Variable selection in microbiome compositional data analysis: tutorial

Antoni Susin, Yiwen Wang, Kim-Anh Lê Cao, M.Luz Calle

2019-10-14

Contents

1 Introduction **3**
1.1 Packages installation and loading . . . 3
1.2 Example datasets . . . 4
1.2.1 Crohn's disease . . . 4
1.2.2 High fat high sugar diet in mice . . . 4

2 CoDA-lasso **5**
2.1 CD data . . . 5
2.2 HFHS-Day1 data . . . 6

3 CLR-lasso **8**
3.1 CD data . . . 8
3.2 HFHS-Day1 data . . . 9

4 Selbal: selection of balances **12**
4.1 CD data . . . 12
4.2 HFHS-Day1 data . . . 12

5 Concordance of variables selected by the three methods **14**
5.1 CD data . . . 14
5.1.1 UpSetR . . . 14
5.1.2 Selbal-like plot . . . 15
5.1.3 plotLoadings . . . 17
5.1.4 Trajectory plots . . . 20
5.2 HFHS-Day1 data . . . 20
5.2.1 UpSetR . . . 20
5.2.2 Selbal-like plot . . . 22
5.2.3 plotLoadings . . . 24
5.2.4 Trajectory plots . . . 26
5.2.5 GraPhlAn . . . 28

Chapter 1

Introduction

This vignette provides a comparison of three multivariate methods for microbiome variable selection that consider the compositional structure of microbiome data:

- CoDA-lasso: penalized regression with constraint coefficients (regression coefficients sum up to zero) (Lu et al., 2019; Lin et al., 2014);
- CLR-lasso: penalized regression after the centered log-ratio (CLR) transformation (Zou and Hastie, 2005; Tibshirani, 1996; Le Cessie and Van Houwelingen, 1992);
- selbal: forward selection to identify the balance between two groups of taxa that is more associated with the response variable (Rivera-Pinto et al., 2018).

1.1 Packages installation and loading

Install then load the following packages:

```
# cran.packages <- c('knitr', 'glmnet', 'ggplot2', 'gridExtra',
#                    'UpSetR', 'ggforce', 'pROC')
# install.packages(cran.packages)
# devtools::install_github(repo = 'UVic-omics/selbal')

library(knitr) # rbookdown, kable
library(glmnet) # glmnet
library(selbal) # selbal
library(ggplot2) # draw selbal
library(gridExtra) # grid.arrange
library(UpSetR) # upset
library(ggforce) # selbal-like plot
library(pROC) # ROC curve

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

1.2 Example datasets

1.2.1 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 used data from a large pediatric CD cohort study (Gevers et al., 2014) to compare the proposed methodologies for identification of microbial signatures.

The microbiome data were generated using 16S rRNA gene sequencing with QIIME 1.7.0 bioinformatics processing, and the processed data 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 (see Table 1.1).

Load the data as follows:

```
load('./datasets/CD_data.RData')
```

1.2.2 High fat high sugar diet in mice

The study was conducted by Dr Lê Cao at the University of Queensland Diamantina Institute that investigated the effect of diet in mice. C57/B6 female black mice were housed in cages (3 animals per cage and fed with a high fat high sugar diet (HFHS) or a normal diet). Stool sampling was performed at Day 0, 1, 4 and 7. Illumina MiSeq sequencing was used to obtain the 16S rRNA sequencing data. The sequencing data were then processed with QIIME 1.9.0. For our analysis, we considered Day 1 only (HFHS-Day1). The OTU (Operational Taxonomy Units) table after OTU filtering included 558 taxa and 47 samples (24 HFHS diet and 23 normal diet) (see Table 1.1). Taxonomy information from Family, Genus, and Species are also available and reported here.

```
load('./datasets/HFHS_data.RData')
```

Table 1.1: **Summary of case studies**

CD data		HFHS-Day1 data	
No. of genera	48	No. of OTUs	558
No. of samples	975	No. of samples	47
No. of patients with CD	662	No. of mice with HFHS diet	24
No. of healthy patients	313	No. of mice with normal diet	23

Chapter 2

CoDA-lasso

First, we illustrate the method **CoDA-lasso**, which is the penalized regression with constraint coefficients (regression coefficients sum up to zero) (Lu et al., 2019; Lin et al., 2014) in compositional data analysis (CoDA). This method was implemented in the function *coda_lasso()*, which was adapted from matlab. In this chapter, we are going to use function *rangLambda2()* and *coda_lasso()*. Both of them have been loaded through file **functions.R**.

2.1 CD data

In CD data, the matrix of microbial absolute abundances (counts) are treated as input **X**. CD data have 975 samples and 48 taxa (genera). The rows of **X** are individuals/samples, the columns are taxa.

```
dim(CD.x)
```

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

For CoDA-lasso, **X** can either be absolute abundances or relative abundances (proportions). However, **X** should not be the matrix of log(counts) or log(proportions), because the method itself performs the log-transformation of the abundances.

Y is a binary outcome, coded as a factor with two levels (CD vs. non-CD).

```
class(CD.y)
```

```
## [1] "factor"
```

```
summary(CD.y)
```

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

To run the method, we need a value of λ. λ is the penalization parameter: the larger the value of λ, the fewer number of variables will be selected.

The default initial λ is **lambdaIni = 1**. When $\lambda = 1$, there is no microbial variable will be selected.

```
rangLambda2(Y = CD.y, X = CD.x, numLambda = 12, lambdaIni = 0.2)
```

```
## $call
## rangLambda2(Y = CD.y, X = CD.x, numLambda = 12, lambdaIni = 0.2)
##
## $ranglambdas
##           lambda numVarSelect
##  [1,] 0.20000000            8
```

```
##  [2,] 0.18181818           11
##  [3,] 0.16363636           20
##  [4,] 0.14545455           20
##  [5,] 0.12727273           16
##  [6,] 0.10909091           29
##  [7,] 0.09090909           20
##  [8,] 0.07272727           23
##  [9,] 0.05454545           26
## [10,] 0.03636364           31
## [11,] 0.01818182           41
## [12,] 0.00000000           48
```

It provides a range of λ values (**lambda**) and corresponding number of microbial variables selected (**numVarSelect**) according to a given number of λs (**numLambda**).

```
CD.results_codalasso <- coda_lasso(Y = CD.y, X = CD.x, lambda = 0.19)
CD.results_codalasso$numVarSelect
```

```
## [1] 11
```

```
CD.results_codalasso$varSelect
```

```
##  [1] "g__Roseburia"                "g__Dialister"
##  [3] "g__Streptococcus"            "g__Aggregatibacter"
##  [5] "f__Peptostreptococcaceae_g__" "g__Eggerthella"
##  [7] "o__Lactobacillales_g__"      "g__Lachnospira"
##  [9] "g__Parabacteroides"          "g__Prevotella"
## [11] "g__Bilophila"
```

The method selects 11 genera with $\lambda = 0.19$.

2.2 HFHS-Day1 data

The analysis on HFHS-Day1 data is similar as CD data.

```
dim(HFHS.x)
```

```
## [1]  47 558
```

```
class(HFHS.y)
```

```
## [1] "factor"
```

```
summary(HFHS.y)
```

```
##   HFHS Normal
##     24     23
```

In HFHS-Day1 data, **Y** is also a factor with two levels (HFHS vs. Normal)

We then test a range of λ and estimate the number of selected OTUs for each λ.

```
rangLambda2(Y = HFHS.y, X = HFHS.x, numLambda = 20, lambdaIni = 0.3)
```

```
## $call
## rangLambda2(Y = HFHS.y, X = HFHS.x, numLambda = 20, lambdaIni = 0.3)
##
## $ranglambdas
##           lambda numVarSelect
```

```
##  [1,] 0.30000000            2
##  [2,] 0.28421053            2
##  [3,] 0.26842105            2
##  [4,] 0.25263158            2
##  [5,] 0.23684211           35
##  [6,] 0.22105263            2
##  [7,] 0.20526316            2
##  [8,] 0.18947368           11
##  [9,] 0.17368421            2
## [10,] 0.15789474           33
## [11,] 0.14210526            7
## [12,] 0.12631579           10
## [13,] 0.11052632            2
## [14,] 0.09473684           50
## [15,] 0.07894737            5
## [16,] 0.06315789           17
## [17,] 0.04736842           20
## [18,] 0.03157895          230
## [19,] 0.01578947          106
## [20,] 0.00000000          558
```

We specify a λ and run the CoDA-lasso.

```
HFHS.results_codalasso <- coda_lasso(Y = HFHS.y, X = HFHS.x, lambda = 0.18)
HFHS.results_codalasso$numVarSelect
```

```
## [1] 11
```

```
HFHS.results_codalasso$varSelect
```

```
##  [1] "192222"  "401384"  "263479"  "348038"  "400599"  "1105860" "4379961"
##  [8] "338796"  "407963"  "462764"  "175272"
```

The method selects 11 OTUs with $\lambda = 0.18$.

We also extract the taxonomic information of these selected OTUs.

```
HFHS.tax_codalasso <- HFHS.taxonomy[which(rownames(HFHS.taxonomy) %in%
                                            HFHS.results_codalasso$varSelect), ]
kable(HFHS.tax_codalasso[ ,2:6], booktabs = T)
```

	Phylum	Class	Order	Family	Genus
192222	Bacteroidetes	Bacteroidia	Bacteroidales	Prevotellaceae	
1105860	Firmicutes	Erysipelotrichi	Erysipelotrichales	Erysipelotrichaceae	Allobaculum
4379961	Firmicutes	Erysipelotrichi	Erysipelotrichales	Erysipelotrichaceae	Allobaculum
338796	Firmicutes	Clostridia	Clostridiales	Ruminococcaceae	Oscillospira
462764	Firmicutes	Clostridia	Clostridiales	Ruminococcaceae	Ruminococcus
263479	Bacteroidetes	Bacteroidia	Bacteroidales	S24-7	
400599	Firmicutes	Clostridia	Clostridiales		
407963	Firmicutes	Clostridia	Clostridiales	Ruminococcaceae	Oscillospira
348038	Bacteroidetes	Bacteroidia	Bacteroidales	S24-7	
401384	Firmicutes	Clostridia	Clostridiales	Ruminococcaceae	Oscillospira
175272	Bacteroidetes	Bacteroidia	Bacteroidales	S24-7	

Chapter 3

CLR-lasso

Then, we illustrate **CLR-lasso**, which is the penalized regression after the centered log-ratio (CLR) transformation (Zou and Hastie, 2005; Tibshirani, 1996; Le Cessie and Van Houwelingen, 1992). This method was applied based on function *glmnet()*. We also generated a wrapper function called *glmnet_wrapper()*. It is based on *glmnet()*, but provides additional outputs. All these functions are uploaded via **functions.R**.

3.1 CD data

As mentioned in the paper, CLR transformation is each log-transformed value subtracted with the arithmetic mean of the log-transformed values.

$$clr(x) = clr(x_1, ..., x_k) = (log(x_1) - M, ..., log(x_k) - M)$$

$$M = \frac{1}{k} \sum_{i=1}^{k} log(x_i)$$

Where $i = 1, ..., k$ microbial variables, x_k is the counts of variable k, M is the arithmetic mean of the log-transformed values.

```
# CLR transformation
CD.z <- log(CD.x)
CD.clrx <- apply(CD.z, 2, function(x) x - rowMeans(CD.z))
```

```
CD.y.num <- as.numeric(CD.y)
```

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

Same as CoDA-lasso, we first need to choose a λ. We fit CLR transformed **X** and numeric **Y** as input of function *glmnet()*. We also need to specify the response family to be **binomial**.

```
CD.test_clrlasso <- glmnet(x = CD.clrx, y = CD.y.num, family = 'binomial')
plot(CD.test_clrlasso, xvar = 'lambda', label = T)
```

In Figure 3.1, each curve corresponds to a variable (e.g. genus). It shows the path of its coefficient against different $log(\lambda)$ values. At each $log(\lambda)$, the shown curves indicate the number of nonzero coefficients. In the plot command, if **label = T**, each curve will be annotated with variable index.

Once we have chosen the λ value, it is input in our new wrapper function *glmnet_wrapper()*.

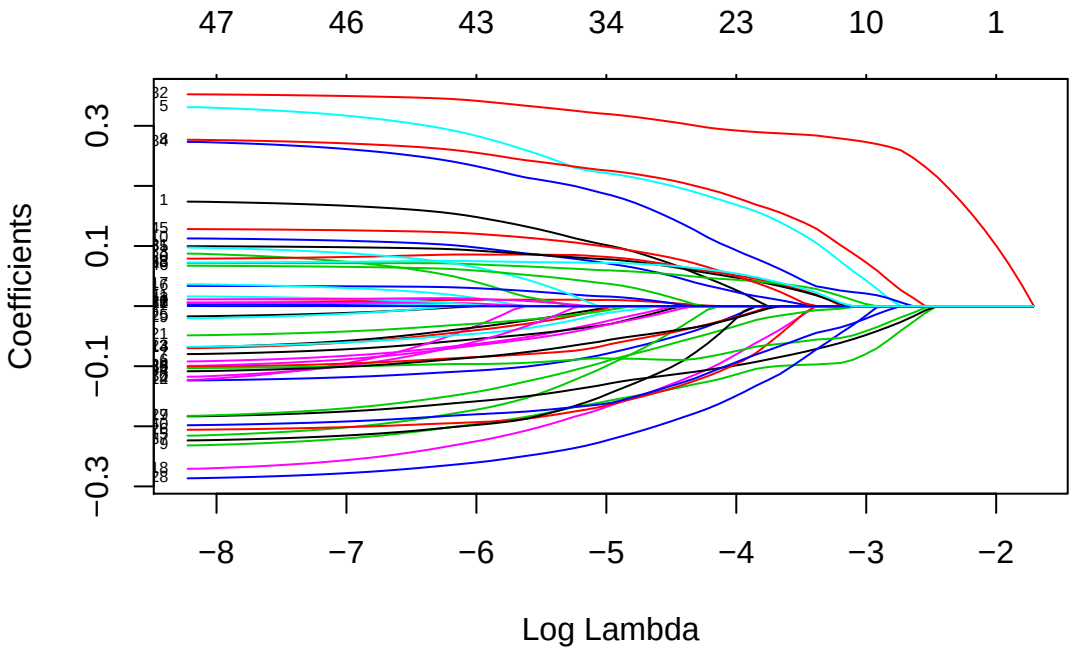

Figure 3.1: Lasso plot of CD data

```
CD.lambda_clr = 0.045
CD.results_clrlasso <- glmnet_wrapper(Y = CD.y.num, X = CD.clrx, family = 'binomial',
                                      lambda = CD.lambda_clr)
CD.results_clrlasso$numVarSelect
```

```
## [1] 11
```

```
CD.results_clrlasso$varSelect
```

```
##  [1] "g__Roseburia"                "g__Eggerthella"
##  [3] "g__Bacteroides"              "f__Peptostreptococcaceae_g__"
##  [5] "g__Dialister"                "g__Streptococcus"
##  [7] "g__Adlercreutzia"            "g__Aggregatibacter"
##  [9] "o__Clostridiales_g__"        "g__Lachnospira"
## [11] "o__Lactobacillales_g__"
```

The method selects 11 genera with $\lambda = 0.045$, and they are listed in the object **CD.results_clrlasso**.

3.2 HFHS-Day1 data

The analysis on HFHS-Day1 data is as similar as CD data.

```
# CLR transformation
HFHS.z <- log(HFHS.x)
HFHS.clrx <- apply(HFHS.z, 2, function(x) x-rowMeans(HFHS.z))
```

X is centered log-ratio (CLR) transformed.

```
HFHS.y.num <- as.numeric(HFHS.y)
```

Y is converted to a numeric vector.

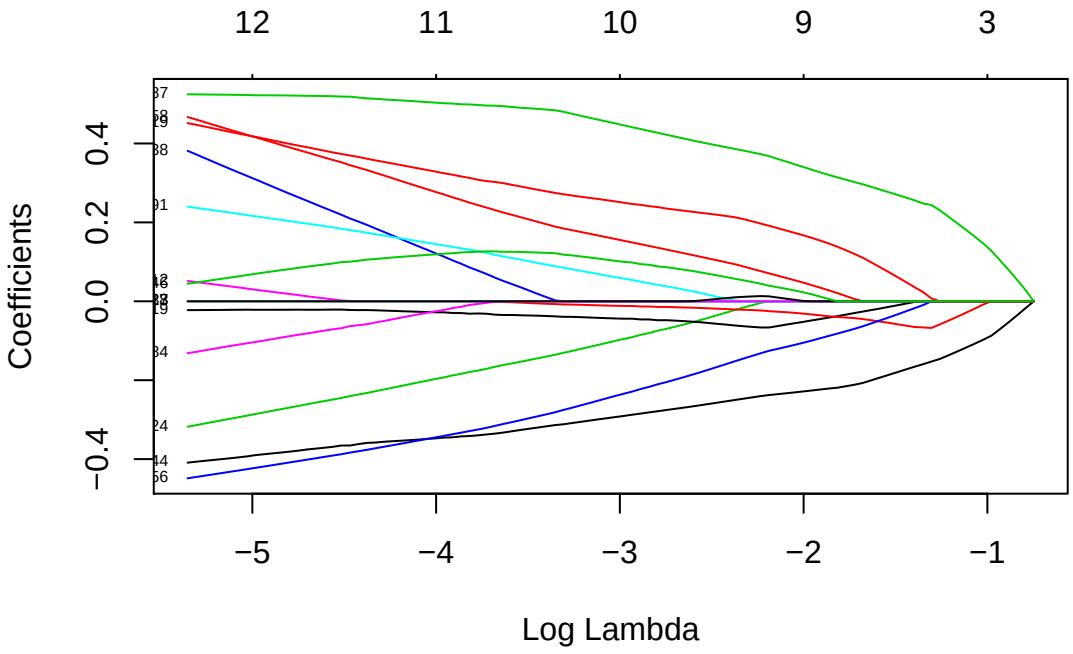

Figure 3.2: Lasso plot of HFHS-Day1 data

```
HFHS.test_clrlasso <- glmnet(x = HFHS.clrx, y = HFHS.y.num, family = 'binomial')
plot(HFHS.test_clrlasso, xvar = 'lambda', label = T)
```

The explanation of Figure 3.2 is the same as Figure 3.1.

Once we have chosen the λ value, we use function *glmnet_wrapper()* with the same input as *glmnet()* and extra input λ.

```
HFHS.lambda_clr = 0.03
HFHS.results_clrlasso <- glmnet_wrapper(Y = HFHS.y.num, X = HFHS.clrx, family = 'binomial',
                                        lambda = HFHS.lambda_clr)
HFHS.results_clrlasso$numVarSelect
```

```
## [1] 10
```

```
HFHS.results_clrlasso$varSelect
```

```
##  [1] "400599"  "192222"  "348038"  "401384"  "290253"  "261265"  "300851"
##  [8] "462764"  "1108745" "265322"
```

The method selects 10 OTUs with $\lambda = 0.03$, and these OTUs are listed in the object **HFHS.results_clrlasso**.

We also extract the taxonomic information of these selected OTUs.

```
HFHS.tax_clrlasso <- HFHS.taxonomy[which(rownames(HFHS.taxonomy) %in%
                                           HFHS.results_clrlasso$varSelect), ]
kable(HFHS.tax_clrlasso[ ,2:6], booktabs = T)
```

	Phylum	Class	Order	Family	Genus
192222	Bacteroidetes	Bacteroidia	Bacteroidales	Prevotellaceae	
290253	Firmicutes	Clostridia	Clostridiales	Ruminococcaceae	Oscillospira
261265	Firmicutes	Clostridia	Clostridiales	Lachnospiraceae	
1108745	Firmicutes	Clostridia	Clostridiales	[Mogibacteriaceae]	
462764	Firmicutes	Clostridia	Clostridiales	Ruminococcaceae	Ruminococcus
265322	Bacteroidetes	Bacteroidia	Bacteroidales	S24-7	
400599	Firmicutes	Clostridia	Clostridiales		
348038	Bacteroidetes	Bacteroidia	Bacteroidales	S24-7	
401384	Firmicutes	Clostridia	Clostridiales	Ruminococcaceae	Oscillospira
300851	Firmicutes	Clostridia	Clostridiales	Ruminococcaceae	Oscillospira

Chapter 4

Selbal: selection of balances

selbal (Rivera-Pinto et al., 2018) relies on the concept of compositional balances, which is the balance between the abundances of two groups of microbial species that is more associated with the response variable. We also generated a wrapper function called *selbal_wrapper()*. It is based on *selbal()*, but provides additional outputs. All these functions are uploaded via **functions.R**.

4.1 CD data

```
class(CD.y)
```

```
## [1] "factor"
```

selbal() requires **Y** to be factor, as it will run logistic regression with a binary outcomes. If **Y** is numeric, *selbal()* implements linear regression.

Besides input **Y** and **X**, we also need to decide how many variables to select (**maxV**). The default performance measure (**logit.acc**) to compute the correlation between **Y** and balances is **AUC**, other opitions can be **"Dev"**, **"Rsq"** or **"Tjur"**.

```
# optimization criteria Deviance
CD.results_selbal <- selbal_wrapper(Y = CD.y, X = CD.x, maxV = 12, logit.acc = 'Dev')
CD.results_selbal$numVarSelect
```

```
## [1] 12
```

```
CD.results_selbal$varSelect
```

```
##  [1] "g__Roseburia"                "g__Eggerthella"
##  [3] "g__Dialister"                "g__Streptococcus"
##  [5] "f__Peptostreptococcaceae_g__" "g__Bacteroides"
##  [7] "g__Aggregatibacter"          "g__Adlercreutzia"
##  [9] "g__Dorea"                    "g__Oscillospira"
## [11] "o__Clostridiales_g__"        "g__Blautia"
```

The method selects 12 genera as we required.

4.2 HFHS-Day1 data

The analysis on HFHS-Day1 data is as similar as CD data.

First, we need to check if **Y** is a factor.

```
class(HFHS.y)
```

```
## [1] "factor"
```

```
# optimization criteria Deviance
HFHS.results_selbal <- selbal_wrapper(Y = HFHS.y, X = HFHS.x, maxV = 2, logit.acc = 'Dev')
HFHS.results_selbal$numVarSelect
```

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

```
HFHS.results_selbal$varSelect
```

```
## [1] "290253" "263479"
```

The method selects 2 OTUs as required.

We also extract the taxonomic information of these selected OTUs.

```
HFHS.tax_selbal <- HFHS.taxonomy[which(rownames(HFHS.taxonomy) %in%
                                         HFHS.results_selbal$varSelect), ]
kable(HFHS.tax_selbal[ ,2:6], booktabs = T)
```

	Phylum	Class	Order	Family	Genus
290253	Firmicutes	Clostridia	Clostridiales	Ruminococcaceae	Oscillospira
263479	Bacteroidetes	Bacteroidia	Bacteroidales	S24-7	

Chapter 5

Concordance of variables selected by the three methods

In this chapter, we are going to use different visualisation approaches to display the variables selected by the three methods:

- **UpSet plot:** highlights overlap of the variables selected by the three methods.
- **Selbal-like plot:** lists the selected variables and displays their discriminating ability with respect to sample groups.
- **plotLoadings:** visualises the variable coefficients and sample groups each variable contributes to.
- **Trajectory plot:** represents the rank of the variables selected by CoDA-lasso and CLR-lasso, and their corresponding regression coefficients
- **GraPhlAn:** displays the taxonomic tree of selected variables (HFHS-Day1 data only).

5.1 CD data

5.1.1 UpSetR

UpSet is a visualisation technique for the quantitative analysis of sets and their intersections (Lex et al., 2014). Before we apply *upset()*, we take the list of variable vectors selected with the three methods and convert them into a data frame compatible with *upset()* using *fromList()*. We then assign different color shcemes for each variable selection.

```
CD.select <- list(CoDA_lasso = CD.results_codalasso$varSelect,
                  CLR_lasso = CD.results_clrlasso$varSelect,
                  selbal = CD.results_selbal$varSelect)


CD.select.upsetR <- fromList(CD.select)

upset(as.data.frame(CD.select.upsetR), main.bar.color = 'gray36',
      sets.bar.color = color[c(1:2,5)], matrix.color = 'gray36',
      order.by = 'freq', empty.intersections = 'on',
      queries = list(list(query = intersects, params = list('CoDA_lasso'),
                          color = color[5], active = T),
                     list(query = intersects, params = list('CLR_lasso'),
                          color = color[2], active = T),
                     list(query = intersects, params = list('selbal'),
                          color = color[1], active = T)))
```

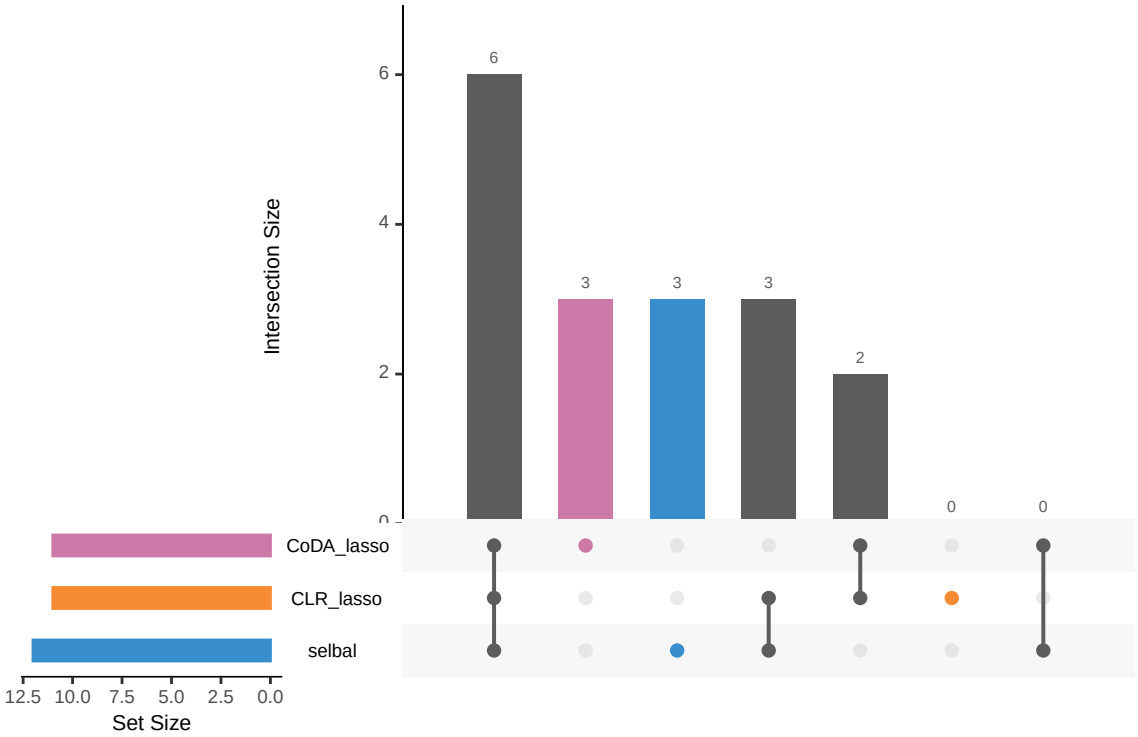

Figure 5.1: UpSet plot showing overlap between variables selected with different methods.

In Figure 5.1, the left bars show the number of variables selected by each method. The right bar plot combined with the scatterplot show different intersection situations and their aggregates. For example, in the first column, three points are linked with one line, and the intersection size of the bar is 6. This means that 6 variables are selected by all these three methods. While in the second column, 3 variables are only selected by the method CoDA-lasso.

5.1.2 Selbal-like plot

Selbal-like plot is an extension of the plot proposed by Rivera-Pinto et al. (2018).

```
# CoDA_lasso
CD.coda_pos <- CD.results_codalasso$posCoefSelect
CD.coda_neg <- CD.results_codalasso$negCoefSelect
selbal_like_plot(pos.names = names(CD.coda_pos), neg.names = names(CD.coda_neg),
                 Y = CD.y, X = CD.x)
```

In Figure 5.2, the top left panel lists the selected variable names with either negative or positive coefficients. The names are ordered according to their importance (absolute coefficient values). The top right panel is the Receiver Operating Characteristic (ROC) curve based on generalised linear model: $g(E(Y)) = \beta_0 + \beta_1 logX_1 + ... + \beta_p logX_p$ with p selected variables. The Area Under the Curve (AUC) is 0.822, which indicates the ability to discriminate the sample groups. The bottom left boxplot is based on the log mean difference between negative and positive variables: $\frac{1}{p_+}\sum_{i=1}^{p_+} logX_i - \frac{1}{p_-}\sum_{j=1}^{p_-} logX_j$. This log mean difference is calculated for each sample as a balance score, because it is proportionally equal to the balance mentioned in (Rivera-Pinto et al., 2018). The bottom right density plots represent the distributions of the log mean difference scores for CD and non-CD individuals.

```
# CLR_lasso
CD.clr_pos <- CD.results_clrlasso$posCoefSelect
CD.clr_neg <- CD.results_clrlasso$negCoefSelect
```

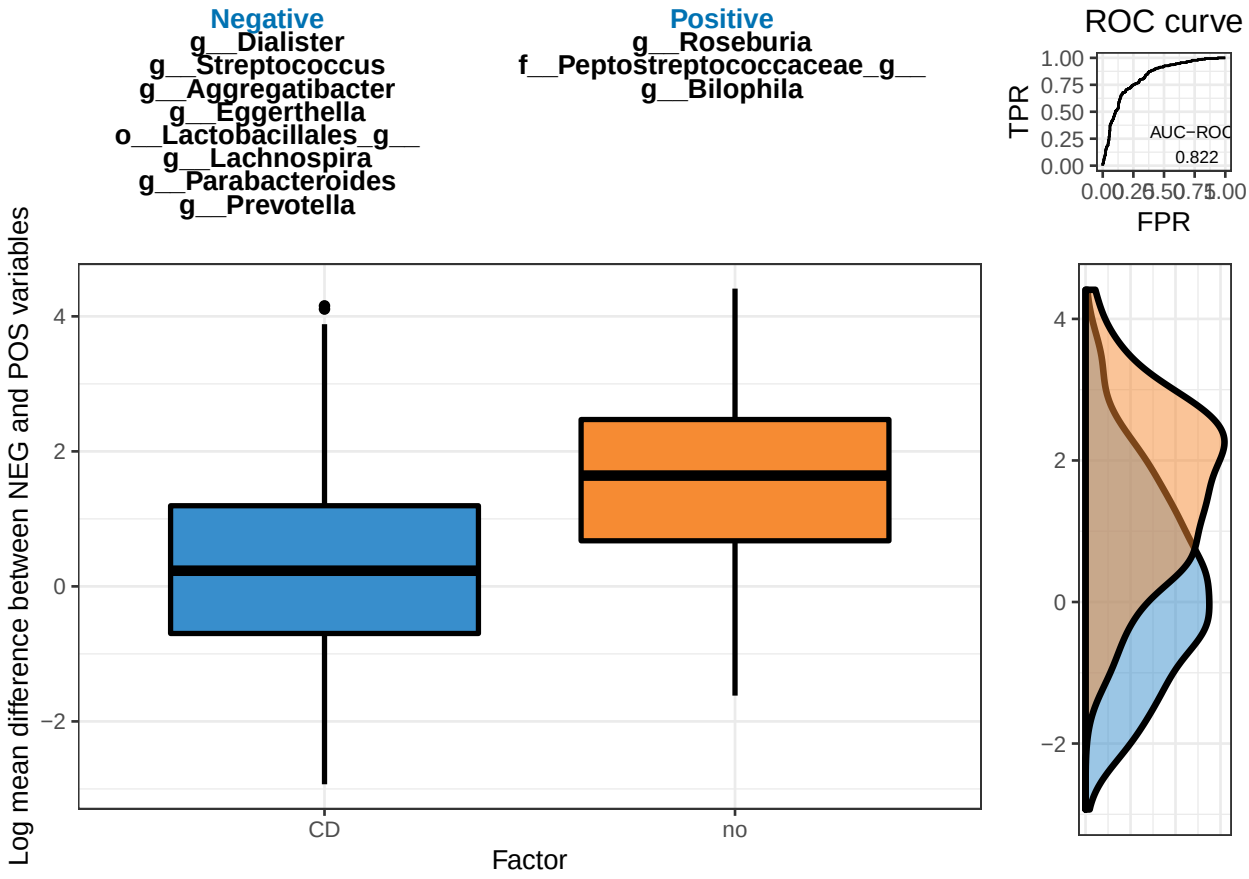

Figure 5.2: Selbal-like plot showing variables selected with method CoDA-lasso and the ability of these variables to discriminate CD and non-CD individuals.

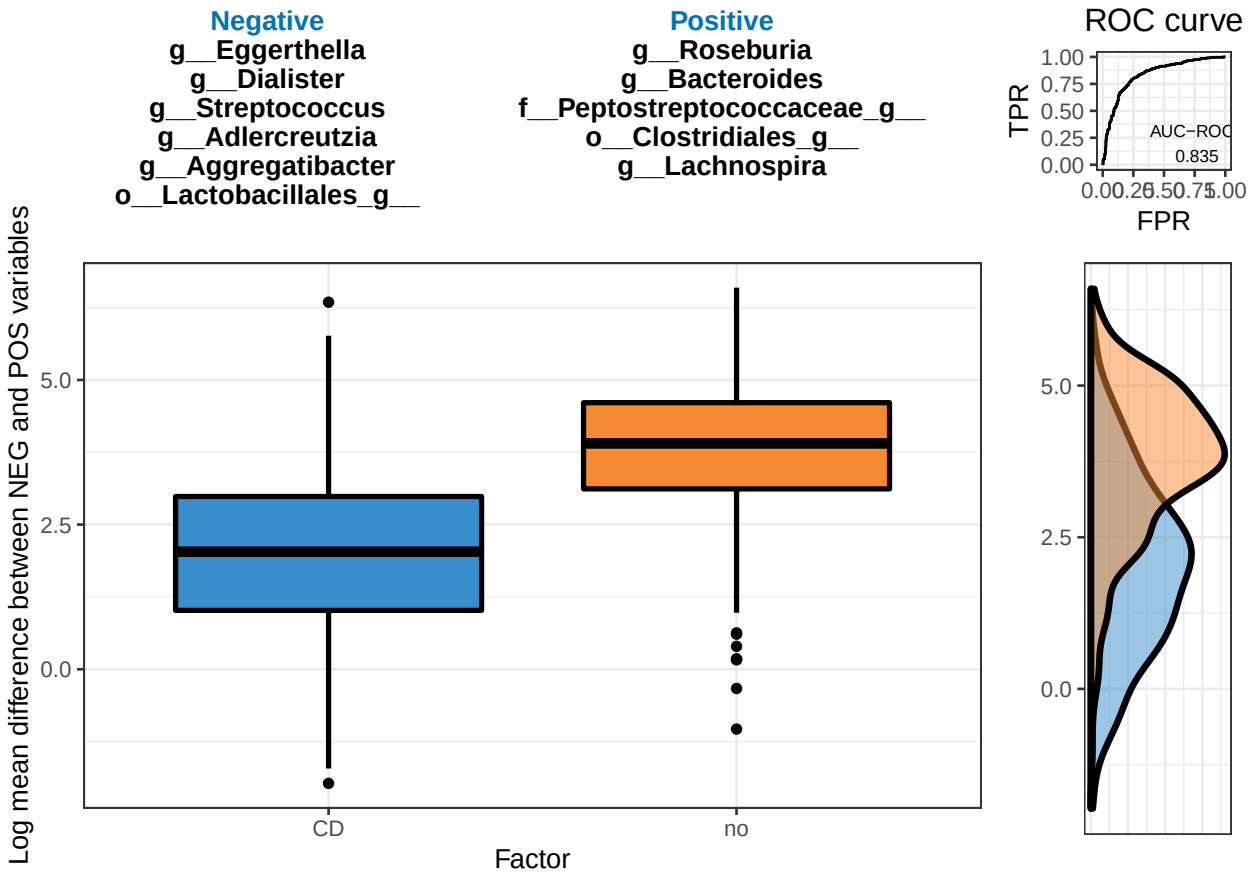

Figure 5.3: Selbal-like plot showing variables selected with method CLR-lasso and the ability of these variables to discriminate CD and non-CD individuals.

```
selbal_like_plot(pos.names = names(CD.clr_pos), neg.names = names(CD.clr_neg),
                 Y = CD.y, X = CD.x)
```

The interpretation of Figure 5.3 is the same as Figure 5.2, but with variables selected with method CLR-lasso.

```
# selbal
CD.selbal_pos <- CD.results_selbal$posVarSelect
CD.selbal_neg <- CD.results_selbal$negVarSelect
selbal_like_plot(pos.names = CD.selbal_pos, neg.names = CD.selbal_neg, Y = CD.y,
                 selbal = TRUE, FINAL.BAL = CD.results_selbal$finalBal)
```

In Figure 5.4, for the selbal method, the two groups of variables that form the global balance are specified at the top of the plot. They are equally important. 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 ($g(E(Y)) = \beta_0 + \beta_1 B(Den, Num)$) with its AUC value (0.841, higher than other methods) and the density curve for each group.

5.1.3 plotLoadings

An easier way to visualise the coefficients of the selected variables is to plot them in a barplot, as proposed by Rohart et al. (2017). We have amended the *plotLoadings()* function from the package *mixOmics* to do so. The argument **Y** specified the sample class, so that the color assigened to each variable represents the class that has maximum CLR mean value (**method = 'mean'** and **contrib.method = 'max'**).

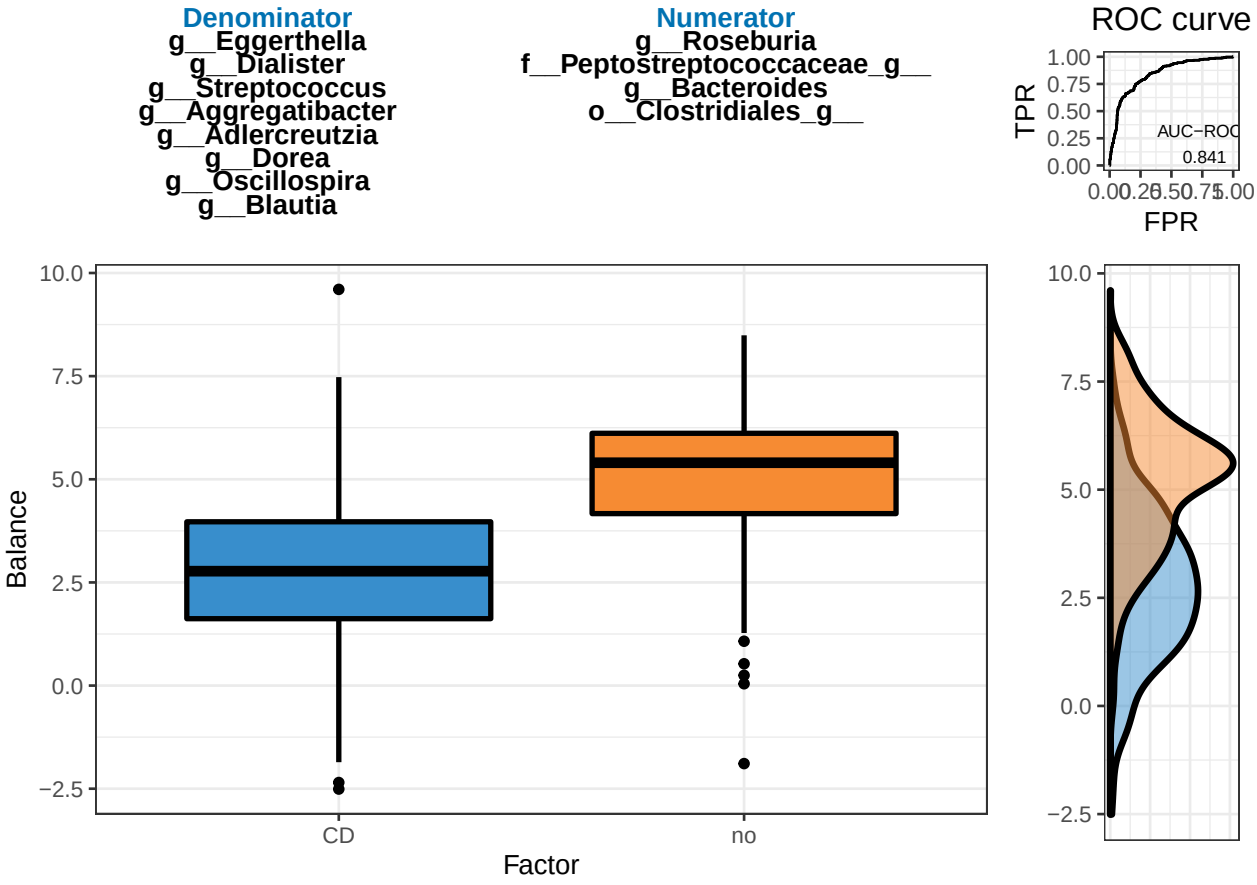

Figure 5.4: Selbal plot showing variables selected with method selbal and the ability of these variables to discriminate CD and non-CD individuals.

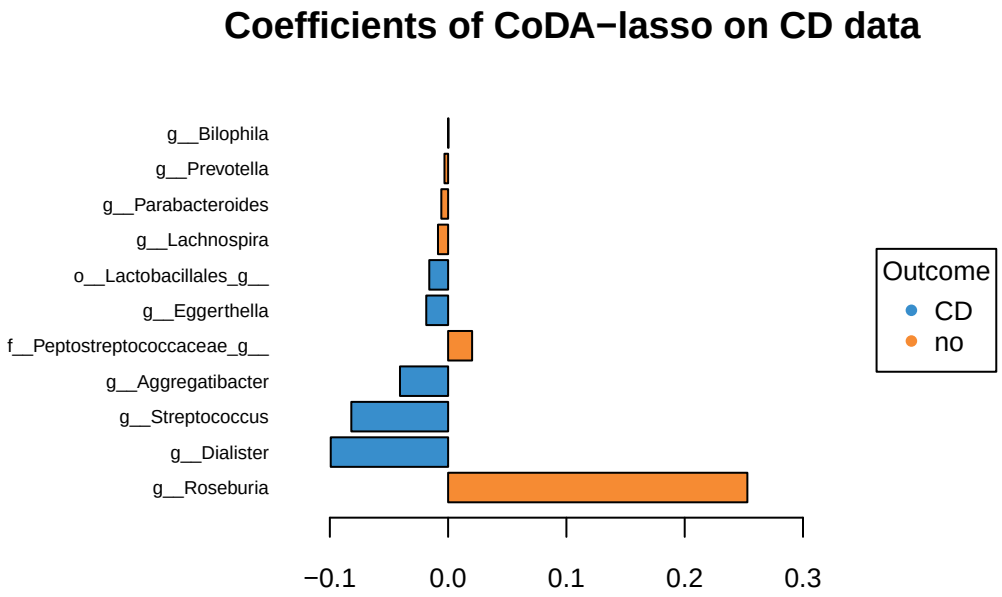

Figure 5.5: The plotLoadings of selected variables with CoDA-lasso.

```
# CoDA_lasso
CD.coda_coef <- CD.results_codalasso$coefficientsSelect
CD.coda_data <- CD.x[ ,CD.results_codalasso$varSelect]

CD.coda.plotloadings <- plotcoefficients(coef = CD.coda_coef,
                                         data = CD.coda_data,
                                         Y = CD.y,
                                         method = 'mean',
                                         contrib.method = 'max',
                                         title = 'Coefficients of CoDA-lasso on CD data')
```

Figure 5.5 shows that the taxa colored in orange have a CLR abundance greater in non-CD samples relative to CD samples (e.g. *Roseburia*), while the blue ones have a greater CLR abundance in CD samples relative to non-CD samples (e.g. *Dialister*). It is based on their mean per group (CD vs. non-CDs). The bar indicates the coefficient. As we can see, several taxa have a greater CLR abundance in non-CD group, but with a negative coefficient. It means this model is not optimal at some extent.

```
# CLR_lasso
CD.clr_coef <- CD.results_clrlasso$coefficientsSelect
CD.clr_data <- CD.x[ ,CD.results_clrlasso$varSelect]
CD.clr.plotloadings <- plotcoefficients(coef = CD.clr_coef,
                                        data = CD.clr_data,
                                        Y = CD.y,
                                        method = 'mean',
                                        contrib.method = 'max',
                                        title = 'Coefficients of clr-lasso on CD data')
```

Same as Figure 5.5, we can interpret Figure 5.6 as follows. All selected variables with a greater CLR abundance in non-CD samples have been assigned a positive coefficient. Compared to Figure 5.5, where the sign of the coefficients is not consisitent across the sample class, this may suggest the CLR-lasso is better at identifying discriminative variables than CODA-lasso. Both *Roseburia* and *Peptostreptococcaceae* selected by CoDA-lasso and CLR-lasso have a greater CLR abundance in non-CD group and were assigned with the same coeffcient rank. But both *Eggerthella* and *Dialister* selected by two methods were assigned with very different coefficient rank.

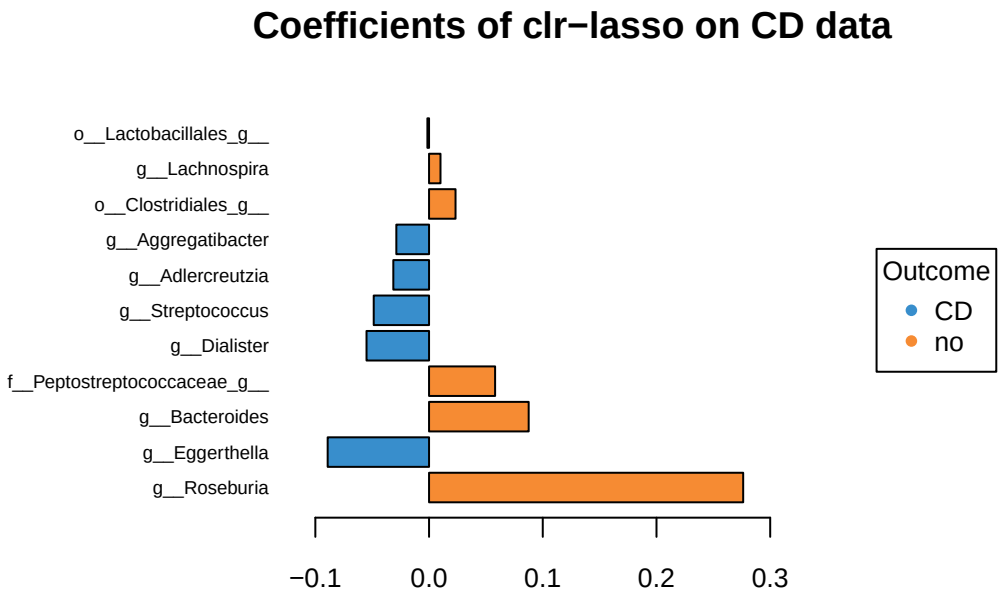

Figure 5.6: The plotLoadings of selected variables with CLR-lasso.

5.1.4 Trajectory plots

To visualise the change of variable coefficients and their ranks in the selection between different methods, we use trajectory plots.

```
TRAJ_plot(selectVar_coef_method1 = CD.coda_coef, selectVar_coef_method2 = CD.clr_coef,
          selectMethods = c('CoDA-lasso', 'CLR-lasso'))
```

Figure 5.7 shows the selected variables ordered by their rank in the selection (according to their coefficient absolute values) between CoDA-lasso and CLR-lasso, with the thickness of the lines representing the coefficient absolute values.

In this plot, we can visualise the rank change of each selected variable between CoDA-lasso and CLR-lasso selection. For example, the rank of *Dialister* is lower in CLR-lasso compared to CoDA-lasso. Moreover, we can detect the variables (e.g. *Bacteroides*) that are selected by one method (e.g. CLR-lasso) with high coefficient rank, but not selected by the other method (e.g. CoDA-lasso).

5.2 HFHS-Day1 data

Guidance on how to interpret the following plots is detailed in previous **section: CD data**.

5.2.1 UpSetR

```
HFHS.select <- list(CoDA_lasso = HFHS.results_codalasso$varSelect,
                    CLR_lasso = HFHS.results_clrlasso$varSelect,
                    selbal = HFHS.results_selbal$varSelect)

HFHS.select.upsetR <- fromList(HFHS.select)

upset(as.data.frame(HFHS.select.upsetR), main.bar.color = 'gray36',
      sets.bar.color = color[c(5,2,1)], matrix.color = 'gray36',
```

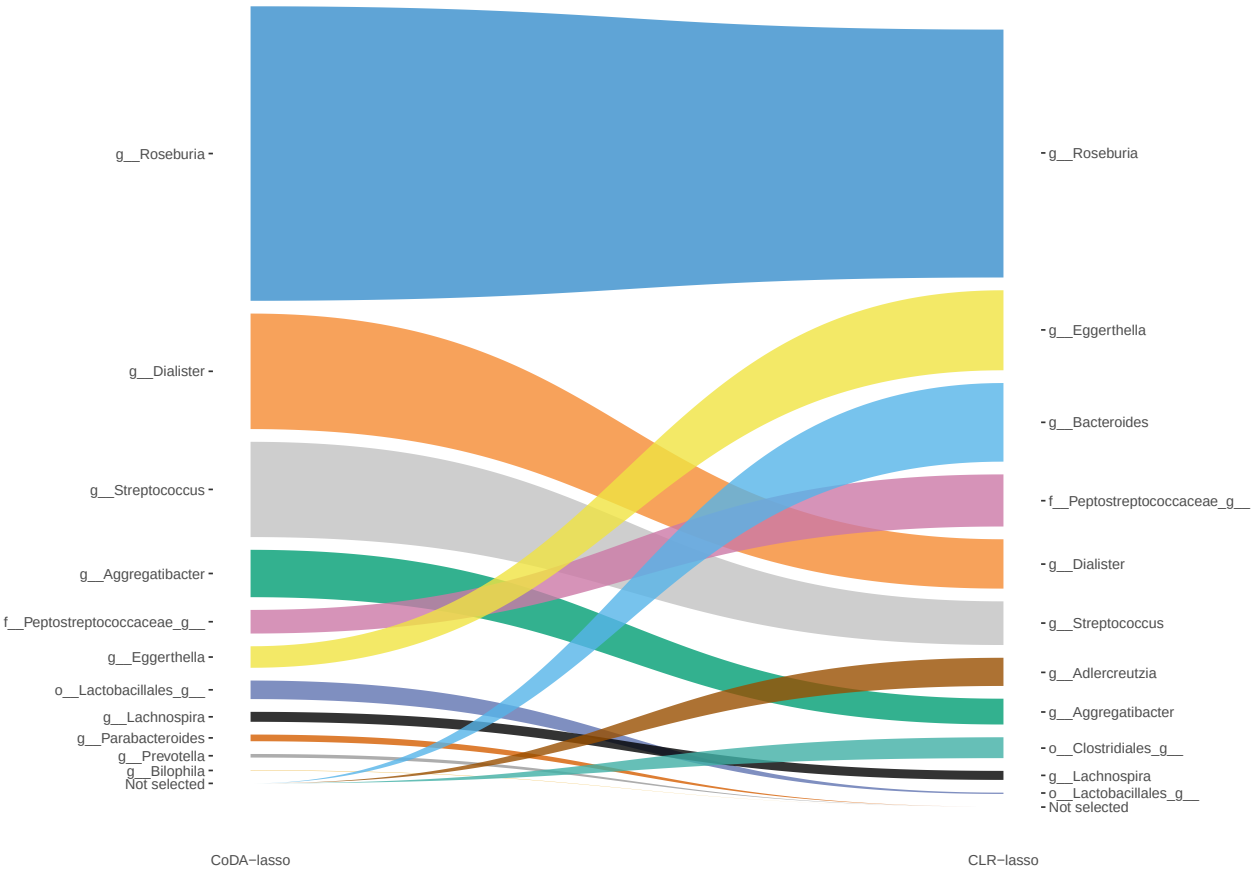

Figure 5.7: Trajectory plots of selected variables from both CoDA-lasso and CLR-lasso in CD data.

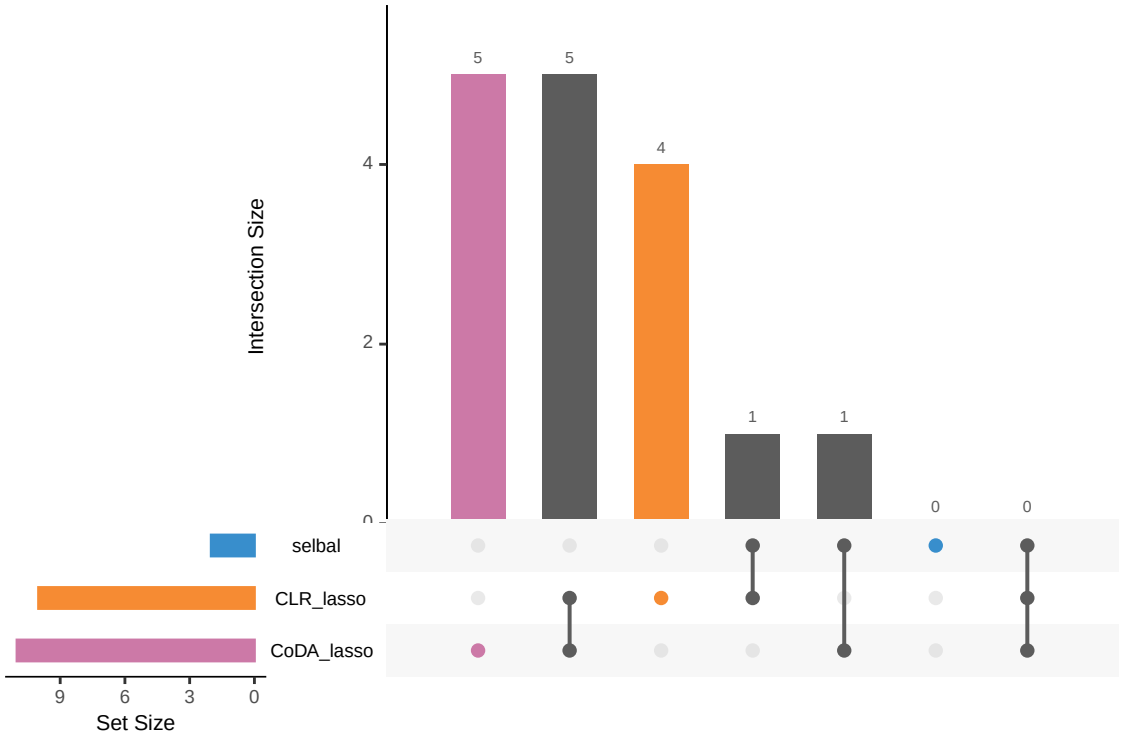

Figure 5.8: UpSet plot showing overlap between variables selected with different methods.

```
      order.by = 'freq', empty.intersections = 'on',
      queries = list(list(query = intersects, params = list('CoDA_lasso'),
                          color = color[5], active = T),
                     list(query = intersects, params = list('CLR_lasso'),
                          color = color[2], active = T),
                     list(query = intersects, params = list('selbal'),
                          color = color[1], active = T)))
```

Figure 5.8 shows that 5 OTUs are only selected with CoDA-lasso, 5 OTUs are selected both with CoDA-lasso and CLR-lasso, 4 OTUs are only selected with CLR-lasso, 1 OTUs is selected with both CLR-lasso and selbal, and 1 is selected both with CoDA-lasso and selbal. Among three methods, CoDA-lasso selected the most OTUs and selbal the least.

5.2.2 Selbal-like plot

```
# CoDA_lasso
HFHS.coda_pos <- HFHS.results_codalasso$posCoefSelect
HFHS.coda_neg <- HFHS.results_codalasso$negCoefSelect
selbal_like_plot(pos.names = names(HFHS.coda_pos), neg.names = names(HFHS.coda_neg),
                 Y = HFHS.y, X = HFHS.x, OTU = T, taxa = HFHS.taxonomy)
```

Note: *S24-7* is a family from order *Bacteroidales*.

```
# CLR_lasso
HFHS.clr_pos <- HFHS.results_clrlasso$posCoefSelect
HFHS.clr_neg <- HFHS.results_clrlasso$negCoefSelect
selbal_like_plot(pos.names = names(HFHS.clr_pos), neg.names = names(HFHS.clr_neg),
                 Y = HFHS.y, X = HFHS.x, OTU = T, taxa = HFHS.taxonomy)
```

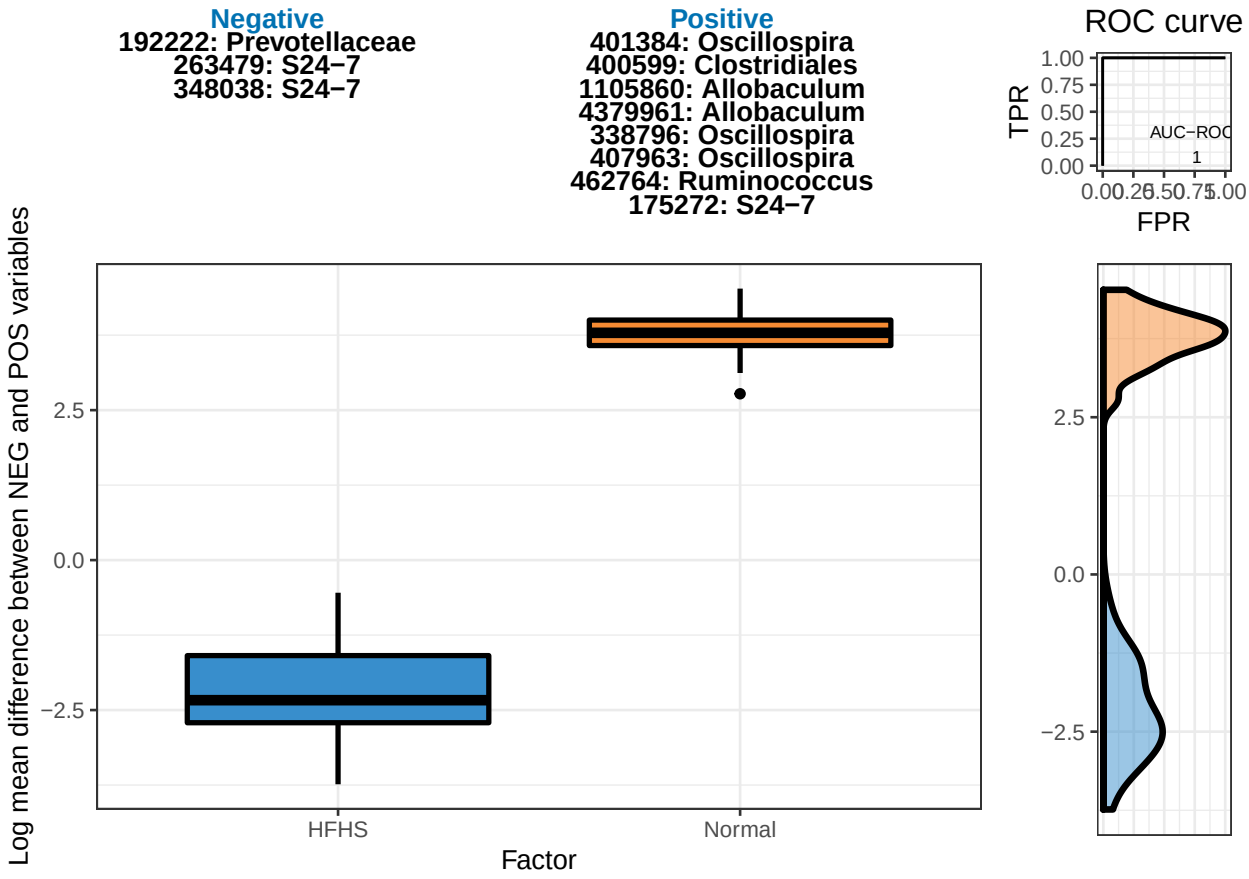

Figure 5.9: Selbal-like plot showing variables selected with method CoDA-lasso and the ability of these variables to discriminate HFHS and normal individuals.

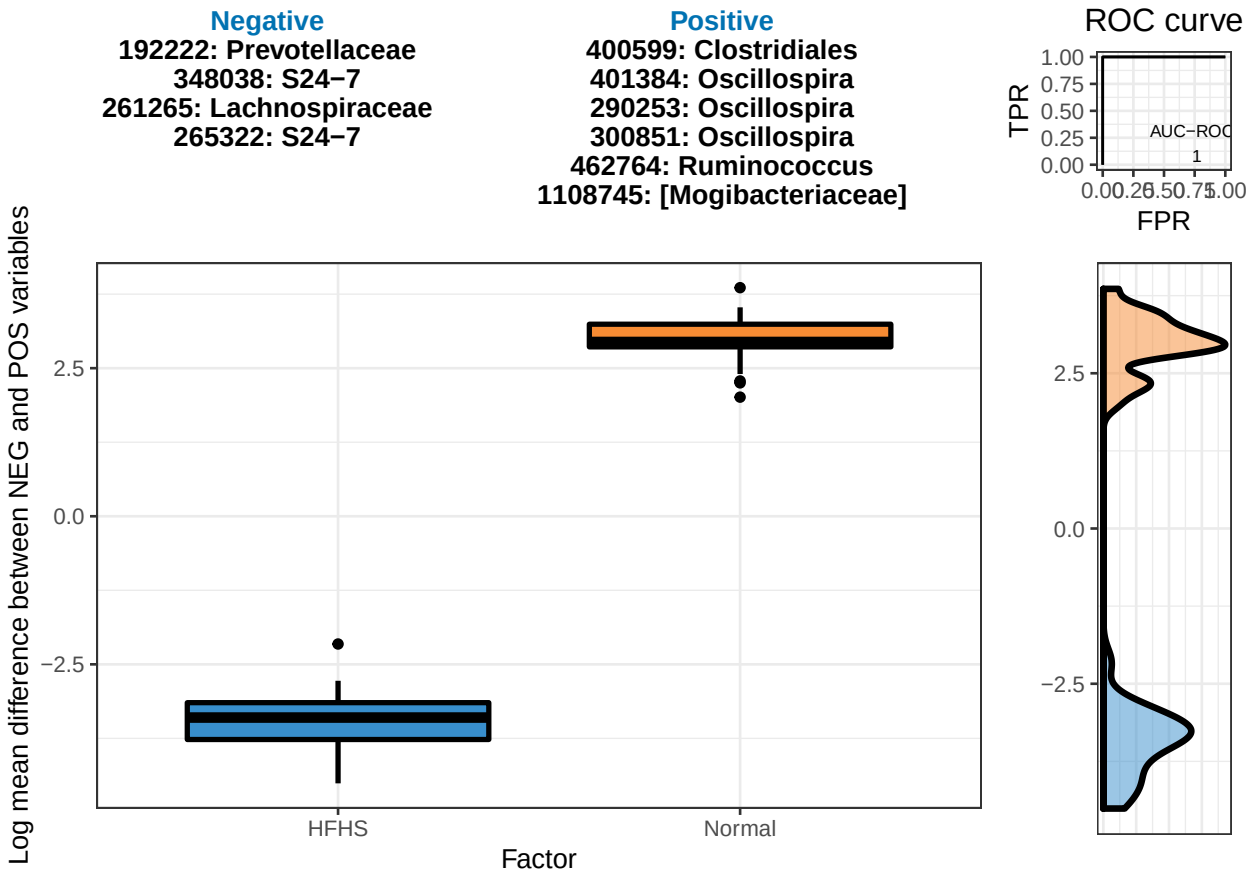

Figure 5.10: Selbal-like plot showing variables selected with method CLR-lasso and the ability of these variables to discriminate HFHS and normal individuals.

```
# selbal
HFHS.selbal_pos <- HFHS.results_selbal$posVarSelect
HFHS.selbal_neg <- HFHS.results_selbal$negVarSelect
selbal_like_plot(pos.names = HFHS.selbal_pos, neg.names = HFHS.selbal_neg, Y = HFHS.y,
                 selbal = TRUE, FINAL.BAL = HFHS.results_selbal$finalBal,
                 OTU = T, taxa = HFHS.taxonomy)
```

These plots show that the variables selected from three different methods all have a maximum discrimination (AUC = 1) between HFHS samples and normal ones. Among these methods, selbal only needs two OTUs to build a balance, it also means the association between microbiome composition and diet is very strong.

5.2.3 plotLoadings

```
# CoDA_lasso
HFHS.coda_coef <- HFHS.results_codalasso$coefficientsSelect
HFHS.coda_data <- HFHS.x[ ,HFHS.results_codalasso$varSelect]

HFHS.coda.plotloadings <- plotcoefficients(coef = HFHS.coda_coef,
                                           data = HFHS.coda_data,
                                           Y = HFHS.y,
```

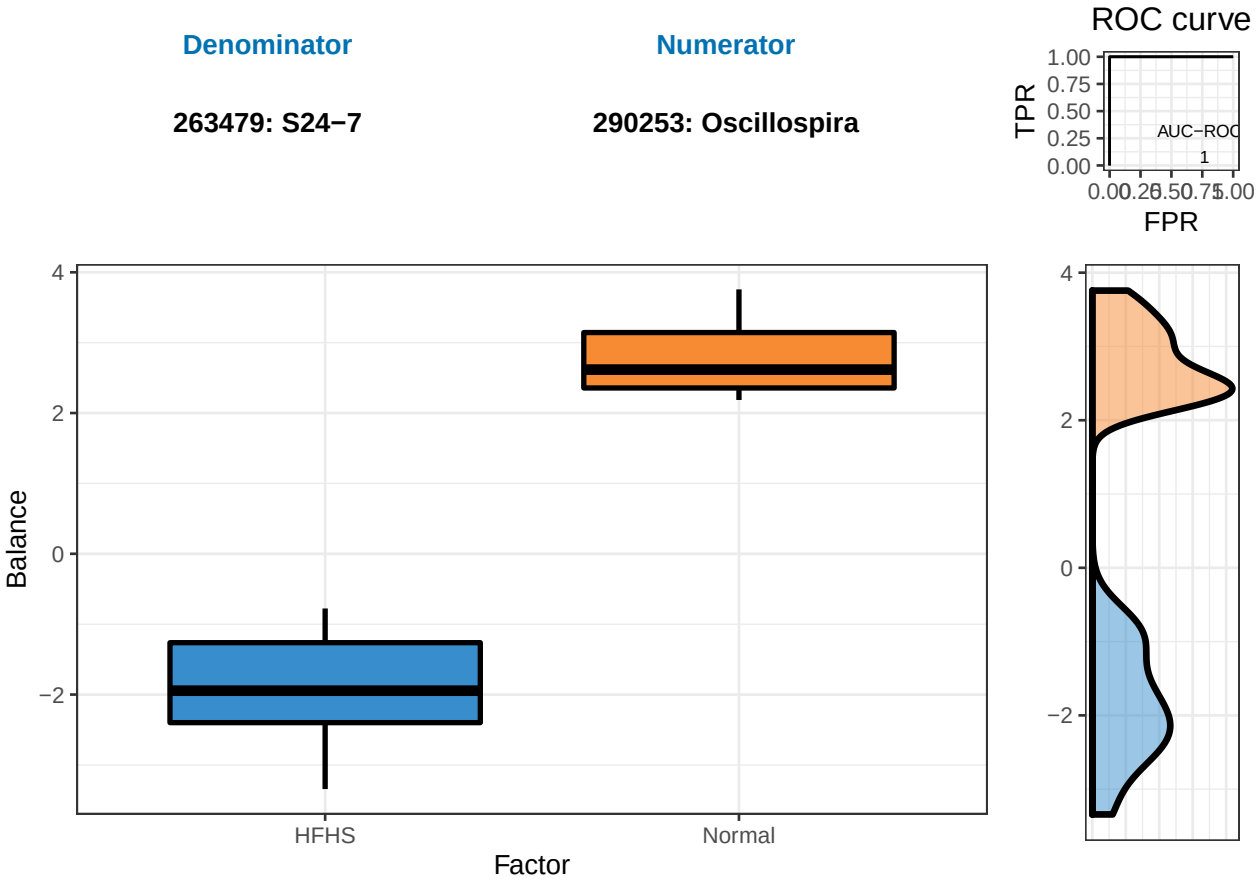

Figure 5.11: Selbal plot showing variables selected with methd selbal and the ability of these variables to discriminate HFHS and normal individuals.

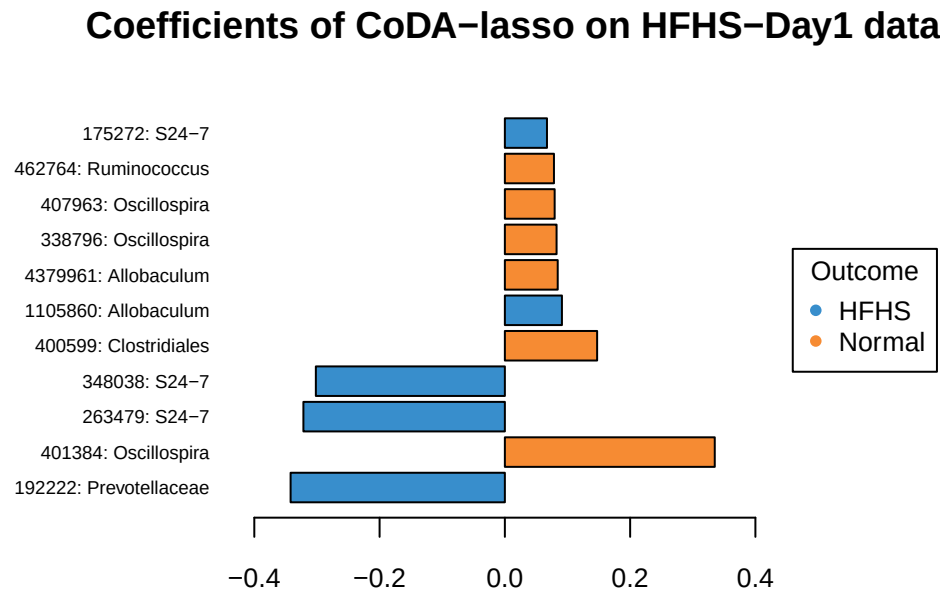

Figure 5.12: The plotLoadings of selected variables with CoDA-lasso.

```
                                        method = 'mean',
                                        contrib.method = 'max',
                                        title = 'Coefficients of CoDA-lasso on HFHS-Day1 data',
                                        OTU = T,
                                        taxa = HFHS.taxonomy)
```

In Figure 5.12, two OTUs **1105860: Allobaculum** and **175272: S24-7** have a greater CLR abundance in HFHS group but were assigned with positive coefficients.

```
# CLR_lasso
HFHS.clr_coef <- HFHS.results_clrlasso$coefficientsSelect
HFHS.clr_data <- HFHS.x[ ,HFHS.results_clrlasso$varSelect]
HFHS.clr.plotloadings <- plotcoefficients(coef = HFHS.clr_coef,
                                          data = HFHS.clr_data,
                                          Y = HFHS.y,
                                          title = 'Coefficients of clr-lasso on HFHS-Day1 data',
                                          method = 'mean',
                                          contrib.method = 'max',
                                          OTU = T,
                                          taxa = HFHS.taxonomy)
```

5.2.4 Trajectory plots

```
TRAJ_plot(selectVar_coef_method1 = HFHS.coda_coef, selectVar_coef_method2 = HFHS.clr_coef,
          selectMethods = c('CoDA-lasso', 'CLR-lasso'), OTU = T, taxa = HFHS.taxonomy)
```

In Figure 5.14, top four OTUs selected with CLR-lasso are also selected as top OTUs from CoDA-lasso but with different order. The other OTUs are either selected by CoDA-lasso or CLR-lasso (except OTU 462764: Ruminococcus).

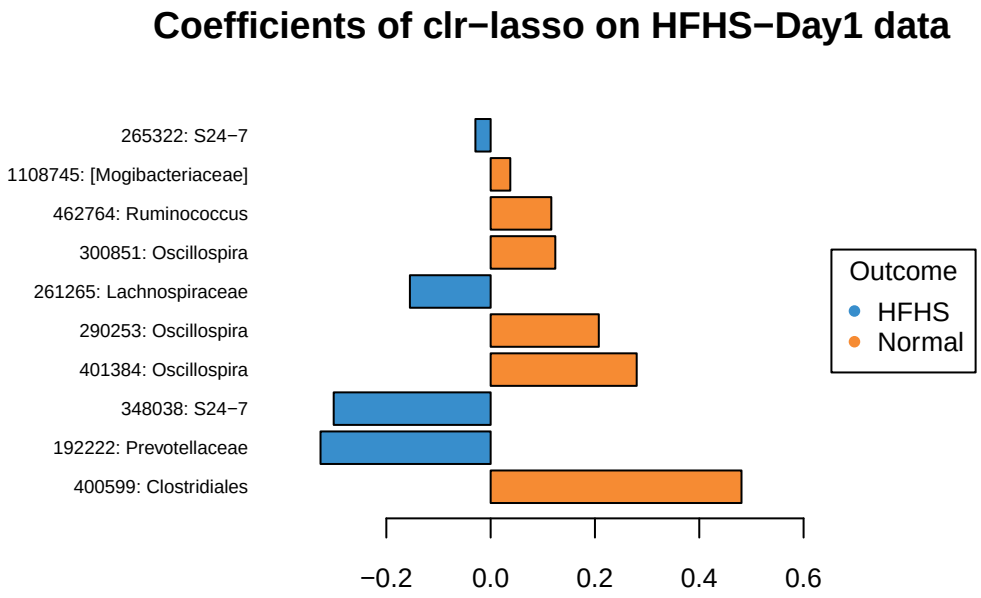

Figure 5.13: The plotLoadings of selected variables with CLR-lasso.

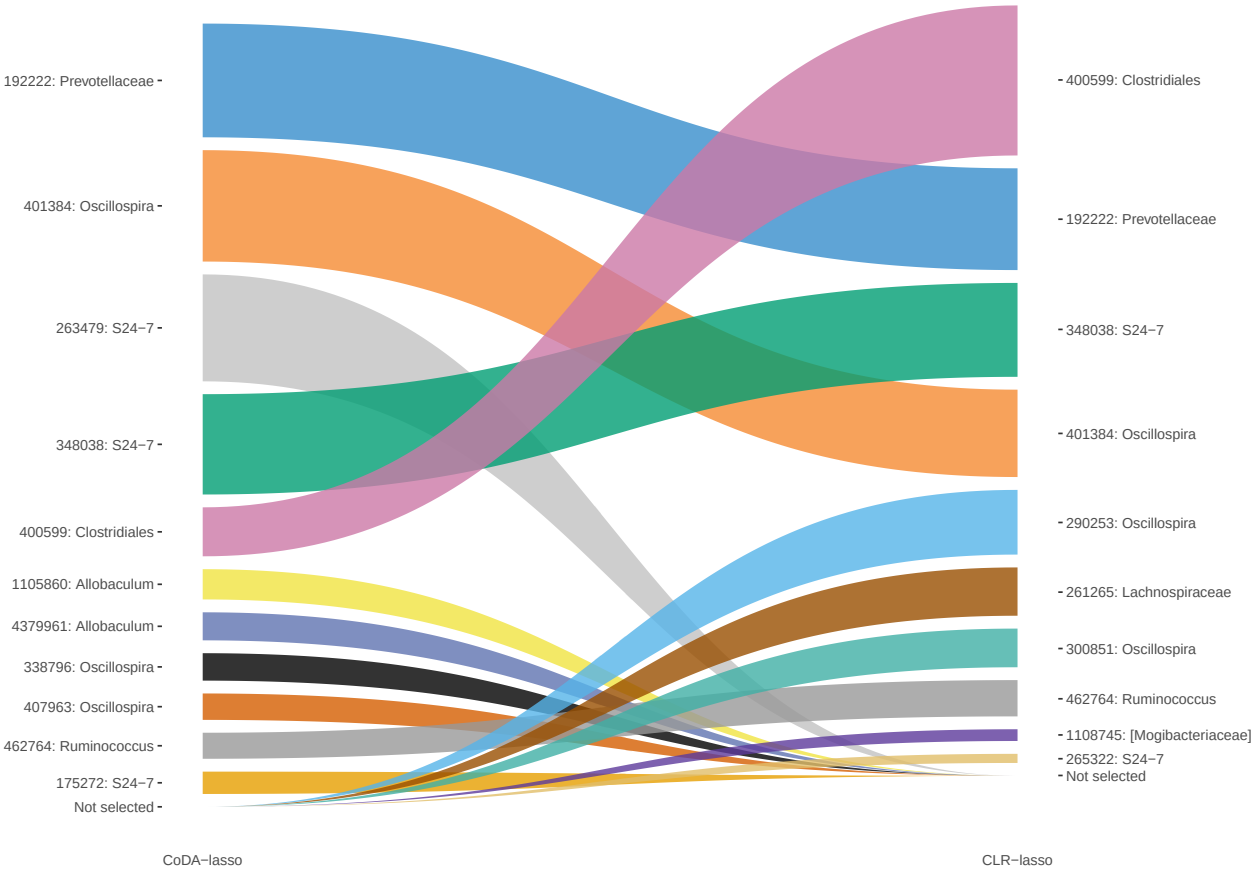

Figure 5.14: Trajectory plots of selected variables with both CoDA-lasso and CLR-lasso in HFHS-Day1 data.

5.2.5 GraPhlAn

As we also have the taxonomic information of HFHS-Day1 data, we use GraPhlAn to visualise the taxonomic information of the selected OTUs. GraPhlAn is a software tool for producing high-quality circular representations of taxonomic and phylogenetic trees (https://huttenhower.sph.harvard.edu/graphlan). It is coded in Python.

We first remove empty taxa (e.g. species) and aggregate all these selected variables into a list. Then we use function *graphlan_annot_generation()* to generate the input files that graphlan python codes require. In the **save_folder**, there are two existing files: **annot_0.txt** and **graphlan_all.sh**. After we generate our input files **taxa.txt** and **annot_all.txt**, we only need to run the **graphlan_all.sh** in the bash command line to generate the plot.

```
# remove empty columns
HFHS.tax_codalasso <- HFHS.tax_codalasso[,-7]
HFHS.tax_clrlasso <- HFHS.tax_clrlasso[,-7]
HFHS.tax_selbal <- HFHS.tax_selbal[,-7]

HFHS.select.tax <- list(CoDA_lasso = HFHS.tax_codalasso,
                        CLR_lasso = HFHS.tax_clrlasso,
                        selbal = HFHS.tax_selbal)

graphlan_annot_generation(taxa_list = HFHS.select.tax,
save_folder = '/Users/yiwenw5/Documents/GitHub/Microbiome_variable_selection_tutorial/Microbiome_variabl
```

In Figure 5.15, the inner circle is a taxonomic tree of selected OTUs. The outside circles indicate different selection methods. If a proportion of a circle is coloured, it means that the corresponding OTU is selected by the method labeled on the circle. If the bottom nodes are coloured in gray, it indicates the OTUs are only selected by one method.

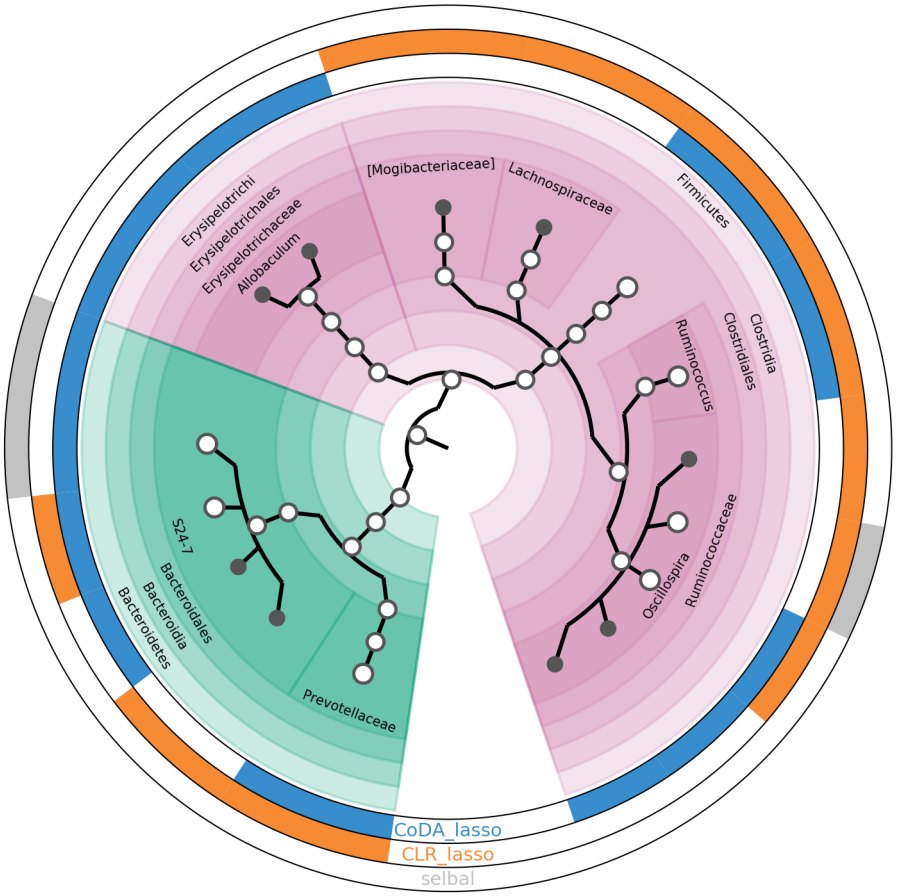

Figure 5.15: GraPhlAn of selected taxa from different methods in HFHS-Day1 data.

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.

Le Cessie, S. and Van Houwelingen, J. C. (1992). Ridge estimators in logistic regression. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 41(1):191–201.

Lex, A., Gehlenborg, N., Strobelt, H., Vuillemot, R., and Pfister, H. (2014). Upset: visualization of intersecting sets. *IEEE transactions on visualization and computer graphics*, 20(12):1983–1992.

Lin, W., Shi, P., Feng, R., and Li, H. (2014). Variable selection in regression with compositional covariates. *Biometrika*, 101(4):785–797.

Lu, J., Shi, P., and Li, H. (2019). Generalized linear models with linear constraints for microbiome compositional data. *Biometrics*, 75(1):235–244.

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.

Rohart, F., Eslami, A., Matigian, N., Bougeard, S., and Le Cao, K.-A. (2017). Mint: a multivariate integrative method to identify reproducible molecular signatures across independent experiments and platforms. *BMC bioinformatics*, 18(1):128.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1):267–288.

Zou, H. and Hastie, T. (2005). Regularization and variable selection via the elastic net. *Journal of the royal statistical society: series B (statistical methodology)*, 67(2):301–320.