mixOmics sRDA integration - Melbourne project 2019-2020

Attila Csala

March 4, 2020

Contents

1 Introduction to sRDA in mixOmics

Redundancy analysis (RDA) is a directional, latent variable based regression method, often referred to as the multivariate equivalent of multiple regression [Legendre, 2012]. RDA accounts for the dependency between data sets, which distinguishes it from other latent variable based methods in the mixOmics package [Rohart et al., 2017]. RDA can be used when the goal of the analysis is to find a combination of *explanatory variables* from a explanatory data set that explain the most variance in all the *response variables* in a response data set (see Figure 1).

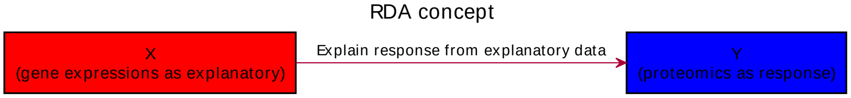

RDA concept
X
(gene expressions as explanatory)
Explain response from explanatory data
Y
(proteomics as response)

Figure 1: Conceptual framework of RDA

RDA was first described by Wollenberg [van den Wollenberg, 1977], its development was motivated by expending the ordinary least squares multiple regression to multiple response variables. It has been extensively used in ecology and chemo-metrics, and lately has been developed for high-dimensional data analysis in the form of sparse Redundancy Analysis (sRDA) [Csala et al., 2017]. sRDA can be used when the objective is to find a combination of the subset of *explanatory variables* (also called the latent variable) that explain the most variance in the *response variables*. In omics data analyses, for example, one might be interested to find a combination of *multiple mRNA levels* that explain the most variation in *multiple proteomics levels*, measured in the sample of patients with a particular disease. Applying sRDA in this setting, the extracted variables can be interpreted as bio-markers for the given disease. Moreover, the *explanatory* − *response* properties of the variables are preserved, thus indicating that changing (some of) the mRNA levels that form the latent variable would result in a change in the proteomics levels. sRDA also provides an indication of the strength of "dependency" of all response variables on the latent variable in terms of correlation coefficients, which can be interpreted as the multiple regression correlation coefficient per response variable.

1.1 Biological questions to answer with sRDA

sRDA can be applied to answer the following biological questions:

I have two omics data sets measured on the same patients, one data set measured on mRNA levels and a second data set measured on protein levels.

1. I assume that mRNAs are *explanatory variables* for protein levels (i.e. *response variables*). I am interested to find out which are the "important" mRNAs that explain the most variance in the protein levels?

2. And which are the proteins that response most strongly and therefore are most affected by the "important" mRNAs?

3. And what is the quantified effect of the "important" mRNAs on the proteins?

As a motivating example, we apply sRDA to a filtered breast cancer data from he Cancer Genome Atlas (TCGA, `http://cancergenome.nih.gov/`) to answer these questions. The filtered TCGA data is available in mixOmics, here are *expression levels of* 200 *mRNA* and the *abundance of* 142 *proteins* are available on 150 patients with breast cancer.

```
#load sRDA and helper functions for mixOmics from Github
library(sRDA)
library(devtools)
url <- "https://raw.githubusercontent.com/acsala/2019_mixOmics_RDA/master/00_helper_fun
source_url(url)
```

Listing 1: Load RDA and the helper functions from CRAN and Github

First we load sRDA from CRAN and its mixOMICS helper functions from Github (Listing 1) and then apply sRDA to the TCGA data from mixOmics (Listing 2).

In this example, we applied sRDA to extract two latent variables (LVs) from the mRNA measurements (data set X) to explain the variance in the protein levels (data set Y). The two LVs represent independent sets of the combination of 20 mRNAs that explain the most variance in the protein levels. With these results, we can answer our biological questions.

1.1.1 Which are the "important" explanatory variables that explain the most variance in the response variables?

We explicitly defined that we wish to obtain 20 mRNAs that best explain the variance in all the protein variables (Listing 2). In order to analyse the

```r
# load cancer data from mixOmics package
library(mixOmics)
data(breast.TCGA)

#set seed for reproducibility
set.seed(100)

X <- breast.TCGA$data.train$mrna
Y <- breast.TCGA$data.train$protein

#call the sRDA function to run sRDA on data sets X and Y
res_sRDA <- sRDA_mixOmics(X = X, Y = Y,
                          keepX = 20,
                          ridge_penalty = 1,
                          penalty_mode = "enet",
                          ncomp = 5,
                          scale = TRUE)
```

Listing 2: sRDA code to analyse the TCGA data set from mixOmics

results, we can print out the loadings for the mRNAs and proteins, which indicates the strength of the particular mRNAs in the latent variable, and gives the multiple regression coefficient (in respect to the latent variable) for each protein variables (see Table 1 for the selected variables and their loadings, where we printed out only the top 20 absolute loadings of the protein variables).

1. I assume that mRNAs are *explanatory variables* for protein levels (i.e. *response variables*). I am interested to find out which are the "important" mRNAs that explain the most change in the protein levels?

   We can see these important mRNAs in Table 1.

1.1.2 Which response variables are affected most strongly by the "important" explanatory variables?

1. And which are the proteins that response most strongly and therefore are most affected by the "important" mRNAs?

   We can find these proteins by ordering our results based on the absolute

```
lv = 1
protein_names <-
  names(sort(abs(res_sRDA$loadings$Y[,lv]), decreasing = T)[1:20])
mRNA_names <-
  names(sort(abs(res_sRDA$loadings$X[,lv]), decreasing = T)[1:20])

print(cbind(mRNA_names,
            round(res_sRDA$loadings$X[,lv][mRNA_names],
                  digits = 3),
            protein_names,
            round(res_sRDA$loadings$Y[,lv][protein_names],
                  digits = 3)))
```

Listing 3: Code to extract the results after the sRDA analysis

Table 1: mRNA and protein variables extracted in the first latent variable from the TCGA data set with sRDA

mRNA$_{names}$		protein$_{names}$	
CCNA2	0.152	ER-alpha	-0.813
ZNF552	-0.126	GATA3	-0.768
KDM4B	-0.107	Cyclin$_{B1}$	0.745
PREX1	-0.107	ASNS	0.716
LRIG1	-0.093	PR	-0.704
E2F1	0.052	Cyclin$_{E1}$	0.704
C4orf34	-0.05	AR	-0.692
ASPM	0.047	JNK2	-0.654
NTN4	-0.038	INPP4B	-0.654
FUT8	-0.033	CDK1	0.65
TTC39A	-0.03	Bcl-2	-0.558
STC2	-0.025	ER-alpha$_{pS118}$	-0.527
SLC19A2	-0.022	PDK1$_{pS241}$	-0.515
MEX3A	0.017	p53	0.512
C18orf1	-0.016	Chk2	0.51
MTL5	-0.015	DJ-1	-0.5
NCAPG2	0.014	Chk2$_{pT68}$	0.475
MED13L	-0.01	Caveolin-1	-0.466
FAM63A	-0.009	P-Cadherin	0.463
SEMA3C	-0.006	p27$_{pT198}$	0.46

loadings of the proteins. The top 20 most affected proteins can be found in Table 1.

1.1.3 What is the "quantified" effect of the explanatory variables on the response variables

We can find regression weights for the mRNA variables (i.e. the regression weight between the mRNA and the latent variable) and the) multiple regression correlation coefficient for the protein levels (i.e. the multiple regression coefficient between the protein levels and the LV) in Table 1 that helps us to quantify the relationships we found. ER-alpha protein levels have a strong negative correlation with the latent variable (with correlation coefficient -0.813), thus one unit change in the latent variable correlates with a .813 decrees in the (scaled) protein levels (see Figure 2). We can also read that mRNA CCNA2 has a 0.152 coefficient with the latent variable, thus a unit change in CCNA2 correlates with a .152 increase in the latent variable (see Table 1).

```
ER_lm <- lm(Y[,"ER-alpha"] ~ res_sRDA$variates$X[,1])

plot_cus(res_sRDA$variates$X[,1], Y[,"ER-alpha"],
         ylab = "ER-alpha level", xlab = "Latent variable",
         ylim = c(-4,4))

abline(c(ER_lm), col=c(regression_line_col), lwd=c(2.5))
```

Listing 4: Code to plot the relationship between ER-alpha protein levels and the latent variable

1. And what is the quantified effect of the "important" mRNAs on the proteins?

   We can read from the results that ER-alpha protein levels have a strong negative correlation with the latent variable (with correlation coefficient -0.813), thus one unit change in the latent variable correlates with a .813 decrees in the (scaled) protein levels (see Figure 2). We can also read that mRNA CCNA2 has a 0.152 coefficient with the latent variable, thus a unit change in CCNA2 correlates with a .152 increase in the latent variable (see Table 1).

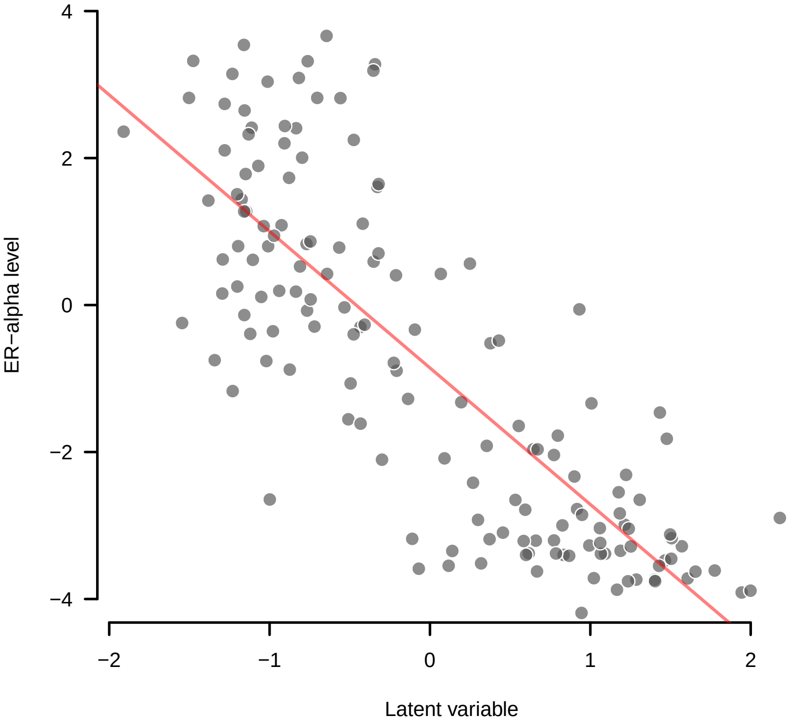
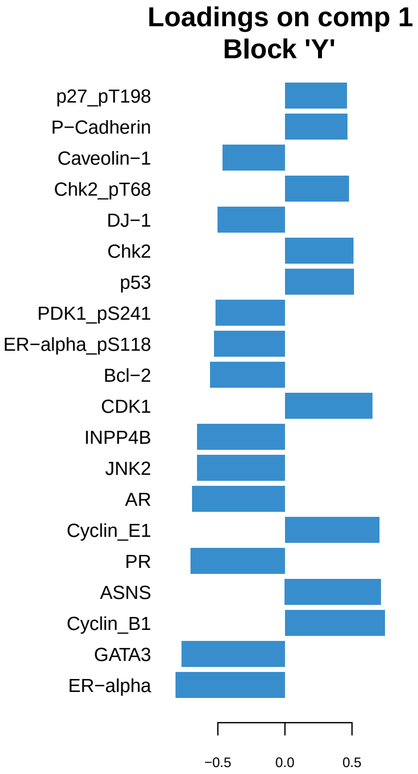


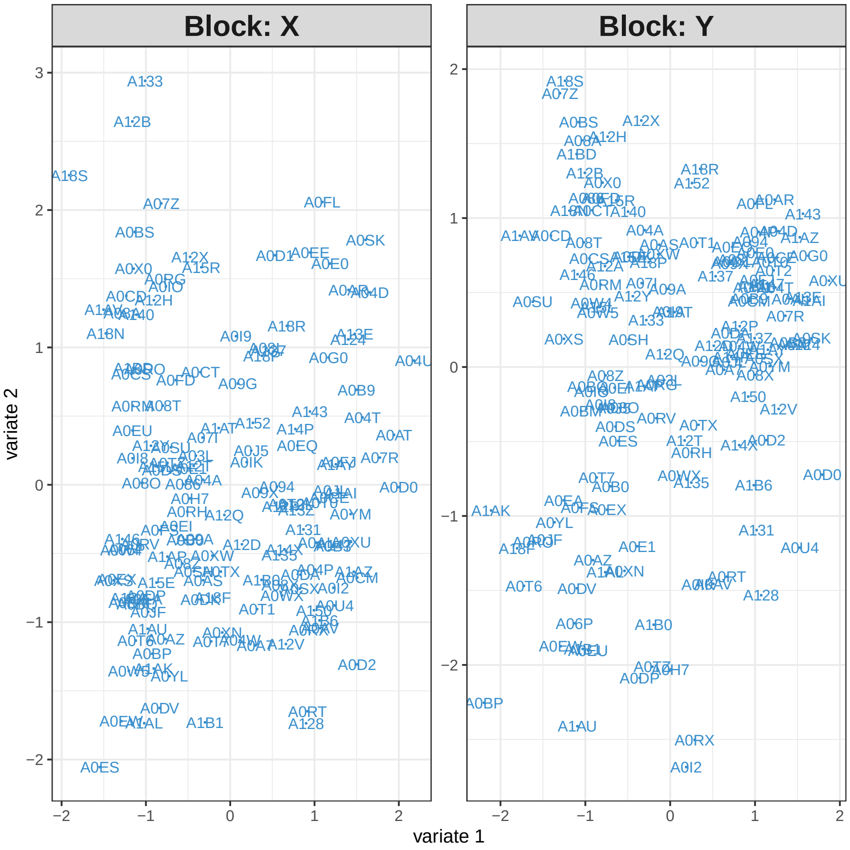
Figure 2: ER-alpha level (on the y-axes) has a strong negative correlation with the linear combination of the 20 selected mRNA levels (Latent variable; on the x-axes)

Additionally, we can use the mixOmics' plots, such as plotLoadings(), plotIndiv(), plotVar() or cim() (the detailed descriptions for these plots can be found in mixOmics' vignette).

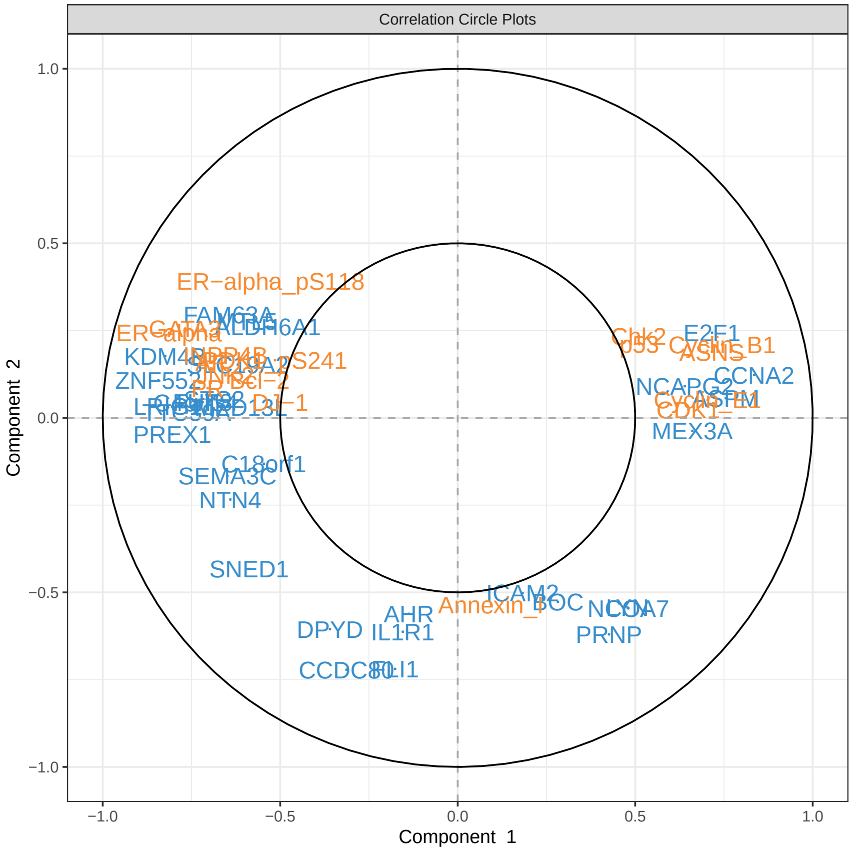
```
plotLoadings(res_sRDA, ndisplay = 20, comp = 1, size.name = rel(0.9),
             contrib = "max", size.title = rel(1.3))
```



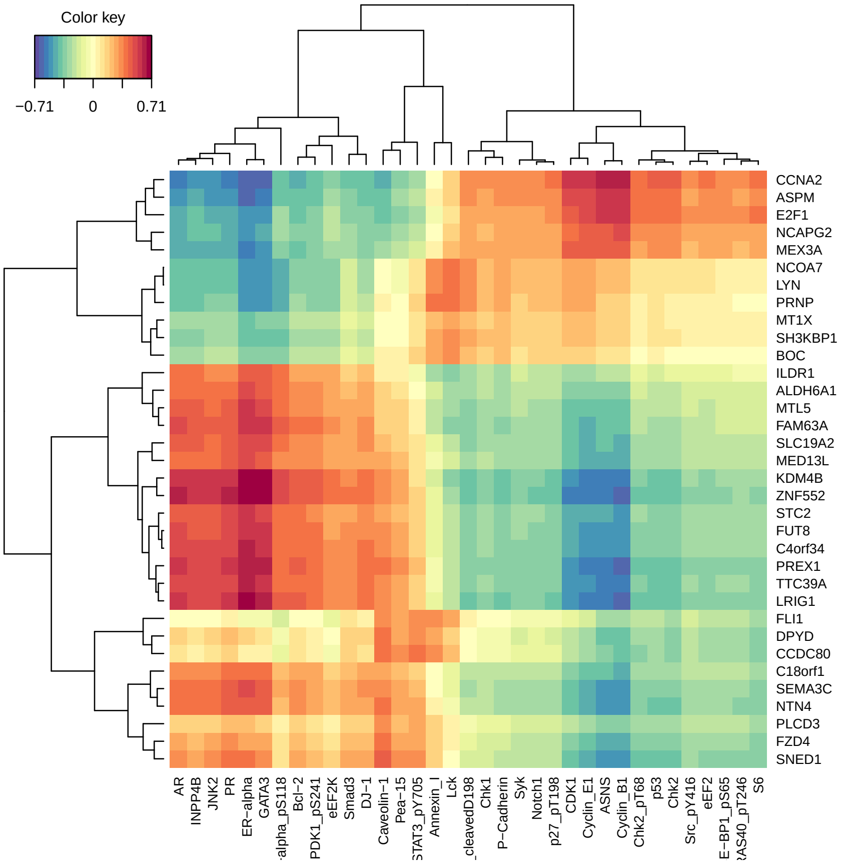
```
plotIndiv(res_sRDA)
```



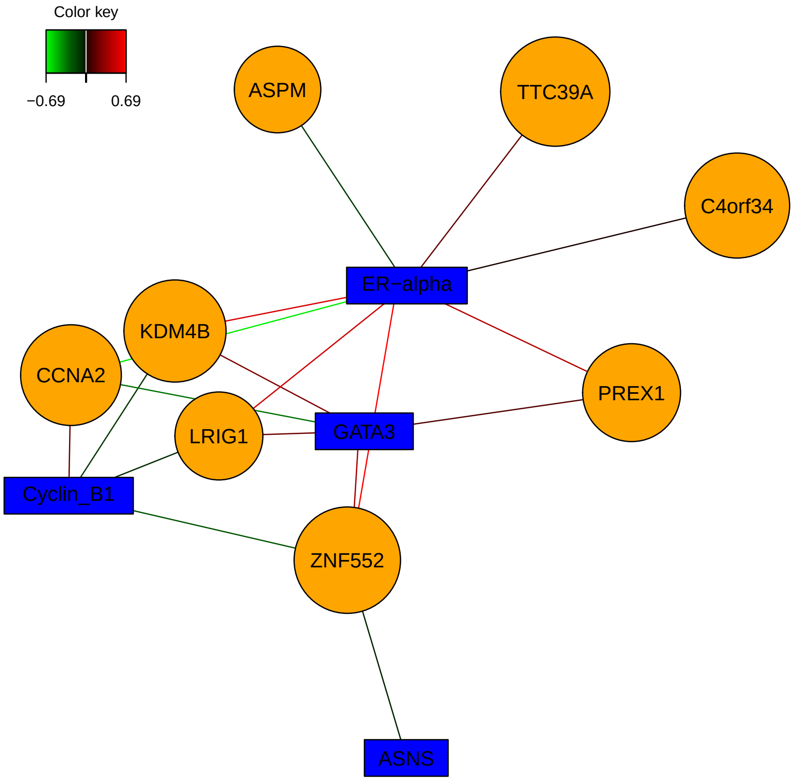
```
plotVar(res_sRDA, cutoff = 0.5)
```



```
cim(res_sRDA, scale = T, threshold = 0.35)
```



```
network(res_sRDA,
        color.node = c("orange","blue"),
        cutoff = 0.6,
        row.names = T,
        col.names = T,
        comp = 1)
```



In this section we described when and how to use sRDA for omics data analysis. In the following we describe the differences between sRDA and other multivariate methods available from mixOmics in terms of objective function and also in terms of bilogical interpretability.

2 Comparison of latent variable based methods in terms of objective functions

Partial least squares (PLS), Canonical correlation analysis (CCA), Principal component analysis (PCA) and RDA are closely related multivariate latent variable based methods, all available in the mixOmics package. In this section, we describe their relationship in terms of objective functions, which also applies to the penalized and sparse versions of these methods.

Given $\mathbf{X} \in \mathbb{R}^{n \times p}$, a centered and scaled (with columns of mean zero and standard deviation one, respectively) predictor data set and $\mathbf{Y} \in \mathbb{R}^{n \times q}$, a centered and scaled response data set. Our goal is to look for the first pair of loading vectors, $\mathbf{w} \in \mathbb{R}^p$ for $\mathbf{X}$ and $\mathbf{v} \in \mathbb{R}^q$ for $\mathbf{Y}$, then these methods maximize the following quantities:

$$\begin{aligned}
\text{PCA}: &\quad \text{Var}(\mathbf{Xw}) && (1)\\
\text{CCA}: &\quad \text{Corr}^2(\mathbf{Xw}, \mathbf{Yv}) && (2)\\
\text{RDA}: &\quad \text{Corr}^2(\mathbf{Xw}, \mathbf{Yv}) \cdot \text{Var}(\mathbf{Yv}) && (3)\\
\text{PLS}: &\quad \text{Var}(\mathbf{Xw}) \cdot \text{Corr}^2(\mathbf{Xw}, \mathbf{Yv}) \cdot \text{Var}(\mathbf{Yv}) = \text{Cov}^2(\mathbf{Xw}, \mathbf{Yv}) && (4)
\end{aligned}$$

PCA is a unsupervised method which aims to maximize the variance of the linear combinations of all variables from one data set (Eq. (1)). These linear combinations are also called latent variables. By maximising the variance of latent variables, the aim is to capture all the variation in the original data set in the latent variables. By doing so, often nearly all information from the original data can be preserved in a lower dimensional latent variable space.

CCA and PLS are close related to PCA, they aim to extract latent variables that explain variation in the original data. They are applied to two (or multiple) data sets. CCA aims to maximize the squared correlations between the latent variables (Eq. (2)), which results of a linear combination of variables from $\mathbf{X}$ that has the maximum correlation with a linear combination of variables from $\mathbf{Y}$. PLS aims to maximize the squared covariance between linear the latent variables (Eq. (4)). Both PLS and CCA are non-directional methods, that is the optimum of the objective function doesn't change by interchanging the input data sets. These methods can be used to explore linear combinations of variables that have the highest correlation or covariance (respectively) with each other. CCA is preferred if the relationship wished to be characterized in correlations instead of covariances (one is standardized the other is in the scale of the original variables).

RDA is similar to the aforementioned methods, in the sense that it is based on latent variable extraction. The objective function of RDA is to maximise the squared correlation between a linear combination of the predictor variables and the response data set (Eq. (3)). Therefore, RDA is a directional method that distinguishes between the predictor and response data sets and the optimum of its objective function changes if one interchanges the input data sets. RDA and can be used to explore linear combinations of explanatory variables that explain the most variance in an outcome data set.

3 Cross validated analysis

3.1 Cross validate over a grid of non-zeros

We can (and we should) cross-validate for the optimum penalty parameters. There is Elastic net (enet) and Univariate soft thresholding (ust) available for sRDA ([Csala et al., 2017]), below we use enet with a Ridge penalty of 0.1 and a grid of non-zeros between 5 and 150 (the longer the grid the longer it takes to run the cross validation) (Listing 2). Under this setting the cross-validation ran for 2.5 minutes and selected 25 as the optimum non-zero parameter for the mRNA data source (Figure 3).

Table 2: sRDA code and results for cross validated analysis

mRNA$_{\text{names}}$		protein$_{\text{names}}$	
CCNA2	0.209	ER-alpha	-0.797
ZNF552	-0.141	GATA3	-0.753
PREX1	-0.099	Cyclin$_{\text{B1}}$	0.751
KDM4B	-0.098	ASNS	0.713
LRIG1	-0.054	Cyclin$_{\text{E1}}$	0.701
C18orf1	-0.029	PR	-0.693
NTN4	-0.025	AR	-0.681
E2F1	0.023	CDK1	0.652
C4orf34	-0.016	JNK2	-0.648
MTL5	-0.005	INPP4B	-0.625

3.2 Cross validate over a grid of non-zeros and a grid of ridge penalties

4 Compare sRDA and sPLS canonical mode with 5 components on mixOmics breast.TCGA data

In order to compare sRDA to sPLS, Both methods are applied on a dataset with 150 patients and 200 mrna and 142 protein measurement.

5 components are extracted with enforcing 25 non-zero variables (no optimization procedure).

4.1 Overlapping variables

```r
#set seed for reproducibility
set.seed(100)

X <- breast.TCGA$data.train$mrna
Y <- breast.TCGA$data.train$protein

#call the sRDA function to run sRDA on data sets X and Y
res_sRDA <- sRDA_mixOmics(X = X, Y = Y,
                          penalty_mode = "enet",
                          ncomp = 1,
                          scale = TRUE,
                          cross_validate = TRUE,
                          CV_nonzer_grid = c(5,10,15,20,25,50,75,100,150),
                          CV_ridge_grid = c(0.1),
                          parallel_CV = TRUE)


lv = 1
protein_names <-
  names(sort(abs(res_sRDA$loadings$Y[,lv]), decreasing = T)[1:res_sRDA$keepX])
mRNA_names <-
  names(sort(abs(res_sRDA$loadings$X[,lv]), decreasing = T)[1:res_sRDA$keepX])

print(cbind(mRNA_names,
            round(res_sRDA$loadings$X[,lv][mRNA_names],
                  digits = 3),
            protein_names,
            round(res_sRDA$loadings$Y[,lv][protein_names],
                  digits = 3)))
```

Listing 5: Code for sRDA cross validation

```r
plot_CV_results(res_sRDA, ylim = c(43,45),spline_df = 6)
```

Listing 6: Code to plot the cross validation results

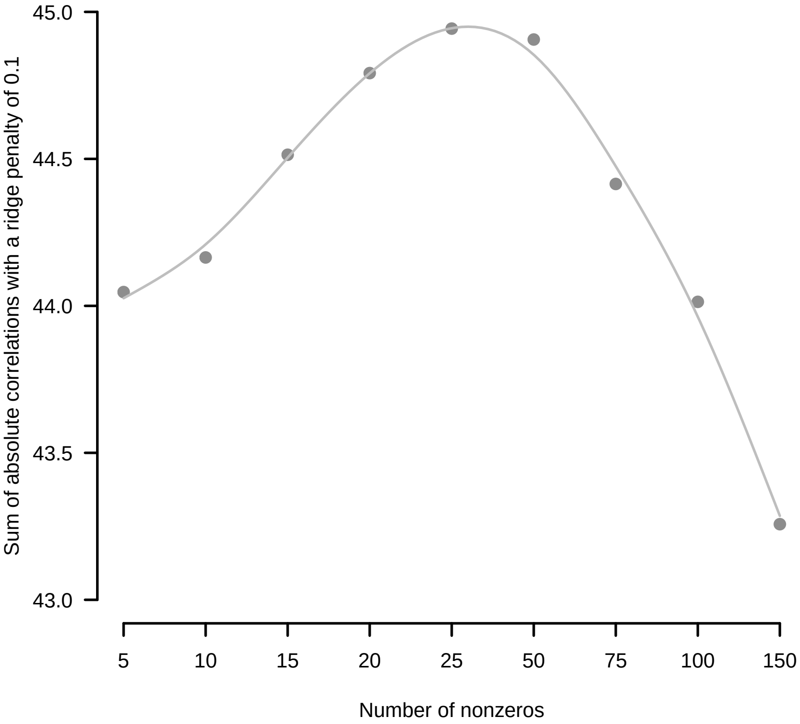


Figure 3: Cross validation results. The y-axis represents the sum of absolute correlations between the latent variable and the outcome variables, and the x-axis represents the number of selected non-zeros.

```
#set seed for reproducibility
set.seed(100)

X <- breast.TCGA$data.train$mrna
Y <- breast.TCGA$data.train$protein

#call the sRDA function to run sRDA on data sets X and Y
res_sRDA2 <- sRDA_mixOmics(X = X, Y = Y,
                          penalty_mode = "enet",
                          ncomp = 1,
                          scale = TRUE,
                          cross_validate = TRUE,
                          CV_nonzer_grid = c(5,10,15,20),
                          CV_ridge_grid = c(0.1,0.2),
                          parallel_CV = TRUE)


lv = 1
protein_names <-
  names(sort(abs(res_sRDA2$loadings$Y[,lv]), decreasing = T)[1:res_sRDA2$keepX])
mRNA_names <-
  names(sort(abs(res_sRDA2$loadings$X[,lv]), decreasing = T)[1:res_sRDA2$keepX])

print(cbind(mRNA_names,
            round(res_sRDA2$loadings$X[,lv][mRNA_names],
                  digits = 3),
            protein_names,
            round(res_sRDA2$loadings$Y[,lv][protein_names],
                  digits = 3)))
```

Listing 7: Code for sRDA cross validation for multiple ridges

```
plot_CV_results(res_sRDA2, ylim = c(43,45),spline_df = 4, which_ridge = 0.2)
```

Listing 8: Code to plot the cross validation results

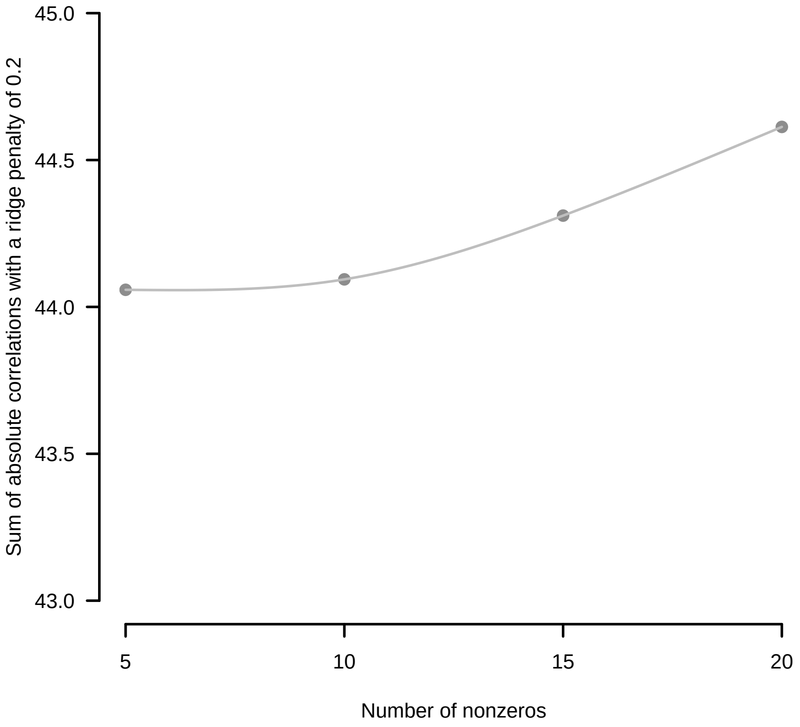


Figure 4: Second cross validation results. The y-axis represents the sum of absolute correlations between the latent variable and the outcome variables (for a specific ridge penalty), and the x-axis represents the number of selected non-zeros.

```r
set.seed(100)
ncomp <- 5
nr_nonz <- 10
res_all <- get_PLS_CCA_RDA_results(X = X, Y = Y,
                                 nr_nonz = nr_nonz,
                                 nr_comp = ncomp,
                                 pls_mode = "canonical",
                                 penalty_mode = "ust",
                                 CCA = F)

res_sRDA <- res_all$res_sRDA
res_spls <- res_all$res_spls

#loadings.star and loadngs are the same for pls?
res_spls$loadings$X[,1] == res_spls$loadings.star[[1]][,1]

# Compare the variables selection
res_sRDA <- get_nonzero_variables(res_object = res_sRDA)
res_spls <- get_nonzero_variables(res_object = res_spls)

#res_sRDA$nz_loading_names[["X"]]
#res_spls$nz_loading_names[["X"]]

plot_cus(1:ncomp,get_nr_of_common_components(res_sRDA, res_spls,
                                       ncomp = ncomp),
   ylab = "Nr of common variables", xlab = "Component",
   ylim = c(0,nr_nonz))
```

Listing 9: Code to assess the overlap between the LVs of sPLS and sRDA

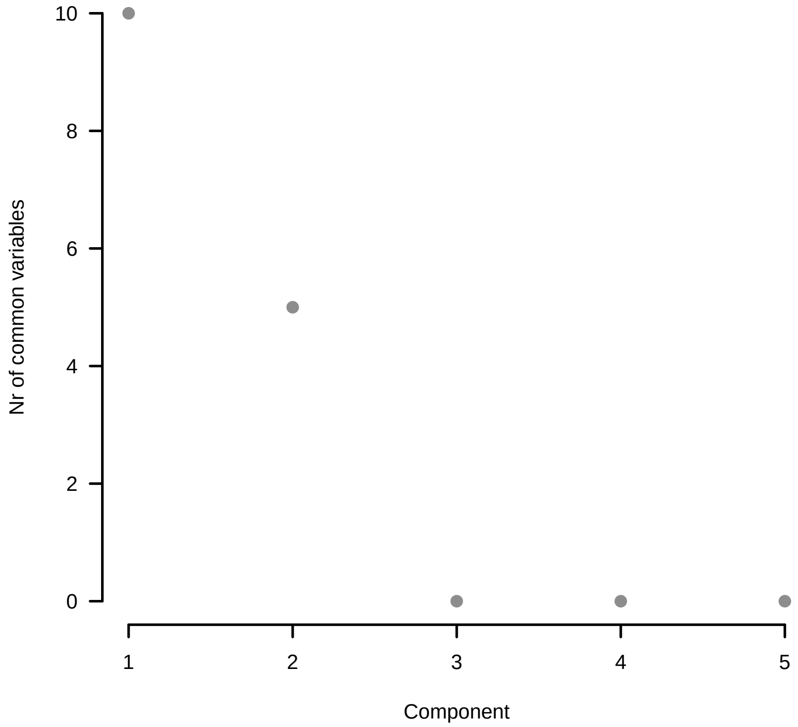


Figure 5: Overlap of the original variables preserved in the LVs by sPLS and sRDA.

Table 3: sRDA code and results for second cross validated analysis

$\text{mRNA}_{\text{names}}$		$\text{protein}_{\text{names}}$	
CCNA2	0.205	ER-alpha	-0.808
ZNF552	-0.133	GATA3	-0.763
PREX1	-0.097	$\text{Cyclin}_{\text{B1}}$	0.749
KDM4B	-0.084	ASNS	0.723
LRIG1	-0.055	$\text{Cyclin}_{\text{E1}}$	0.7
C18orf1	-0.046	PR	-0.697
NTN4	-0.045	AR	-0.679
E2F1	0.038	JNK2	-0.659
MTL5	-0.029	CDK1	0.656
SLC19A2	-0.02	INPP4B	-0.64
C4orf34	-0.017	Bcl-2	-0.565
MEX3A	0.017	p53	0.526
PLCD4	-0.009	$\text{ER-alpha}_{\text{pS118}}$	-0.519
GMDS	0.005	$\text{PDK1}_{\text{pS241}}$	-0.511
MED13L	-0.005	Chk2	0.504

The variables selected in the first component are identical for PLS and RDA, there are 5 overlapping variables in the second component, and there are no overlapping variables in consecutive latent variables (Figure 5).

4.2 Benchmark correlation

```r
# we can look at explained variances, they are about the
explained_Y <- cbind(res_spls$explained_variance$Y,
res_sRDA$explained_variance$Y)

colnames(explained_Y) <- c("Explined by sPLS",
"Explained by sRDA")

explained_X <- cbind(res_spls$explained_variance$X,
res_sRDA$explained_variance$X)

colnames(explained_X) <- c("Explined by sPLS",
"Explained by sRDA")

total_in_Y <- apply(explained_Y,2,sum)
explained_Y <- rbind(explained_Y, total_in_Y)
```

```
total_in_X <- apply(explained_X,2,sum)
explained_X <- rbind(explained_X, total_in_X)

round(explained_Y,2)
```

	Explined by sPLS	Explained by sRDA
comp 1	0.13	0.18
comp 2	0.12	0.04
comp 3	0.06	0.02
comp 4	0.07	0.04
comp 5	0.04	0.02
total$_{\text{inY}}$	0.41	0.29

```
round(explained_X,2)
```

	Explined by sPLS	Explained by sRDA
comp 1	0.17	0.17
comp 2	0.12	0.12
comp 3	0.04	0.05
comp 4	0.05	0.04
comp 5	0.05	0.03
total$_{\text{inX}}$	0.43	0.41

5 References

References

[Csala et al., 2017] Csala, A., Voorbraak, F. P. J. M., Zwinderman, A. H., and Hof, M. H. (2017). Sparse redundancy analysis of high-dimensional genetic and genomic data. *Bioinformatics*, 33(20):3228–3234.

[Legendre, 2012] Legendre, P. (2012). *Numerical ecology*. Elsevier, Oxford.

[Rohart et al., 2017] Rohart, F., Gautier, B., Singh, A., and Lê Cao, K.-A. (2017). mixomics: an r package for ‘omics feature selection and multiple data integration.

[van den Wollenberg, 1977] van den Wollenberg, A. L. (1977). Redundancy analysis an alternative for canonical correlation analysis. *Psychometrika*, 42(2):207–219.