Bioinformatics, 38(1), 2022, 255–256
doi: 10.1093/bioinformatics/btab519
Advance Access Publication Date: 14 July 2021
Applications Note

Genome analysis

Optimal linkage disequilibrium splitting

Florian Privé

National Centre for Register-Based Research, Aarhus University, Aarhus 8210, Denmark

Associate Editor: Tobias Marschall

Received on March 7, 2021; revised on May 22, 2021; editorial decision on July 5, 2021; accepted on July 9, 2021

Abstract

Motivation: A few algorithms have been developed for splitting the genome in nearly independent blocks of linkage disequilibrium. Due to the complexity of this problem, these algorithms rely on heuristics, which makes them suboptimal.
Results: Here, we develop an optimal solution for this problem using dynamic programming.
Availability: This is now implemented as function snp_ldsplit as part of R package bigsnpr.
Contact: florian.prive.21@gmail.com
Supplementary information: Supplementary data are available at *Bioinformatics* online.

Introduction

A few algorithms have been developed for splitting the genome in nearly independent blocks of linkage disequilibrium (Berisa and Pickrell, 2016; Kim *et al.*, 2018). Dividing the genome in multiple smaller blocks has many applications. One application is to report signals from independent regions of the genome (Berisa and Pickrell, 2016; Ruderfer *et al.*, 2018; Wen *et al.*, 2017). Another application is for the development of statistical methods, e.g. for deriving polygenic scores (Ge *et al.*, 2019; Mak *et al.*, 2017; Zhou and Zhao, 2020), estimating genetic architecture and performing other statistical genetics analyses (Shi *et al.*, 2016; Wen *et al.*, 2016). Indeed, most statistical methods based on summary statistics also use a correlation matrix (between variants), and these methods often perform computationally expensive operations such as inversion and eigen decomposition of this correlation matrix. These operations are often quadratic, cubic or even exponential with the number of variants. However, if we can decompose the correlation matrix in nearly independent blocks, then we can apply these expensive operations to smaller matrices with less variants, making these operations much faster, and parallelizable. For instance, inverting a block-diagonal matrix requires only inverting each block separately.

Implementation

We aim at optimally splitting the genome into K blocks, where each block has a bounded number of variants (minimum and maximum size). This splitting is optimal in the sense that it minimizes the sum of squared correlations between variants from different blocks (hereinafter denoted as 'cost'). This problem is quite complex, and a naive implementation would be exponential with the number of variants. To solve this problem efficiently, we use dynamic programming, which consists in breaking a problem into subproblems and then recursively finding the optimal solutions to the subproblems. Dynamic programming has been successfully used before to solve related problems such as haplotype block partitioning (Zhang *et al.*, 2002). Here, each subproblem consists in solving

$$C(i,k) = \min_j\{E(i,j) + C(j+1, k-1)\}, \qquad (1)$$

where $C(i, k)$ is the minimum cost for splitting the region from variant i to the last variant into k blocks exactly, and $E(i, j)$ is the error/cost between block (i, j) and the latter blocks. This is illustrated in Figure 1. These subproblems can be solved efficiently by starting with $k = 1$ and with i from the end of the region, and working our way up. Once all costs in the C matrix have been computed, and corresponding splits j have been recorded, the optimal split can be reconstructed from $C(1, K)$, where K is the number of blocks desired.

To efficiently compute $E(i,j) = \sum_{p=i}^{j} \sum_{q=j+1}^{m} R(p,q)^2$, where m is the number of variants and $R(p, q)$ is the correlation between variants p and q, we first compute the matrix L defined as $L(i,j) = \sum_{q=j+1}^{m} R(i,q)^2$. Matrices L and E are sparse. E is the largest matrix and requires approximately $m \times (\max_{\text{size}} - \min_{\text{size}}) \times 4$ bytes to be stored efficiently. For $m = 100\ 000$, $\text{min\_size} = 500$ and $\text{max\_size} = 10\ 000$, this represents 3.5 GB. A description of the parameters of function snp_ldsplit implementing this method can be found in Supplementary section 'Parameters of snp_ldsplit'.

Results

As input, function snp_ldsplit uses a correlation matrix in sparse format from R package Matrix, which can be computed using the available snp_cor function from R package bigsnpr (Privé *et al.*, 2018). This function is fast and parallelized. Then, to run snp_ldsplit using a correlation matrix for 102 451 variants from chromosome 1, it takes <6 min on a laptop to find the optimal split in K blocks (for all $K = 1$ to 133) with a bounded block size between 500 and 10 000 variants. Then, the user can choose the desired number of blocks,

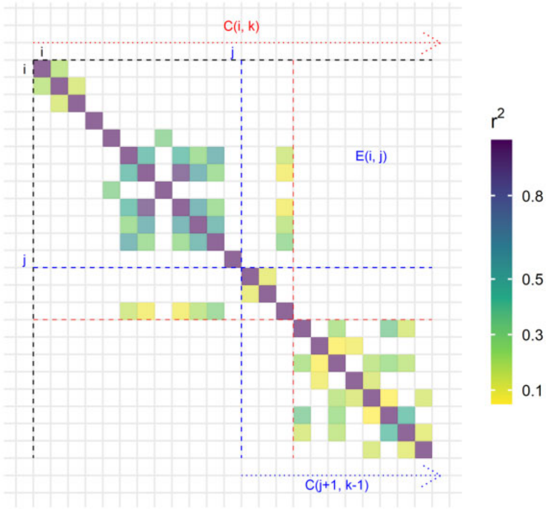

Fig. 1. Illustration of subproblems solved by the algorithm using a small LD matrix. The cost of separating the region starting at variant *i* in *k* blocks exactly, *C*(*i*, *k*), is broken down in two: the error *E*(*i*, *j*), the sum of all squared correlations between variants from block (*i*, *j*) and variants from all the later blocks, and the cost of separating the rest starting at $(j+1)$ using $(k-1)$ blocks. The variant *j* at which the split occurs is chosen so that the cost $\left(E(i,j) + C(j+1, k-1)\right)$ is minimized. The optimal split is highlighted in red here.

which is a compromise between having more (smaller) blocks with a higher overall cost (LD between blocks), and having less (larger) blocks with a smaller cost. For chromosome 1 and Europeans, ldetect report 133 linkage disequilibrium (LD) blocks (Berisa and Pickrell, 2016); however, we find that they can hardly be considered truly independent given the high cost (10 600) of the corresponding split (Supplementary Fig. S1). When splitting chromosome 1 for Europeans using the optimal algorithm we propose here, it can be split into 39 blocks at a cost of 1, in 65 blocks at a cost of 10, and in 133 blocks at a cost of 296 (Supplementary Fig. S1). Similar results are found for other chromosomes, and for Africans and Asians; however, splitting the LD from admixed Americans comes at a high cost (Supplementary Figs S2–S5). Both methods largely pick block boundaries at recombination hotspots (Supplementary Figs S7 and S8). We also provide an application to LD score regression in Supplementary section 'Application to LD score regression', where we show that standard errors for the SNP heritability using nearly independent blocks tend to be larger than when there is substantial LD between blocks, especially for phenotypes with large associations in the HLA (human leukocyte antigen) region (a long-range LD region).

Software, code and data availability

The newest version of R package bigsnpr can be installed from GitHub (see https://github.com/privefl/bigsnpr). All code used for this article is available at https://github.com/privefl/paper-ldsplit/tree/master/code. The HapMap3 variants annotated with 242 blocks can be downloaded at https://www.dropbox.com/s/hdui60p9ohyhvv5/map_blocks.rds?dl=1. LD score regression results are available at https://github.com/privefl/paper-ldsplit/tree/main/ldsc_blocks, with a description of the 245 phenotypes used at https://github.com/privefl/UKBB-PGS/blob/main/phenotype-description.xlsx.

Acknowledgements

We thank Bjarni J. Vilhjálmsson for his feedback on the manuscript, and the two reviewers for their comments and suggestions.

Funding

This work was supported by the Danish National Research Foundation (Niels Bohr Professorship to Prof. John McGrath).

Conflict of Interest: none declared.

References

Berisa,T., and Pickrell,J.K. (2016) Approximately independent linkage disequilibrium blocks in human populations. *Bioinformatics*, **32**, 283–285.

Ge,T. *et al.* (2019) Polygenic prediction via bayesian regression and continuous shrinkage priors. *Nat. Commun.*, **10**, 1–10.

Kim,S.A. *et al.* (2018) A new haplotype block detection method for dense genome sequencing data based on interval graph modeling of clusters of highly correlated snps. *Bioinformatics*, **34**, 388–397.

Mak,T.S.H. *et al.* (2017) Polygenic scores via penalized regression on summary statistics. *Genetic Epidemiol.*, **41**, 469–480.

Privé,F. *et al.* (2018) Efficient analysis of large-scale genome-wide data with two R packages: bigstatsr and bigsnpr. *Bioinformatics*, **34**, 2781–2787.

Ruderfer,D.M. *et al.* (2018) Genomic dissection of bipolar disorder and schizophrenia, including 28 subphenotypes. *Cell*, **173**, 1705–1715.

Shi,H. *et al.* (2016) Contrasting the genetic architecture of 30 complex traits from summary association data. *Am. J. Hum. Genet.*, **99**, 139–153.

Wen,X. *et al.* (2016) Efficient integrative multi-SNP association analysis via deterministic approximation of posteriors. *Am. J. Hum. Genet.*, **98**, 1114–1129.

Wen,X. *et al.* (2017) Integrating molecular qtl data into genome-wide genetic association analysis: probabilistic assessment of enrichment and colocalization. *PLoS Genet.*, **13**, e1006646.

Zhang,K. *et al.* (2002) A dynamic programming algorithm for haplotype block partitioning. *Proc. Natl. Acad. Sci. USA*, **99**, 7335–7339.

Zhou,G., and Zhao,H. (2020). A fast and robust bayesian nonparametric method for prediction of complex traits using summary statistics. bioRxiv.