Efficient management and analysis of large-scale genome-wide data with two R packages: bigstatsr and bigsnpr

Florian Privé [1,*], Hugues Aschard [2,3] and Michael G.B. Blum [1,*]

[1]Université Grenoble Alpes, CNRS, Laboratoire TIMC-IMAG, UMR 5525, France,

[2]Centre de Bioinformatique, Biostatistique et Biologie Intégrative (C3BI), Institut Pasteur, Paris, France

[3]Department of Epidemiology, Harvard T.H. Chan School of Public Health, Boston, Massachusetts, USA.

[*]To whom correspondence should be addressed.

Abstract

Motivation: Genome-wide datasets produced for association studies have dramatically increased in size over the past few years, with modern datasets commonly including millions of variants measured in dozens of thousands of individuals. This increase in data size is a major challenge severely slowing down genomic analyses. Specialized software for every part of the analysis pipeline have been developed to handle large genomic data. However, combining all these software into a single data analysis pipeline might be technically difficult.

Results: Here we present two R packages, bigstatsr and bigsnpr, allowing for management and analysis of large scale genomic data to be performed within a single comprehensive framework. To address large data size, the packages use memory-mapping for accessing data matrices stored on disk instead of in RAM. To perform data pre-processing and data analysis, the packages integrate most of the tools that are commonly used, either through transparent system calls to existing software, or through updated or improved implementation of existing methods. In particular, the packages implement a fast derivation of Principal Component Analysis, functions to remove SNPs in Linkage Disequilibrium, and algorithms to learn Polygenic Risk Scores on millions of SNPs. We illustrate applications of the two R packages by analysing a case-control genomic dataset for the celiac disease, performing an association study and computing Polygenic Risk Scores. Finally, we demonstrate the scalability of the R packages by analyzing a simulated genome-wide dataset including 500,000 individuals and 1 million markers on a single desktop computer.

Availability: https://privefl.github.io/bigstatsr/ and https://privefl.github.io/bigsnpr/

Contact: michael.blum@univ-grenoble-alpes.fr

Supplementary information: Supplementary data are available at *Bioinformatics* online.

1 Introduction

Genome-wide datasets produced for association studies have dramatically increased in size over the past years. A range of software and data formats have been developed to perform essential pre-processing steps and data analysis, often optimizing each of these steps within a dedicated implementation. This diverse and extremely rich software environment has been of tremendous benefit for the genetic community. However, it has two limitations: analysis pipelines become very complex and researchers have limited access to diverse analysis tools due to growing data sizes.

Consider first the basic tools necessary to perform a standard genome-wide analysis. Conversions between standard file formats has become a field by itself with several tools such as VCFtools, BCFtools and PLINK, available either independently or incorporated within large framework (Danecek *et al.* 2011; Li *et al.* 2011; Purcell *et al.* 2007). Similarly, quality control software for genome-wide analysis have been developed such as PLINK and Bioconductor GWASTools (Gogarten *et al.* 2012). Computation of principal components (PCs) of genotypes is commonly performed to account for population stratification in association studies and there are several software available, including EIGENSOFT (SmartPCA and FastPCA) and FlashPCA (Price *et al.* 2006; Galinsky *et al.* 2016; Abraham and Inouye 2014; Abraham *et al.* 2016). Then, implementation of GWAS analyses also depends on the genotype format, e.g. ProbABEL or SNPTEST for dosage data (Aulchenko *et al.* 2010; Marchini and Howie 2010). Finally, there exists a range of tools for Polygenic Risk Scores (PRSs) such as LDpred and PRSice, which provide prediction for quantitative traits or disease risks based on multiple genetic variants (Vilhjálmsson *et al.* 2015; Euesden *et al.* 2015). As a result, one has to make extensive bash/perl/R/python scripts to link these software together and convert between multiple file formats, involving many file manipulations and conversions. Overall, this might be a brake on data exploration.

Secondly, increasing size of genetic datasets is the source of major computational challenges and many analytical tools would be restricted by the amount of memory (RAM) available on computers. This is particularly a burden for commonly used analysis languages such

as R, Python and Perl. Solving the memory issues for these languages would give access to a broad range of tools for data analysis that have been already implemented. Hopefully, strategies have been developed to avoid loading large datasets in RAM. For storing and accessing matrices, memory-mapping is very attractive because it is seamless and usually much faster to use than direct read or write operations. Storing large matrices on disk and accessing them via memory-mapping is available in R through "big.matrix" objects implemented in the R package bigmemory (Kane *et al.* 2013). Thanks to this matrix-like format, algorithms in R/C++ can be developed or adapted for large genotype data.

2 Approach

We developed two R packages, bigstatsr and bigsnpr, that integrate the most efficient algorithms for the pre-processing and analysis of large-scale genomic data while using memory-mapping. Package bigstatsr implements many statistical tools for several types of "big.matrix" objects (raw, char, short, integer, float and double). This includes implementation of multivariate sparse linear models, Principal Component Analysis, matrix operations, and numerical summaries. The statistical tools developed in bigstatsr can be used for other types of data as long as they can be represented as matrices. Package bigsnpr depends on bigstatsr, using a special type of "big.matrix" object to store the genotypes. Package bigsnpr implements algorithms which are specific to the analysis of SNP arrays, such as calls to external software for processing steps, I/O (Input/Output) operations from binary PLINK files, and data analysis operations on SNP data (pruning, testing, plotting).

We use both a real case-control genomic dataset for Celiac disease and large-scale simulated data to illustrate application of the two R packages, including association study and computation of Polygenic Risk Scores. We also compare results from the two R packages with those obtained when using PLINK and EIGENSOFT, and report execution times along with the code to perform major computational tasks.

3 Methods

3.1 Memory-mapped files

The two R packages don't use standard read operations on a file nor load the genotype matrix entirely in memory. They use what we could call an hybrid solution: memory-mapping. Memory-mapping is used to access data, possibly stored on disk, as if it were in memory. This solution is made available within R through an object called "big.matrix", available in R package bigmemory (Kane *et al.* 2013).

We are aware of the software library SNPFile that uses memory-mapped files to store and efficiently access genotype data, coded in C++ (Nielsen and Mailund 2008). With the two packages we developed, we made this solution available in R and in C++ via package Rcpp (Eddelbuettel and François 2011). The major advantage of manipulating genotype data within R, almost as it were a standard matrix in memory, is the possibility of using most of the other tools that have been developed in R (R Core Team 2017). For example, we provide sparse multivariate linear models and an efficient algorithm for Principal Component Analysis (PCA) based on adaptations from R packages biglasso, sparseSVM and RSpectra (Qiu and Mei 2016; Zeng and Breheny 2017).

Usually, memory-mapping provides seamless and faster access than standard read/write operations. When some element is needed, a small chunk of the genotype matrix, containing this element, is accessed in memory. When the system needs more memory, some chunks of the matrix are freed from the memory in order to make space for others. All this is managed by the Operating System so that it is seamless and efficient. It means that if you use the same chunks of data repeatedly, it will be very fast the second time you access it, the third time and so on. Of course, if the memory size of your computer is larger than the size of the dataset, the file could fit entirely in memory and every second access would be fast.

3.2 Data management, preprocessing and imputation

Fast read/write operations from/to bed/bim/fam PLINK files are available so that one should convert data to this format. In bigsnpr, we provide R functions that use system calls to PLINK for the conversion and the Quality Control steps (Figure 1). PLINK files are then read into a "bigSNP" object, which contains the genotype "big.matrix", a data frame with information on samples and another data frame with information on SNPs. We also provide another function which could be used to read from tabular-like text files in order to create a genotype in the format "big.matrix".

We developed a special "big.matrix" object, called "BM.code", that can be used to seamlessly store up to 256 arbitrary different values, while having a relatively efficient storage. Indeed, each element is stored on one byte which requires 8 times less disk storage than double-precision numbers but 4 times more space than the binary PLINK format ".bed". With these 256 values, the matrix can store genotype calls and missing values (4 values), best guess genotypes (3 values) and genotype dosages (likelihoods) rounded to two decimal places (201 values).

Because it is an important part of the preprocessing, we provide two functions for imputing missing values of genotyped SNPs (Figure 2). The first function is a wrapper to PLINK and Beagle which takes bed files as input and return bed files without missing values, and should therefore be used before reading the data in R (Browning and Browning 2008). The second function is a new algorithm we developed in order to have a fast imputation method without losing much of imputation accuracy. This algorithm doesn't use phasing and is very fast, as previous Machine Learning Approaches for genetic imputation (Wang *et al.* 2012). It only relies on some local XGBoost models. XGBoost is an optimized distributed gradient boosting library that can be used in R and provides some of the best results in machine learning competitions (Chen and Guestrin 2016). XGBoost builds decision trees that can detect nonlinear interactions, partially reconstructing phase so that it seems well suited for imputing genotype matrices. For each SNP, we provide an estimation of imputation error by separating non-missing data into training/test sets. The training set is used to build a model for predicting missing data. The prediction model is then evaluated on the test set for which we know the true genotype values, which gives an unbiased estimator of the number of individuals that have been wrongly

imputed for that particular SNP.

3.3 Population structure and SNP thinning based on Linkage Disequilibrium

For computing Principal Components (PCs) of a large-scale genotype matrix, we provide several functions related to SNP thinning and two functions, for computing a partial Singular Value Decomposition (SVD), one based on eigenvalue decomposition, big_SVD, and the other on randomized projections, big_randomSVD (Figure 3).

The function based on randomized projections runs in linear time in all dimensions (Lehoucq and Sorensen 1996). FlashPCA2 and bigstatsr use the same PCA algorithm called Implicitly Restarted Arnoldi Method (IRAM), which is implemented in R package RSpectra. The main difference between the two implementations is that FlashPCA2 computes vector-matrix multiplications with the genotype matrix based on the binary PLINK file whereas bigstatsr computes these multiplications based on the "big.matrix" format, which enables parallel computations.

Fast algorithms for thinning SNPs similar to algorithms provided in PLINK have been developed. For instance, thinning is mandatory when computing PCs of a genotype matrix (Abdellaoui *et al.* 2013). There are at least 3 different options to thin SNPs based on Linkage Disequilibrium. The first option is known as pruning, which is an algorithm that sequentially scan the genome for nearby SNPs in LD, performing pairwise thinning.

A variant of pruning is clumping. Clumping is useful if a statistic is available to sort the SNPs by importance, e.g. association with a phenotype, and for discarding SNPs in LD with a more associated SNP relatively to the phenotype of interest. Furthermore, we advise to always use clumping instead of pruning (by using the minor allele frequency as the statistic of importance, which is the default) because, in some particular cases, pruning can leave regions of the genome without any representative SNP at all[1].

The third option that is generally combined with pruning or clumping consists of removing SNPs in long-range LD regions (Price *et al.* 2008). Long-range LD regions for the human

[1] https://goo.gl/T5SJqM

genome are available as an online table that our packages can use to discard SNPs in long-range LD regions while computing PCs[2]. However, such table is human specific and could also be population specific, so we developed an algorithm that automatically detects these regions and removes them. This algorithm consists in the following steps: first, PCA is performed using a subset of SNP remaining after clumping, then outliers SNPs are detected using Mahalanobis distance as implemented in the R package pcadapt (Luu *et al.* 2017). Finally, the algorithm considers that consecutive outlier SNPs are in long-range LD regions. Indeed, a long-range LD region would cause SNPs in this region to have strong consecutive weights (loadings) in the PCA. This algorithm is implemented in function snp_autoSVD and will be referred by this name in the rest of the paper.

3.4 Association tests and Polygenic Risk Scores

Any test statistic that is based on counts could be easily implemented because we provide fast counting summaries. Among these tests, the Armitage trend test and the MAX3 test statistic are already provided for binary outcome (Zheng *et al.* 2012). Statistical tests based on linear and logistic regressions are also available. For the linear regression, for each SNP j, a t-test is performed on $\beta^{(j)}$ where $\hat{y} = \alpha^{(j)} + \beta^{(j)} SNP^{(j)} + \gamma_1^{(j)} PC_1 + \cdots + \gamma_K^{(j)} PC_K + \delta_1^{(j)} COV_1 + \cdots + \delta_K^{(j)} COV_L$, and K is the number of Principal Components and L is the number of other covariates (such as the age and gender). Similarly, for the logistic regression, for each SNP j, a Z-test is performed on $\beta^{(j)}$ where $\log \frac{\hat{p}}{1-\hat{p}} = \alpha^{(j)} + \beta^{(j)} SNP^{(j)} + \gamma_1^{(j)} PC_1 + \cdots + \gamma_K^{(j)} PC_K + \delta_1^{(j)} COV_1 + \cdots + \delta_K^{(j)} COV_L$, and $\hat{p} = \mathbb{P}(Y = 1)$ and Y denotes the binary phenotype.

The R packages also implement functions to compute Polygenic Risk Scores. First, they implement the widely-used Pruning + Thresholding (P+T) model based on univariate GWAS summary statistics as described in previous equations. Under the P+T model, a coefficient of regression is learned independently for each SNP along with a corresponding p-value. The SNPs are first clumped (P) so that there remains only SNPs that are weakly correlated with each other. Thresholding (T) consists in removing SNPs that are under a certain level of significance (P-value threshold to be determined). Finally, a polygenic risk score is defined as the sum

[2]https://goo.gl/8TngVE

of allele counts of the remaining SNPs weighted by the corresponding regression coefficients (Dudbridge 2013; Chatterjee *et al.* 2013; Golan and Rosset 2014).

Secondly, the two R packages also implement multivariate models to compute risk scores that do not use univariate summary statistics but instead train a model on all the SNPs and covariables at once, optimally accounting for correlation between predictors (Abraham *et al.* 2012). The currently available models are linear and logistic regressions and Support Vector Machine (SVM). These models include lasso and elastic-net regularizations, which reduce the number of predictors (SNPs) included in the predictive models (Tibshirani 1996; Zou and Hastie 2005; Friedman *et al.* 2010). Package bigstatsr provides a fast implementation of these models by using efficient rules to discard most of the predictors (Tibshirani *et al.* 2012). The implementation of these algorithms is based on modified versions of functions available in the R packages sparseSVM and biglasso (Zeng and Breheny 2017). These modifications allow to include covariates in the models and to use these algorithms on the special type of "big.matrix" called "BM.code" used in bigsnpr (see section 3.2).

3.5 Data analyzed

In this paper, two datasets are analyzed: the celiac disease cohort and the POPRES datasets (Dubois *et al.* 2010; Nelson *et al.* 2008). The Celiac dataset is composed of 15,283 individuals of European ancestry genotyped on 295,453 SNPs. The POPRES dataset is composed of 1385 individuals of European ancestry genotyped on 447,245 SNPs.

For comparing computation times of different software, we replicated all the individuals in the Celiac dataset 5 and 10 times in order to have larger datasets while keeping the same population structure and pattern of Linkage Disequilibrium as the original dataset. To assess scalibility of our algorithms for a biobank-scale genotype dataset, we formed another dataset of 500,000 individuals and 1 million SNPs, also by replicating the Celiac dataset.

3.6 Reproducibility

All the code used in this paper along with results, such as execution times and figures, are available as HTML R notebooks in the Supplementary Data.

4 Results

4.1 Overview

We present the results for three different analyses. First, we illustrate the application of R packages bigstatsr and bigsnpr. Secondly, we compare the performance of the R packages to the performance obtained with PLINK and FastPCA (EIGENSOFT). Thirdly, we present results of the two new methods implemented in these packages, one method for the automatic detection and removal of long-range LD regions in PCA and another for the imputation of missing genotypes. We use three types of data: a case-control cohort for the celiac disease, the European population cohort POPRES and simulated datasets using real genotypes from the Celiac cohort. We compare the performance on two computers, a desktop computer with 64GB of RAM and 12 cores, and a laptop with only 8GB of RAM and 4 cores. For the functions that enable parallelism, we use half of the cores available on the corresponding computer.

4.2 Application

We performed an association study and computed a polygenic risk score for the Celiac cohort. The data was preprocessed following steps from figure 1, removing individuals and SNPs which had more than 5% of missing values, non-autosomal SNPs, SNPs with a minor allele frequency lower than 0.05 or a p-value for the Hardy-Weinberg exact test lower than 10^{-10}, and finally, removing one individual in each pair of individuals with a proportion of alleles shared IBD greater than 0.08 (Purcell *et al.* 2007). For the POPRES dataset, this resulted in 1382 individuals and 344,614 SNPs with no missing value. For the Celiac dataset, this resulted in 15,155 individuals and 281,122 SNPs with an overall genotyping rate of 99.96% that was then imputed with the XGBoost method. If we used a standard R matrix to store the genotypes,

this data would take 32GB of memory. On the disk, the ".bed" file takes 1GB and the ".bk" file (storing the "big.matrix") takes 4GB only.

We used bigstatsr and bigsnpr R functions to compute the first Principal Components (PCs) of a genotype matrix and to visualize them (Figure 4). We then performed a Genome-Wide Association Study (GWAS) investigating how Single Nucleotide Polymorphisms (SNPs) are associated with the celiac disease, while accounting for population structure with PCs, and plotted the results as a Manhattan plot (Figure 5). As illustrated in the supplementary data, the whole pipeline is user-friendly and requires only 20 lines of R code.

To illustrate the scalability of the two R packages, we performed a GWAS analysis on 500K individuals and 1M SNPs. The GWAS analysis completed in less than 5 hours using the aforementioned desktop computer. The GWAS analysis was composed of three main steps. First, we removed SNPs in long-range LD regions and used SNP clumping, leaving 93,083 SNPs. Then, the 10 first PCs were computed on the 500K individuals and these remaining SNPs. Finally, on the whole dataset, we made a linear association test for each SNP, using the 10 first PCs as covariables.

4.3 Method Comparison

We first compared the GWAS and PRS computations obtained with the R packages to the ones obtained with PLINK 1.9 and EIGENSOFT 6.1.4. For most functions, multithreading is not available yet in PLINK, nevertheless, PLINK-specific algorithms that use bitwise parallelism (e.g. pruning) are still faster than the parallel algorithms reimplemented in package bigsnpr (Table 1). Overall, the computations with our two R packages for an association study and a polygenic risk score are of the same order of magnitude as when using PLINK and EIGENSOFT (Tables 1 and 2). However, the whole analysis pipeline makes use of R calls only; there is no need to write temporary files and functions have parameters which enable subsetting of the genotype matrix without having to copy it.

On our desktop computer, we compared the computation times of FastPCA, FlashPCA2 to the similar function big_randomSVD implemented in bigstatsr. For each comparison, we used the 93,083 SNPs which were remaining after pruning and we computed 10 PCs. We

used the datasets of growing size simulated from the Celiac dataset. Overall, our function big_randomSVD showed to be twice as fast as FastPCA and FlashPCA2 and almost 10 times as fast when using parallelism with 6 cores (Figure 6). We also compared results in terms of precision by comparing squared correlation between approximated PCs and "true" PCs provided by an exact singular value decomposition obtained with SmartPCA. FastPCA, FlashPCA2 and bigstatsr infer the true first 6 PCs but the squared correlation between true PCs and approximated ones decreases for larger PCs when using FastPCA whereas it remains larger than 0.999 when using FlashPCA2 or bigstatsr (Figure 7).

4.4 Automatic detection of long-range LD regions

For the detection of long-range LD regions during the computation of PCA, we tested the function snp_autoSVD on both the Celiac and POPRES datasets. For the POPRES dataset, the algorithm converged in two iterations. The first iterations found 3 long-range LD regions in chromosomes 2, 6 and 8 (Table S1). We compared the scores (PCs) obtained by this method with the ones obtained by removing pre-determined long-range LD regions[3] and found a mean correlation of 89.6% between PCs, mainly due to a rotation of PC7 and PC8 (Table S3). For the Celiac dataset, we found 5 long-range LD regions (Table S2) and a mean correlation of 98.6% between PCs obtained with snp_autoSVD and the ones obtained by clumping with removing of predetermined long-range LD regions (Table S4).

For the Celiac dataset, we compared results of PCA obtained when using snp_autoSVD and when computing PCA without removing any long range LD region (only clumping at $R^2 > 0.2$). When not removing any long range LD region, we show that PC4 and PC5 corresponds to a long-range LD region in chromosome 8 (Figures S1 and S2). When automatically removing some long-range LD regions with snp_autoSVD, we show that PC4 and PC5 are now only reflecting population structure (Figure S1). Moreover, loadings are more equally distributed among SNPs (Figure S2). This is confirmed by Gini coefficients (measure of dispersion) of each squared loadings that are significantly smaller when computing SVD with snp_autoSVD than when no long-range LD region is removed (Figure S3).

[3] https://goo.gl/8TngVE

4.5 Imputation of missing values for genotyped SNPs

For the fast imputation method based on XGBoost, we compared the imputation accuracy and computation times on the POPRES dataset. The minor allele frequencies (MAFs) are approximately uniformly distributed between 0.05 and 0.5 (Figure S4). We introduced missing values using a Beta-binomial distribution (Figure S5) resulting in approximately 3% of missing values. Imputation was compared between function snp_fastImpute of package bigsnpr and Beagle 4.1 (version of January 21, 2017). Overall, our method made 4.7% of imputation errors whereas Beagle made only 3.1% but it took Beagle 14.6 hours to complete while our method only took 42 minutes (20 times less). For the Celiac dataset, our method took less than 6 hours to complete whereas Beagle didn't finish imputing chromosome 1 in 48 hours. We also show that the estimation of the number of imputation errors is accurate (Figure S6).

5 Discussion

We have developed two R packages, bigstatsr and bigsnpr, which enable multiple analyses of large-scale genotype datasets in a single comprehensive framework. Linkage Disequilibrium pruning, Principal Component Analysis, association tests and computations of polygenic risk scores are made available in this software. Implemented algorithms are both fast and memory-efficient, allowing the use of laptops or desktop computers to make genome-wide analyses. Technically, bigstatsr and bigsnpr could handle any size of datasets. However, if accesses demand the OS to often swap between the file and the memory, this would slow down analysis. For example, the Principal Component Analysis (PCA) algorithm in bigstatsr is iterative so that the matrix has to be sequentially accessed over a hundred times. If the number of samples times the number of SNPs remaining after pruning is larger than the available memory, this slowdown would happen. For instance, a 32GB computer would be slow when computing PCs on more than 100K samples and 300K SNPs remaining after LD thinning.

The two R packages don't use some specific file format nor load the entire matrix in memory but rather use a special type of matrix. Using a matrix-like format makes it easy to develop new functions in order to experiment and develop new ideas. Integration in R makes it possible to

take advantage of all what R has to offer, for example using the widely-used machine learning algorithm XGBoost to easily make a fast and accurate imputation algorithm for genotyped SNPs. Other functions, not presented here, are also available and all the functions available within the package bigstatsr are not specific to SNP arrays, so that they could be used for other omic data or in other fields of research.

We think that the two R packages and the corresponding data format could help researchers to develop new ideas and algorithms to analyze genome-wide data. For example, we wish to use these packages to train much more accurate predictive models than the standard P+T model currently in use when computing PRS. As a second example, multiple imputation has been shown to be a very promising method for increasing statistical power of a GWAS, and it could be implemented with the data format "BM.code", without having to write multiple files (Palmer and Pe'er 2016).

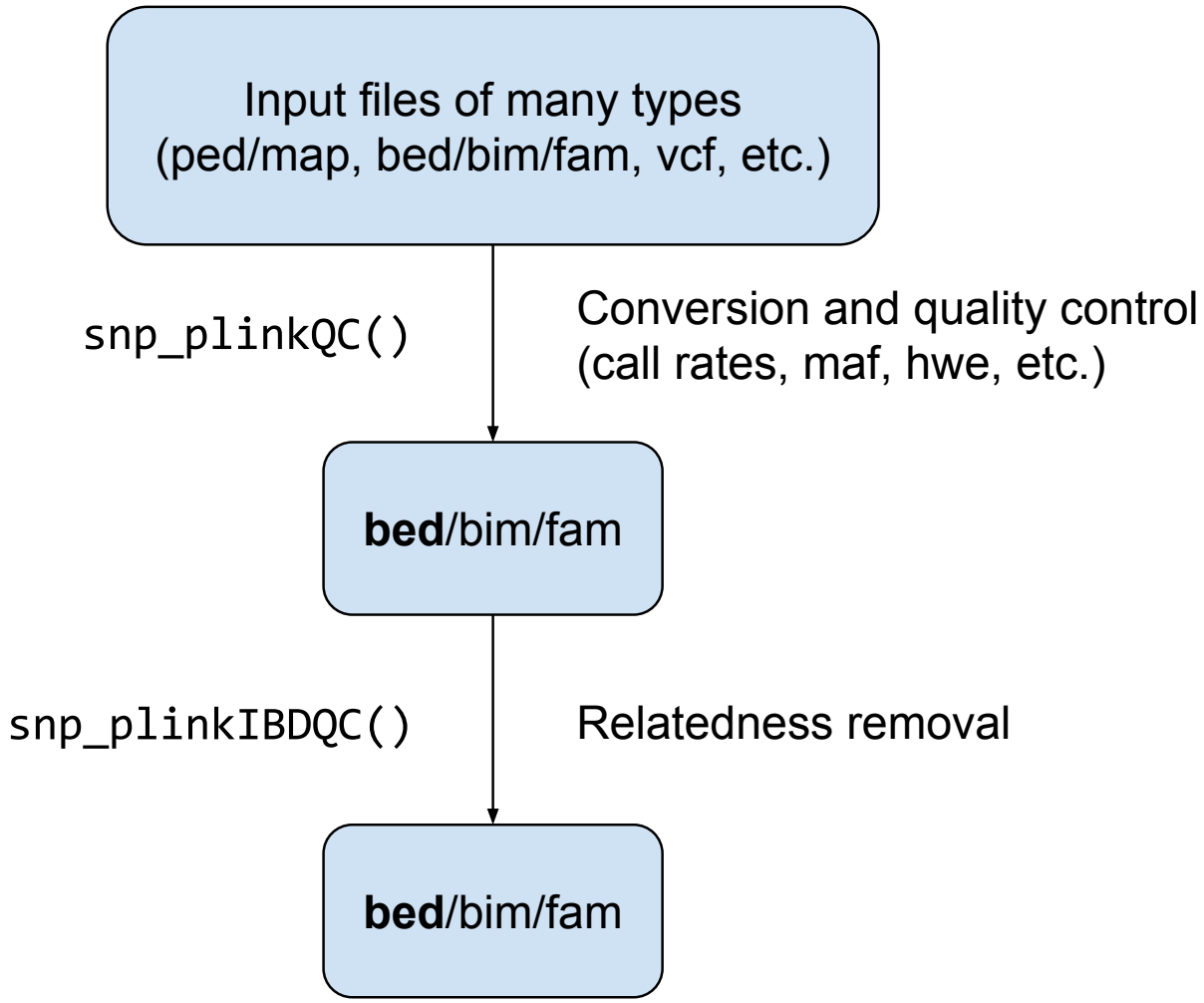

Figure 1: Conversion and Quality Control preprocessing functions available in package bigsnpr via system calls to PLINK.

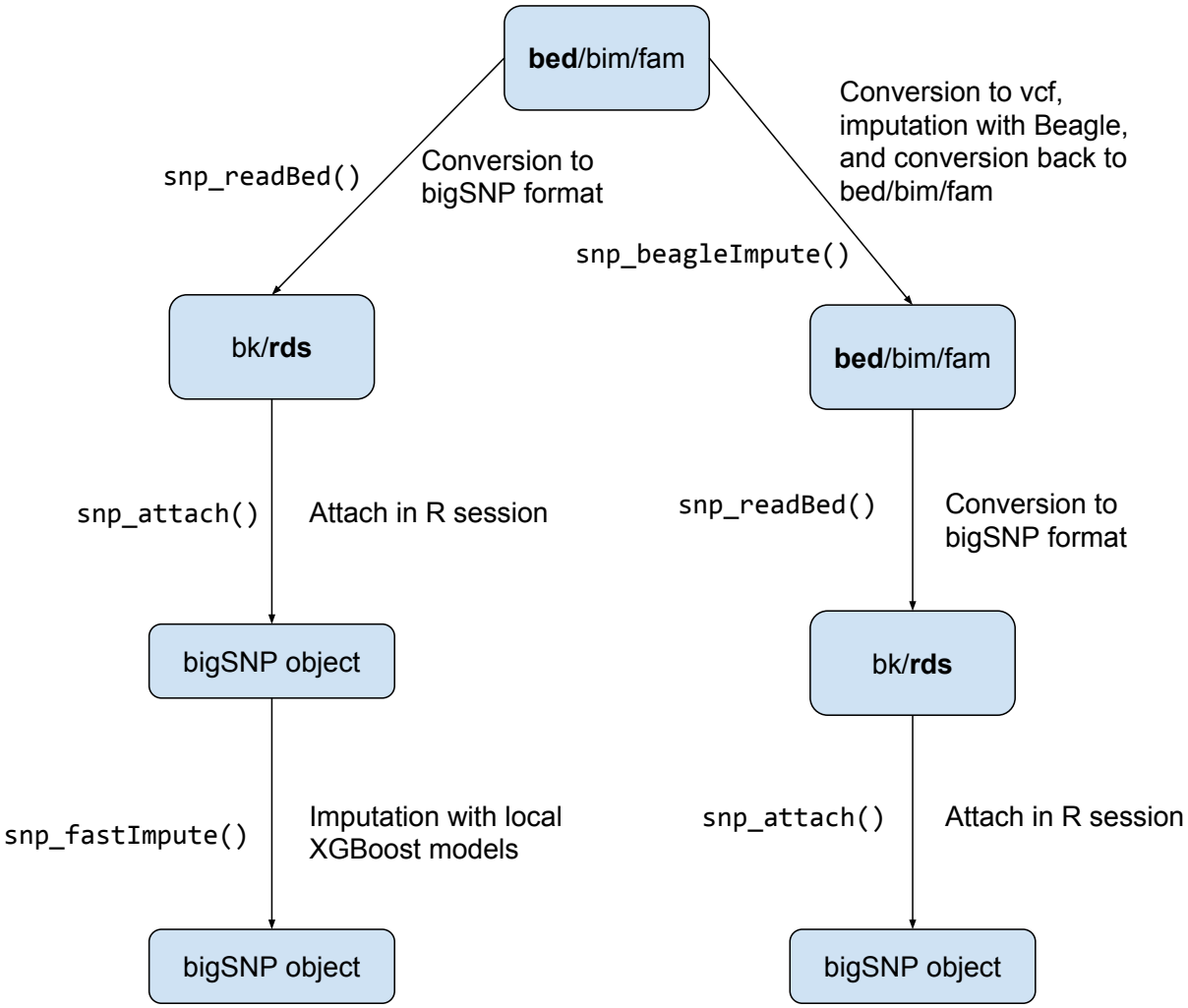

Figure 2: Imputation and reading functions available in package bigsnpr.

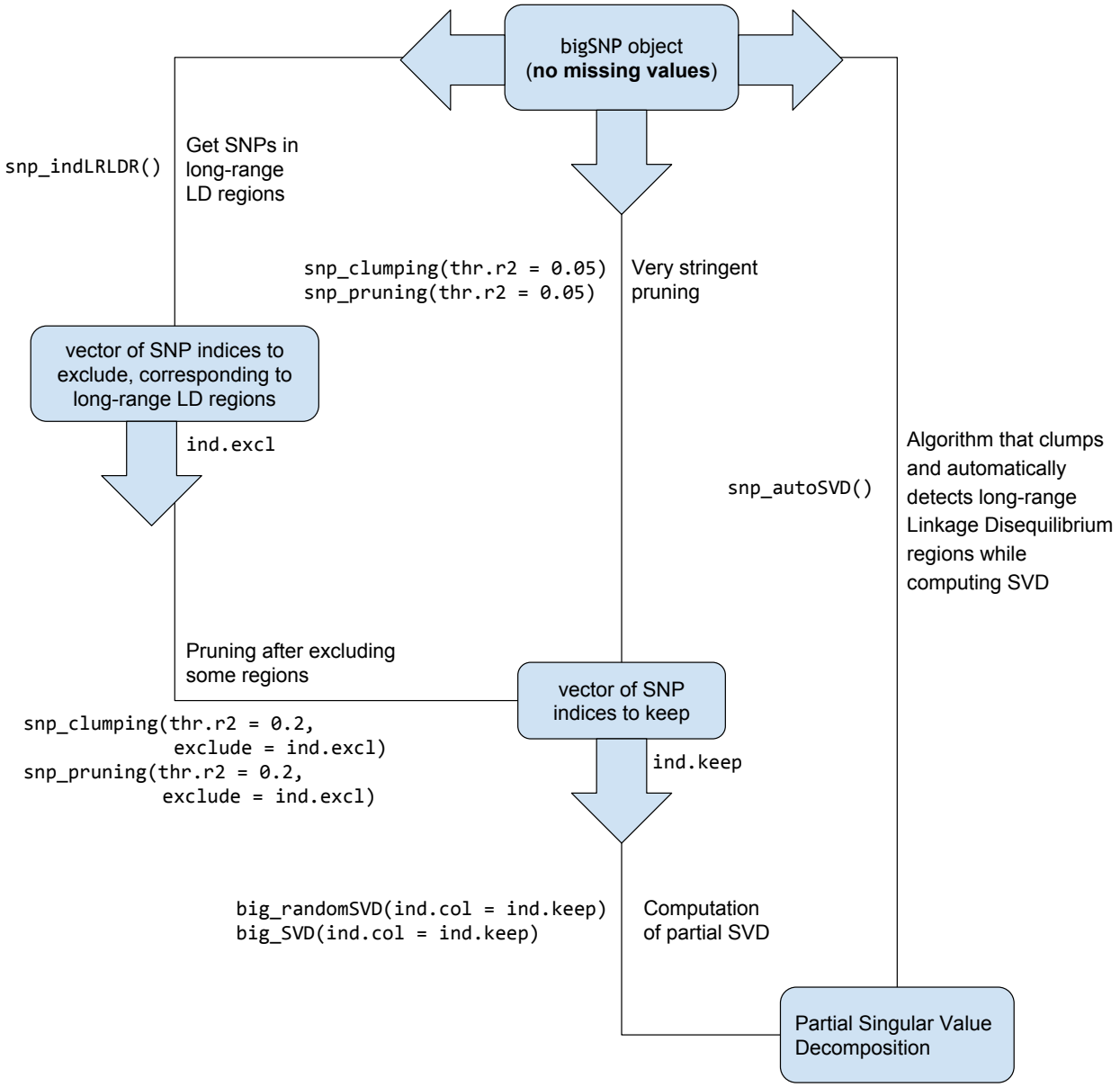

Figure 3: Functions available in packages bigstatsr and bigsnpr for the computation of a partial Singular Value Decomposition of a genotype array, with 3 different methods for thinning SNPs.

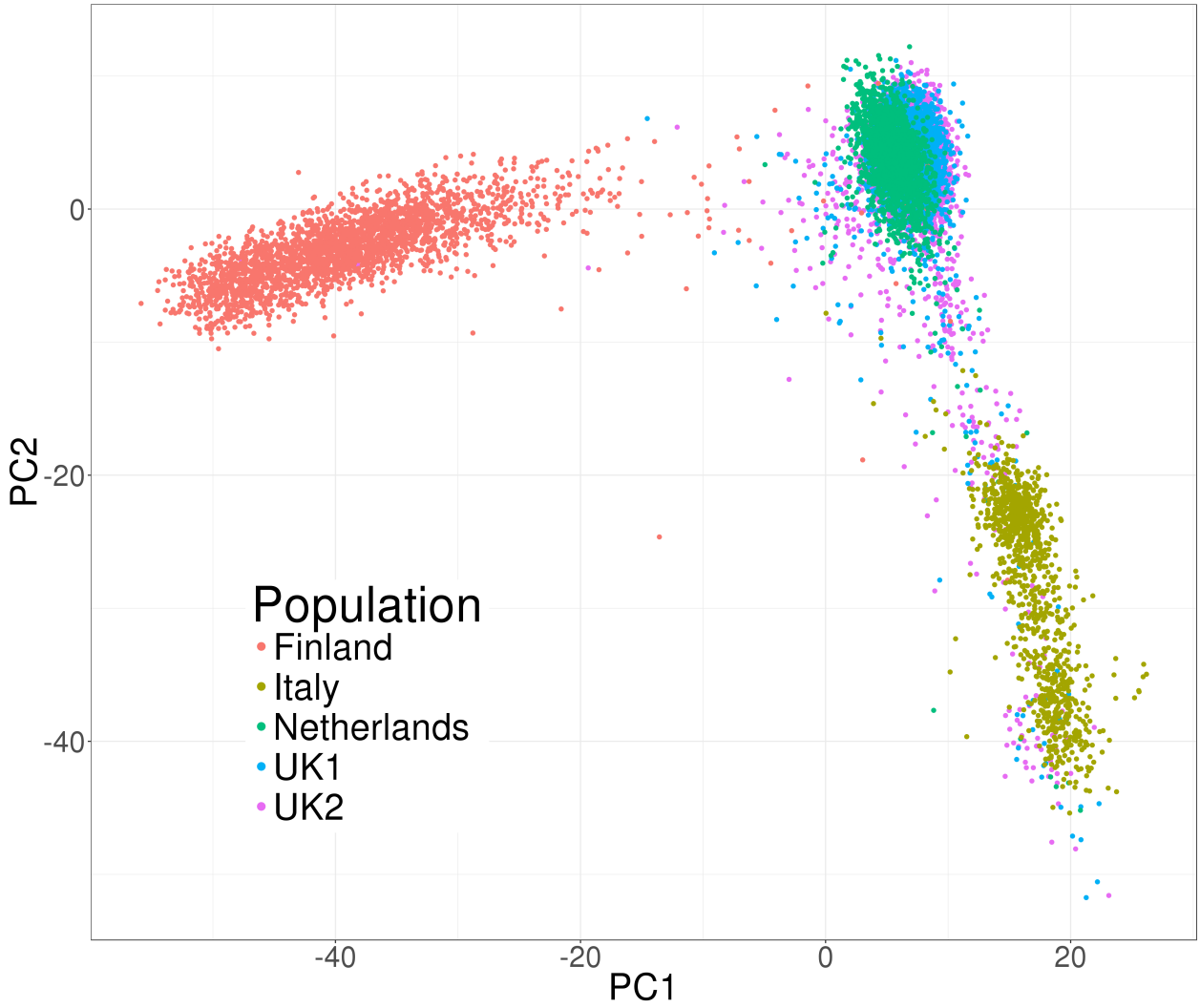

Figure 4: Principal Components of the celiac cohort genotype matrix produced by package bigstatsr.

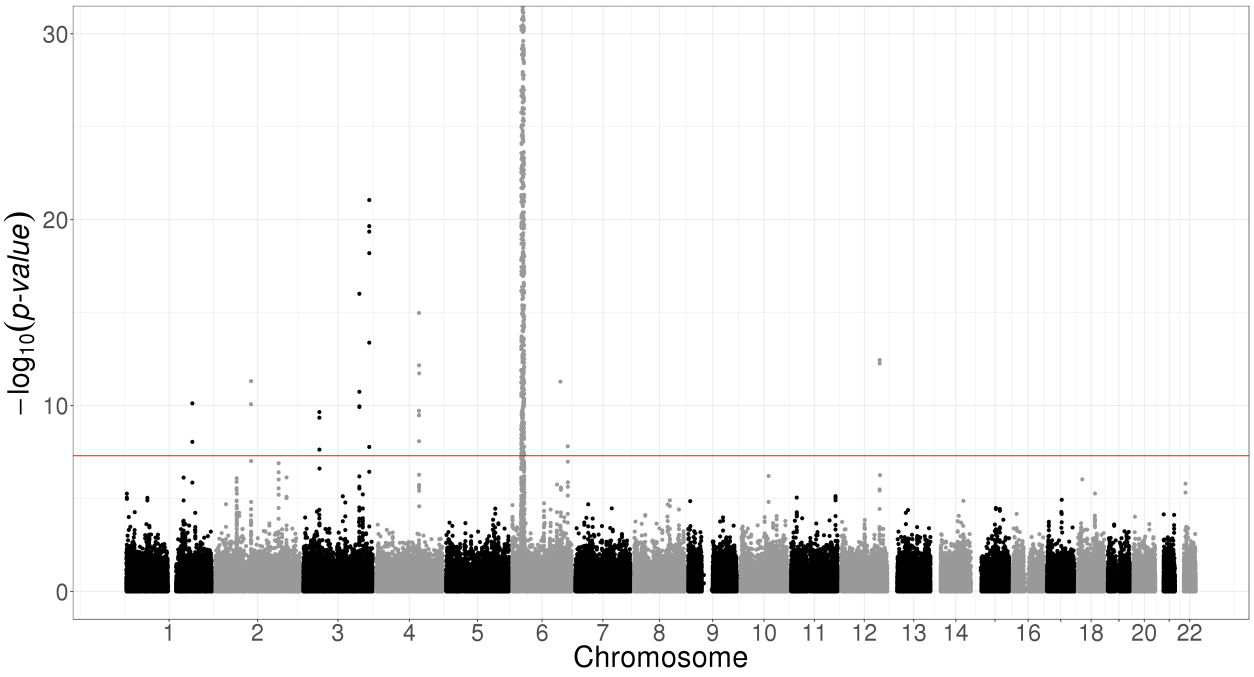

Figure 5: Manhattan plot of the celiac disease cohort produced by package bigsnpr. Some SNPs in chromosome 6 have p-values smaller than the 10^{-10} threshold used for vizualisation purposes.

Operation	Execution times	
	PLINK and FastPCA	bigstatsr and bigsnpr
Reading PLINK files	n/a	5 / 20 sec
Pruning	4 / 6 sec	13 / 46 sec
Computing 10 PCs	6 / 7 min	45 / 136 sec
GWAS (binary phenotype)	5 min	5 / 14.5 min
Total	11 / 12 min	6 / 18 min

Table 1: Execution times with bigstatsr and bigsnpr compared to PLINK and FastPCA for making a GWAS for the Celiac dataset. The first time is with a desktop computer and the second time is with a laptop computer.

Operation	Execution times	
	PLINK	bigstatsr and bigsnpr
GWAS (binary phenotype)	4 / min	3 / min
Clumping	49 / sec	9 / sec
PRS	9 / sec	4 / sec
Total	5 / min	3 / min

Table 2: Execution times with bigstatsr and bigsnpr compared to PLINK and FastPCA for making a PRS for the Celiac dataset. The first time is with a desktop computer and the second time is with a laptop computer.

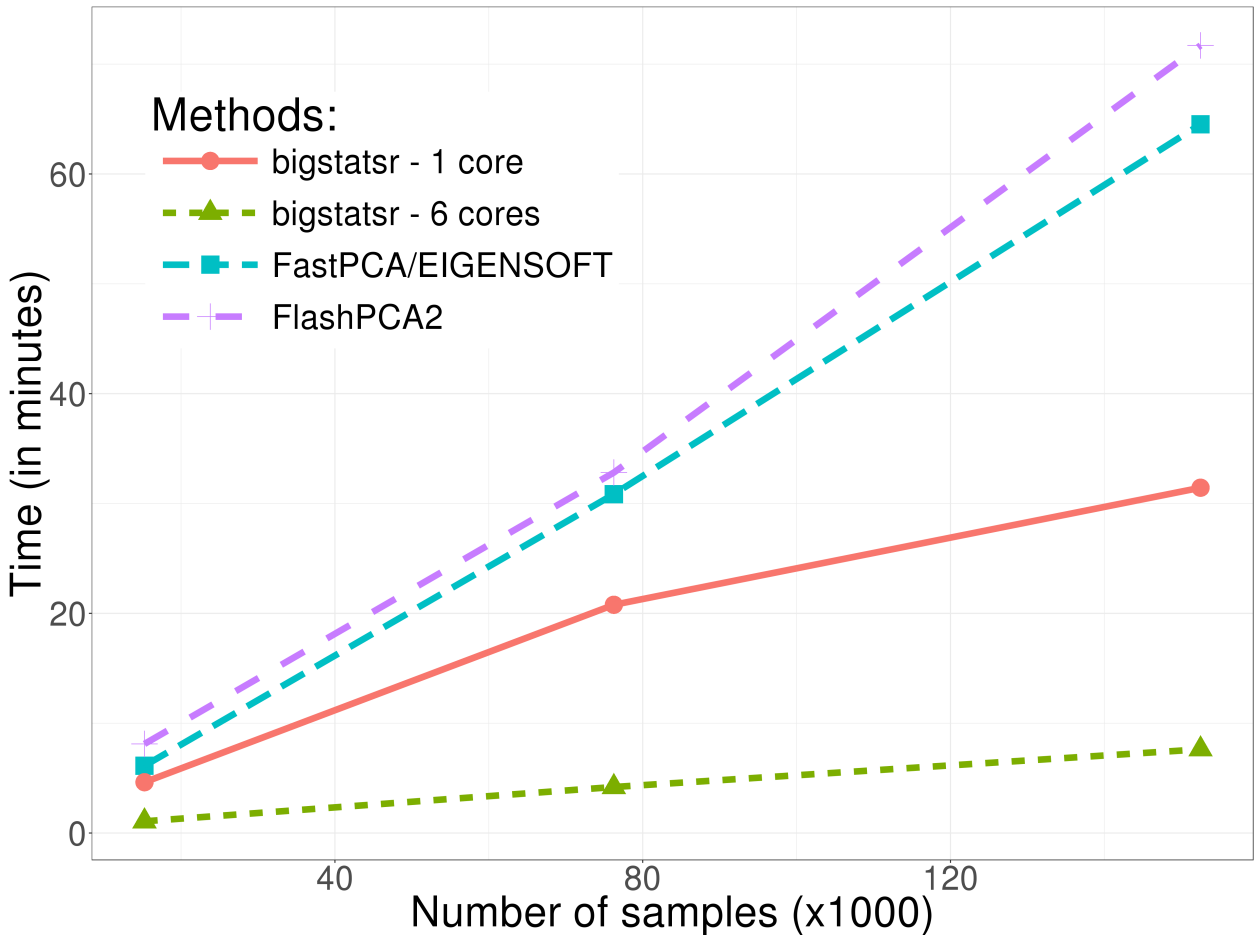

Figure 6: Benchmark comparisons between randomized Partial Singular Value Decomposition available in FlashPCA2, FastPCA (fast mode of SmartPCA/EIGENSOFT) and package bigstatsr. It shows the computation time in minutes as a function of the number of samples. The computation corresponds to 10 Principal Components and 93,083 SNPs which remain after thinning.

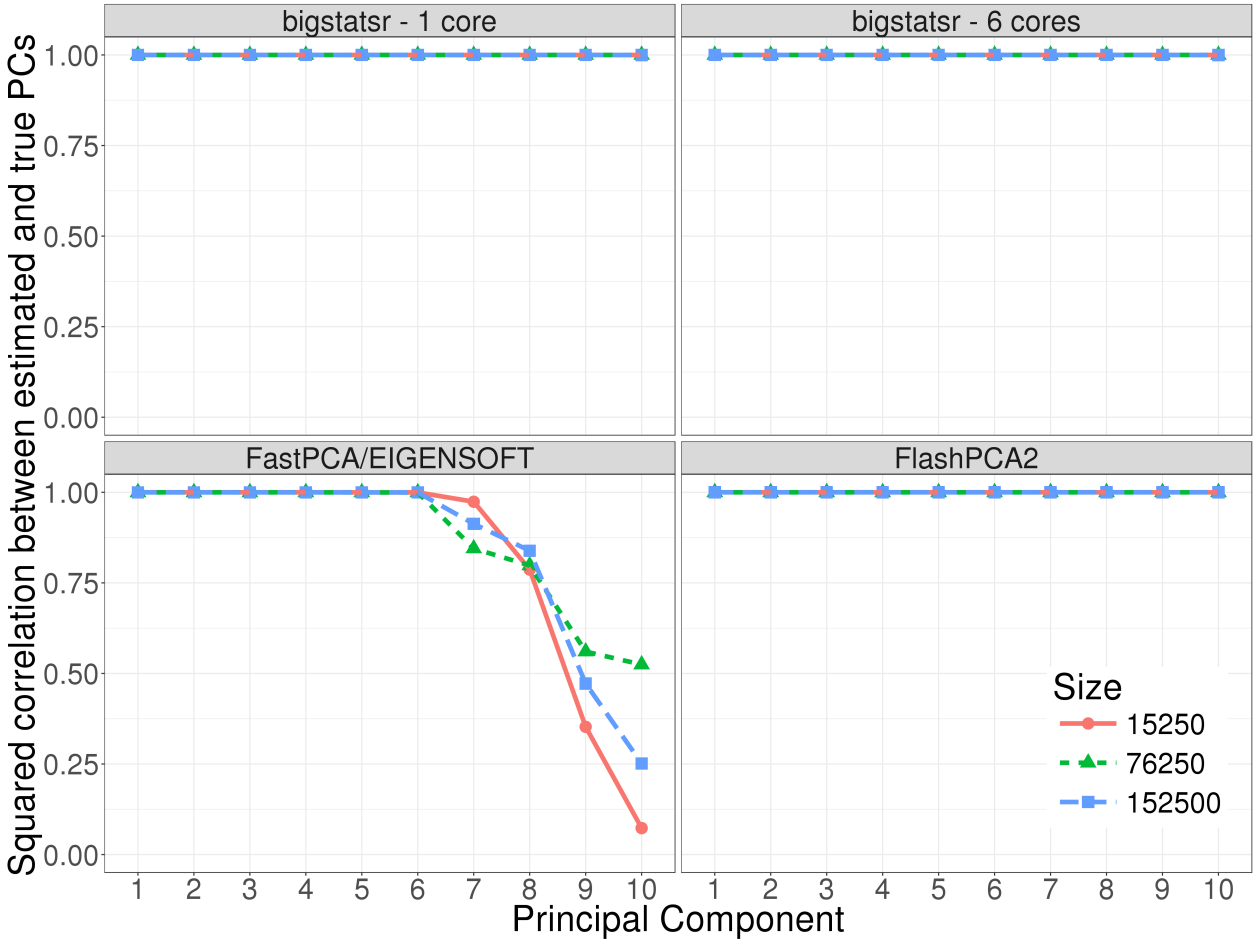

Figure 7: Precision comparisons between randomized Partial Singular Value Decomposition available in FlashPCA2, FastPCA (fast mode of SmartPCA/EIGENSOFT) and package bigstatsr. It shows the squared correlation between approximated PCs and "true" PCs (given by the slow mode of SmartPCA) of the Celiac dataset (whose individuals have been repeated 1, 5 and 10 times).

Acknowledgements

Authors acknowledge Grenoble Alpes Data Institute, supported by the French National Research Agency under the "Investissements d?avenir" program (ANR-15-IDEX-02) and the LabEx PERSYVAL-Lab (ANR-11-LABX-0025-01).

References

Abdellaoui, A., Hottenga, J.-J., De Knijff, P., Nivard, M. G., Xiao, X., Scheet, P., Brooks, A., Ehli, E. A., Hu, Y., Davies, G. E., Hudziak, J. J., Sullivan, P. F., Van Beijsterveldt, T., Willemsen, G., De Geus, E. J., Penninx, B. W., and Boomsma, D. I. (2013). Population structure, migration, and diversifying selection in the Netherlands. *European Journal of Human Genetics*, **21**(10), 1277–1285.

Abraham, G. and Inouye, M. (2014). Fast principal component analysis of large-scale genome-wide data. *PLoS ONE*, **9**(4), e93766.

Abraham, G., Kowalczyk, A., Zobel, J., and Inouye, M. (2012). SparSNP: Fast and memory-efficient analysis of all SNPs for phenotype prediction. *BMC Bioinformatics*, **13**(1), 88.

Abraham, G., Qiu, Y., and Inouye, M. (2016). FlashPCA2 : principal component analysis of biobank-scale genotype datasets. *bioRxiv*, **12**, 2014–2017.

Aulchenko, Y. S., Struchalin, M. V., and van Duijn, C. M. (2010). ProbABEL package for genome-wide association analysis of imputed data. *BMC Bioinformatics*, **11**(1), 134.

Browning, B. L. and Browning, S. R. (2008). A unified approach to genotype imputation and haplotype-phase inference for large data sets of trios and unrelated individuals. *American Journal of Human Genetics*, **84**(2), 210–223.

Chatterjee, N., Wheeler, B., Sampson, J., Hartge, P., Chanock, S. J., and Park, J.-H. (2013). Projecting the performance of risk prediction based on polygenic analyses of genome-wide association studies. *Nature genetics*, **45**(4), 400–5, 405e1–3.

Chen, T. and Guestrin, C. (2016). XGBoost : Reliable Large-scale Tree Boosting System. *arXiv*, pages 1–6.

Danecek, P., Auton, A., Abecasis, G., Albers, C. A., Banks, E., DePristo, M. A., Handsaker, R. E., Lunter, G., Marth, G. T., Sherry, S. T., McVean, G., and Durbin, R. (2011). The variant call format and VCFtools. *Bioinformatics*, **27**(15), 2156–2158.

Dubois, P. C. A., Trynka, G., Franke, L., Hunt, K. A., Romanos, J., Curtotti, A., Zhernakova, A., Heap, G. A. R., Adány, R., Aromaa, A., Bardella, M. T., van den Berg, L. H., Bockett, N. A., de la Concha, E. G., Dema, B., Fehrmann, R. S. N., Fernández-Arquero, M., Fiatal, S., Grandone, E., Green, P. M., Groen, H. J. M., Gwilliam, R., Houwen, R. H. J., Hunt, S. E., Kaukinen, K., Kelleher, D., Korponay-Szabo, I., Kurppa, K., MacMathuna, P., Mäki, M., Mazzilli, M. C., McCann, O. T., Mearin, M. L., Mein, C. A., Mirza, M. M., Mistry, V., Mora, B., Morley, K. I., Mulder, C. J., Murray, J. A., Núñez, C., Oosterom, E., Ophoff, R. A., Polanco, I., Peltonen, L., Platteel, M., Rybak, A., Salomaa, V., Schweizer, J. J., Sperandeo, M. P., Tack, G. J., Turner, G., Veldink, J. H., Verbeek, W. H. M., Weersma, R. K., Wolters, V. M., Urcelay, E., Cukrowska, B., Greco, L., Neuhausen, S. L., McManus, R., Barisani, D., Deloukas, P., Barrett, J. C., Saavalainen, P., Wijmenga, C., and van Heel, D. A. (2010). Multiple common variants for celiac disease influencing immune gene expression. *Nature genetics*, **42**(4), 295–302.

Dudbridge, F. (2013). Power and Predictive Accuracy of Polygenic Risk Scores. *PLoS Genetics*, **9**(3).

Eddelbuettel, D. and François, R. (2011). Rcpp : Seamless R and C ++ Integration. *Journal Of Statistical Software*, **40**, 1–18.

Euesden, J., Lewis, C. M., and O'Reilly, P. F. (2015). PRSice: Polygenic Risk Score software. *Bioinformatics*, **31**(9), 1466–1468.

Friedman, J., Hastie, T., and Tibshirani, R. (2010). Regularization Paths for Generalized Linear Models via Coordinate Descent. *Journal of Statistical Software*, **33**(1), 1–22.

Galinsky, K. J., Bhatia, G., Loh, P. R., Georgiev, S., Mukherjee, S., Patterson, N. J., and Price, A. L. (2016). Fast Principal-Component Analysis Reveals Convergent Evolution of ADH1B in Europe and East Asia. *American Journal of Human Genetics*, **98**(3), 456–472.

Gogarten, S. M., Bhangale, T., Conomos, M. P., Laurie, C. A., McHugh, C. P., Painter, I., Zheng, X., Crosslin, D. R., Levine, D., Lumley, T., Nelson, S. C., Rice, K., Shen, J., Swarnkar, R., Weir, B. S., and Laurie, C. C. (2012). GWASTools: An R/Bioconductor package for quality control and analysis of genome-wide association studies. *Bioinformatics*, **28**(24), 3329–3331.

Golan, D. and Rosset, S. (2014). Effective genetic-risk prediction using mixed models. *American Journal of Human Genetics*, **95**(4), 383–393.

Kane, M. J., Emerson, J. W., and Weston, S. (2013). Scalable Strategies for Computing with Massive Data. *Journal of Statistical Software*, **55**(14), 1–19.

Lehoucq, R. B. and Sorensen, D. C. (1996). Deflation Techniques for an Implicitly Restarted Arnoldi Iteration. *SIAM Journal on Matrix Analysis and Applications*, **17**(4), 789–821.

Li, L., Rakitsch, B., and Borgwardt, K. (2011). ccSVM: correcting Support Vector Machines for confounding factors in biological data classification. *Bioinformatics (Oxford, England)*, **27**(13), i342–8.

Luu, K., Bazin, E., and Blum, M. G. B. (2017). pcadapt: an R package to perform genome scans for selection based on principal component analysis. In *Molecular Ecology Resources*, volume 17, pages 67–77.

Marchini, J. and Howie, B. (2010). Genotype imputation for genome-wide association studies. *Nature Reviews. Genetics*, **11**(7), 499–511.

Nelson, M. R., Bryc, K., King, K. S., Indap, A., Boyko, A. R., Novembre, J., Briley, L. P., Maruyama, Y., Waterworth, D. M., Waeber, G., Vollenweider, P., Oksenberg, J. R., Hauser, S. L., Stirnadel, H. A., Kooner, J. S., Chambers, J. C., Jones, B., Mooser, V., Bustamante, C. D., Roses, A. D., Burns, D. K., Ehm, M. G., and Lai, E. H. (2008). The Population Reference Sample, POPRES: A Resource for Population, Disease, and Pharmacological Genetics Research. *American Journal of Human Genetics*, **83**(3), 347–358.

Nielsen, J. and Mailund, T. (2008). SNPFile–a software library and file format for large scale association mapping and population genetics studies. *BMC bioinformatics*, **9**(1), 526.

Palmer, C. and Pe'er, I. (2016). Bias Characterization in Probabilistic Genotype Data and Improved Signal Detection with Multiple Imputation. *PLoS Genetics*, **12**(6), e1006091.

Price, A. L., Patterson, N. J., Plenge, R. M., Weinblatt, M. E., Shadick, N. A., and Reich, D. (2006). Principal components analysis corrects for stratification in genome-wide association studies. *Nature genetics*, **38**(8), 904–9.

Price, A. L., Weale, M. E., Patterson, N., Myers, S. R., Need, A. C., Shianna, K. V., Ge, D., Rotter, J. I., Torres, E., Taylor, K. D. D., Goldstein, D. B., and Reich, D. (2008). Long-Range LD Can Confound Genome Scans in Admixed Populations.

Purcell, S., Neale, B., Todd-Brown, K., Thomas, L., Ferreira, M. A. R., Bender, D., Maller, J., Sklar, P., de Bakker, P. I. W., Daly, M. J., and Sham, P. C. (2007). PLINK: a tool set for whole-genome association and population-based linkage analyses. *American journal of human genetics*, **81**(3), 559–75.

Qiu, Y. and Mei, J. (2016). *RSpectra: Solvers for Large Scale Eigenvalue and SVD Problems*. R package version 0.12-0.

R Core Team (2017). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.

Tibshirani, R. (1996). Regression Selection and Shrinkage via the Lasso.

Tibshirani, R., Bien, J., Friedman, J., Hastie, T., Simon, N., Taylor, J., and 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.

Vilhjálmsson, B. J., Yang, J., Finucane, H. K., Gusev, A., Lindström, S., Ripke, S., Genovese, G., Loh, P.-R., Bhatia, G., Do, R., Hayeck, T., Won, H.-H., Kathiresan, S., Pato, M., Pato, C., Tamimi, R., Stahl, E., Zaitlen, N., Pasaniuc, B., Belbin, G., Kenny, E. E., Schierup, M. H., De Jager, P., Patsopoulos, N. A., McCarroll, S., Daly, M., Purcell, S., Chasman, D., Neale, B., Goddard, M., Visscher, P. M., Kraft, P., Patterson, N., and Price, A. L. (2015). Modeling Linkage Disequilibrium Increases Accuracy of Polygenic Risk Scores. *The American Journal of Human Genetics*, **97**(4), 576–592.

412 Wang, Y., Cai, Z., Stothard, P., Moore, S., Goebel, R., Wang, L., and Lin, G. (2012). Fast accurate missing SNP genotype local imputation. *BMC*
413 *research notes*, **5**(1), 404.

414 Zeng, Y. and Breheny, P. (2017). The biglasso Package: A Memory- and Computation-Efficient Solver for Lasso Model Fitting with Big Data in R.

415 Zheng, G., Yang, Y., Zhu, X., and Elston, R. C. (2012). *Analysis of Genetic Association Studies*. Statistics for Biology and Health. Springer US,
416 Boston, MA.

417 Zou, H. and Hastie, T. (2005). Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society. Series B: Statistical*
418 *Methodology*, **67**(2), 301–320.

Supplementary Data

5.1 Long-range LD regions

	Chromosome	Start (Mb)	Stop (Mb)
1	2	134.7 (134.5)	137.3 (138)
2	6	27.5 (25.5)	33.1 (33.5)
3	8	6.6 (8)	13.2 (12)

Table S1: Regions found by snp_autoSVD for the POPRES dataset. In parentheses are regions referenced in (Price *et al.* 2008).

	Chromosome	Start (Mb)	Stop (Mb)
1	2	134.4 (134.5)	138.1 (138)
2	6	23.8 (25.5)	35.8 (33.5)
3	8	6.3 (8)	13.5 (12)
4	3	163.1 (n/a)	164.9 (n/a)
5	14	46.6 (n/a)	47.5 (n/a)

Table S2: Regions found by snp_autoSVD for the celiac dataset. In parentheses are regions referenced in (Price *et al.* 2008).

	PC1	PC2	PC3	PC4	PC5	PC6	PC7	PC8	PC9	PC10
PC1	100.0	-0.1	-0.0	0.1	-0.1	0.0	0.0	0.0	-0.0	-0.0
PC2	0.1	100.0	-0.0	0.1	-0.0	-0.0	-0.0	0.2	-0.1	-0.0
PC3	0.0	-0.0	99.9	0.9	0.1	-0.1	-0.3	0.2	0.4	0.1
PC4	-0.1	-0.1	-0.9	99.7	-1.0	0.7	0.6	0.2	0.3	0.9
PC5	0.1	0.0	-0.1	1.1	99.3	1.3	-0.8	1.3	-4.2	-2.4
PC6	-0.0	0.0	0.1	-0.7	-1.0	97.7	-3.5	6.1	7.9	-6.2
PC7	-0.0	-0.1	0.2	-0.3	-1.7	0.3	58.3	73.2	-25.9	9.1
PC8	0.1	-0.1	-0.3	0.4	-0.5	-5.3	-73.5	59.5	15.8	13.2
PC9	0.0	0.1	-0.4	-0.8	5.0	-7.6	27.8	11.0	91.9	9.0
PC10	0.1	0.0	0.0	-0.9	1.6	10.2	3.9	-19.6	-6.3	89.2

Table S3: Correlation between scores of PCA for the POPRES dataset when automatically removing long-range LD regions and when removing them based on a predefined table.

	PC1	PC2	PC3	PC4	PC5	PC6	PC7	PC8	PC9	PC10
PC1	100.0	-0.1	-0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0
PC2	0.1	100.0	0.0	0.0	-0.0	-0.0	0.0	-0.0	-0.0	-0.0
PC3	0.1	-0.0	99.9	0.2	-0.0	0.1	0.1	0.1	0.0	-0.1
PC4	-0.0	-0.0	-0.3	99.9	-0.1	0.1	-0.1	0.0	0.1	0.1
PC5	0.0	0.0	0.0	0.1	99.7	0.9	-0.3	0.1	-0.8	-0.6
PC6	-0.0	0.0	-0.1	-0.2	-0.8	99.6	0.5	-0.5	-0.2	-0.4
PC7	-0.0	0.0	-0.1	0.0	0.5	-0.4	98.9	3.1	0.7	1.6
PC8	0.0	0.0	-0.2	-0.0	-0.2	0.5	-3.2	98.4	-4.5	-1.5
PC9	-0.0	-0.0	-0.0	0.0	0.6	0.1	-0.7	4.6	96.9	-10.7
PC10	-0.0	-0.0	0.1	-0.1	0.3	0.1	-1.2	1.5	8.6	92.7

Table S4: Correlation between scores of PCA for the Celiac dataset when automatically removing long-range LD regions and when removing them based on a predefined table.

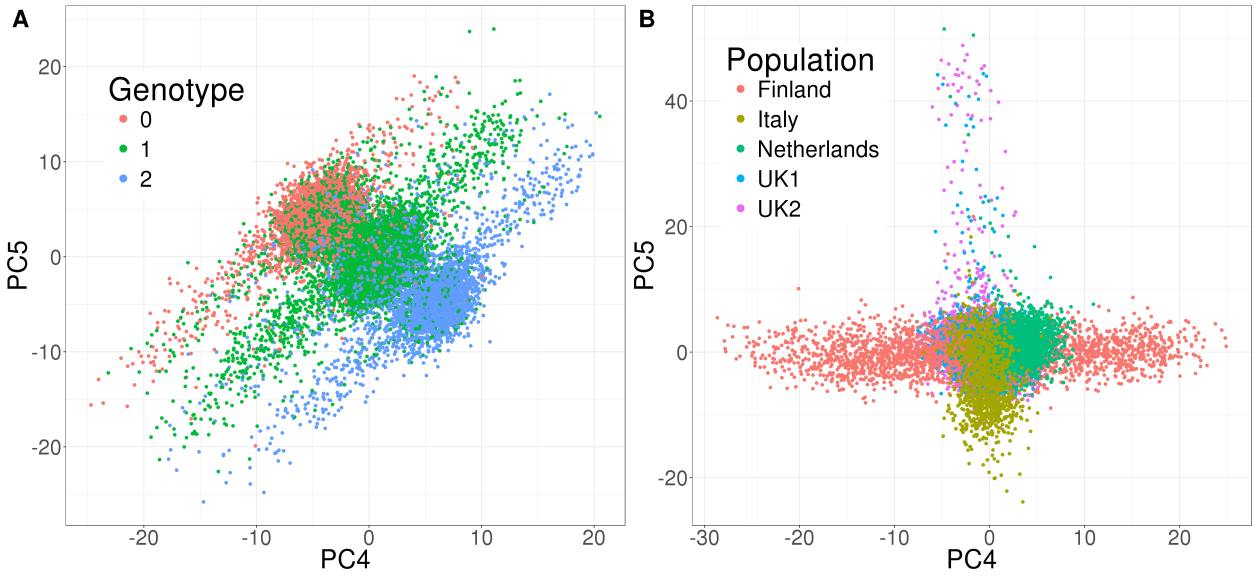

Figure S1: PC4 and PC5 of the celiac disease dataset. Left panel, PC scores obtained without removing any long range LD region (only clumping at $R^2 > 0.2$). Individuals are coloured according to their genotype at the SNP that has the highest loading for PC4. Right panel, PC scores obtained with the automatic detection and removal of long-range LD regions. Individuals are coloured according to their population of origin.

5.2 Imputation

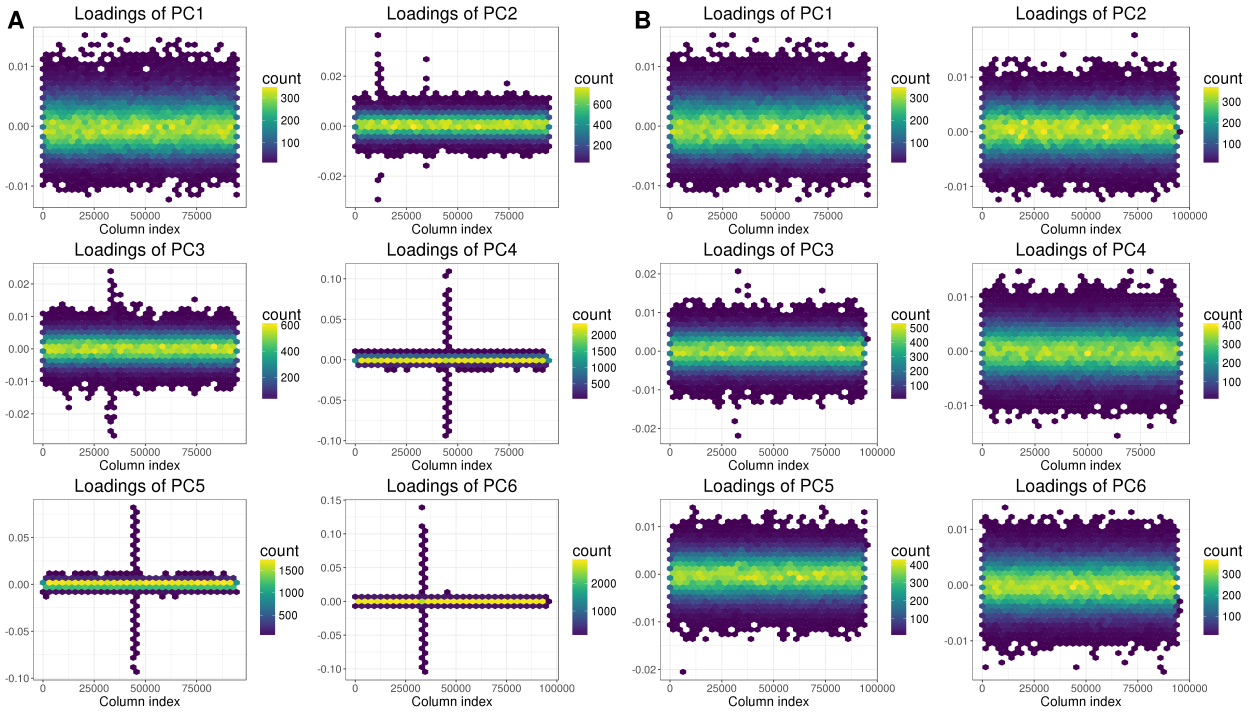

Figure S2: Loadings of first 6 PCs of the celiac disease dataset plotted as hexbins (2-D histogram with hexagonal cells). On the left, without removing any long range LD region (only clumping at $R^2 > 0.2$). On the right, with the automatic detection and removal of long-range LD regions.

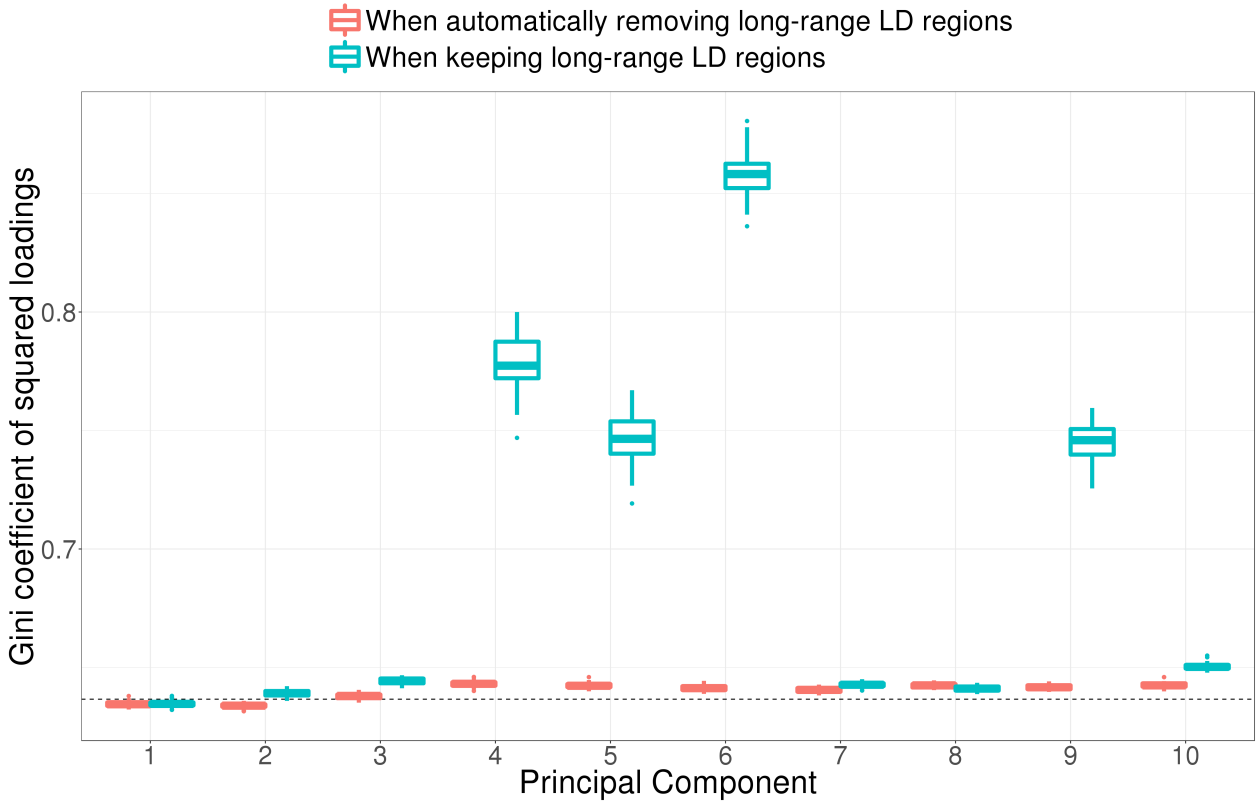

Figure S3: Boxplots of 1000 bootstrapped Gini coefficients (measure of statistical dispersion) of squared loadings without removing any long range LD region (only clumping at $R^2 > 0.2$) and with the automatic detection and removal of long-range LD regions. The dashed line corresponds to the theoretical value for gaussian loadings.

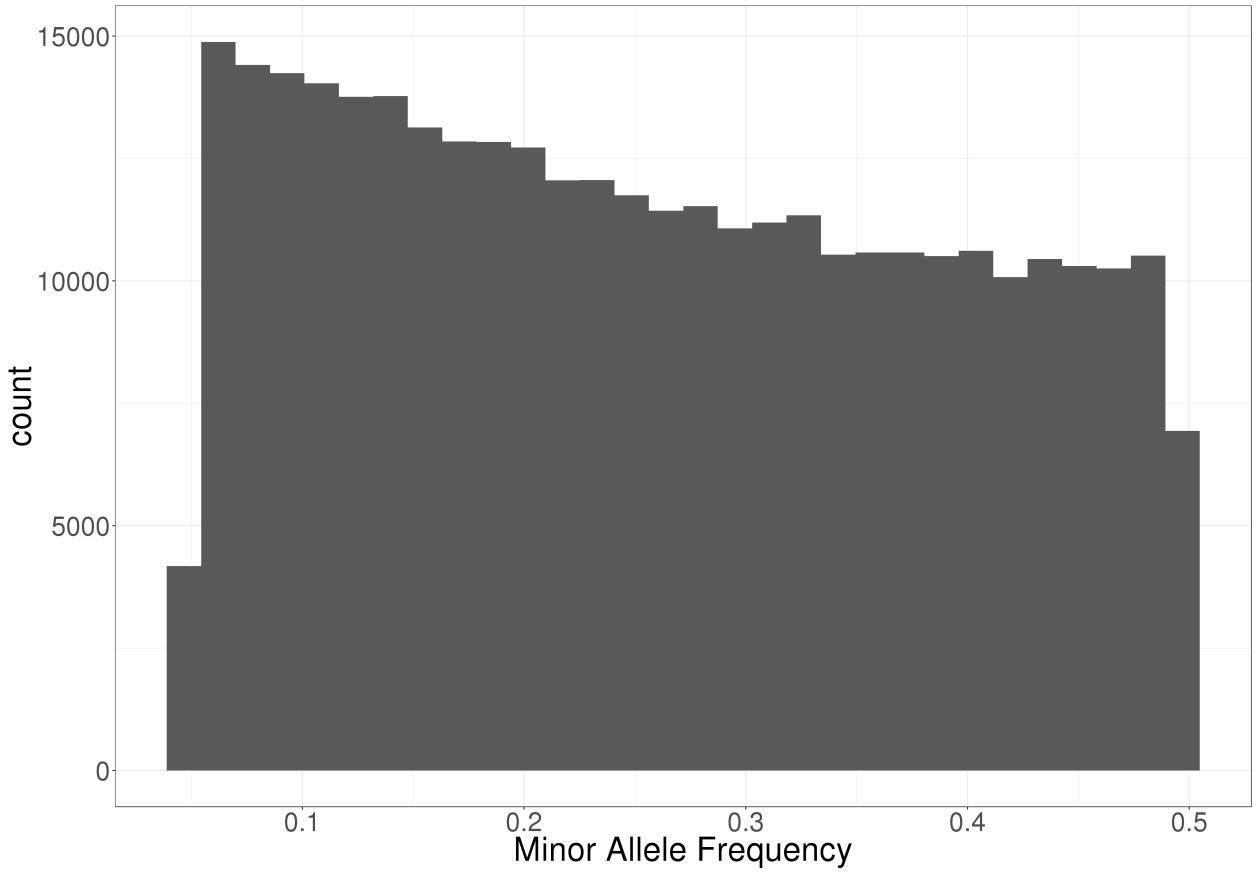

Figure S4: Histogram of the minor allele frequencies of the POPRES dataset used for comparing imputation methods.

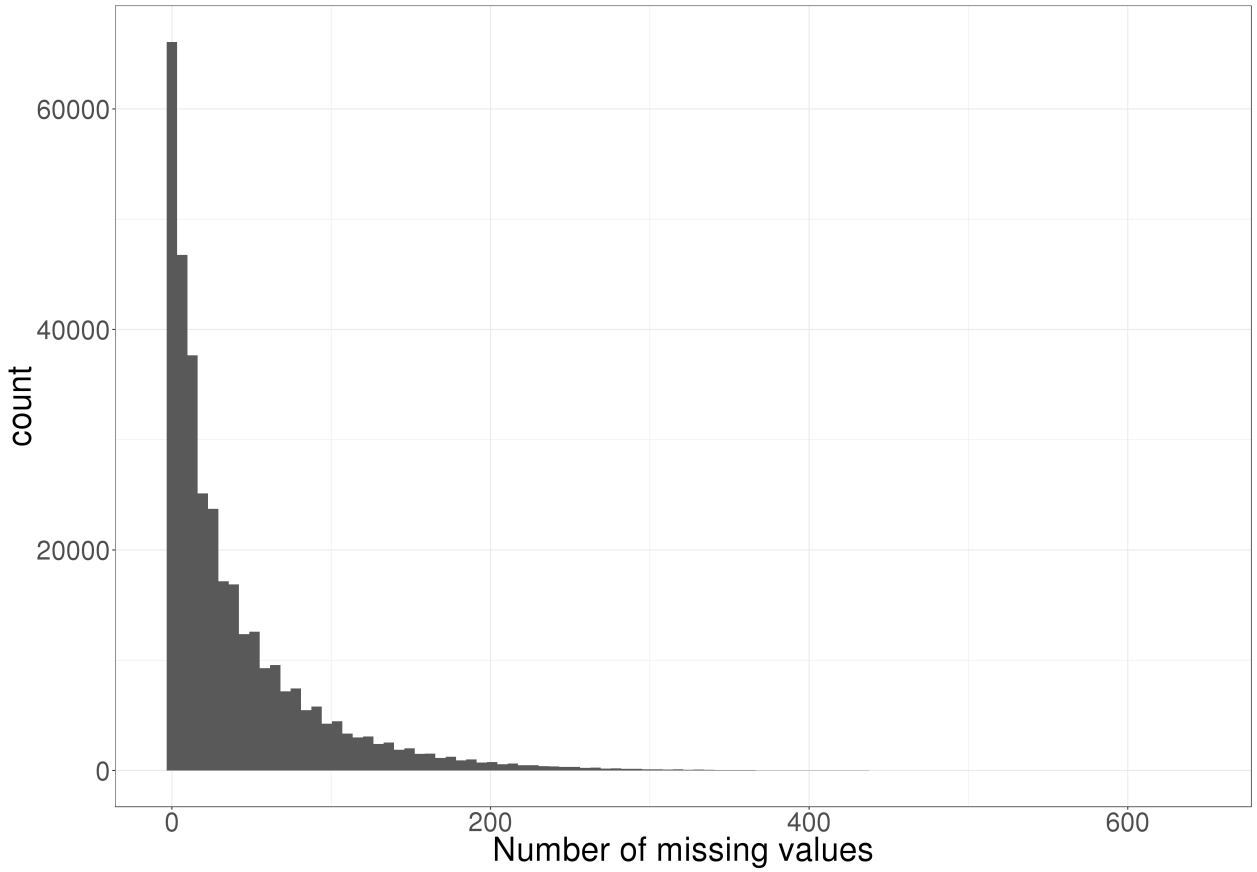

Figure S5: Histogram of the number of missing values by SNP. These numbers were generated using a Beta-binomial distribution.

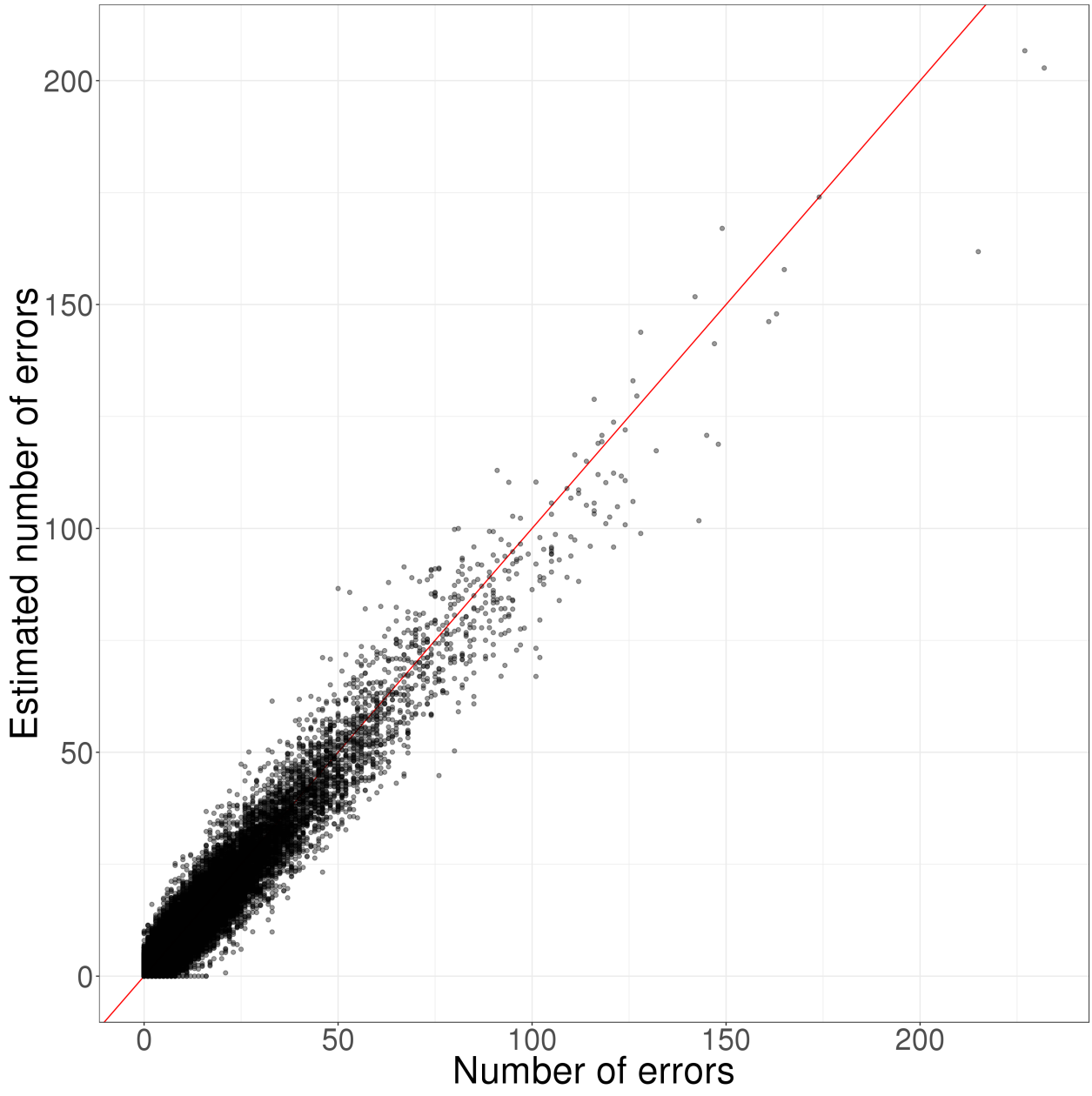

Figure S6: Number of imputation errors vs the estimated number of imputation errors by SNP. For each SNP with missing data, the number of imputation errors corresponds to the number of individuals for which imputation is incorrect. The estimated number of errors is a quantity that is returned when imputing with snp_fastimpute, which is based on XGBoost (Chen and Guestrin 2016).