The RGCCA package for Regularized/Sparse Generalized Canonical Correlation Analysis

Etienne CAMENEN

2019-12-18

Contents

1 Multiblock data analysis with the RGCCA package

We consider J data matrices X1 ,. . . , XJ. Each n × pj data matrix Xj = [xj1, . . . , xjpj] is called a block and represents a set of pj variables observed on n individuals. The number and the nature of the variables may differ from one block to another, but the individuals must be the same across blocks. We assume that all variables are centered. The objective of RGCCA is to find, for each block, a weighted composite of variables (called block component) yj = Xj . aj, j = 1 ,. . . , J (where aj is a column-vector with pj elements) summarizing the relevant information between and within the blocks. The block components are obtained such that (i) block components explain well their own block and/or (ii) block components that are assumed to be connected are highly correlated. In addition, RGCCA integrates a variable selection procedure, called SGCCA, allowing the identification of the most relevant features.

2 Load the inputs

2.1 Load the blocks

The blocks are loaded with the function `load_blocks`. The first argument of this function (`superblock`) required a bolean giving the presence (TRUE) / absence (FALSE) of a superblock. The second one corresponds to a character giving the list of the file path separated by a comma (argument `file`). By default, the name of the blocks corresponds to those of the files (`names` argument) and could be set. By default, the tabulation is used as a column separator (`sep` argument) and the first row is considered as a header (`header` parameter).

```
# Warning : separators by default are tabulation
blocks = load_blocks(file = "data/agriculture.tsv,data/industry.tsv,data/politic.tsv")
```

2.2 Load the groups of response and the connection between blocks

The connection between the blocks will be used by the RGCCA and must be set by `set_connection` function. A group of samples will be used to color them in the samples plot and must be set by `load_response` function. For both functions, the `blocks` parameter, set at the previous step, is required. The other parameters are optional. The user

could import a file containing either (`file` parameter) : (i) a symmetric matrix with 1 giving a connection between two blocs, or 0 otherwise; (ii) a univariate vector (qualitative or quantitative) or a disjunctive table for the response. By default, the column separator is the tabulation and could be set (`sep` argument). For the `load_response`, a header could be specified (`header` parameter).

```
# Optional parameters
response <- connection <- NULL

# Uncomment the parameters below to try without this settings
response = load_response(blocks = blocks, file = "data/response.tsv")
connection = load_connection(file = "data/connection.tsv")
```

2.3 View the inputs

Table 1: agriculture

	gini	farm	rent
Argentina	86.3	98.2	3.52
Australia	92.9	99.6	3.27
Austria	74	97.4	2.46
Belgium	58.7	85.8	4.15
Bolivia	93.8	97.7	3.04
Brasil	83.7	98.5	2.31

Table 2: industry

	gnpr	labo
Argentina	5.92	3.22
Australia	7.1	2.64
Austria	6.28	3.47
Belgium	6.92	2.3
Bolivia	4.19	4.28
Brasil	5.57	4.11

Table 3: politic

	inst	ecks	death	demostab	demoinst	dictator
Argentina	0.07	4.06	5.38	0	1	0
Australia	0.01	0	0	1	0	0
Austria	0.03	1.61	0	0	1	0
Belgium	0.45	2.2	0.69	1	0	0
Bolivia	0.37	3.99	6.5	0	0	1
Brasil	0.45	3.91	0.69	0	1	0

Table 4: response

Argentina	demoinst
Australia	demostab
Austria	demoinst

Belgium	demostab
Bolivia	dictator
Brasil	demoinst

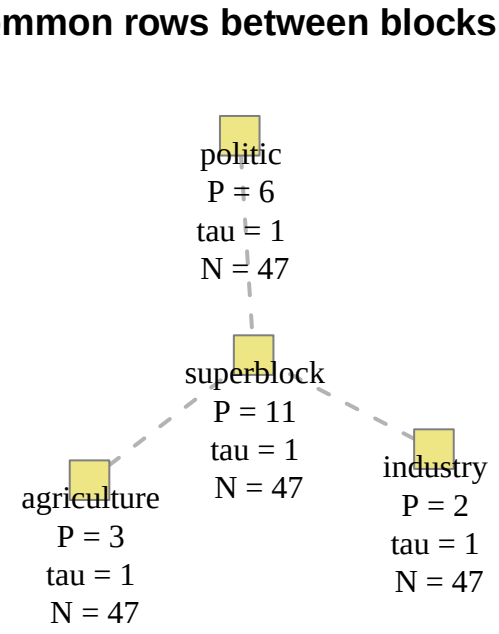
Table 5: connection

0	0	0	1
0	0	0	1
0	0	0	1
1	1	1	0

3 Run S/RGCCA

SGCCA is run from the RGCCA package by using two components for a biplot visualization. The S/RGCCA function doesn't names the blocks in their outputs. This step is required to generate biplots.

```
sgcca_out = rgcca.analyze(blocks = blocks)
plot_network(sgcca_out)
```



4 Vizualise the analysis

Both the samples and the variables could be visualized by using biplots functions (respectively `plot_ind` and `plot_var_2D`). Histograms are used to visualized in decreasing order the variables with the higher weights and the blocks with the higher Average Variance Explained (AVE).

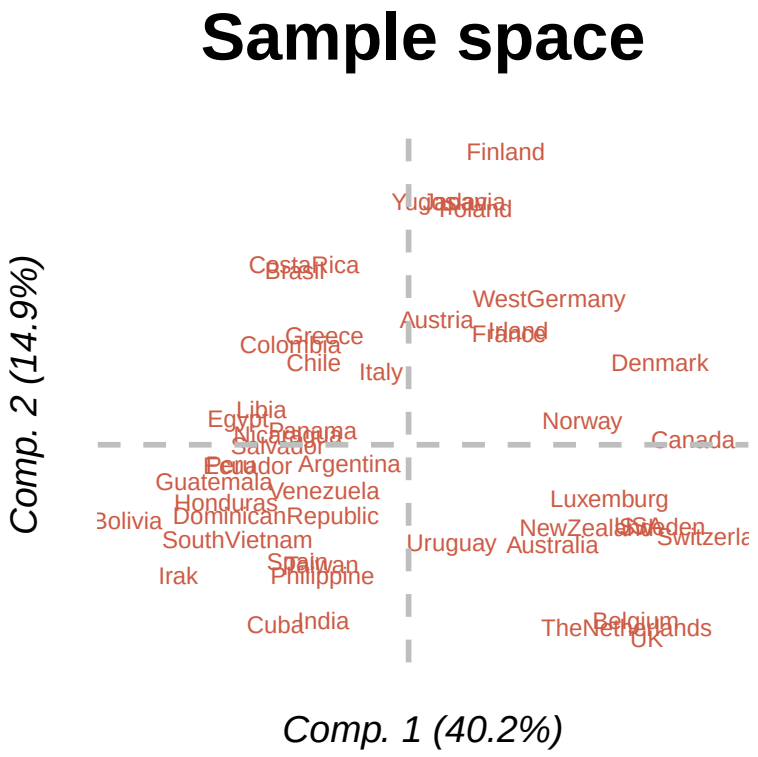
These functions take the results of a sgcca or a rgcca (`rgcca` parameter) and the components to visualize : either `compx` and `compy` for biplots or `comp` for histograms. By default, `compx` = `comp` = 1 and `compy` = 2. The presence or the absence of a superblock among the analysis could be specified for `plot_var_2D` and `plot_var_1D` to color the variables according to their blocks. By default, the last block is plotted, corresponding to the superblock if selected (`i_block` parameter). `plot_var_2D`, which is a corcircle plot, required the `blocks` for the correlation with the selected component. `plot_var_2D` could use the response variable to color the samples by groups. By default, the first 100th higher weights are used for the `plot_var_1D` and could be set by using the `n_mark` argument.

```
comp1 = 1
comp2 = 2
nmark = 100
```

4.1 With the superblock, by default on the first and the second components

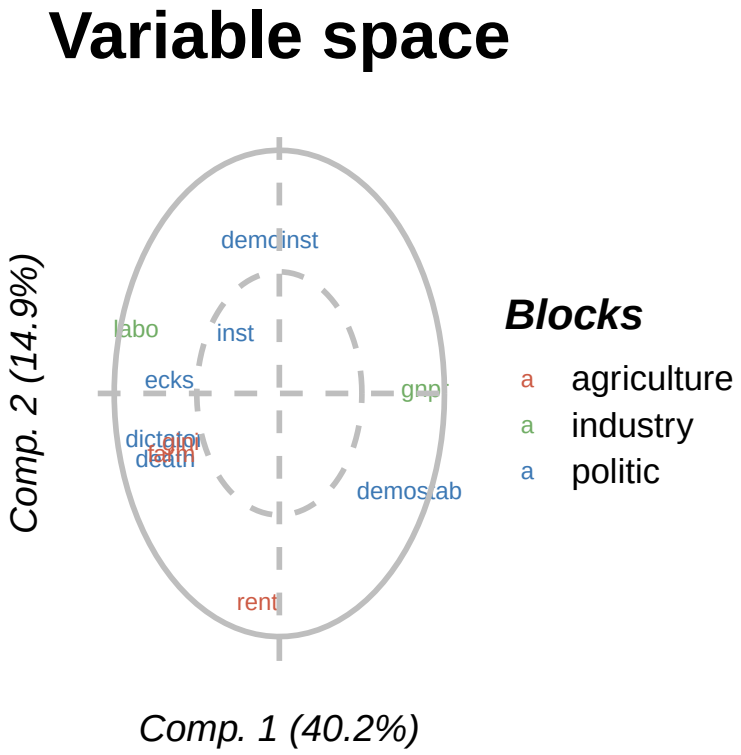
4.1.1 Samples plot

```
plot_ind(sgcca_out)
```



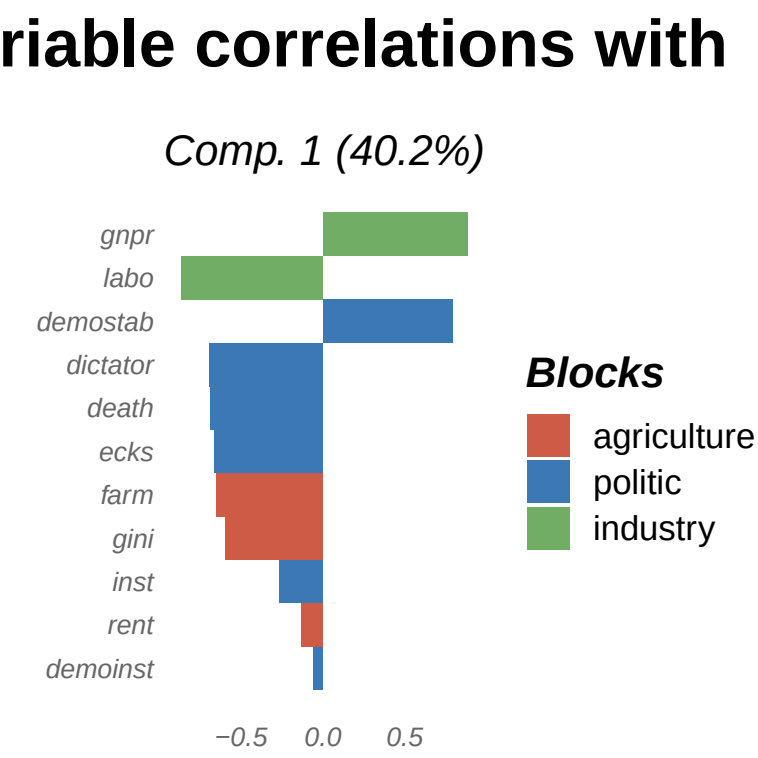
4.1.2 Corcircle plot

```
plot_var_2D(sgcca_out)
```



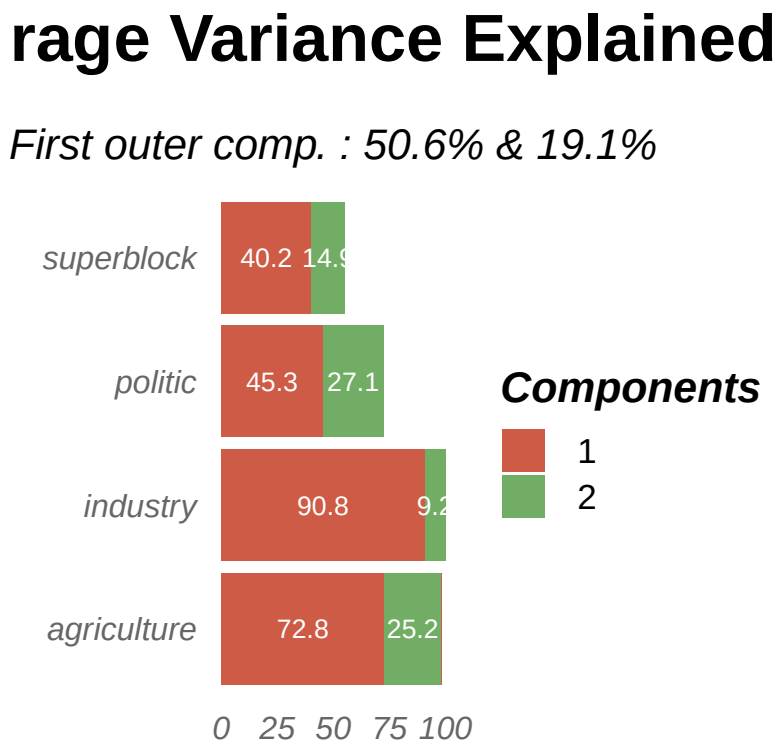
4.1.3 Fingerprint plot

```
plot_var_1D(sgcca_out)
```



4.1.4 Best explained blocks

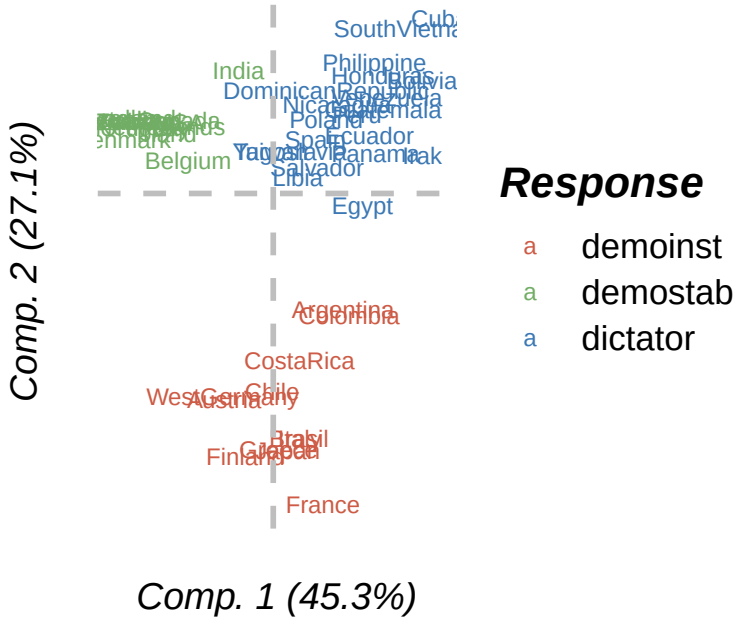
```
plot_ave(sgcca_out)
```



4.2 With the politic block, on the 2nd and the 3rd components

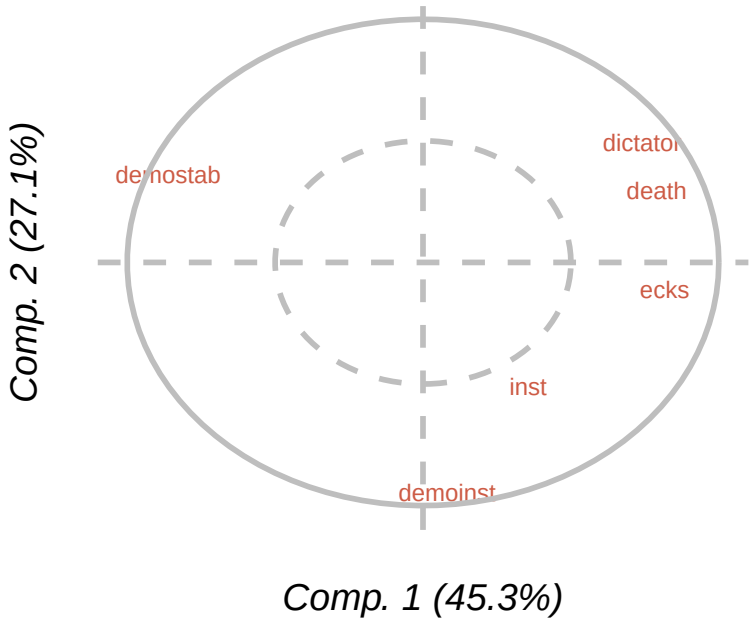
4.2.1 Samples plot

```
plot_ind(
    rgcca = sgcca_out,
    resp = response,
    compx = comp1,
    compy = comp2,
    i_block = 3)
```



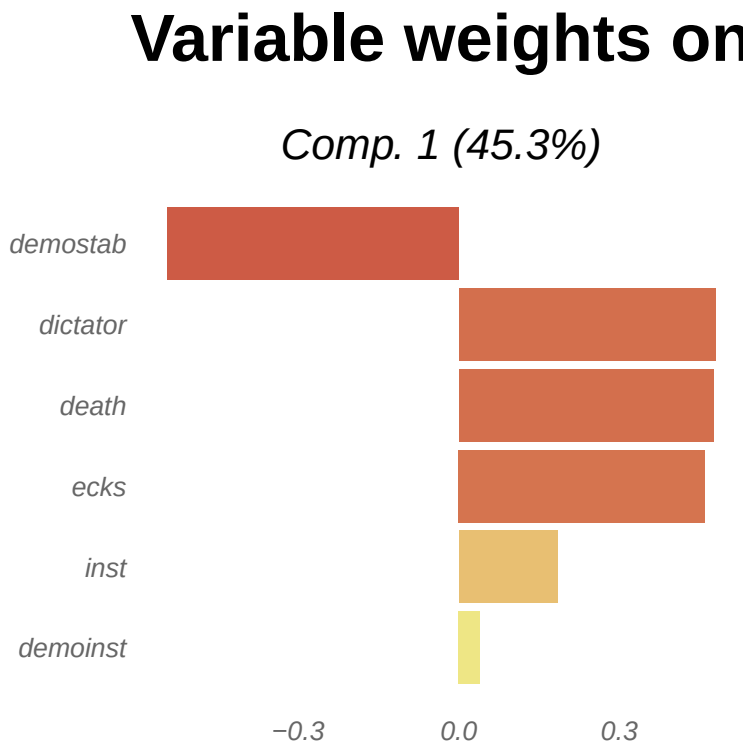
4.2.2 Corcircle plot

```
plot_var_2D(
    rgcca = sgcca_out,
    compx = comp1,
    compy = comp2,
    i_block = 3)
```



4.2.3 Fingerprint plot

```
plot_var_1D(
    rgcca = sgcca_out,
    comp = comp1,
    n_mark = nmark,
    i_block = 3,
    type = "weight")
```


Variable weights on
Comp. 1 (45.3%)
demostab
dictator
death
ecks
inst
demoinst
−0.3
0.0
0.3