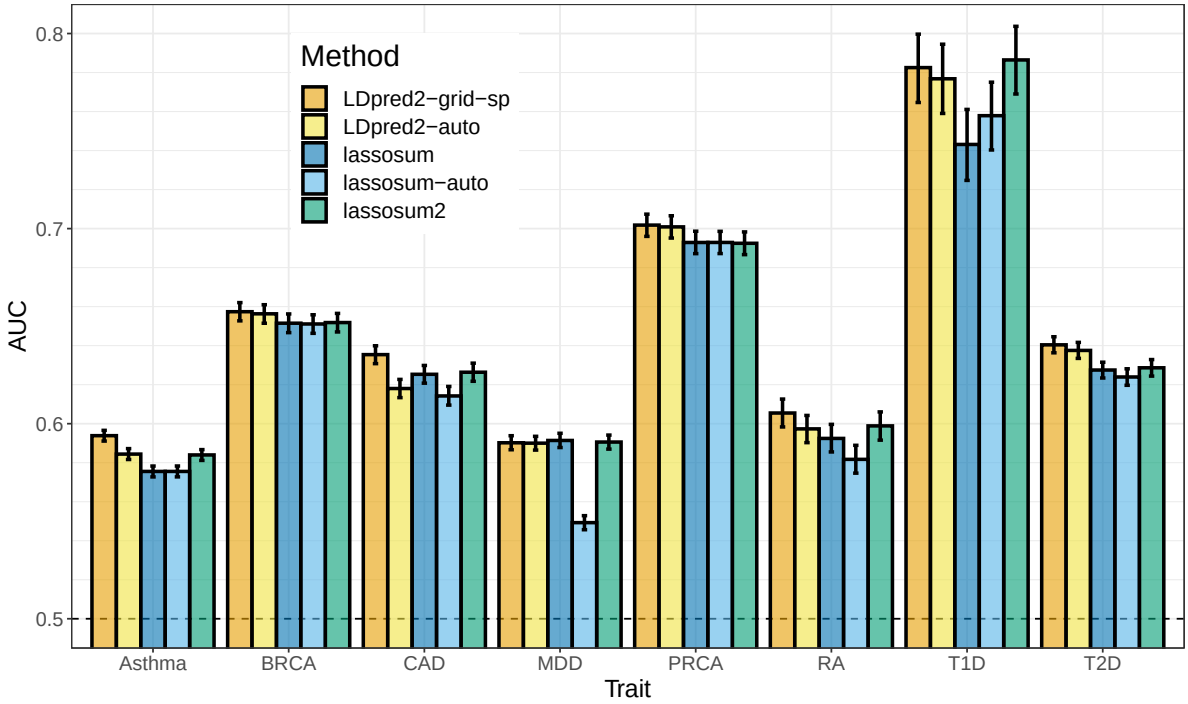

Figure 1: lassosum2 is compared with LDpred2, LDpred2-auto, lassosum and lassosum-auto using published external summary statistics and the UKBB data. Bars present AUC values on the test set of UKBB (mean and 95% CI from 10,000 bootstrap samples). For additional comparisons to C+T, SCT, PRS-CS, PRS-CS-auto and SBayesR (Privé *et al.* 2019; Ge *et al.* 2019; Lloyd-Jones *et al.* 2019), the reader is referred to figures 4 and S1 of Privé *et al.* (2020).

as LDpred2 and is very fast. We also show that it generally provides very good predictive performance, sometimes better than LDpred2, e.g. in the case of severe sample size misspecification. Finally, we would like to mention that, except for the GWAS summary statistics provided by the Psychiatric Genomics Consortium (PGC), per-variant GWAS sample sizes are often not reported in GWAS summary statistics. We show here that lassosum is robust to this misspecification (i.e. when using the total sample size for all variants), but that it can dramatically affect predictive performance of LDpred2; we may wonder which other statistical genetics methods could be affected by this missing information? Although this sample size misspecificaition can possibly be detected from the quality control plot we propose in Privé *et al.* (2020) (Figure S1), we urge researchers to report the per-variant sample sizes from their GWAS meta-analyses.

Code and results availability

All code used for this paper is available at `https://github.com/privefl/paper-lassosum2/tree/master/code`. Latest versions of R package bigsnpr can be installed from GitHub. A tutorial on running lassosum2 along with LDpred2 using R package bigsnpr is available at `https://privefl.github.io/bigsnpr-extdoc/polygenic-scores-pgs.html`. We have extensively used R packages bigstatsr and bigsnpr (Privé *et al.* 2018) for analyzing large genetic data, packages from the future framework (Bengtsson 2020) for easy scheduling and parallelization of analyses on the HPC cluster, and

packages from the tidyverse suite (Wickham *et al.* 2019) for shaping and visualizing results.

Acknowledgements

Authors thank GenomeDK and Aarhus University for providing computational resources and support that contributed to these research results. This research has been conducted using the UK Biobank Resource under Application Number 58024.

Funding

F.P. and B.J.V. are supported by the Danish National Research Foundation (Niels Bohr Professorship to Prof. John McGrath). B.J.V. is also supported by a Lundbeck Foundation Fellowship (R335-2019-2339).

Declaration of Interests

The authors declare no competing interests.

References

Bengtsson, H. (2020). A unifying framework for parallel and distributed processing in R using futures. *arXiv preprint arXiv:2008.00553*.

Bycroft, C., Freeman, C., Petkova, D., Band, G., Elliott, L. T., Sharp, K., Motyer, A., Vukcevic, D., Delaneau, O., O'Connell, J., *et al.* (2018). The UK Biobank resource with deep phenotyping and genomic data. *Nature*, **562**(7726), 203–209.

Ge, T., Chen, C.-Y., Ni, Y., Feng, Y.-C. A., and Smoller, J. W. (2019). Polygenic prediction via Bayesian regression and continuous shrinkage priors. *Nature Communications*, **10**(1), 1776.

Kulm, S., Marderstein, A., Mezey, J., and Elemento, O. (2021). A systematic framework for assessing the clinical impact of polygenic risk scores. *medRxiv*, pages 2020–04.

Lloyd-Jones, L. R., Zeng, J., Sidorenko, J., Yengo, L., Moser, G., Kemper, K. E., Wang, H., Zheng, Z., Magi, R., Esko, T., *et al.* (2019). Improved polygenic prediction by Bayesian multiple regression on summary statistics. *Nature Communications*, **10**(1), 1–11.

Mak, T. S. H., Porsch, R. M., Choi, S. W., Zhou, X., and Sham, P. C. (2017). Polygenic scores via penalized regression on summary statistics. *Genetic epidemiology*, **41**(6), 469–480.

Pain, O., Glanville, K. P., Hagenaars, S. P., Selzam, S. P., Fürtjes, A. E., Gaspar, H. A., Coleman, J. R., Rimfeld, K., Breen, G., Plomin, R., *et al.* (2020). Evaluation of polygenic prediction methodology within a reference-standardized framework. *bioRxiv*.

Privé, F. (2021). Optimal linkage disequilibrium splitting. *bioRxiv*.

Privé, F., Aschard, H., Ziyatdinov, A., and Blum, M. G. B. (2018). Efficient analysis of large-scale genome-wide data with two R packages: bigstatsr and bigsnpr. *Bioinformatics*, **34**(16), 2781–2787.

Privé, F., Vilhjálmsson, B. J., Aschard, H., and Blum, M. G. B. (2019). Making the most of clumping and thresholding for polygenic scores. *The American Journal of Human Genetics*, **105**(6), 1213–1221.

Privé, F., Arbel, J., and Vilhjálmsson, B. J. (2020). LDpred2: better, faster, stronger. *Bioinformatics*. btaa1029.

Privé, F., Aschard, H., Carmi, S., Folkersen, L., Hoggart, C., O'Reilly, P. F., and Vilhjálmsson, B. J. (2021). High-resolution portability of 245 polygenic scores when derived and applied in the same cohort. *medRxiv*.

Wickham, H., Averick, M., Bryan, J., Chang, W., McGowan, L. D., François, R., Grolemund, G., Hayes, A., Henry, L., Hester, J., *et al.* (2019). Welcome to the tidyverse. *Journal of Open Source Software*, **4**(43), 1686.