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

Florian Privé (@privefl)

eRum 2018

# About

I'm a PhD Student (2016-2019) in **Predictive Human Genetics** in Grenoble.

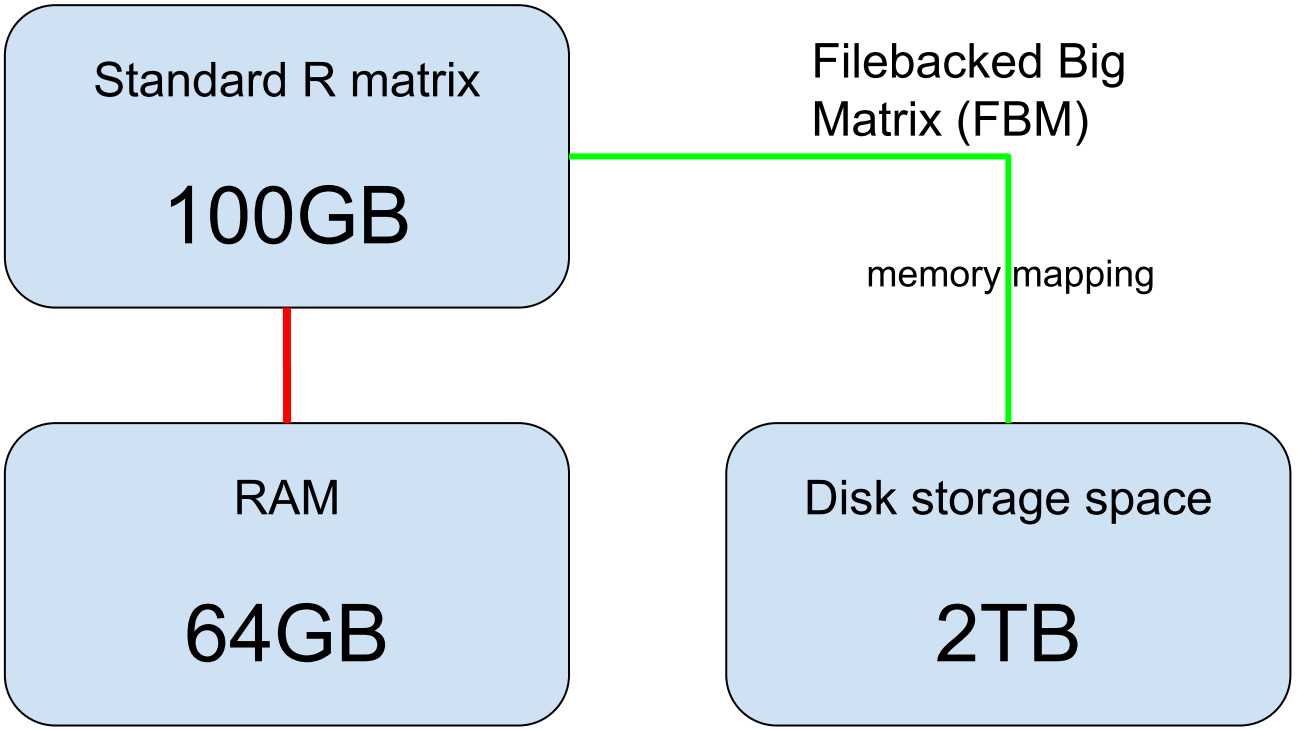
$$\boxed{\text{Disease} \sim \text{DNA mutations} + \cdots}$$

# 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




`FBM` is very similar to `filebacked.big.matrix` from package {bigmemory}.

# Similar accessor as R matrices

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

```
X$backingfile
```

```
## [1] "/home/privef/Bureau/eRum-2018/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
```

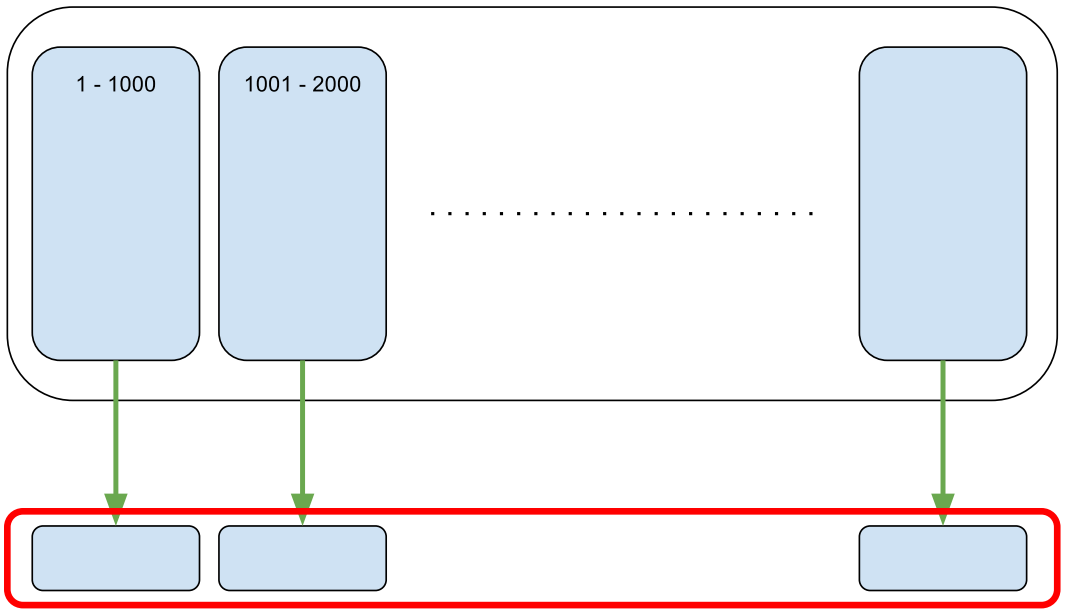
# Similar accessor as R matrices

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

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

# Split-(par)Apply-Combine Strategy

## Apply standard R functions to big matrices (in parallel)



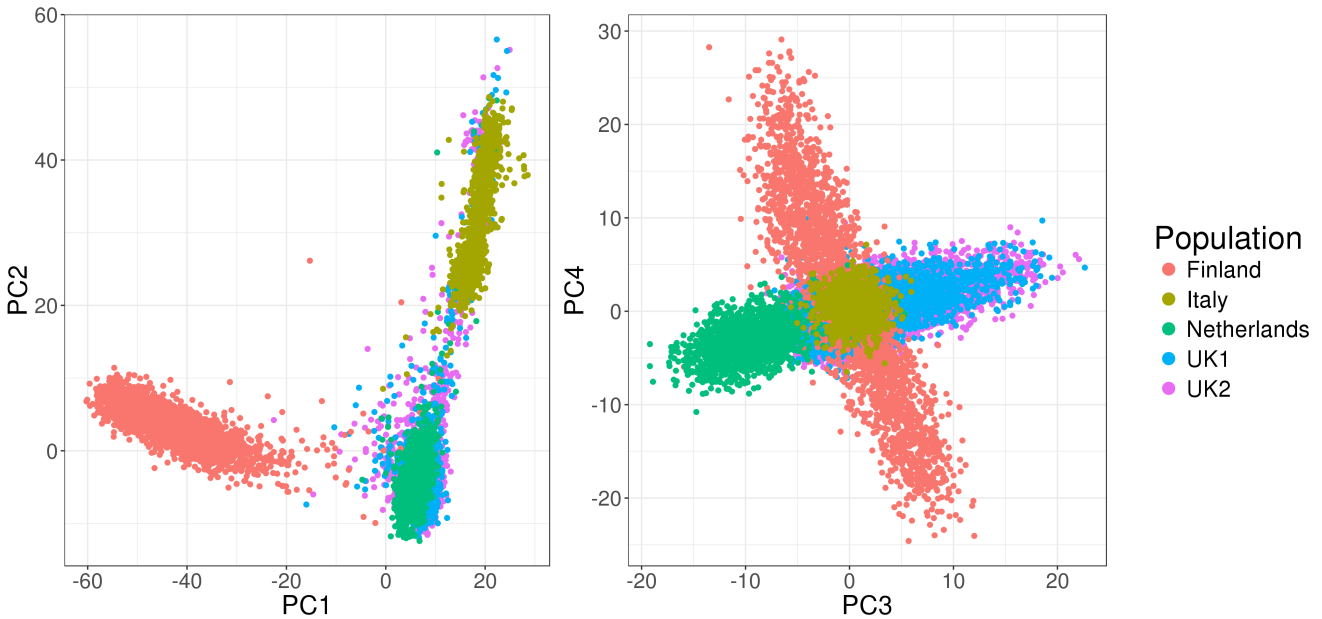

Implemented in `big_apply()`.

# 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;
}
```

# Partial Singular Value Decomposition

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

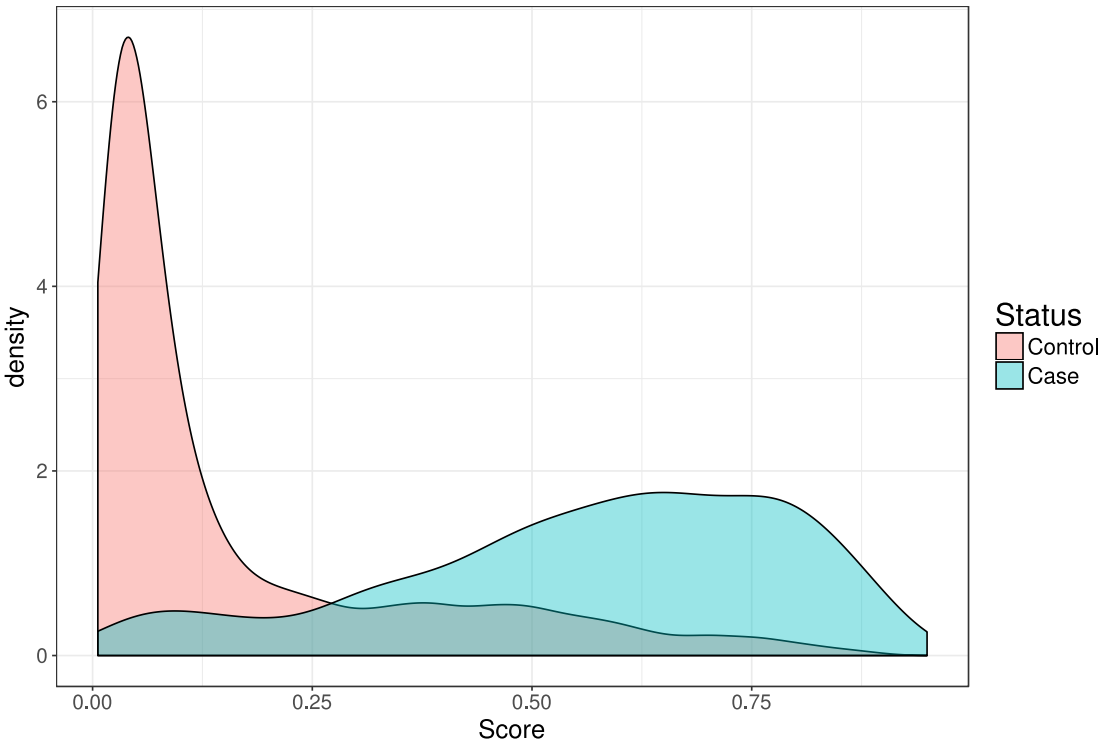



Implemented in `big_randomSVD()`, powered by R packages {RSpectra} and {Rcpp}.

# Sparse linear models

## Predicting complex diseases with a penalized logistic regression

15K $\times$ 280K -- 6 cores -- **2 min**



# Other functions

- matrix operations
- association of each variable with an output
- plotting functions
- read from text files
- many other functions..

## Parallel

- most of the functions are parallelized (memory-mapping makes it easy!)
- you can parallelize you own functions with `big_parallelize()`

I'm now able
to run algorithms
on 100GB of data
on my computer

# 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!

# Thanks!

Presentation: https://privefl.github.io/eRum-2018/slides.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**.