Bioinformatics
doi.10.1093/bioinformatics/xxxxxx
Advance Access Publication Date: Day Month Year
Manuscript Category

Subject Section

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

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

[1] UniversitÃ© Grenoble Alpes, CNRS, Laboratoire TIMC-IMAG, UMR 5525, France,
[2] Centre de Bioinformatique, Biostatistique et Biologie IntÃ©grative (C3BI), Institut Pasteur, Paris, France and
[3] Department of Epidemiology, Harvard T.H. Chan School of Public Health, Boston, Massachusetts, USA.

*To whom correspondence should be addressed.

Associate Editor: XXXXXXX

Received on XXXXX; revised on XXXXX; accepted on XXXXX

Abstract

Motivation: Genome-wide genetic 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 for the genetic community, severely slowing down genetic analyses. This increase in data size has led to the emergence of a range of specialized software for every part of the analysis pipeline. Yet, it is often difficult to choose which piece of software to use and how to combine all these software.

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 the 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 our packages by analyzing a simulated Genome-Wide dataset including 500,000 individuals and 1M markers on a single desktop computer.

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

Contact: name@bio.com

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 available either independently or incorporated within large framework (VCFtools, BCFtools, PLINK...). Similarly genome-wide analysis quality control software have been developed (e.g. Bioconductor GWASTools and PLINK). Regarding

computation of principal components (PCs) of genotypes, commonly performed to account for population stratification in association studies, there are also several software available including different implementations of Principal Component Analysis (PCA) in EIGENSOFT (SmartPCA and FastPCA) and FlashPCA (4 REFs). Then, GWAS analysis itself depends on the genotype format (ProbABEL or SNPTEST for dosage data). Finally, there exists a range of tools for Polygenic Risk Scores (PRSs) such as LDpred, PRSice and other. 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 means that researchers are usually restricted on how they can manipulate and analyse the data they have access to.

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, already implemented by the scientific community. 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/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. It uses 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)).

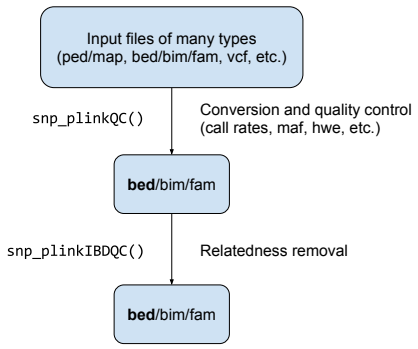

Fig. 1. Conversion and Quality Control preprocessing functions available in package bigsnpr via system calls to PLINK.

We are aware of another work that used memory-mapped files to store and efficiently access genotype data, coded in C++ (REF). With the two packages we developed, we made this solution available in R and in C++ via package Rcpp (REF). 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. 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 (REFs).

Usually, memory-mapping provides faster and seamless 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 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 the 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. One should use PLINK to 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 (use of one byte per element, 8 times less disk storage space 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 D). The first function is a wrapper to PLINK and Beagle which takes PLINK files as input and return PLINK files without missing values, and should therefore be used before reading the data in R. The second function is a new algorithm we developed in order to have a fast imputation method without

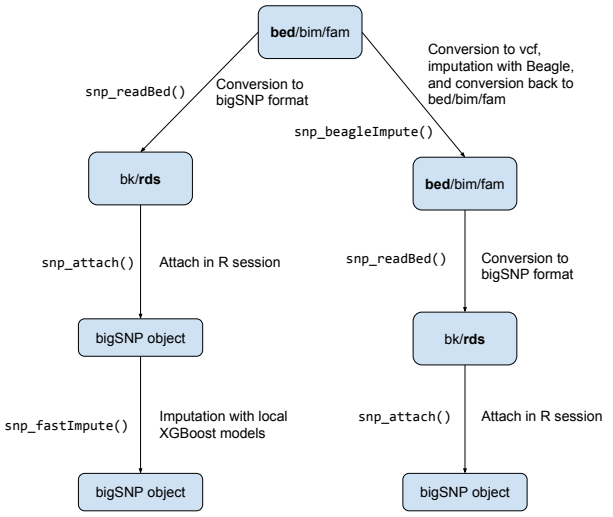

Fig. 2. Imputation and reading functions available in package bigsnpr.

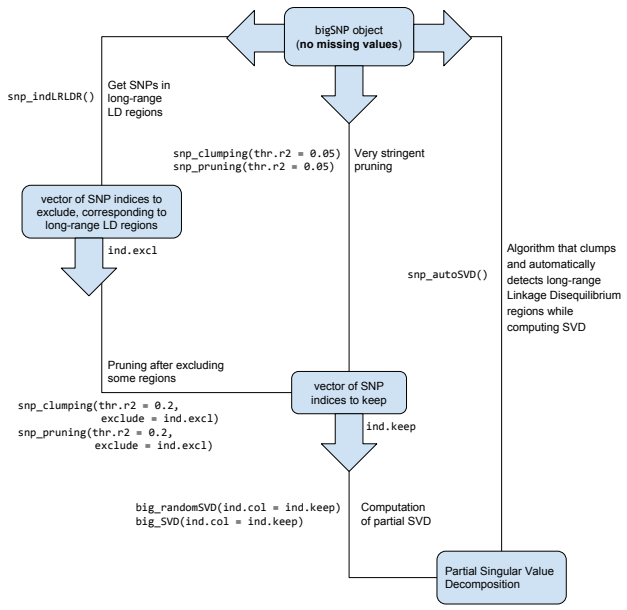

Fig. 3. Functions available in packages bigstatsr and bigsnpr for the computation of a partial Singular Value Decomposition of a genotype array.

losing much of imputation accuracy. This algorithm doesn't use phasing and is very fast. 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 (REF). XGBoost build 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 the computation of a partial Singular Value Decomposition (SVD), one based on eigenvalue decomposition and the other on randomized projections (Figure 2).

The function based on randomized projections runs in linear time in all dimensions (REF). FlashPCA2 and bigstatsr use the same PCA algorithm called Implicitly Restarted Arnoldi Method (IRAM), which is implemented in R package RSpectra (REF). 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 the multiplication 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 (REF). There are 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 MAF 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 (URL).

The third option that is generally combined with pruning or clumping consists of removing SNPs in long-range LD regions (REF). Long-range LD regions for the human genome are available as an online table that our packages can use to discard SNPs in these regions while computing PCs (URL). However, such table is human specific and could also be population specific, so we developed an algorithm that automatically detects these regions and remove 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 (REF). Finally, the algorithm keeps only consecutive outlier SNPs which is considered as the sign of long-range LD regions by the algorithm. Indeed, a long-range LD region would cause SNPs in this region to have a strong weight (loadings) in the PCA and we can differentiate these from true outliers because they are consecutive. 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

For association purposes, statistical tests based on linear and logistic regressions are available. 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 (REF).

$\forall j \in \{1, \dots, m\}$,

- for the linear regression: $\hat{y} = \alpha^{(j)} + \beta^{(j)} SNP^{(j)} + \gamma_1^{(j)} PC_1 + \cdots + \gamma_K^{(j)} PC_K + \delta_1^{(j)} COV_1 + \cdots + \delta_K^{(j)} COV_L$, where m is the number of SNPs, K is the number of Principal Components and L is the number of other covariates (such as Age or Gender).
- for the logistic regression: $\log \frac{\hat{p}}{1-\hat{p}} = \alpha^{(j)} + \beta^{(j)} SNP^{(j)} + \gamma_1^{(j)} PC_1 + \cdots + \gamma_K^{(j)} PC_K$, where $\hat{p} = \mathbb{P}(Y = 1)$.

The hypothesis that is tested is $\beta^{(j)} = 0$ against the alternative $\beta^{(j)} \neq 0$.

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 equations X and Y (REF evans2009). 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 of allele counts of the remaining SNPs weighted by the corresponding regression coefficients.

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. The currently available models are linear and logistic regressions and Support Vector Machine (SVM). These models include lasso (REF) and elastic-net (REF) regularizations, which reduce the number of predictors (SNPs) included in the predictive models. Package bigstatsr provides a fast implementation of these models by using efficient rules to discard most of the predictors (REF). The implementation of these algorithms is based on modified versions of functions available in the R packages sparseSVM and biglasso (REF). 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

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 compared the performance of the packages against standard software. 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: Celiac, a real case-control cohort, POPRES, a general population cohort and simulated datasets using real genotypes from the Celiac cohort. We compare the performance on two computers, a desktop computer with 64GB of memory and 12 cores, and a laptop with only 8GB of memory 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 have been pre-processed following steps from figure 1, resulting in 15,250 individuals and 281,122 SNPs. We note that if we used a standard R matrix to store the genotypes, it would take 32GB of memory. On the disk, the ".bed" file takes 1GB and the ".bk" file (storing the "big.matrix") takes 4GB.

We used bigstatsr and bigsnpr R functions to get the first Principal Components (PCs) of a genotype matrix and to visualize them (Figure 3). 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 4). As illustrated in the supplementary data, the whole pipeline is user-friendly and takes only 20 lines of code.

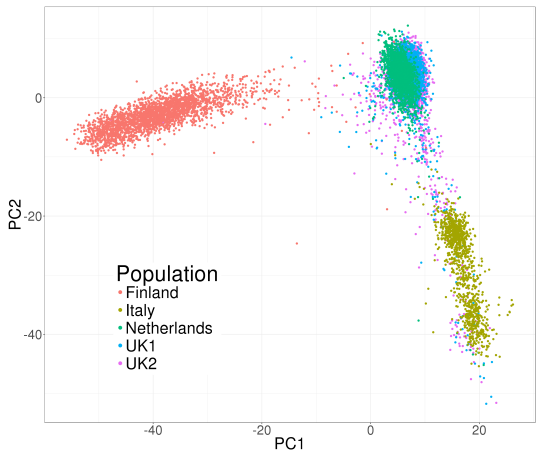

Fig. 4. Principal Components of the celiac cohort genotype matrix produced by package bigstatsr.

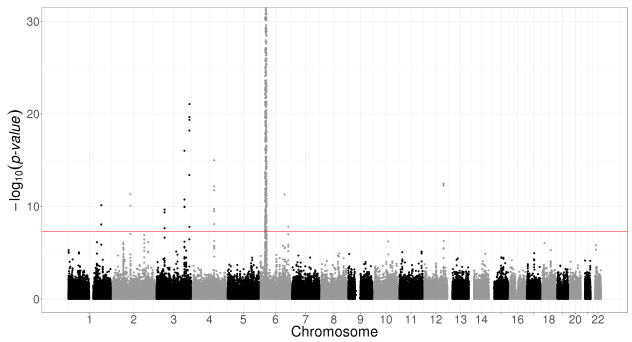

Fig. 5. Manhattan plot of the celiac disease cohort produced by package bigsnpr. The y-axis has been cut in order to not see only the very strong effects on chromosome 6.

The Celiac dataset is relatively small as compared to modern genetic cohorts. 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 with the package against PLINK and EIGENSOFT.

(TABLEAU GWAS (feuille 1) et PRS (feuille 2))

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 of order of magnitude as when using PLINK and EIGENSOFT (Table x). However, the whole analysis pipeline make 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. The code used

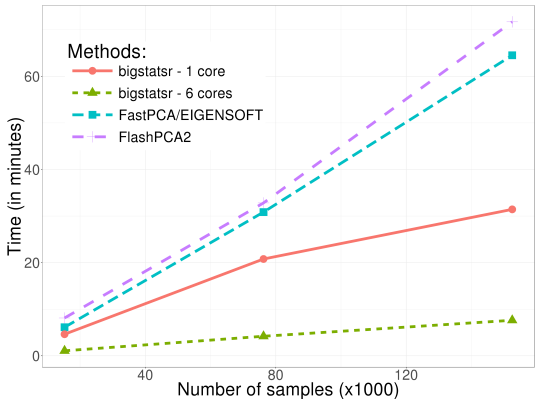

Fig. 6. Benchmark comparisons between randomized Partial Singular Value Decomposition available in FlashPCA2, FastPCA (fast mode of SmartPCA/EIGENSOFT) and package bigstatsr. It shows the computation time in minutes as a function of the number of samples. The computation corresponds to 10 Principal Components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

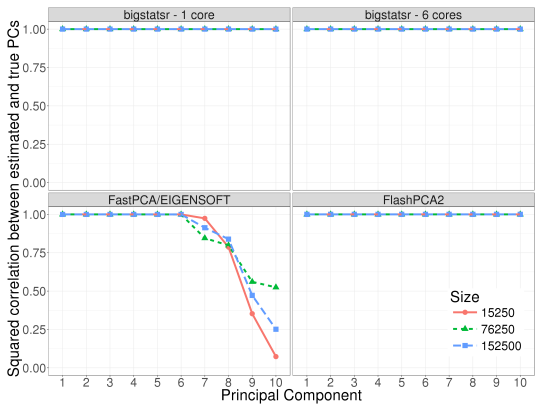

Fig. 7. Precision comparisons between randomized Partial Singular Value Decomposition available in FlashPCA2, FastPCA (fast mode of SmartPCA) and package bigstatsr. It shows the squared correlation between approximated PCs and "true" PCs (given by the slow mode of SmartPCA) of the celiac disease dataset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

for these two comparisons is available as Supplementary Material, as long as the code for the following comparisons.

On our desktop computer, we compared the computation times between FlashPCA2 and the similar function implemented in bigstatsr, big_randomSVD. For each comparison, we used the 93,083 SNPs which were remaining after pruning and we computed 10 PCs. We replicated the individuals of the celiac disease cohort in order to have three datasets of growing sizes: 15,250, 76,250 and 152,500 individuals. 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 5). 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 (e.g. 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 99.9% when using FlashPCA2 or bigstatsr (Figure 25).

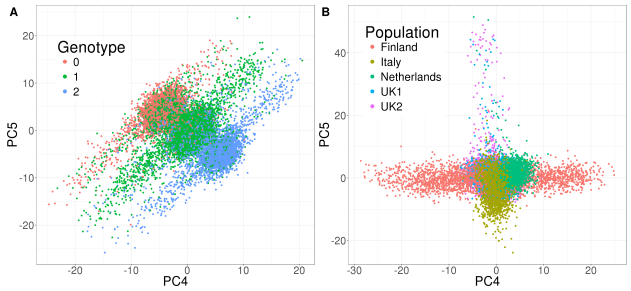

Fig. 8. PC4 and PC5 of the celiac disease dataset. Left panel, PC scores obtained without removing any long range LD region (only clumping at $R^2 > 0.2$). Individuals are coloured according to their genotype at the SNP that has the highest loading for PC4. Right panel, PC scores obtained with the automatic detection and removal of long-range LD regions. Individuals are coloured according to their population of origin.

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 (REFs). 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 SA). We compared the scores (PCs) obtained by this method with the ones obtained by removing pre-determined long-range LD regions (URL) and found a mean correlation of 89.6% between PCs mainly due to a rotation of PC7 and PC8 (Table SA2). For the celiac dataset, we found 5 long-range LD regions and a mean correlation of 98.6% between PCs obtained with snp_autoSVD and the ones obtained by clumping followed by the removing of predetermined long-range LD regions (Table SB and SB2).

For the celiac dataset, we compare 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 E and SE). When automatically removing some long-range LD regions with snp_autoSVD, we show that PC4 and PC5 are now only reflecting population structure (Figure E). Moreover, loadings are more equally distributed among SNPs (Figure SE) which is confirmed by Gini coefficients (measure of dispersion) of each squared loadings reported in Figure SF, which are all around the theoretical value of $2/pi$ (for gaussian loadings) and significantly smaller when computing SVD with snp_autoSVD than when no long-range LD region is removed.

4.5 Imputation of missing values for genotyped SNPs

For the fast imputation method based on XGBoost, we compared on the POPRES dataset (REF) composed of 1382 individuals and 344,614 SNPs in both terms of imputation accuracy and computation times. The minor allele frequencies (MAFs) are approximately uniformly distributed between 0.05 and 0.5 (Figure S1). We introduced missing values using a Beta-binomial distribution with a mean of 3% of missing values. Imputation was compared between function snp_fastImpute of package bigsnpr and Beagle (v21Jan17). Overall, our method made 4.7% of imputation errors whereas Beagle made only 3.1% but it took Beagle 14.6h to complete while our method only took 42 minutes (20 times less). For the Celiac disease dataset, our method took less than 6 hours to complete whereas Beagle didn't finish imputing chromosome 1 in 48h. We also show that our method's estimation of the number of imputation errors is accurate (Figure X).

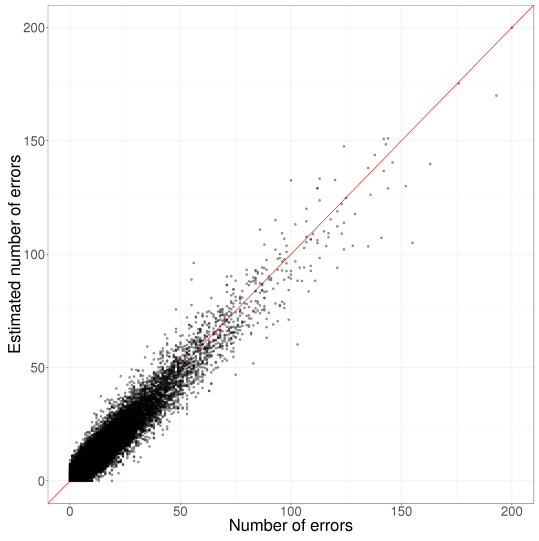

Fig. 9. Number of imputation errors vs the estimated number of imputation errors by SNP. For each SNP with missing data, the number of imputation errors corresponds to the number of individuals for which imputation is incorrect. The estimated number of errors is a quantity that is returned when imputing with snp_fastimpute, which is based on XGBoost (REF).

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 your analysis down. 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 SNP 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 pruning.

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 excellent machine learning algorithm XGBoost to easily make a fast yet 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 completely different 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 (REF P+T). 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 (REF).

Acknowledgements

Text Text Text Text Text Text Text Text.

Funding

This work has been supported by the... Text Text Text Text.

References

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.

Supplementary results

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 1. Regions found by snp_autoSVD for the POPRES dataset. In parentheses are regions referenced in (REF)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 2. Regions found by snp_autoSVD for the celiac dataset. In parentheses are regions referenced in (REF)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 3. 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 4. 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.

5.2 Imputation

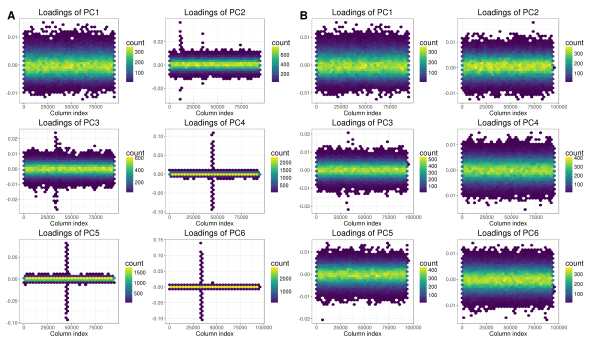

Fig. 10. Loadings of first 6 PCs of the celiac disease dataset plotted as hexbins (2-D histogram with hexagonal cells). On the left, without removing any long range LD region (only clumping at $R^2 > 0.2$). On the right, with the automatic detection and removal of long-range LD regions.

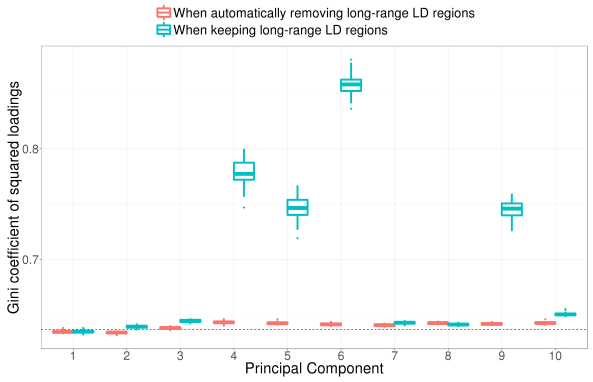

Fig. 11. Boxplots of 1000 bootstrapped Gini coefficients (measure of statistical dispersion) of squared loadings without removing any long range LD region (only clumping at $R^2 > 0.2$) and with the automatic detection and removal of long-range LD regions. The dashed line corresponds to the theoretical value for gaussian loadings.

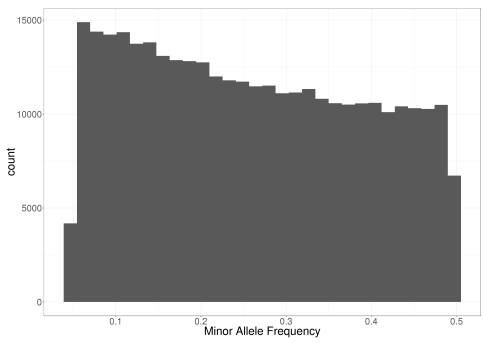

Fig. 12. Histogram of the minor allele frequencies of the POPRES dataset used for comparing imputation methods.

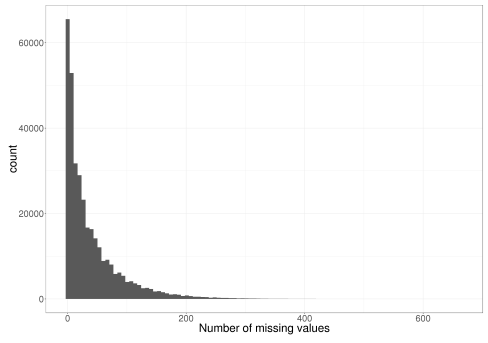

Fig. 13. Histogram of the number of missing values by SNP. These numbers were generated using a Beta-binomial distribution.