Methods Comparison

Malu Calle Rosingana
Coding: Yiwen Wang

2019-05-14

Contents

Chapter 1

Introduction

This vignette provides a comparison of the 12 variables selected by the 3 methods:

- logistic_lasso + linear constraints;
- centered logratio transformation (CLR) + logistic_lasso;
- selbal.

First, you will need to install then load the following packages:

1.1 Packages installation and loading

```r
#cran.packages <- c('knitr', 'MASS', 'VennDiagram', 'gplots', 'glmnet')
#install.packages(cran.packages)
#devtools::install_github(repo = "UVic-omics/selbal")

library(knitr) # rbookdown
library(MASS) # ginv
library(VennDiagram) # venn.diagram
library(gplots) # venn
library(glmnet) # glmnet
library(selbal) # selbal

# build in functions
source(file = 'functions.R')
```

1.2 Simulations

The way we simulate datasets.

1.3 Example datasets

We have two case studies.

1.3.1 Microbiome and Crohn's disease

Crohn's disease (CD) is an inflammatory bowel disease that has been linked to microbial alterations in the gut (Gevers et al., 2014; Øyri et al., 2015). We use data from a large pediatric CD cohort study (Gevers et al., 2014) to compare the proposed methodologies for identification of microbial signatures.

Microbiome data from 16S rRNA gene sequencing and QIIME 1.7.0 bioinformatics processing were downloaded from Qiita (Rivera-Pinto et al., 2018). Only patients with Crohn's disease (n = 662) and those without any symptoms (n = 313) were analyzed. The OTU table was agglomerated to the genus level, resulting in a matrix with 48 genera and 975 samples.

We load the data that are provided in advance.

```r
load("./datasets/Crohn_data.rda")
# dim(x_Crohn)
# summary(y_Crohn)
```

Table 1.1: **CD data summary**

No. of genera	48
No. of samples	975
No. of patients with CD	662
No. of healthy patients	313

1.3.2 BEME day1 data

content need to be added.

Chapter 2

CODA_LOGISTIC_LASSO

First, we illustrate the method lasso with linear constraints.

2.1 CD data

```
x <- x_Crohn
```

x is the matrix of microbiome abundances, either absolute abundances (counts) or relative abundances (proportions). However, x should not be the matrix of log(counts) or log(proportions). The method itself performs the log-transformation of the abundances.

```
y <- y_Crohn
summary(y)
```

```
##  CD  no
## 662 313
```

In the CD data, y is the binary outcome, can be numerical (values 0 and 1), factor (2 levels) or categorical (2 categories).

```
dim(x)
```

```
## [1] 975  48
```

The rows of x are individuals/samples, the columns are taxa.

To run the method, we need a value of λ.

```
rangLambda(y, x,numVar = 12, lambdaIni = 0.15)
```

```
## [1]  0.00  0.65 48.00 17.00
## [1]  0.000  0.325 48.000  2.000
## [1]  0.1625  0.3250 12.0000  2.0000
```

```
## $`rang lambdas`
## [1] 0.1625 0.3250
##
## $`num selected variables`
## [1] 12  2
```

It provides a rang of λ values corresponding to a given number of variables to be selected (numVar).

The default initial λ is lambdaIni=1.

```
results_codalasso <- coda_logistic_lasso(y, x, lambda = 0.19)
```

λ is the penalization parameter: the larger the value of λ, the fewer number of variables will be selected.

```
results_codalasso
```

```
## $`number of iterations`
## [1] 23
##
## $`number of selected taxa`
## [1] 11
##
## $`indices of taxa with non-zero coeff`
##  [1]  0  2  5  9 19 27 31 32 33 39 40 48
##
## $`taxa with non-zero coeff`
##  [1] "beta0"                          "g__Parabacteroides"
##  [3] "f__Peptostreptococcaceae_g__" "g__Eggerthella"
##  [5] "g__Dialister"                   "o__Lactobacillales_g__"
##  [7] "g__Prevotella"                  "g__Roseburia"
##  [9] "g__Lachnospira"                 "g__Streptococcus"
## [11] "g__Aggregatibacter"             "g__Bilophila"
##
## $`beta non-zero coefficients`
##  [1] -1.0023255478 -0.0057464016  0.0202907034 -0.0184875313 -0.0992393846
##  [6] -0.0159418386 -0.0030842274  0.2530005235 -0.0085670215 -0.0818265310
## [11] -0.0406982632  0.0002999724
##
## $`proportion of explained deviance`
## [1] 0.1542276
##
## $betas
##  [1] -1.0023255478  0.0000000000 -0.0057464016  0.0000000000  0.0000000000
##  [6]  0.0202907034  0.0000000000  0.0000000000  0.0000000000 -0.0184875313
## [11]  0.0000000000  0.0000000000  0.0000000000  0.0000000000  0.0000000000
## [16]  0.0000000000  0.0000000000  0.0000000000  0.0000000000 -0.0992393846
## [21]  0.0000000000  0.0000000000  0.0000000000  0.0000000000  0.0000000000
## [26]  0.0000000000  0.0000000000 -0.0159418386  0.0000000000  0.0000000000
## [31]  0.0000000000 -0.0030842274  0.2530005235 -0.0085670215  0.0000000000
## [36]  0.0000000000  0.0000000000  0.0000000000  0.0000000000 -0.0818265310
## [41] -0.0406982632  0.0000000000  0.0000000000  0.0000000000  0.0000000000
## [46]  0.0000000000  0.0000000000  0.0000000000  0.0002999724
```

```
selected_codalasso <- results_codalasso[[4]][-1]

columns_selected_codalasso <- results_codalasso[[3]][-1]

write.csv(data.frame(columns_selected_codalasso,selected_codalasso),
          "./Generated_datasets/results_codalasso_Crohn12.csv")
```

We extract the name of selected genera and their column indices and then save them in a csv file.

2.2 BEME day1 data

content need to be added.

Chapter 3

CLR_LOGISTIC_LASSO

Then, we illustrate a combined method with centered logratio transformation (CLR) and lasso.

3.1 CD data

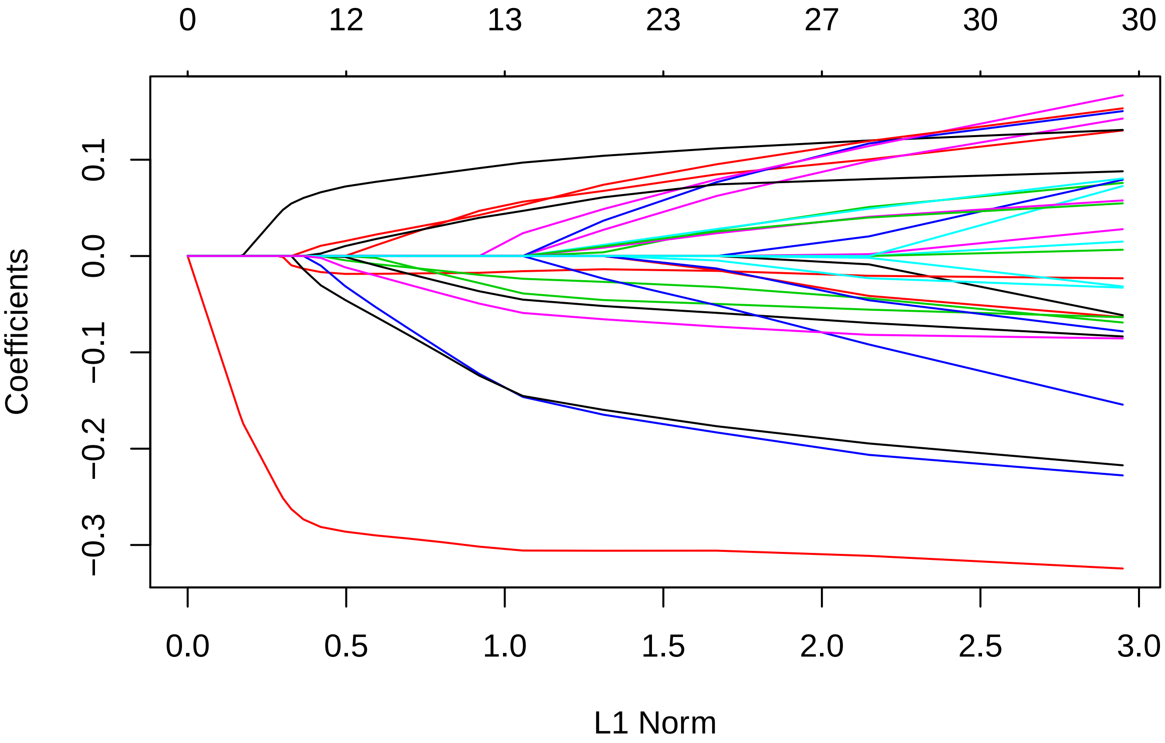
```r
x <- as.matrix(x_Crohn)
y <- y_Crohn_numeric
```

Here, y is numeric, because *glmnet()* requires y to be numeric.

```r
# CLR transformation Z=log(x)
z <- log(x)
clrx <- apply(z, 2, function(x) x-rowMeans(z))
# rowMeans(clrx)
```

x (the matrix of microbiome abundances) is then CLR transformed.

```r
clrlasso <- glmnet(clrx, y, standardize = FALSE, alpha = 1,family = "binomial",
                   lambda = seq(0.015, 2, 0.01))
plot(clrlasso)
```



```
#print(clrlasso)
clrlasso_coef <- coef(clrlasso, s = 0.10)
sum(abs(clrlasso_coef) > 0)
```

```
## [1] 13
```

```
selected_clrlasso <- which(as.numeric(abs(coef(clrlasso, s = 0.1))) > 0)[-1];
```

We obtain the indices of selected variables (abs(coef) > 0) from lasso.

```
selected_clrlasso <- selected_clrlasso - 1
taxa_id <- colnames(x_Crohn)[selected_clrlasso]

write.csv(data.frame(selected_clrlasso, taxa_id),
          "./Generated_datasets/results_clrlasso_Crohn12.csv")
```

The name of selected genera and their indices are also saved in a csv file.

3.2 BEME day1 data

content need to be added.

Chapter 4

Selbal: selection of balances

The last method to illustrate is selbal, see details in (Rivera-Pinto et al., 2018).

4.1 CD data

```r
x <- x_Crohn
colnames(x) <- (1:ncol(x))

y <- y_Crohn
```

For binary outcomes (logistic regression), *selbal()* requires y to be factor. If y is numeric, *selbal()* implements linear regression.

```r
selbal_Crohn <- selbal(x = x, y = y, logt=T, maxV=12)
```

We use **A**rea **U**nder the receiver operating characteristic **C**urve (AUC) to deciede what is the optimal number of variables. Once the first two-taxon balance is selected, the algorithm performs a forward selection process. At each step, a new taxon is added to the existing balance such that AUC is improved, the algorithm stops when there is no additional variable that increase the current AUC or when the maximun number of variables to be included in the balance is achieved.

In Figure 4.1, the two groups of taxa that form the global balance are specified at the top of the plot. The box plot represents the distribution of the balance scores for CD and non-CD individuals. The right part of the figure contains the ROC curve with its AUC value and the density curve for each group.

```r
selected_selbal <- as.numeric(c(selbal_Crohn[[6]][,1], selbal_Crohn[[6]][,2]))
id.na <- which(is.na(selected_selbal))
selected_selbal <- selected_selbal[-id.na]
selected_selbal <- as.character(selected_selbal)

columns_selected_selbal <- which(colnames(x)%in% selected_selbal)
taxa_id <- colnames(x_Crohn)[columns_selected_selbal]

write.csv(data.frame(columns_selected_selbal, taxa_id),
          "./Generated_datasets/results_selbal_Crohn12.csv")
```

The name of selected genera and their indices are also saved as the other method did.

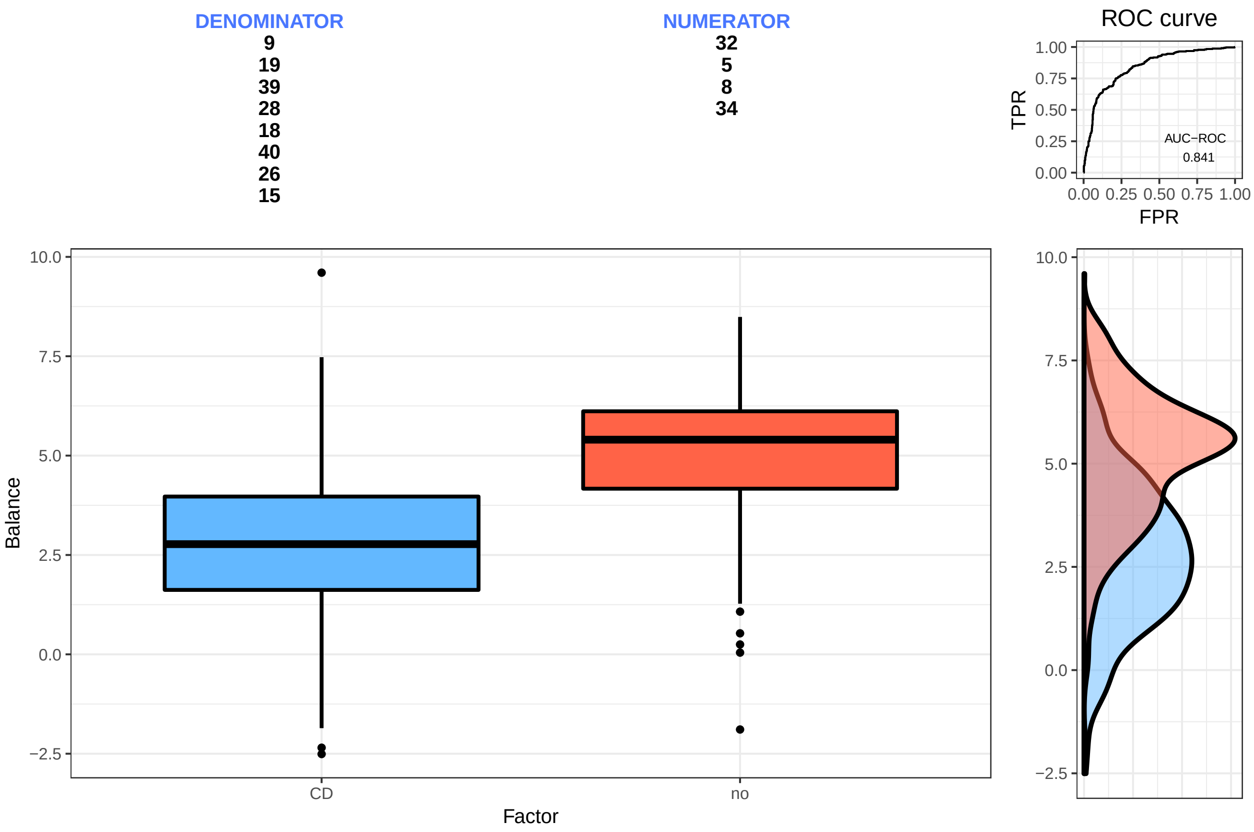


Figure 4.1: Description of the global balance for CD.

4.2 BEME day1 data

content need to be added.

Chapter 5

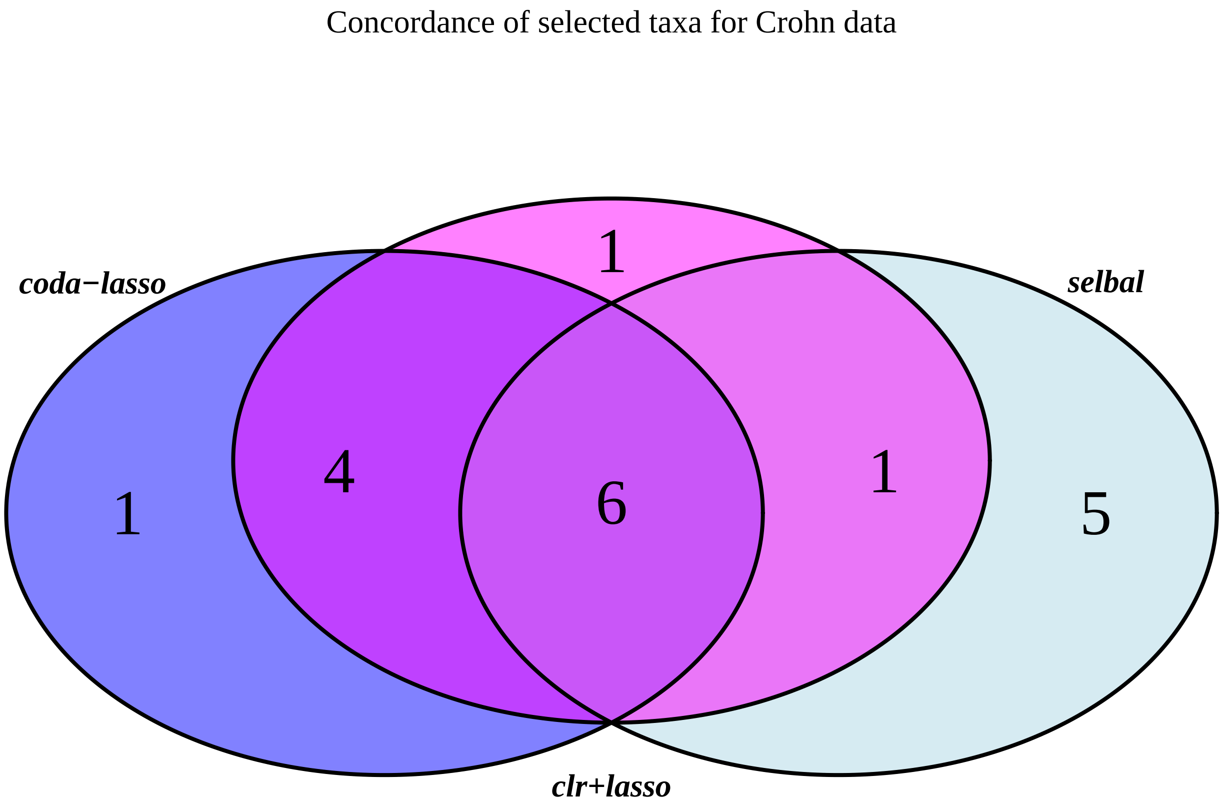
Concordance of variables selected by the three methods

5.1 CD data

```r
d_selbal <- read.csv("./Generated_datasets/results_selbal_Crohn12.csv", header = T)
d_clrlasso <- read.csv("./Generated_datasets/results_clrlasso_Crohn12.csv", header = T)
d_codalasso <- read.csv("./Generated_datasets/results_codalasso_Crohn12.csv", header = T)

taxa_selected <- list(d_clrlasso[,2], d_codalasso[,2], d_selbal[,2])
taxa.id_selected <- list(d_clrlasso[,3], d_codalasso[,3], d_selbal[,3])


venn.plot <- venn.diagram(taxa_selected, NULL, fill = c("magenta", "blue", "lightblue"),
                          alpha=c(0.5,0.5,0.5), cex = 2, cat.fontface=4,
                          category.names = c("clr+lasso", "coda-lasso", "selbal"),
                          main = "Concordance of selected taxa for Crohn data")
grid.draw(venn.plot)
```



Concordance of selected taxa for Crohn data

```
taxa.id <- venn(taxa.id_selected, show.plot=FALSE)
taxa <- venn(taxa_selected, show.plot=FALSE)

inters_taxa.id <- attr(taxa.id, "intersections")
inters_taxa <- attr(taxa, "intersections")

lapply(inters_taxa, head)
```

```
## $A
## [1] "10"
##
## $B
## [1] "27"
##
## $C
## [1] "15" "18" "26" "28" "34"
##
## $`A:B`
## [1] "2"  "31" "33" "48"
##
## $`A:C`
## [1] "8"
##
## $`A:B:C`
## [1] "5"  "9"  "19" "32" "39" "40"
```

```
lapply(inters_taxa.id, head)
```

```
## $A
## [1] "g__Faecalibacterium"
##
## $B
## [1] "o__Lactobacillales_g__"
##
## $C
## [1] "g__Blautia"           "g__Dorea"             "g__Oscillospira"
## [4] "g__Adlercreutzia"     "o__Clostridiales_g__"
##
## $`A:B`
## [1] "g__Parabacteroides" "g__Prevotella"      "g__Lachnospira"
## [4] "g__Bilophila"
##
## $`A:C`
## [1] "g__Bacteroides"
##
## $`A:B:C`
## [1] "f__Peptostreptococcaceae_g__" "g__Eggerthella"
## [3] "g__Dialister"                 "g__Roseburia"
## [5] "g__Streptococcus"             "g__Aggregatibacter"
```

need more explanation of the venn diagram.

5.2 BEME day1 data

content need to be added.

Bibliography

Gevers, D., Kugathasan, S., Denson, L. A., Vázquez-Baeza, Y., Van Treuren, W., Ren, B., Schwager, E., Knights, D., Song, S. J., Yassour, M., et al. (2014). The treatment-naive microbiome in new-onset crohn's disease. *Cell host & microbe*, 15(3):382–392.

Øyri, S. F., Műzes, G., and Sipos, F. (2015). Dysbiotic gut microbiome: a key element of crohn's disease. *Comparative immunology, microbiology and infectious diseases*, 43:36–49.

Rivera-Pinto, J., Egozcue, J., Pawlowsky-Glahn, V., Paredes, R., Noguera-Julian, M., and Calle, M. (2018). Balances: a new perspective for microbiome analysis. *MSystems*, 3(4):e00053–18.