SCGLR - An R Package for Supervised Component Generalized Linear Regression

Frédéric Mortier*, Catherine Trottier†‡, Guillaume Cornu* and Xavier Bry‡

March 7, 2016

Summary:
The objective of this paper is to present an *R* package, **SCGLR**, implementing a new PLS regression approach in the multivariate generalized linear framework. The method allows the joint modeling of random variables from different exponential family distributions, searching for common PLS-type components. We discuss several of the functions in the package focusing in particular on the two main ones: `scglr` and `scglrCrossVal`. The former constructs the components and performs the parameter estimation, while the latter selects the approriate number of components by cross-validation. The package is illustrated on an appropriate ecological dataset through which we aim at predicting the abundance of multiple tree genera given a large number of geo-referenced environmental variables.
Key words: Multivariate generalized linear model, partial least squares, Fisher Scoring, *R*

1 Introduction

The classical generalized linear model (GLM), used for modeling random variables from exponential family distributions, suffer from different limitations: (i) it does not allow modeling of more than one outcome at a time; (ii) for want of regularization, it cannot deal with many correlated regressors - whatever relevant causal factors they may represent - and thus requires some preliminary selection of regressors; (iii) the degree of explanatory realism of the model and the robustness of the prediction may be highly influenced by this selection.

We developed the supervised component generalized linear regression (SCGLR) method to overcome these limitations [1, 2]. SCGLR is a multivariate extension of partial least squares (PLS) regression to the generalized linear framework. It allows the relevant information contained in the data to be summarized in a few common components that can predict, as best as possible, the multivariate outcomes. The method was motivated by ecological applications where there is

*UPR Biens et Services des Ecosystèmes Forestiers tropicaux (B&SEF), Département Environnements et Sociétés du CIRAD, Campus International de Baillarguet, TA C-105/D, FR-34398 Montpellier Cedex 5

†UFR IV, Université Paul Valéry, Route de Mende, FR-34199 Montpellier Cedex 5

‡UMR I3M, Equipe Probabilités et Statistique, Place Eugène Bataillon, FR-34095 Montpellier Cedex 5

interest in understanding how communities of tree species are structured based on environmental traits. Because species data can be collected through different measurement processes, the outcomes arise from several types of distributions. For example, some species may just be measured through presence/absence and others through count data (e.g., binomial or Poisson distributions). The originality of the SCGLR approach is to allow the simultaneous modeling of distributions from exponential family; Bernoulli, binomial, Gaussian and Poisson distributions can currently be handled in the **SCGLR** package.

SCGLR is based on a multivariate GLM and performs a PLS regression on each step of the GLM estimation algorithm. It uses both the responses (e.g., species abundances) and the regressors to calculate common components. Components are constructed sequentially: the first one maximizes some trade-off between its variance and the goodness of fit of the GLM that takes it as sole regressor (cf below for more details), the second one is its complement in the space orthogonal to the first component, etc, until we get a set of K complementary and mutually independent components, just as in principal component analysis (PCA). Ultimately, these components are used in a GLM as covariates, allowing them to have specific effects on each response. The optimal number of components on which to base the linear predictors is the one that allows the best prediction in cross-validation. The quality of prediction is assessed through various well-known criteria.

In this paper, we introduce an *R*-package [4] that performs SCGLR. We first briefly review the mathematical basis of the method, then describe the program's features and usage. We illustrate **SCGLR** on a dataset built from the CoForChange[1] and CoForTips[2] databases. It gives the abundance of 27 common tree genera in the tropical moistforest of the Congo-Basin and measurements on 40 geo-referenced environmental variables for one thousand 8 by 8 km plots (observations). Each plot's data were obtained by aggregating data measured on a variable number of previously sampled 0.5 ha sub-plots. Geo-referenced environmental variables were used to describe the physical factors as well as vegetation characteristics.

2 Description of the SCGLR statistical approach

Let $Y = \left(y^1, \ldots, y^q\right)$ be a matrix of q responses we want to model. For this, regressors are classified into two groups. Let $X = \left(x^1, \ldots, x^p\right)$ be a matrix whose column-vectors code p regressors (possibly including dummy variables coding nominal covariates). These many regressors contain redundancy and hence there is a specific need for regularisation of the coefficients. Let also T be a matrix whose column-vectors code additional covariates, which are to be included as they are without any regularisation. SCGLR assumes that the q responses are dependent on an unknown number of mutually orthogonal components (linear combinations of the covariates X), along with covariates T. The components are assumed common to all the responses in that they play some role in the GLM fit of each response. Moreover, the components are designed to stay rather close to the principal directions of the covariates, that is, stray from the noise contained in the group of regressors.

[1] more information on CoForChange is available at `http://www.coforchange.eu`

[2] and for CoForTips `http://www.fordev.ethz.ch/research/active/CoForTips`

Let u be a p-coefficient vector. Just as in PCA, the structural strength of a component $f = Xu$ is measured through its variance under a unit-norm constraint on u. The components are determined sequentially. The first component $f^1 = Xu^1$ optimizes a trade-off between the goodness-of-fit of a multivariate GLM using f^1 as common explanatory variable along with T, and the variance of f^1. To be precise, the Fisher scoring algorithm (FSA) used to estimate the GLM of Y on $< f^1, T >$ has been altered in its Generalized Least Squares (GLS) step of the current linearised model according to two alternative approaches.

LPLS approach: the first one corresponds to a Local Partial Least Squares procedure (LPLS) and maximises the criterion:

$$\sum_{k=1}^{q} ||z_k||^2_{W_k} cos^2_{W_k}(z_k; < Xu, T >) ||Xu||^2_{W_k}$$

where z_k are the working variables in the current linearised model,

- $||z_k||^2_{W_k} cos^2_{W_k}(z_k; < Xu, T >)$ is a measure of goodness-of-fit,
- $||Xu||^2_{W_k}$ is a measure of structural relevance.

A Singular Value Decomposition algorithm is used to optimize criterion LPLS. Taking f^1's variance into account regularises coefficient vector u^1, which the standard GLS procedure does not. Covariates in T are considered in the regression step, but not taken into account in component f^1. Full details on this method can be found, more formally expressed, in [2].

SR approach: the second one takes the Structural Relevance (SR) more explicitely into account and maximises the criterion:

$$\psi(u)^{1-s} \phi(u)^s$$

with $\psi(u) = \sum_{k=1}^{q} ||z_k||^2_{W_k} cos^2_{W_k}(z_k; < Xu, T >)$ contains the goodness-of-fit measure, and two options for function ϕ:

- Component Variance (CV): $\phi(u) = ||Xu||^2_W$
- Variable Powered Inertia (VPI): $\phi(u) = \left(\sum_{j=1}^{p} < Xu | x^j >^{2l}_W \right)^{\frac{1}{l}}$, when X consists of p standardised numeric variables x^j.

Tuning-parameter l allows to draw components towards more or less local variable bundles. Tuning-parameter s weights the structural relevance in the criterion. This criterion is maximised using an Iterated Normed Gradient algorithm. We recommend the second criterion (SR) for being more flexible. Default is VPI.

Once f^1 is obtained, then X is deflated on f^1, that is, projected onto its orthogonal space, yielding residual predictor matrix X^1, and the second component f^2 is sought in X^1 ($f^2 = X^1 u^2$) according to the same trade-off optimization, but including f^1 amongst extra-covariates T, from there on. And so forth for higher rank components. So, current component $f^r = X^{r-1} u^r$ is based on the matrix of residuals obtained by projecting the original variables X onto

the space orthogonal to all previous components $\{f^1, ..., f^{r-1}\}$, and the estimation procedure of the linearized model within the modified FSA step takes into account covariates $T \cup \{f^1, ..., f^{r-1}\}$.

Finally, given some integer R, a multivariate GLM of the responses is performed on the set $F^R = \{f^1, ..., f^R\}$ of the first R components, along with covariates T, yielding a coefficient vector for each response y_k, with corresponding linear predictor $\eta_k = F^R\gamma_k + T\delta_k$ $(k = 1, \ldots, q)$. Now, each component f^r can be expressed as a linear combination of the original predictors: $f^r = Xv^r$. Hence, in matrix form, we have: $F^R = XV^R$. Thus, we can express each linear predictor as a linear combination of the regressors: $\eta_k = X\beta_k + T\delta_k$ with $\beta_k = V^R\gamma_k$.

The coefficients β_k and δ_k can be used in cross-validation to determine the optimal number R of components in order to avoid overfitting. Given the number of components R under trial, the observations are repeatedly partitioned into 2 sub-samples: C (for calibration) and S (for testing). On each partition, C is used to calculate the R components, and hence the β's and δ's, which in turn are used to predict the expectation of the responses on S. An appropriate criterion of predictive power is then calculated (depending on the distribution of the responses) and averaged over all (C, S) partitions considered. Eventually, we consider and select the value of R yielding the best performance.

3 Program description and usage

3.1 Main description

SCGLR is developed using $R \geq 3.0$ version [5]. **SCGLR** is a set of R functions illustrated on a floristic data set, *genus*. `scglr()` and `scglrCrossval()` are the two main high level functions, which are respectively dedicated to fitting the model and selecting the number of components. `print()`, `summary()` and `plot()` methods are also available for the `scglr()` function.

The call to `scglr()` has the following structure:

```
results.scglr <- scglr(formula,data,family,K,size,offset,subset,
                       na.action,crit,method)
```

The *formula*, *data*, *family* and K arguments are required and *size* must be specified if binomial variables are used. The *formula* object of the *Formula* class [7] is composed of two or three terms. The first term describes the dependent variables whereas the second term describes the regressors used to construct components, and the third term describes additional covariates to be included in the model but not used in the linear combination giving the components. The first two terms should be separated by a $\sim$ symbol as classical R formula objects, whereas the second and third terms, if any, should be separated by a | symbol. All the elements in each term are separated by a + sign. The formula can be written out explicitly or provided using the `multivariateFormula()` function. For example, if $ny = (\text{“}y1\text{”}, \text{“}y2\text{”})$ contains the names of the dependent variables, $nx = (\text{“}x1\text{”}, \ldots, \text{“}x5\text{”})$ the names of the regressors used to construct the component, and $nz = (\text{“}z1\text{”}, \text{“}z2\text{”}, \text{“}z3\text{”})$ the names of the additional regressors

```
myformula <- multivariateFormula(ny,nx,nz)
myformula

## y1 + y2 ~ x1 + x2 + x3 + x4 + x5 | z1 + z2 + z3
```

The *data* argument is an object of the *data.frame* class. The *family* is a vector of characters describing the family of each dependent variable. In `SCGLR`, "bernoulli", "binomial", "poisson" or "gaussian" are allowed. For Poisson outcomes, the *offset* argument is either a vector or a matrix of size: number of observations $\times$ number of Poisson dependent variables, allowing a different offset for each dependent variable. If binomial dependent variables are included in the model, *size* must be specified as a matrix describing the number of trials.

The output of the `scglr` function is an object of class **SCGLR** made of:

- u: matrix of size: number of regressors $\times$ number of components, contains the component-loadings, i.e., the coefficients of the regressors in the linear combination giving each component.
- comp: matrix of size : number of statistical units $\times$ number of components, having the components as column vectors.
- compr: matrix of size : number of statistical units $\times$ number of components, having the standardized components as column vectors.
- gamma: list of length number of dependant variables. Each element is a matrix of coefficients, standard errors, z-values and p-values.
- beta: matrix of size: number of regressors + 1 (intercept) $\times$ number of dependent variables, contains the coefficients of the regression on the original regressors X.
- lin.pred: data.frame of size: number of statistical units $\times$ number of dependent variables, the fitted linear predictor.
- xFactors: data.frame containing the nominal regressors.
- xNumeric: data.frame containing the quantitative regressors.
- inertia: matrix of size: number of components $\times$ 2, contains the percentage and cumulative percentage of the overall regressors' variance, captured by each component.
- deviance: vector of length: number of dependent variables, gives the deviance of each y_k's GLM on the components.

The `print()` method gives the values of inertia and deviance. `summary()` gives inertia, deviance, and three additional tables. The first one contains the square correlations between X's and the components, along with two columns highlighting the plane on which the regressors are best projected and their associated square correlations. The second table presents the square correlations between fitted linear predictors and components, with two more columns corresponding to the plane on which the regressors are best projected and their associated square correlations. These two tables summarize how well the regressors

and the dependent variables, through their linear predictors, are represented on the planes. The third table presents the γ values obtained from the GLM; only γ's with p-values lower than a given cutoff (default 0.05) are printed.

3.2 Plots

Several specialized plot commands are available to show the results of `scglr()`. They are all based on the **ggplot2** package developed by Wickham, H. [6] and as such can be further customized (i.e., one can add more layers or labels for example).

- `plot()`: general function to produce various plots from the `scglr()` output by selecting elements to draw. This selection is specified by parameters whose names can be abbreviated (e.g. *pred.col* will be understood as *predictors.color*). Options can be set globally using `options("plot.SCGLR")`. It will then provide default values that can be further overriden by giving explicit parameter value.
- `barplot()`: takes an **SCGLR** object as input and produces a barplot of the inertia per component.
- `pairs()`: takes an **SCGLR** object and produces an array plot for pairwise combinations of components (all components or a selected subset).

3.3 Selecting the number of components

The appropriate number of components to best predict dependent variables remains unknown and must be selected. We propose a cross-validation approach using different criteria to determine the number of components. The call to the `scglrCrossVal()` function shares the same arguments as the `scglr()` function with two additional arguments *nfolds* and *type*:

```
scglrCrossVal(formula,data,family,K,nfolds,types,size,
              offset,subset,na.action,crit,method)
```

nfolds is the number of subsamples to be used in the cross-validation - default is 5. Although *nfolds* can be as large as the sample size (leave-one-out CV), this is not recommended for large datasets. *type* is the criterion to use for cross-validation. Currently five options are available in a general setting: "mspe" (Mean Squared Prediction Error), "likelihood", "aic", "bic" and "aicc". When all dependent variables are Bernoulli, the option "auc" (area under ROC curve) enables to measure the prediction performance. The output of the procedure is a $(q \times (K+1))$ matrix containing the criterion values for each response variable and each model. The first column corresponds to the model without any component.

4 Examples

4.1 Floristic data set

We illustrate **SCGLR** using the data *genus*. This example highlights the use of the multivariate Poisson count distribution with an offset.

genus is a dataset built from the CoForChange database. It gives the abundance of 27 common tree genera in the tropical moistforest of the Congo-Basin and measurements on 40 geo-referenced environmental variables for one thousand 8 by 8 km plots (observations). Data on each plot were obtained by aggregating the data measured on a variable number of previously sampled 0.5 ha sub-plots. The geo-referenced environmental variables were used to describe 16 physical factors pertaining to the description of topography, geology and rainfall and the remaining variables give the vegetation characteristics defined through 16-days enhanced vegetation index (EVI).

```
data("genus")
dim(genus)

## [1] 1000   69
```

```
names(genus)

##  [1] "gen1"       "gen2"       "gen3"       "gen4"
##  [5] "gen5"       "gen6"       "gen7"       "gen8"
##  [9] "gen9"       "gen10"      "gen11"      "gen12"
## [13] "gen13"      "gen14"      "gen15"      "gen16"
## [17] "gen17"      "gen18"      "gen19"      "gen20"
## [21] "gen21"      "gen22"      "gen23"      "gen24"
## [25] "gen25"      "gen26"      "gen27"      "altitude"
## [29] "pluvio_yr"  "forest"     "pluvio_1"   "pluvio_2"
## [33] "pluvio_3"   "pluvio_4"   "pluvio_5"   "pluvio_6"
## [37] "pluvio_7"   "pluvio_8"   "pluvio_9"   "pluvio_10"
## [41] "pluvio_11"  "pluvio_12"  "geology"    "evi_1"
## [45] "evi_2"      "evi_3"      "evi_4"      "evi_5"
## [49] "evi_6"      "evi_7"      "evi_8"      "evi_9"
## [53] "evi_10"     "evi_11"     "evi_12"     "evi_13"
## [57] "evi_14"     "evi_15"     "evi_16"     "evi_17"
## [61] "evi_18"     "evi_19"     "evi_20"     "evi_21"
## [65] "evi_22"     "evi_23"     "lon"        "lat"
## [69] "surface"
```

We chose to use the covariate "geology" as an additional factor not directly used in the component construction because of the demonstrated importance of the geological substrates on the spatial distribution of tree species in the Congo Basin [3]. We also used the covariate "surface" as an offset and we added the product $I(lon * lat)$ as a new covariate.

```
ny <- names(genus)[1:27]
sx <- which(names(genus) %in% c("geology","surface"))
nx <- names(genus)[-c(1:27,sx)]
family <- rep("poisson",length(ny))
formula <- multivariateFormula(ny,c(nx,"I(lon*lat)"),"geology")
formula

## gen1 + gen2 + gen3 + gen4 + gen5 + gen6 + gen7 + gen8 + gen9 +
##     gen10 + gen11 + gen12 + gen13 + gen14 + gen15 + gen16 + gen17 +
##     gen18 + gen19 + gen20 + gen21 + gen22 + gen23 + gen24 + gen25 +
```

```
##     gen26 + gen27 ~ altitude + pluvio_yr + forest + pluvio_1 +
##     pluvio_2 + pluvio_3 + pluvio_4 + pluvio_5 + pluvio_6 + pluvio_7 +
##     pluvio_8 + pluvio_9 + pluvio_10 + pluvio_11 + pluvio_12 +
##     evi_1 + evi_2 + evi_3 + evi_4 + evi_5 + evi_6 + evi_7 + evi_8 +
##     evi_9 + evi_10 + evi_11 + evi_12 + evi_13 + evi_14 + evi_15 +
##     evi_16 + evi_17 + evi_18 + evi_19 + evi_20 + evi_21 + evi_22 +
##     evi_23 + lon + lat + I(lon * lat) | geology

offset <- genus$surface
```

```
K <- 4
genus.cv <- scglrCrossVal(formula=formula,data=genus,family=family,
                          K=K,nfolds=5,type="mspe",offset=offset,
                          method=methodSR(l=1, s=1/2))
```

Concerning the selection procedure, in order to produce comparable values for possibly very different response variables, we used the following heuristic. For each response and each of the $K+1$ models (one model for each number K of components and one for no component), divide the criterion value by its median over all the models. Then calculate for each number of components the mean of the standardized values over the different response variables. Alternatively, the mean can be used to normalize instead of the median.

```
criterion <- t(apply(genus.cv,1,function(x) x/mean(x)))
criterion.mean <- apply(criterion,2,mean)
K.cv <- which.min(criterion.mean)-1
```

In the expression of *K.cv*, the minus 1 enables to relabel the output such that it matches the actual number of components used. Plotting *criterion* values (see Figure˜1) displays the change in the selection criterion as the number of components increases. Here, the *criterion* is minimized for 3 components. We can therefore call `scglr()` with $K = 3$.

```
plot(0:K,criterion.mean, type="l",
     xlab="K, number of components", ylab="Criterion (MSPE)")
Axis(side=1,at=0:K)
abline(v=K.cv,col=2)
```



```
genus.scglr<-scglr(formula=formula,data=genus,family=family,
                   K=K.cv,size=NULL,offset=offset,
                   method=methodSR(l=1,s=1/2))
```

Printing *genus.scglr*:

```
print(genus.scglr)

##
## Call:  scglr(formula = formula, data = genus, family = family, K = K.cv,
##     size = NULL, offset = offset, method = methodSR(l = 1, s = 1/2))
##
## Inertia:
##       sc1        sc2        sc3
```

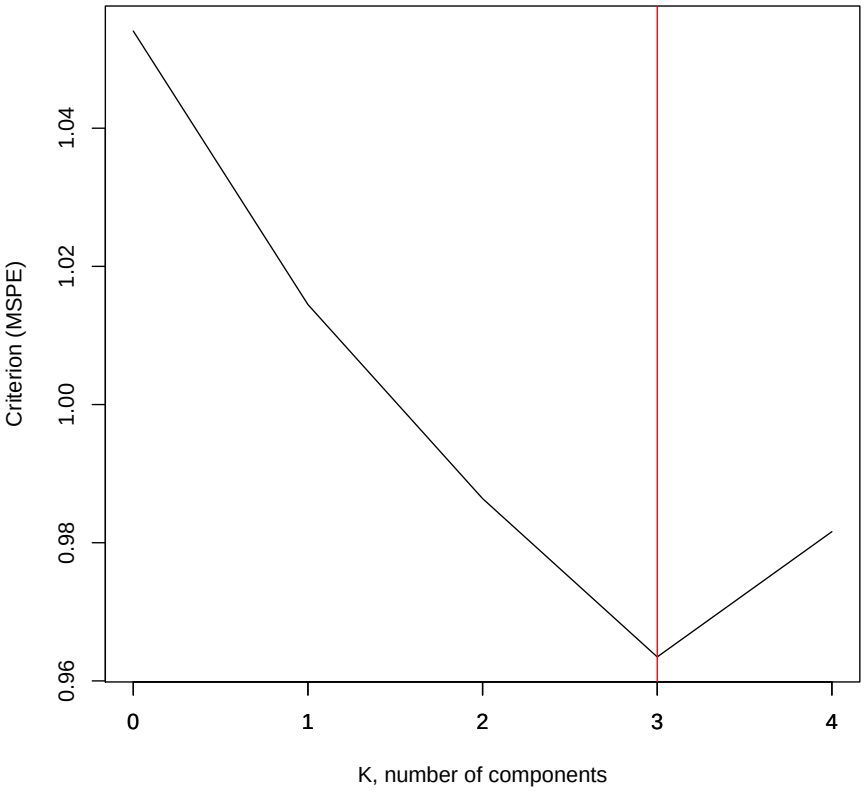


Figure 1: Mean Squared Prediction Error (MSPE) as a function of the number of components.

```
## 0.3525894  0.2160546  0.1314694
##
## Deviance:
##      gen1       gen2       gen3       gen4       gen5
##  2348.607   2792.147   1597.398   1455.839   1474.662
##      gen6       gen7       gen8       gen9      gen10
##  1864.100   2696.955  16973.876   1646.440  17886.477
##     gen11      gen12      gen13      gen14      gen15
##  5715.859   7954.563   8755.673   9460.521   2815.295
##     gen16      gen17      gen18      gen19      gen20
##  2128.786   2973.116   1536.648  28310.115   2145.009
##     gen21      gen22      gen23      gen24      gen25
##  3668.294   1478.181   3883.473   9905.367  12437.319
##     gen26      gen27
## 10503.350  12792.850
```

Inertia of the 3 components (see Figure˜2):

```
barplot(genus.scglr)
```

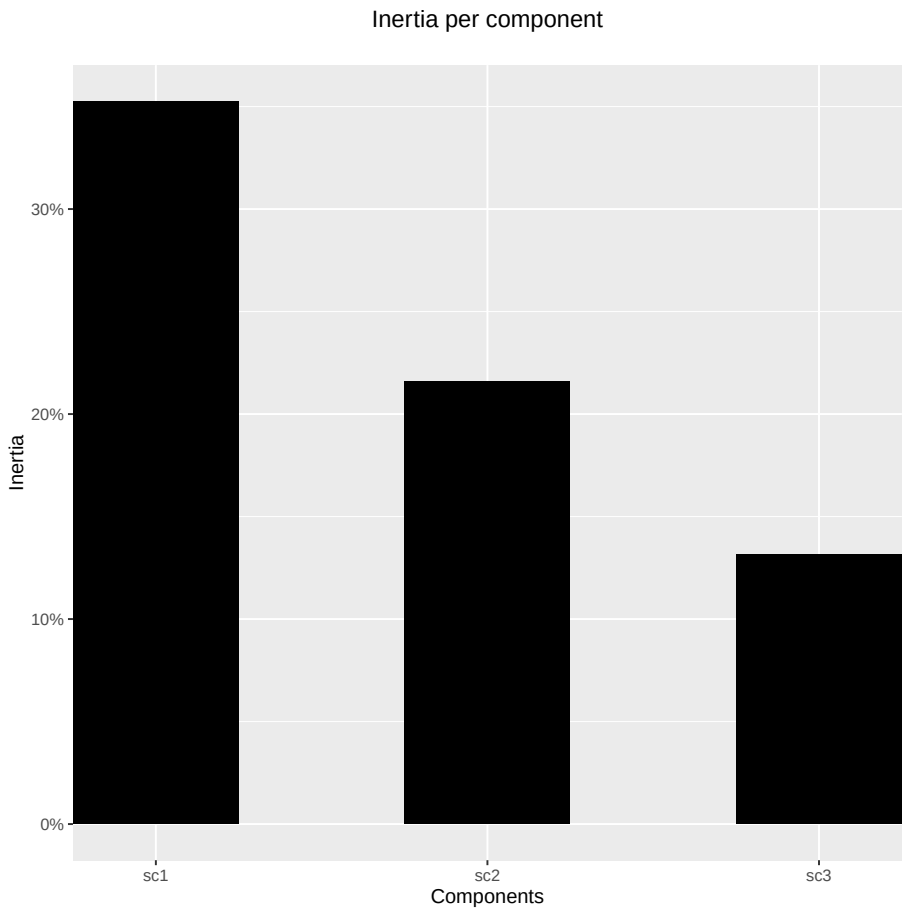


Figure 2: Barplot of inertia per component

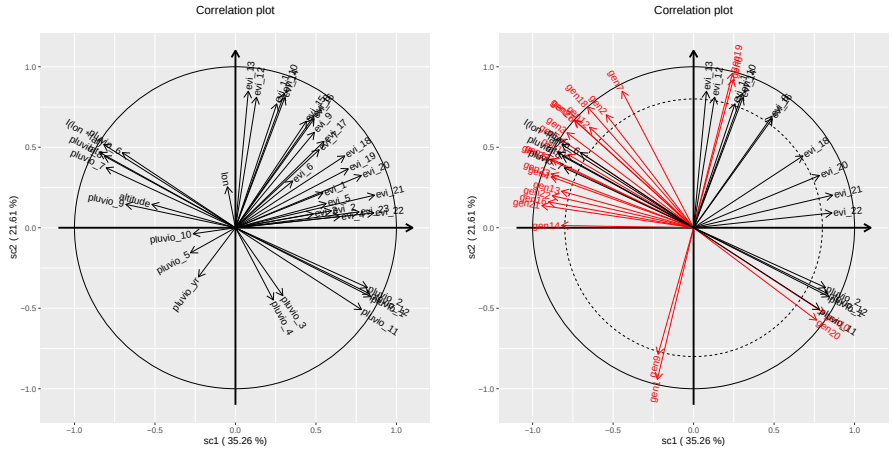
The following two commands create the plots in Figure˜3. The first one gives a simple correlation plot (see Figure˜3a)

The second correlation plot (see Figure˜3b) displays only the linear predictors and covariates whose norms in the selected plane exceed the threshold specified by the "thr" *styles* element.

```
plot(genus.scglr)
plot(genus.scglr, threshold=0.8, predictors=TRUE)
```

Finally, we present the *pairs* plot on the planes spanned by components (see Figure˜4):

```
pairs(genus.scglr,ncol=2,label.size=0.5)
```



(a) Simple correlation plot

(b) Correlation plot with linear predictors and covariates passing a threshold of 0.8

Figure 3: Two sample plots

q \ n	100	1000	10000
10	0.075	1.11	17.10
100	2.942	5.644	31.60

Table 1: Mean user times (in seconds) to calculate one SCGLR component for 10 or 100 dependent variables and 100, 1000 or 10,000 observations

```
## NULL
```

5 Conclusion

The main features of the *R* package **SCGLR** have been explained and illustrated in this paper using the data set *genus* provided with the package. Contrary to existing PLS-dedicated packages that can only handle Gaussian data, **SCGLR** provides a unified framework to deal with multivariate outcomes arising from any exponential family distribution. The computational time required to run `scglr` depends on the dimension of the problem. Table˜1 provides the mean user times required to run 100 simulations of the *scglr()* algorithm using one component with $p = 100$ covariates and a varying number of dependent variables ($q = 10$ and 100) and varying sample sizes ($n = 100, 1000$ and $10,000$). These results highlight the efficiency of the **SCGLR** package.

6 Acknowledgments

This research was supported by ITG-SEITA and was part of the CoForChange project (`www.coforchange.eu`), funded by the ERA-Net BiodivERsA, with the

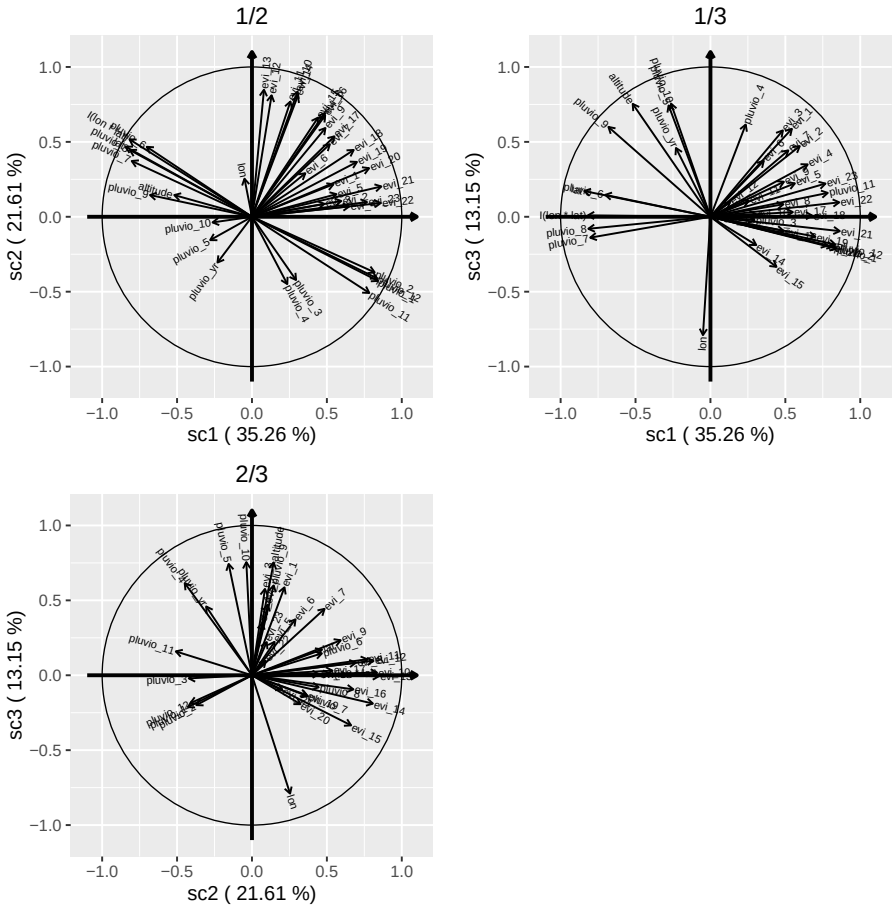


Figure 4: Correlation plots on planes spanned by components

national funders ANR (France) and NERC (UK), part of the 2008 BiodivERsA call for research proposals, involving 16 European, African and international partners including a number of timber companies (see the list on the website, `http://www.coforchange.eu/partners`), and of the CoForTips project, funded by the ERA-Net BiodivERsA, with the national funders FWF (Austria), BelSPO (Belgium) and ANR (France), part of the 2011-2012 BiodivERsA call for research proposals (`http://www.fordev.ethz.ch/research/active/CoForTips`).

References

[1] Xavier Bry, Catherine Trottier, Thomas Verron, and Frédéric Mortier. Supervised component generalized linear regression using a pls-extension of the fisher scoring algorithm. In *COMPSTAT 2012 (Cyprus)*, pages 121–129, 2012.

[2] Xavier Bry, Catherine Trottier, Thomas Verron, and Frédéric Mortier. Su-

pervised component generalized linear regression using a pls-extension of the fisher scoring algorithm. *Journal of Multivariate Analysis*, 119(0):47 – 60, 2013.

[3] Adeline Fayolle, Bettina Engelbrecht, Vincent Freycon, Frédéric Mortier, Mike Swaine, Maxime Réjou-Méchain, Jean-Louis Doucet, Nicolas Fauvet, Guillaume Cornu, and Sylvie Gourlet-Fleury. Geological substrates shape tree species and trait distributions in African moist forests. *Plos One*, 2012.

[4] Frédéric Mortier, Catherine Trottier, Guillaume Cornu, and Xavier Bry. **SCGLR***: Supervised Component Generalized Linear Regression (SCGLR)*, 2013. *R* package version 1.1.

[5] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2013.

[6] Hadley Wickham. *ggplot2: elegant graphics for data analysis*. Springer-Verlag New York, 2009.

[7] Achim Zeileis and Yves Croissant. Extended model formulas in *R*: Multiple parts and multiple responses. *Journal of Statistical Software*, 34(1):1–13, 2010.