BALD Vignette

Alia Dehman

February 19, 2015

1 The BALD package

The `BALD` package (Blockwise Approach using Linkage Disequilibrium) arose out of the need to provide friendly data generation functions and an efficient analysis method of whole genome association studies in R.
With recent advances in high-throughput genotyping technology, genome-wide association studies (GWAS) have become a tool of choice for identifying genetic markers underlying a variation in a given phenotype - typically complex human diseases and traits. Whole-genome single nucleotide polymorphism (SNP) data are collected for many thousands of SNP markers, leading to high-dimensional regression problems where the number of predictors greatly exceeds the number of observations. Moreover, these predictors are statistically dependent, in particular due to linkage disequilibrium (LD).
The main function of this package `grplassoCWard` implements a *proposed three-step approach* [1] that explicitly takes advantage of the grouping structure induced by LD: in the first step, LD blocks are inferred by performing a clustering of LD estimates with an adjcency constraint [4]. In the second step, the Gap statistic model selection approach [3] is applied to estimate the number of groups and finally the Group Lasso regression [5] is performed on the inferred LD blocks.

2 Generating genotype and phenotype data

Let N be the number of individuals of our GWA study and p the number of variables (SNPs). The `BALD` package allows to generate block-structured genotype data and associated continuous phenotype according to the linear regression model:

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon},$$

where $\mathbf{y} \in \mathbb{R}^N$ is the phenotype vector, $\mathbf{X} \in \{1, 2, 3\}^{N \times p}$ the SNP genotypes matrix and $\boldsymbol{\epsilon} \in \mathbb{R}^N$ a gaussian error term. The columns of $\mathbf{X}$ are assumed to be block-structured on `nBlocks` non-overlapping blocks.
In order to simulate such GWAS data, we will use the two simulation functions `simBeta` and `simulation`. We will first simulate the association vector $\boldsymbol{\beta}$ using the function `simBeta` as follows:

```
set.seed(2)
blockSizes <- c(2,4,5,3,2,4)
p <- sum(blockSizes)
sig.blocks <- c(3,5)
nb.per.block <- c(2,3)
betas <- simBeta(blockSizes, sig.blocks, nb.per.block)
```

The first element of the output `betas`:

```
betas$blockSizes

## [1] 4 3 5 2 4 2
```

contains the effective block sizes used for the simulation of $\boldsymbol{\beta}$. The second element of `betas`

```
str(betas$betaMat)

##  num [1:20, 1:3] 0 0 0 0 1 -1 0 -1 -1 1 ...
##  - attr(*, "dimnames")=List of 2
##   ..$ : NULL
##   ..$ : chr [1:3] "betaSNP" "betaBl" "groups"
```

is a 20×3 numeric matrix. The first column of the matrix contains the regression vector for the 20 predictors (SNPs) structured in `length(blockSizes)`=6 blocks. We can check that only 5 SNPs were simulated as associated which are the first two (resp. three) SNPs contained in the first block of size 3 (resp. 5). The second column of `betas$betaMat` contains the "block regression vector".

```
betas$betaMat[,"betaBl"]

##  [1] 0 0 0 0 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0
```

As we can see, all the coefficients of the predictors contained in `sig.blocks` have been simulated nonzero (TRUE).
Finally, the third column of `betas$betaMat` corresponds to the vector defining the grouping of the variables effectively used for the simulation of the regression vector $\boldsymbol{\beta}$.

Using the regression coefficients in `betas`, we can then simulate the genotype and phenotype data. We begin by fixing the number of individuals of our study `N`, the level of correlation between the SNPs of each block `corr` and the coefficient of determination of the problem `r2`:

```
N <- 50
corr <- 0.5
r2 <- 0.7
sim <- simulation(N, betas$betaMat[, "betaSNP"], betas$blockSizes,
                  corr, r2)
X <- sim$X
str(X)

##  num [1:50, 1:20] 1 3 1 3 2 3 3 3 1 3 ...

y <- sim$y
str(y)

##  num [1:50, 1] 0.8686 2.6876 5.0238 -0.0499 1.101 ...

sim$r2  ## effective coefficient of determination

## [1] 0.707
```

3 The three-step method

Now that we have generated our genotype and phenotype data, we can apply the proposed three-step method using the following command line:

```
nlambda <- 30
B <- 50
## assessing the within-cluster measures for the B reference datasets
traceWB <- lapply(1:B, FUN=function(ii){
  getTraceW(N, p)
})
traceWB <- simplify2array(traceWB)
gl <- grplassoCWard(sim$X, sim$y, groups=NULL, nlambda=nlambda,
                    max.nc=p-1, min.nc=1, traceWB=traceWB)
str(gl)

## List of 2
##  $ coefs : num [1:30, 1:20] 0 0 0 -0.00823 -0.08501 ...
##  $ groups: num [1:20] 1 1 1 1 2 2 2 3 3 3 ...
```

Through the default value `NULL` for the argument `groups`, the user indicates that the group structure needs to be inferred using the constrained Ward's incremental method and the Gap statistic model selection approach. The inferred group structure is returned as a vector of integers from `1` to `nBlocks`. SNPs sharing the same number belong to the same group. Therefore, we can check if the two first steps of inferring groups have well estimated the block sizes:

```
betas$blockSizes

## [1] 4 3 5 2 4 2

gl$groups

##  [1] 1 1 1 1 2 2 2 3 3 3 3 3 4 4 5 5 5 5 6 6

tab <- table(gl$groups)
dimnames(tab) <- NULL
tab

## [1] 4 3 5 2 4 2
```

The block sizes were in effect well estimated but this is not always the case above all when the correlation level is less that 0.4.
The second element of the output corresponds to the `nlambda` $\times$ `p` matrix of the Group Lasso coefficients.

4 Compared to other approaches

The `BALD` package allows the application of several regression methods using the function `select`:

```
coefsGL <- select("groupLasso", X, y, groups=NULL,  nlambda=nlambda,
                  max.nc=p-1, min.nc=2, B=100)
coefsOGL <- select("groupLasso", X, y, groups=betas$groups,
                   nlambda=nlambda)   ## Oracle Group Lasso !
coefsL <- select("lasso", X, y, nlambda=nlambda)
coefsEN <- select("elastic.net", X, y, lambda2=0.5, nlambda=nlambda)
pvalsUniv <- select("univ", X, y, nlambda=nlambda)
```

See vignette in the path :

```
system.file("evaluation/evaluation.Rnw", package="BALD")
```

for an example of performance comparison of different methods.

5 Representations of the results

The `BALD` package allows two different representations of the results: the first plot function `plotHeatmap` allows a Heatmap of the linkage disequilibrium blocks within a given region [2] and possibly to highlight selected blocks/SNPs. The second plot function `plotGroupsGL` provides a graphical display for interpreting selected blocks in function of the univariate p-values of the SNPs

contained in these blocks.

Based on the regression results of the models Group Lasso and Lasso on the previously simulated data set, we can represent the first 3 blocks selected by the Group Lasso and the first SNPs selected by the Lasso as follows:

```
## "true" beta
betas$betaMat[,"betaSNP"]

##  [1]  0  0  0  0  1 -1  0 -1 -1  1  0  0  0  0  0  0  0  0  0  0

groups <- gl$groups
coefsGL <- select("groupLasso", X, y, groups=groups,  nlambda=nlambda)
selSNP <- as.matrix(t(coefsGL)!=0)

## blocks selected by GL at each level of regularization
selBl <- as.matrix(aggregate(selSNP, list(groups=groups), sum)[, -1])
str(selBl)

##  int [1:6, 1:30] 0 3 0 0 0 0 0 3 5 0 ...
##  - attr(*, "dimnames")=List of 2
##   ..$ : NULL
##   ..$ : chr [1:30] "V1" "V2" "V3" "V4" ...

selBl[,1:3]

##      V1 V2 V3
## [1,]  0  0  0
## [2,]  3  3  3
## [3,]  0  5  5
## [4,]  0  0  0
## [5,]  0  0  0
## [6,]  0  2  2

## first 3 blocks selected by GL
firstBl <- which(selBl[,2]!=0)

## first 5 SNPs selected by the Lasso
coefsL <- select("lasso", X, y, nlambda=nlambda)
firstSNPs <- which(coefsL[4,]!=0)

## heatmap plot
blockSizes <- betas$blockSizes
```

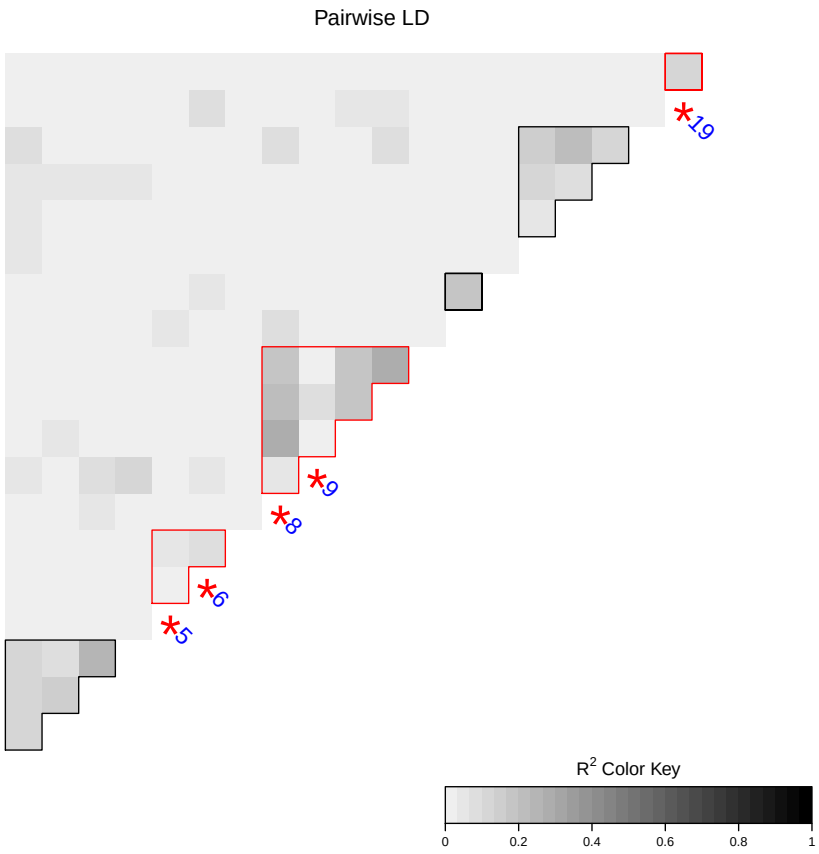


Figure 1: Heatmap plot of the LD blocks.

```
plotHeatmap(X, blockSizes, selBlocks=firstBl, snpNames=as.character(1:p),
            snpStar=as.character(firstSNPs))
```

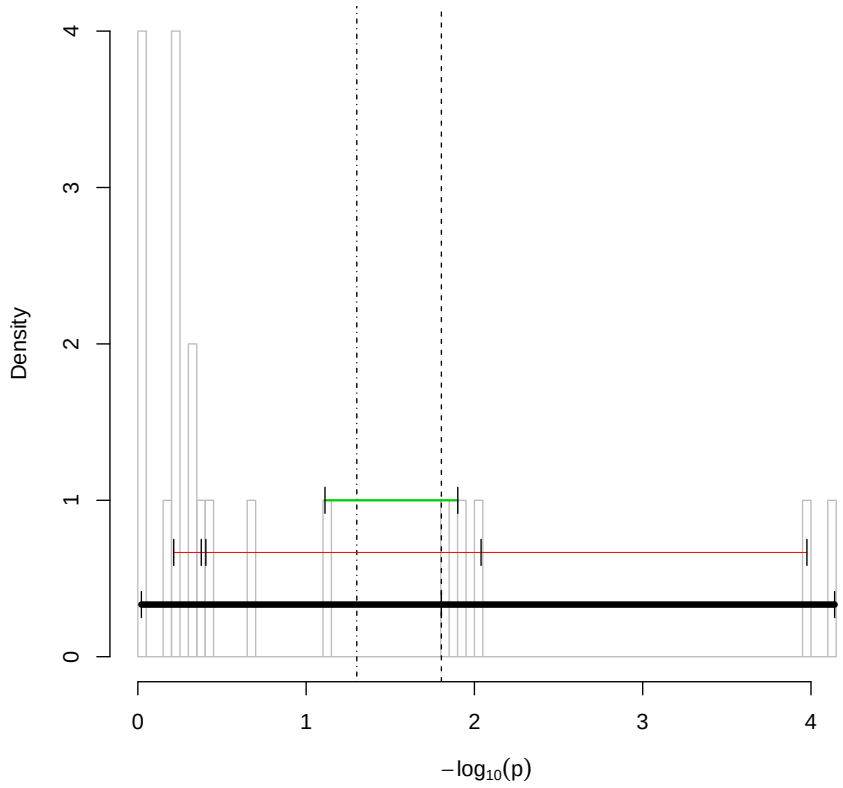
Figure 1 displays the linkage disequilibrium measures of the set of 20 contiguous markers. The SNPs shown with a red star (*) correspond to the first markers (in the regularization path) selected by the Lasso. The local block structure inferred by the clustering and model selection steps of the proposed method is also highlighted and the first 3 blocks (in the regularization path) selected by the Group Lasso are delimited by a red outline.

Finally, in order to have a more accurate idea about the relevance of the blocks selected by the Group Lasso, we can display the univariate p-values of the SNPs within them. To do this, we will use the function `plotGroupsGL`

that takes as arguments regression coefficients matrix of the Group Lasso, the number of groups to be displayed and the univariate p-values of the 20 markers:

```
## univariate p-values
pvals <- apply(X, 2, FUN=function(vect){
  pv <- summary(lm(y ~ vect))$coefficients[2,4]
})

## first 3 blocks displayed
plotGroupsGL(coefsGL, nbGroup=3, pvals)
```



Each of the first 3 blocks selected by the Group Lasso is represented by a colored horizontal segment ranging from the largest to the smallest univariate p-value of the block. The vertical black segments indicate the univariate p-values of each SNP in these LD blocks and the vertical line highlights the significance threshold (defaults to `t=0.25`).

6 Session information

```
sessionInfo()

## R version 3.1.0 (2014-04-10)
## Platform: x86_64-apple-darwin10.8.0 (64-bit)
##
## locale:
## [1] C/fr_FR.UTF-8/fr_FR.UTF-8/C/fr_FR.UTF-8/fr_FR.UTF-8
##
## attached base packages:
## [1] grid      stats     graphics  grDevices utils     datasets  methods
## [8] base
##
## other attached packages:
## [1] knitr_1.6  BALD_0.1.1
##
## loaded via a namespace (and not attached):
##  [1] BiocGenerics_0.10.0 LDheatmap_0.99-1    MASS_7.3-34
##  [4] Matrix_1.1-4        ROC_1.40.0          Rcpp_0.11.2
##  [7] colorspace_1.2-4    digest_0.6.4        evaluate_0.5.5
## [10] formatR_1.0         ggplot2_1.0.0       grplasso_0.4-4
## [13] gtable_0.1.2        highr_0.3           lattice_0.20-29
## [16] munsell_0.4.2       parallel_3.1.0      plyr_1.8.1
## [19] proto_0.3-10        quadrupen_0.2-4     reshape2_1.4
## [22] scales_0.2.4        snpStats_1.14.0     splines_3.1.0
## [25] stringr_0.6.2       survival_2.37-7     tools_3.1.0
```

References

[1] A. Dehman, C. Ambroise, and P. Neuvial. Performance of a Blockwise Approach in Variable Selection using Linkage Disequilibrium Information. *BioMed Central Bioinformatics*, 2014.

[2] Shin, Ji-Hyung and Blay, Sigal and McNeney, Brad and Graham, Jinko. LDheatmap: an R function for graphical display of pairwise linkage disequilibria between single nucleotide polymorphisms. *Journal of Statistical Software*, 16(3):1–10, 2006.

[3] Tibshirani, Robert and Walther, Guenther and Hastie, Trevor. Estimating the number of clusters in a data set via the gap statistic. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 63(2):411–423, 2001.

[4] Ward Jr, Joe H. Hierarchical grouping to optimize an objective function. *Journal of the American statistical association*, 58(301):236–244, 1963.

[5] Yuan, Ming and Lin, Yi. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 68(1):49–67, 2005.