The R package {bigstatsr}: memory- and computation-efficient tools for big matrices stored on disk

Florian Privé (@privefl)

Rencontres R 2018

Slides: https://privefl.github.io/RR18/bigstatsr.html

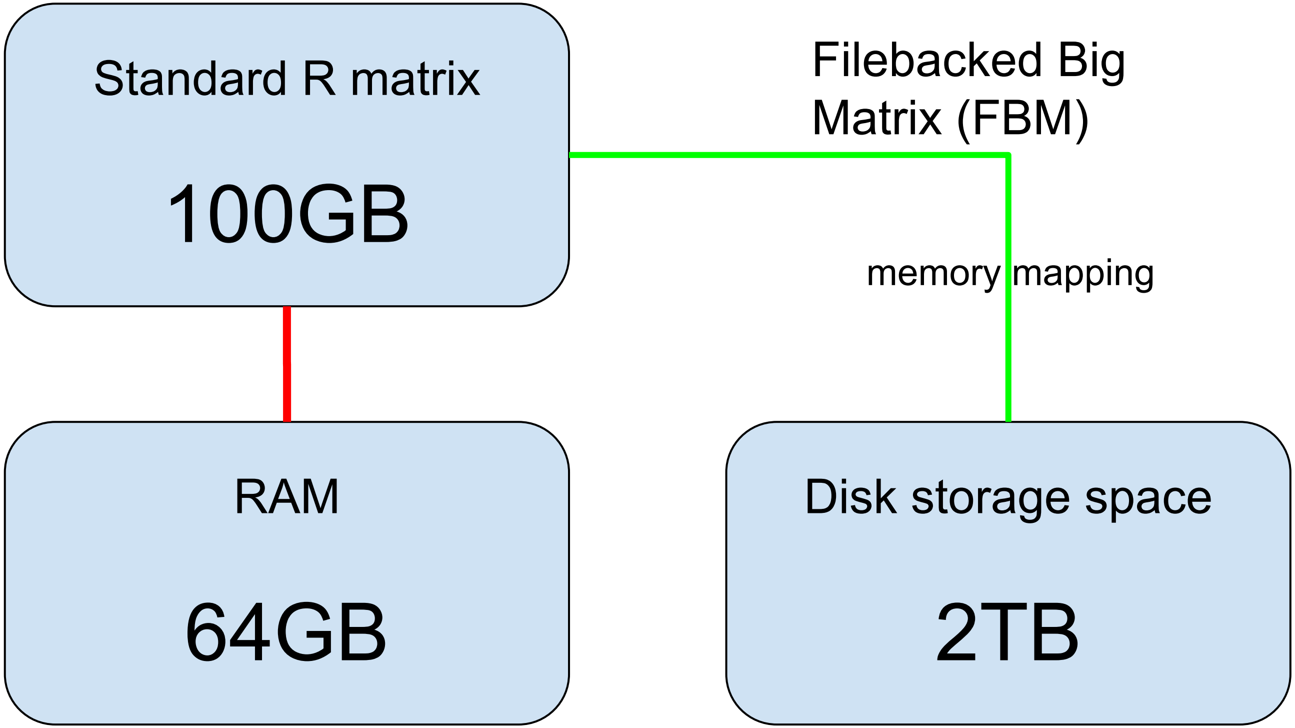
Motivation

Analyze very large genotype matrices

- previously: 15K x 280K, celiac disease (~30GB)
- currently: 500K x 500K, UK Biobank (~2TB)

But I still want to use R..

The solution I found



Format `FBM` is very similar to format `filebacked.big.matrix` from package {bigmemory} (details in this vignette).

Simple accessors

Similar accessor as R matrices

```
X <- FBM(2, 5, init = 1:10, backingfile = "test")
```

```
X$backingfile
```

```
## [1] "/home/privef/Bureau/RR18/test.bk"
```

```
X[, 1]   ## ok
```

```
## [1] 1 2
```

```
X[1, ]   ## bad
```

```
## [1] 1 3 5 7 9
```

```
X[]      ## super bad
```

```
##      [,1] [,2] [,3] [,4] [,5]
## [1,]    1    3    5    7    9
## [2,]    2    4    6    8   10
```

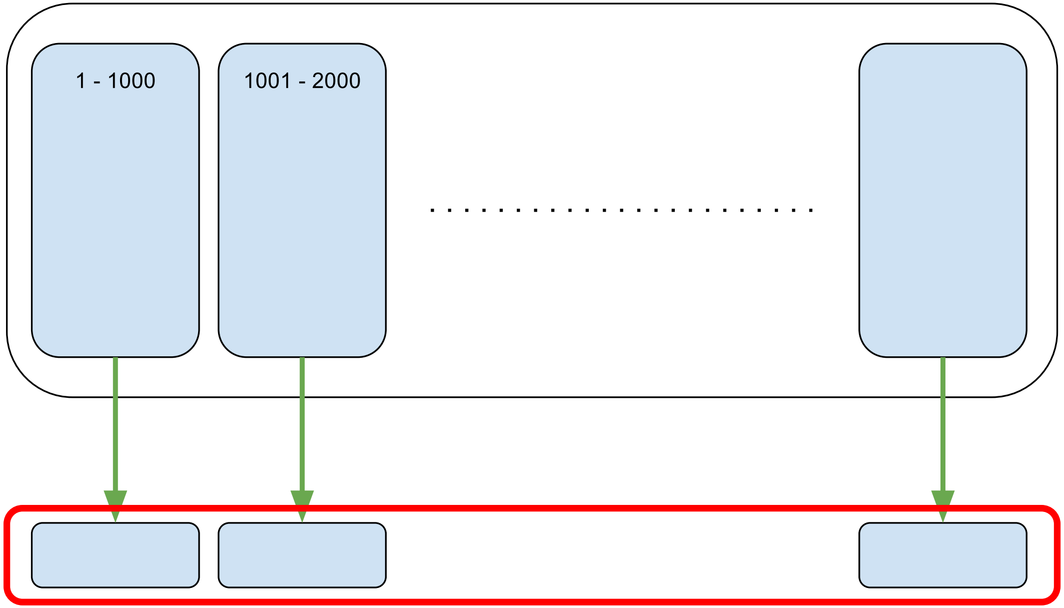
Similar accessor as R matrices

```r
colSums(X[])  ## super bad
```

```
## [1]  3  7 11 15 19
```

Split-(par)Apply-Combine Strategy

Apply standard R functions to big matrices (in parallel)



SPLIT
(e.g. every 1000 columns)

APPLY
(e.g. `colSums()`)

COMBINE
(e.g. `c()`)

Implemented in `big_apply()`.

Similar accessor as Rcpp matrices

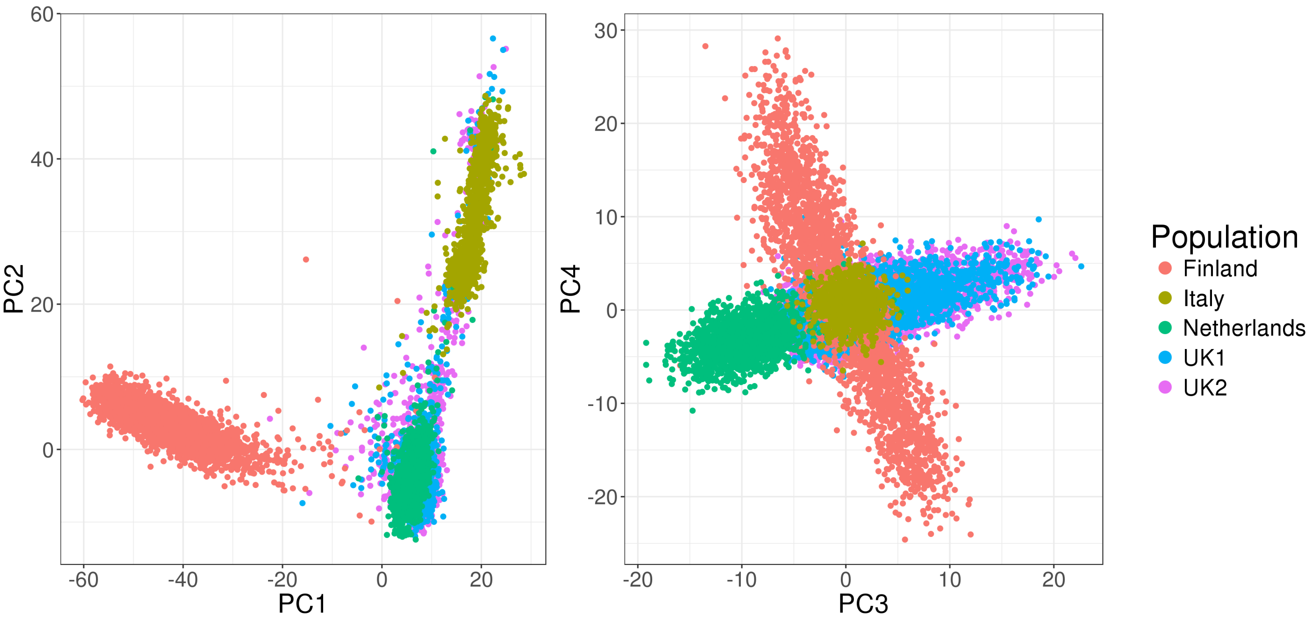
```cpp
// [[Rcpp::depends(BH, bigstatsr)]]
#include <bigstatsr/BMAcc.h>

// [[Rcpp::export]]
NumericVector big_colsums(Environment BM) {

  XPtr<FBM> xpBM = BM["address"];
  BMAcc<double> macc(xpBM);

  size_t n = macc.nrow();
  size_t m = macc.ncol();

  NumericVector res(m);

  for (size_t j = 0; j < m; j++)
    for (size_t i = 0; i < n; i++)
      res[j] += macc(i, j);

  return res;
}
```

Some examples

from my work

Partial Singular Value Decomposition

15K × 100K -- 10 first PCs -- 6 cores -- **1 min** (vs 2h in base R)

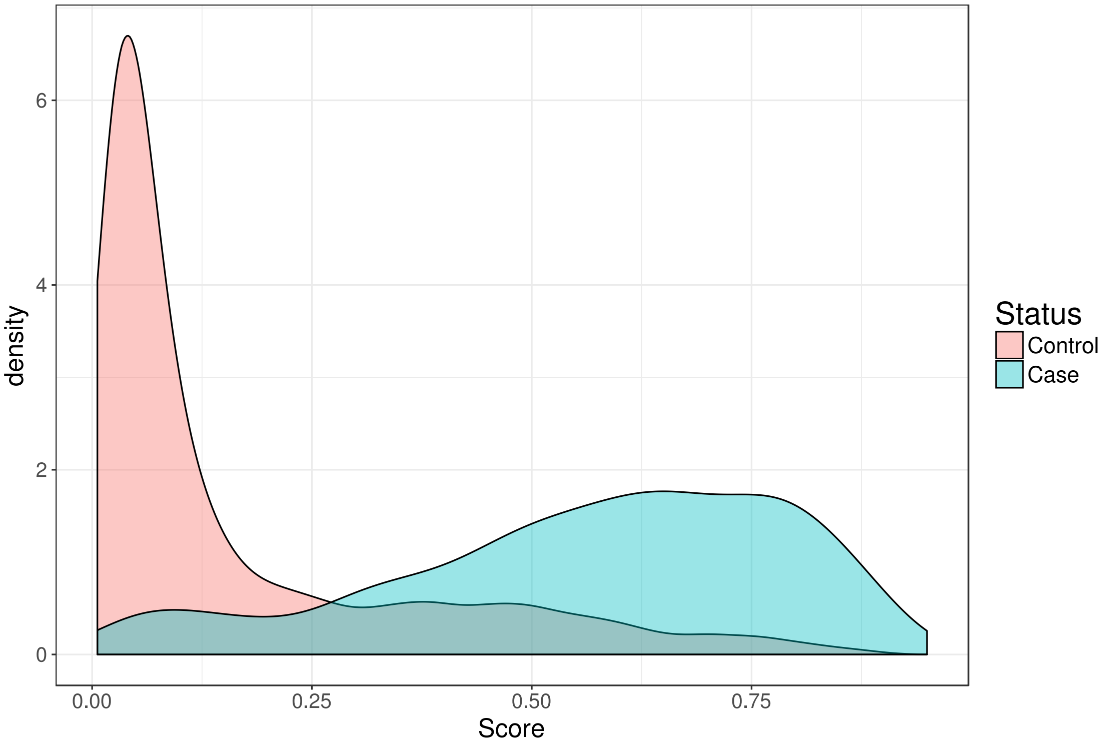


Implemented in `big_randomSVD()`, powered by R packages {RSpectra} and {Rcpp}.

Sparse linear models

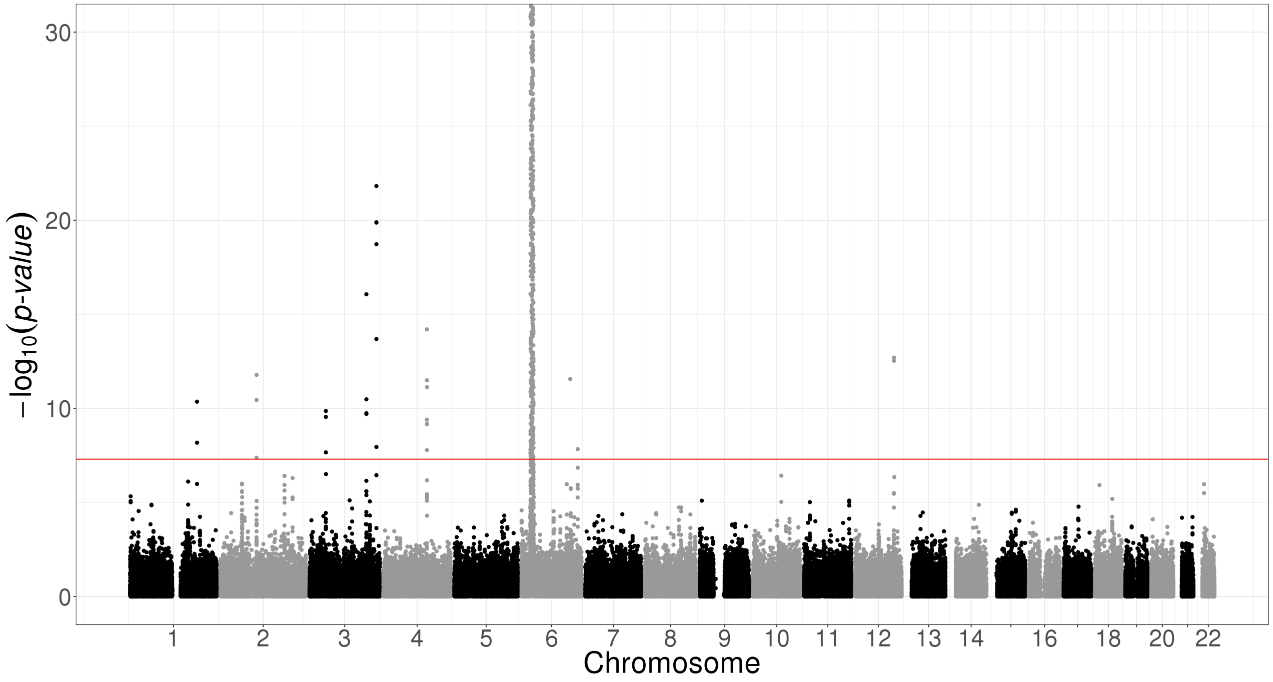
Predicting complex diseases via penalized logistic regression

15K × 280K -- 6 cores -- **2 min**



Multiple association testing

Which DNA mutations are associated with one disease?



Conclusion

I'm able to run algorithms

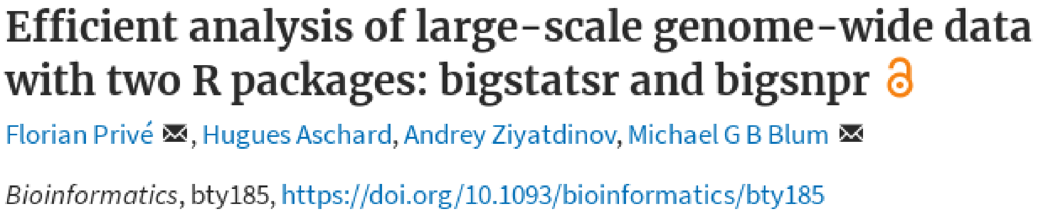
on 100GB of data

in R on my computer

Advantages of using FBM objects

- you can apply algorithms on **data larger than your RAM**,
- you can easily **parallelize** your algorithms because the data on disk is shared,
- you write **more efficient algorithms** (you do less copies and think more about what you're doing),
- you can use **different types of data**, for example, in my field, I’m storing my data with only 1 byte per element (rather than 8 bytes for a standard R matrix). See the documentation of the FBM class for details.

R Packages



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

Florian Privé ✉, Hugues Aschard, Andrey Ziyatdinov, Michael G B Blum ✉

Bioinformatics, bty185, https://doi.org/10.1093/bioinformatics/bty185

- {bigstatsr}: to be used by any field of research
- {bigsnpr}: algorithms specific to my field of research

Contributors are welcomed!

Make sure to grab an hex sticker

Thanks!

Presentation: https://privefl.github.io/RR18/bigstatsr.html

Package's website: https://privefl.github.io/bigstatsr/

DOI: 10.1093/bioinformatics/bty185

privefl privefl F. Privé

Slides created via the R package **xaringan**.