

A flexible framework for sparse simultaneous component based data integration

Katrijn Van Deun^{*1} and Tom F Wilderjans² and Robert A van den Berg^{1,3} and Anestis Antoniadis⁴ and Iven Van Mechelen¹

¹Center for Computational Systems Biology SymbioSys, Katholieke Universiteit Leuven, 3000 Leuven, Belgium

²Department of Psychology, Katholieke Universiteit Leuven, 3000 Leuven, Belgium ³Current address: GlaxoSmithKline Biologicals,1330 Rixensart, Belgium ⁴Laboratoire Jean Kuntzmann, Université Joseph Fourier, 38041 Grenoble, France

Email: Katrijn Van Deun^{*} - katrijn.vandeun@psy.kuleuven.be; Tom F Wilderjans - tom.wilderjans@psy.kuleuven.be; Robert A van den Berg - robert.x.van-den-berg@gskbio.com; Anestis Antoniadis - anestis.antoniadis@imag.fr; Iven Van Mechelen - iven.vanmechelen@psy.kuleuven.be;

^{*}Corresponding author

Abstract

1 Background

High throughput data are complex and methods that reveal structure underlying the data are most useful. Principal component analysis, frequently implemented as a singular value decomposition, is a popular technique in this respect. Nowadays often the challenge is to reveal structure in several sources of information (e.g., transcriptomics, proteomics) that are available for the same biological entities under study. Simultaneous component methods are most promising in this respect. However, the interpretation of the principal and simultaneous components is often daunting because contributions of each of the biomolecules (transcripts, proteins) have to be taken into account.

2 Results

We propose a sparse simultaneous component method that makes many of the parameters redundant by shrinking them to zero. It includes principal component analysis, sparse principal component analysis, and ordinary simultaneous component analysis as special cases. Several penalties can be tuned that account in different ways for the block structure present in the integrated data. This yields known sparse approaches as the

lasso, the ridge penalty, the elastic net, the group lasso, sparse group lasso, and elitist lasso. In addition, the algorithmic results can be easily transposed to the context of regression. Metabolomics data obtained with two measurement platforms for the same set of *Escherichia coli* samples are used to illustrate the proposed methodology and the properties of different penalties with respect to sparseness across and within data blocks.

3 Conclusion

Sparse simultaneous component analysis is a useful method for data integration: First, simultaneous analyses of multiple blocks offer advantages over sequential and separate analyses and second, interpretation of the results is highly facilitated by their sparseness. The approach offered is flexible and allows to take the block structure in different ways into account. As such, structures can be found that are exclusively tied to one data platform (group lasso approach) as well as structures that involve all data platforms (Elitist lasso approach).

4 Availability

The additional file contains a MATLAB implementation of the sparse simultaneous component method.

Background

The integrated analysis of multiple data sets obtained for the same biological entities under study, has become one of the major challenges for data analysis in bioinformatics and computational biology. Two main causes for this trend are the availability of complementary measurement platforms and the systemic approach to biology; in both cases, multiple data sets are obtained on the same set of samples (e.g., culture samples, tissues). First, examples where several measurement platforms are included are the study of the metabolome composition of *Escherichia coli* (*E. coli*) using several analytical chemical methods to screen for metabolites [1] and the combination of cDNA and Affymetrix chips applied to sixty cancer cell lines [2]. In both examples, there is overlap in the metabolites or genes screened but also complementarity. Second, the modern systemic approach to biology leads to a probing of the biological system on different levels in the cellular organization, such as for example the transcript, protein, and metabolite level [3]. These approaches lead to situations where several data blocks are obtained that are coupled in the sense that they were obtained for the same set of samples. A key issue in integrative data analysis is to analyze such

data simultaneously instead of separately or sequentially as this yields an aggregated view. In this respect, simultaneous component methods, that are an extension of principal component analysis (PCA) to the case of multiple coupled data blocks, were proposed and successfully used [4–7].

However, a drawback of component based methods like PCA is their lack of sparseness: Processes underlying the data are revealed by a weighted combination of all variables (these are the genes, transcripts, proteins, metabolites in the aforementioned examples). From an interpretational point of view, this is not very attractive and it also does not reflect that biological processes are expected to be governed by a limited number of genes [8]. The problem holds even more for simultaneous component methods as these involve multiple large sets of variables. To deal with this issue, sparse approaches have been proposed mainly within the context of regression analysis (e.g., [9, 10]) but also for principal component analysis [8, 11–14]: These select a limited number of variables by shrinking many of the weights to zero which is accomplished by proper penalization of these (regression) weights. A favorable characteristic of such penalty based methods is that the selection is built-in (in contrast to, for example, first filtering and then doing the regression/PCA). Here, we extend sparse PCA to sparse simultaneous component methods, accounting for the fact that the data are structured in several data blocks holding both shared and complementary information. The estimation procedure used is efficient and the associated MATLAB code can be found in the additional file.

First, we present the sparse simultaneous component model, starting from ordinary principal component analysis and sparse PCA. A generic modeling framework is introduced that incorporates several types of penalties. Then we present some results for metabolomics data obtained with two measurement platforms for the same set of *E. coli* samples and we validate the method by means of simulated data.

Results

Algorithm

Notation

We will make use of the following formal notation: matrices are denoted by bold uppercases, vectors by bold lower case, the transpose by the superscript T , and the cardinality by the capital of the letter used to run the index (e.g., this paper deals with K data matrices \mathbf{X}_k with k running from 1 to K), see [15].

Throughout the paper, we suppose that all variables are mean-centered and scaled to norm one.

Model

Simultaneous component analysis is an extension of **principal component analysis (PCA)** to the case of multiple coupled data matrices. Consider the PCA of a single data block \mathbf{X}_k containing the scores of I_k objects (e.g., batches, arrays) on J_k variables (e.g., metabolites, genes). In a first model formulation [16] based on component scores, PCA decomposes the data as follows,

$$\mathbf{X}_k = \mathbf{T}_k \mathbf{P}_k^T + \mathbf{E}_k \quad (1)$$

with \mathbf{T}_k the component scores of the I_k objects on the R components, \mathbf{P}_k (of size $J_k \times R$) the loadings, and \mathbf{E}_k (of size $I_k \times J_k$) the matrix of residuals. To identify the model, usually the constraints are imposed that the axes have a principal axes orientation and that the component scores are orthogonal: $\mathbf{T}_k^T \mathbf{T}_k = \mathbf{I}$. Another formulation of the PCA model is based on component weights as follows [17],

$$\mathbf{X}_k = \mathbf{X}_k \mathbf{W}_k \mathbf{P}_k^T + \mathbf{E}_k \quad (2)$$

with \mathbf{W}_k (of size $J_k \times R$) the component weights. Note that we can write $\mathbf{T}_k = \mathbf{X}_k \mathbf{W}_k$ resulting in the equivalence of models (1) and (2). However, usually (2) is constrained to have orthogonal weights:

$\mathbf{W}_k^T \mathbf{W}_k = \mathbf{I}$. Note that under a least squares approach to PCA, $\mathbf{P}_k = \mathbf{W}_k$ and thus also $\mathbf{P}_k^T \mathbf{P}_k = \mathbf{I}$.

The principal components are interpreted by considering the contribution of the variables to the components. For the score-based model (1) this is based on the fact that the loadings are equal to the correlation of the variables with the components (we suppose the variables to be mean-centered and scaled to norm one). Let \mathbf{x}_{jk} be the j th variable in data block k and \mathbf{t}_{rk} the r th component for block k , then

$$r(\mathbf{x}_{jk}, \mathbf{t}_{rk}) = p_{jrk}, \quad (3)$$

with $r(\cdot, \cdot)$ used as a notation for correlation and p_{jrk} the loading of the j th variable on the r th component of block k . In the weight-based model (2), interpretation of the components is based on the weights as these express each component as a weighted linear combination of the variables,

$$\mathbf{t}_{rk} = \mathbf{X}_k \mathbf{w}_{rk}. \quad (4)$$

For both model formulations this implies that for each component a total of J_k correlations or weights have to be taken into account in the interpretation. Especially in the case of omics data, that usually consist of thousands of variables, there is a need for methods that facilitate interpretation. To that end, [14] proposed a **sparse PCA method** for the weight based model (2), that shrinks a (large) number of

component weights to zero. Their method is based on a least-squares approach to PCA model (2) in which the objective function is augmented with an l_1 penalty (also named lasso) and an l_2^2 (ridge) penalty:

Minimize with respect to \mathbf{W}_k and \mathbf{P}_k

$$L(\mathbf{W}_k, \mathbf{P}_k) = \|\mathbf{X}_k - \mathbf{X}_k \mathbf{W}_k \mathbf{P}_k^T\|^2 + \lambda_L \|\mathbf{W}_k\|_1 + \lambda_R \|\mathbf{W}_k\|_2^2, \quad (5)$$

such that $\mathbf{P}_k^T \mathbf{P}_k = \mathbf{I}$ and with $\lambda_L \geq 0$ and $\lambda_R \geq 0$ tuning parameters for the lasso and ridge penalties respectively, $\|\mathbf{W}_k\|_1 = \sum_{j_k, r} |w_{j_k r}|$ and $\|\mathbf{W}_k\|_2^2 = \sum_{j_k, r} w_{j_k r}^2$. The *lasso*, tuned by the parameter λ_L , has the property to simultaneously shrink coefficients and select variables, keeping only those variables with the highest coefficients. The higher λ_L , the stronger the shrinkage and selection. Note that the selection is done in an unstructured way meaning that correlations between variables are not taken into account. The *ridge* penalty, tuned by λ_R , only shrinks the coefficients and does not perform variable selection (none of the coefficients becomes zero). It is often introduced when it is of interest to group correlated variables [10] or in case of ill-conditioned optimization problems (see [18]) to solve the non-uniqueness of the parameter estimates. A particular case is regression analysis with more variables than objects, $J_k > I_k$, which yields an under determined estimation problem. In the context of PCA, this is of relevance for model (5) because the estimation of the component weights boils down to a regression analysis. Adding the ridge penalty with $\lambda_R > 0$ solves the non-uniqueness; in addition, with the appropriate normalization, the ridge ensures that the solution of (5) yields the principal components in case $\lambda_L = 0$ (see [14]).

The **simultaneous component decomposition** of K coupled data blocks \mathbf{X}_k having a common set of samples (so $I_1 = \dots = I_K = I$) is given by imposing the constraint that all \mathbf{T}_k are equal. Applied to the score based model this gives:

$$\mathbf{X}_k = \mathbf{T} \mathbf{P}_k^T + \mathbf{E}_k, \quad (6)$$

for all k and under the constraints of a principal axes orientation and orthogonality of the component scores: $\mathbf{T}^T \mathbf{T} = \mathbf{I}$. Applying the idea of a common matrix of component scores to the weight based model as used by [14], can be realized as follows,

$$\begin{aligned} [\mathbf{X}_1 \dots \mathbf{X}_K] &= \\ &[\mathbf{X}_1 \dots \mathbf{X}_K] [\mathbf{W}_1^T \dots \mathbf{W}_K^T]^T [\mathbf{P}_1^T \dots \mathbf{P}_K^T] \\ &+ [\mathbf{E}_1 \dots \mathbf{E}_K] \end{aligned} \quad (7)$$

$$= \mathbf{T}[\mathbf{P}_1^T \dots \mathbf{P}_K^T] + [\mathbf{E}_1 \dots \mathbf{E}_K], \quad (8)$$

under the constraint of a principal axes orientation and orthogonal loadings: $[\mathbf{P}_1^T \dots \mathbf{P}_K^T][\mathbf{P}_1^T \dots \mathbf{P}_K^T]^T = \mathbf{I}$. Simultaneous component model (7) shows that the common component scores \mathbf{T} lie in the space spanned by all variables, this is from all data blocks. For ease of notation, we will use the shorthand notation $\mathbf{X}_c = [\mathbf{X}_1 \dots \mathbf{X}_K]$ (of size $I \times \sum_k J_k$) and $\mathbf{P}_c = [\mathbf{P}_1^T \dots \mathbf{P}_K^T]^T$ and $\mathbf{W}_c = [\mathbf{W}_1^T \dots \mathbf{W}_K^T]^T$ (both of size $\sum_k J_k \times R$). Note that several simultaneous component models were proposed in the literature: [6] gives an overview that emphasizes the different ways of weighting the data blocks in connection to different principles to realize a fair integration of the data.

The problem that a lot of variables have to be taken into account when interpreting the components is exacerbated in the case of simultaneous component analysis as this involves several blocks of variables. To solve for this problem, we propose to go for a **sparse simultaneous component method** by penalizing either the loadings (in the context of the score based model) or the component weights (in the context of the weights based model) within a least-squares approach. One possibility, in line with sparse PCA, is to use the lasso penalty if necessary in conjunction with a ridge penalty (when grouping of correlated variables is of interest or when $\sum_k J_k > I$). However, other types of penalties can be used that, when selecting variables, explicitly take into account that variables belong to (pre-defined) groups/blocks by selecting variables within blocks only, between blocks only (by setting all weights/loadings of an entire block to zero, i.e. dropping an entire group of variables at once), or both within and between blocks.

A penalty that introduces selection only within each group is *Elitist lasso* (mixed $l_{1,2}$ norm), defined for the r th component as

$$\lambda_E \sum_k \|\mathbf{w}_{rk}\|_{1,2} = \lambda_E \sum_k \left(\sum_{j_k} |w_{j_k rk}| \right)^2. \quad (9)$$

Elitist lasso was introduced by [19] in the context of regression analysis. The behavior of this penalty can be understood by observing that it behaves as the lasso within blocks and as the ridge between blocks, resulting in shrinkage and a selection of the variables with the highest coefficients within each block (lasso) and a shrinkage but with no selection between blocks (ridge).

To select entire (pre-defined) groups of variables, the *group lasso* [20] was introduced. It uses the Euclidean norm (also known as a mixed $l_{2,1}$ norm; see [19]) of the group coefficients as a penalty,

$$\lambda_G \sum_k \sqrt{J_k} \|\mathbf{w}_{rk}\|_2 = \lambda_G \sum_k \sqrt{J_k \sum_{j_k} (w_{j_k rk}^2)}. \quad (10)$$

This penalty behaves as the lasso at the block level and as the ridge within blocks: within blocks shrinkage

and grouping of correlated variables occurs however with no selection (behavior of the ridge penalty); between blocks selection of those blocks with the highest sum of squared coefficients occurs while other blocks are dropped (behavior of the lasso). The group lasso applied to groups consisting of one variable only is the same as the lasso. (Note that taking the square root of a squared value is the same as taking the absolute value.) To obtain also sparsity within the groups that are not dropped by the group lasso, [21] proposed the *sparse group lasso* that blends the lasso with the group lasso and implies shrinkage and selection both within and between groups. The behavior of each of the four penalties and associated norms is summarized in Table 1.

We propose the following **generic functions that combine all penalties**: First, for the approach based on sparse component weights,

$$\begin{aligned}
L(\mathbf{W}_k, \mathbf{P}_k) &= \\
&\sum_k \left(\|\mathbf{X}_k - \mathbf{X}_k \mathbf{W}_k \mathbf{P}_k^T\|^2 + \lambda_L \|\mathbf{W}_k\|_1 \right) \\
&+ \sum_k \left(\lambda_R \|\mathbf{W}_k\|_2^2 + \lambda_G \sqrt{J_k} \|\mathbf{W}_k\|_2 \right) \\
&+ \sum_k \left(\lambda_E \|\mathbf{W}_k\|_{1,2} \right) \\
&= \|\mathbf{X}_c - \mathbf{X}_c \mathbf{W}_c \mathbf{P}_c^T\|^2 + \lambda_L \|\mathbf{W}_c\|_1 \\
&+ \lambda_R \|\mathbf{W}_c\|_2^2 + \sum_k \left(\lambda_G \sqrt{J_k} \|\mathbf{W}_k\|_2 \right) \\
&+ \sum_k \left(\lambda_E \|\mathbf{W}_k\|_{1,2} \right), \tag{11}
\end{aligned}$$

which has to be minimized with respect to \mathbf{W}_k and \mathbf{P}_c under the constraint that $\mathbf{P}_c^T \mathbf{P}_c = \mathbf{I}$. Second, for the approach based on sparse component loadings,

$$\begin{aligned}
L(\mathbf{T}, \mathbf{P}_k) &= \|\mathbf{X}_c - \mathbf{T} \mathbf{P}_c^T\|^2 + \lambda_L \|\mathbf{P}_c\|_1 + \lambda_R \|\mathbf{P}_c\|_2^2 \\
&+ \sum_k \left(\lambda_G \sqrt{J_k} \|\mathbf{P}_k\|_2 + \lambda_E \|\mathbf{P}_k\|_{1,2} \right), \tag{12}
\end{aligned}$$

which has to be minimized with respect to \mathbf{T} and \mathbf{P}_k under the constraint that $\mathbf{T}^T \mathbf{T} = \mathbf{I}$. Note that estimation of the loadings is not a regression problem. Therefore, unlike the model based on sparse weights, unique solutions are obtained when $J_k > I$. This is the case even when $\lambda_R = 0$.

The generic loss functions (11) and (12) allow for a flexible use of all these approaches to sparseness. All combinations of the four penalties are made possible. However, often some prior idea about the structure

(selection within blocks, between blocks, both within and between blocks) exists such that it is not necessary to consider all possible combinations. Furthermore, some combinations are not advisable. For example the combination of the group lasso and elitist lasso does not seem useful because the behavior of the one interferes with the behavior of the other. By setting the appropriate tuning parameters in the objective functions to zero, particular known sparse approaches can be obtained. For example, with $\lambda_G = \lambda_E = 0$ the extension of sparse PCA to simultaneous component analysis is obtained and with $\lambda_R = \lambda_E = 0$ a sparse simultaneous component version of the sparse group lasso in linear regression is obtained. With all four tuning parameters set equal to zero, the ordinary simultaneous component analysis model results. $K = 1$ leads to principal component analysis and setting $\lambda_G = \lambda_E = 0$ yields sparse PCA as proposed by [14]. In Table 1 a summary is given of these different existing sparse approaches in terms of which penalties are active.

Algorithm

Given fixed values for the different tuning parameters ($\lambda_l, \lambda_R, \lambda_G$, and λ_E) and a fixed number of components R , we make use of an alternating scheme to minimize (11) or (12) with respect to \mathbf{W}_c (or \mathbf{T}) and \mathbf{P}_c : \mathbf{W}_c (or \mathbf{T}) and \mathbf{P}_c are alternately updated, conditional on fixed values for the other parameters. For example, focusing on (11):

- Step 1: Initialize \mathbf{W}_c
- Step 2: Conditional on the current estimate of \mathbf{W}_c , obtain the optimal least-squares estimate of \mathbf{P}_c under the orthogonality constraint as follows (see [22]): $\mathbf{P}_c = \mathbf{U}\mathbf{V}^T$ with $\mathbf{U}\mathbf{S}\mathbf{V}^T$ the singular value decomposition of $\mathbf{W}_c^T \mathbf{X}_c^T \mathbf{X}_c$
- Step 3: Check the stop criteria: 1) Is the difference in loss with the previous iteration smaller than $1e - 12$ or, 2) is a maximum of 5000 iterations reached? If yes, terminate, and else continue.
- Step 4: Conditional on the current estimate of \mathbf{P}_c , obtain the update of \mathbf{W}_c using a majorization minimization procedure (see [23–25] for a general introduction); see the Methods Section for a derivation of the estimate. Return to Step 2.

This particular scheme guarantees that the loss is a non-increasing function of the iterations. Due to the convexity (not strict) and the fact that the loss function is bounded from below by zero, the procedure will converge to a fixed point for suitable starting values. The majorization minimization (MM) procedure has

a linear rate of convergence; this slow convergence rate may, however, be compensated for by the efficiency of the calculations (see for example [26]). To account for the problem that the fixed point may represent a local minimum instead of the global optimum, a multistart procedure can be used. See the Methods Section for details on the algorithm used to minimize (12). MATLAB code implementing the algorithms can be found in the supplementary material.

Testing and implementation

In this section we apply the proposed approach both to empirical and simulated data. The application to empirical data (metabolomics) is mainly for illustrative purposes. The simulated data are used to check how the different penalties (and their interactions) behave under various conditions, and to compare the sparse component weights and sparse component loadings modeling approaches.

Metabolomics data

As an illustrative case, we use empirical data on the metabolome composition of 28 samples of *E. coli*. The different samples refer to different environmental conditions and different elapsed fermentation times. Mass spectrometry (MS) in combination with on the one hand gas chromatography (GC) and on the other hand liquid chromatography (LC) as a separation method was used, resulting in two coupled data blocks: a GC-MS block with the peak areas of 144 metabolites in the 28 conditions and a LC-MS block with the peak areas of 44 metabolites in these same conditions. Simultaneous component analysis was previously successfully applied describing the data well by five components (see [5, 6]). However, a better understanding of the processes underlying the data may be obtained by a sparse simultaneous component analysis (SCA) approach as this characterizes the components by a few instead of all metabolites and thus facilitates interpretation.

Our proposed method allows to model the data in several ways, depending on the one hand on the choice of penalizing either the weights or the loadings and on the other hand on the particular values of the different tuning parameters. Therefore, we will analyze the data under different options, namely either under model (11) or under model (12) and, for both models, with several combinations of values for the different tuning parameters. Here we explain how we chose a suitable range of values for the tuning parameters using the notation for the model with penalized weights. The different values of λ_L , λ_G , λ_E , and λ_R were chosen in a way that reflects the balance between lack-of-fit and strength of the penalty by setting them as a fraction of $\|\mathbf{X}_c\|^2$ (maximal lack-of-fit) and $|\mathbf{W}_c|_{p,q}$ with \mathbf{W}_c obtained from the ordinary

SCA solution (maximal value of the penalty). Let $\lambda_{p,q}$ denote the tuning parameter of the penalty corresponding to the (mixed) $l_{p,q}$ norm, then this yields $\lambda_{p,q} = f\|\mathbf{X}_c\|^2/|\mathbf{W}_c|_{p,q}$ with f taking values $0, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 0.2, 0.5$, and 1 . We only consider those combinations of non-zero values for the tuning parameters that were considered in the regression literature, namely the lasso, elastic net, group lasso, sparse group lasso, and elitist lasso (see Table 1). Note that the case with all tuning parameters equal to zero corresponds to regular simultaneous component analysis.

First we discuss the results for the approach based on penalized weights, then the approach based on penalized loadings, followed by a brief comparison of the two approaches. We end the empirical application section with a discussion on the choice and interpretation of a particular sparse simultaneous component analysis.

Penalized weights

Table 2 summarizes the results for the approach with a penalty on the component weights and with only one of the tuning parameters different from zero (the ridge penalty on its own is not considered as it does not induce sparsity). Five components are assumed ($R = 5$). For the three resulting types of sparse simultaneous component analyses we report on the one hand the fit of the model to the data and on the other hand the percentage of component weights that are zero. The fit is defined as

$1 - \|\mathbf{X}_c - \mathbf{X}_c\mathbf{W}_c\mathbf{P}_c^T\|^2/\|\mathbf{X}_c\|^2$. As could be expected, it holds that increasing the tuning parameter results in a decrease of fit and an increase of the proportion of zero component weights. Comparing the lasso and Elitist lasso, we see that the lasso has a better fit for a similar proportion of zeros which may be attributed to the fact the lasso is less constrained because it does not have to reflect the block structure in the variable selection. Both for the lasso and Elitist lasso the proportion of zeros is very high, even for small values of the tuning parameter. This could be expected as the number of variables is larger than the number of samples and warrants the inclusion of a ridge penalty (see also [14] for the case of the lasso), else non-unique solutions are obtained. Also, at most I non-zero weights will be obtained for each component and this may be too sparse, e.g. in the case of micro-array gene expression data obtained for a limited number of (tissue) samples. Such solutions with only I non-zero values fit as well as the regular simultaneous component model. To understand this, consider a model with one component: $\mathbf{t} = \mathbf{X}\mathbf{w}$ which represents an underdetermined system in case $I < \sum_k J_k$ that can be solved exactly by taking only I non-zero weights. The group lasso operates at the level of the block and therefore does not show this effect. The effect of adding a ridge penalty to the lasso and elitist lasso is visualized in Figure 1: The higher the

value of the parameter that tunes the ridge penalty, the lower the fit and the lower the proportion of zeros. In Figure 2 the results for the sparse group lasso (i.e., combination of lasso and ridge penalty) are summarized. The lines express the fit and the proportion of zero weights in function of the lasso tuning parameter with different lines referring to different values of the group lasso. As illustrated by the figure, there is a qualitative interaction between the two types of penalties in the sense that lower values of the lasso parameter have a strong effect on the number of zeros when the group lasso parameter takes lower values while, conversely, higher values of the lasso parameter have a strong effect when the group lasso parameter takes higher values: As the group lasso shrinks the component weights, the penalty for the lasso becomes lower and hence low values of the lasso tuning parameter are ineffective. Addition of a ridge penalty to the lasso and group lasso parameters may be considered when grouping is important (e.g., as is usual with gene expression data to find modules of co-expressed genes; [27]).

Penalized loadings

A summary of the results obtained for the approach with sparse loadings is given in Table 3. The general result that increasing the tuning parameters yields a decrease in fit and an increase in sparsity also holds here. A comparison between Tables 2 and 3 shows that for an equal proportion of zeros, the fit of models with sparse loadings is (much) lower. This can be understood from the fact that the loadings contribute more directly to the reconstruction of the data than the component weights (compare equations (1) and (2)): for example, in a model with one component, a zero loading results in a zero vector for the reconstructed variable. This also explains why different from the approach based on penalizing the weights with an L_1 penalty, the number of non-zeros is not bounded by I . Table 3 also shows that to obtain zero loadings with the group lasso, high values of the tuning parameter are needed.

Reflections on penalizing the weights versus the loadings

As illustrated, the results obtained under the model with penalized loadings are different from the results obtained under the model with penalized weights. In our view, the most important differences are at the level of data reconstruction and at the level of interpretation. With respect to data reconstruction, the model based on weights yields a better fit while the model with sparse loadings may yield many zero vectors for the reconstructed data. Also, in this respect, the components based on sparse weights have a higher correlation with the components of the ordinary SCA solution than the components resulting from a model with sparse loadings. With respect to interpretation of the underlying components, for the model based on sparse weights this is done in a regression-like way, while for the model based on sparse loadings

it is based on considering loadings as correlations of the variables with the component. In ordinary SCA, the loadings are the correlations and in the sparse model we observed a close connection in that zero loadings represent close to zero correlations and higher loadings represent higher correlations. The weights do not have such a relation with the correlation between the variable and the component.

Selection and interpretation of the sparse SCA solution

As has been illustrated in the previous Results Section, the data can be analyzed in many ways depending on choices made with respect to the generic model ((11) or (12)) and with respect to the values of the different tuning parameters. Selection of the appropriate model is of key importance and substantive issues may form a good point of departure. First, concerning the choice of the generic model, the model with penalized weights seems more appropriate for the data at hand because all metabolites can be considered to be involved in the biological processes underlying the data. For applications of component models with sparse loadings to microarray gene expression data, see [28] and [13]. Second, to choose appropriate values for the tuning parameters we consider the properties of the associated penalties. Having components for which the interpretation is tied exclusively to one type of analytical platform (corresponding to the block structure) is convenient. Also, because for each platform many metabolites result, sparseness within each platform/block is needed. This means that we are interested in selection both across and within groups. Recently, there has been a growing interest for methods that perform such a selection [29,30], with particular interest for the group lasso that has been extended and applied in several ways [31–33]. Therefore, we will restrict ourselves to a group lasso type of simultaneous component model, however, including a ridge penalty to account for the fact that grouping is useful (because, within each analytical method, the metabolites belong to several classes of strongly related compounds): $\lambda_L > 0$, $\lambda_R > 0$, $\lambda_G > 0$, and $\lambda_L = 0$. Then, we eliminate solutions that 1) yield components with all weights equal to zero, 2) yield components having non-zero weights for both data blocks, and 3) solutions that do not fit well ($\text{fit} < .40$) or that are not sparse (less than 50 percent of zero weights in a block). The remaining solutions are summarized in Table 4 in terms of the fit and the number of zeros per component. The solutions in bold, with high values for the lasso tuning parameter ($f_L = 0.5$ or 1) and low values for both the ridge and group lasso parameters ($f_R = 0.0001$ and $f_G = 0.1$; or $f_R = 0.001$ and $f_G = 0.001$), show the best tradeoff between fit and sparsity. We select these solutions for an interpretation.

The metabolites with non-zero component weights are displayed for both selected solutions in Table 5; Table 6 contains the component scores corresponding to the weights of the solution with $f_L = 0.5$. Observe

that the solution with $f_L = 1$ is a further selection of the metabolites in the solution with $f_L = 0.5$. The first component shows an effect of phenyllactate, 3,5-dihydroxypentanoate, and two aromatic amino acids (phenylalanine and tyrosine), together with two branched-chain amino acids (isoleucine and valine); the corresponding component scores (see C1 in Table 6) show a clear increasing linear effect of fermentation time. The second component is made up by metabolites like fumarate, malate, aspartate and are associated to succinate catabolism (see C2 in Table 6) making biological sense as these metabolites are close to succinate in central metabolism. For C3, we find non-zero weights for a large number of (unidentified) disaccharides and pyruvate and lactate and high scores in the oxygen related conditions. The identification of pyruvate and lactate could be indicative of a changing, i.e. reduced, dissolved oxygen concentration in the course of the fermentation as pyruvate can be converted into lactate during anaerobic growth. The fourth component is made up by nucleotides important for the energy metabolism in a cell (i.e. ADP, GDP, UDP) and is associated to the growth condition with an elevated pH at the early (16hrs) phase. Finally, C5 seems specific for the wild type strain, although the relation to the metabolites guanine and thymine (both nucleobases) and the other metabolites is not very clear.

Simulated data

To validate the proposed sparse simultaneous component method, we make use of simulated data. The general **setup** is that data are generated under some specific conditions and with known structure; after addition of noise, the performance of the method in terms of recovering the underlying structure is assessed. Here, we are particularly interested in two aspects: A first one is whether the penalties reflect the structure in the selection of the variables (i.e., between data blocks; within data blocks; or both between and within data blocks); a second one is the behavior of the method in function of the model (i.e., sparse weights or sparse loadings). We also manipulated the amount of error in the data (5 and 30 percent) and the degree of sparseness (50 and 90 percent of zero weights/loadings). All factors were fully crossed and for each of the resulting $2 \times 3 \times 2 \times 2 = 24$ conditions, 5 data sets were generated, resulting in a total of 120 data sets. To obtain a realistic simulation, we **generated the data** using the metabolomics data described in the previous section. 28 samples were sampled with replacement from the original data; then a singular value decomposition was performed to obtain three components: the three loading and weight vectors were obtained as the three right singular vectors corresponding to the three largest singular values and multiplied by these, the three component score vectors were set equal to the corresponding left singular vectors. Sparseness was imposed by setting either weights or loadings equal to zero as follows: In case of

sparseness between blocks, all weights/loadings of the first component that correspond to the first data block (the first 144 weights/loadings) were set equal to zero and for the second and third component the weights/loadings corresponding to the second data block (the last 44 weights/loadings) were set equal to zero; in case of sparseness within blocks, 50 or 90 percent of variable indices were randomly sampled and their corresponding weights/loadings were set equal to zero; in case of sparseness within and between data blocks, the two previous strategies were combined. The resulting component loadings and weights were used to generate the true data part using the model part of expressions (1) and (2) (i.e., without the addition of the residual matrices). Noise was then added to this true part of the data with the noise being generated from a normal distribution with mean zero and variance such that these residual matrices account for 5 or 30 percent of the total variation [34]. Each of the data sets was analyzed under both models (sparse weights or sparse loadings) and with varying values for the tuning parameters (f equal to $0, 10^{-3}, 0.1, 0.5$, and 10). The Elitist lasso penalty was only combined with the ridge penalty because it interferes with the lasso and group lasso (see earlier).

In the **discussion of the results of the simulation study**, we first focus on the conditions where the data are generated and analyzed under the same model (either sparse weights or sparse loadings), the error amounting to 30 percent of the total variation in the data, and the ridge penalty set equal to the smallest non-zero value. Figures 3 and 4 display boxplots of the proportion of variables correctly classified (selected versus dropped) in function of the value of the tuning parameter. Figure 3 refers to the case with 50 percent zero weights/loadings, Figure 4 to the case with 90 percent zero weights/loadings. In each Figure, the different panels refer to the different combinations of structure in the variable selection (from top to bottom: within blocks, between blocks, within and between blocks) and of sparseness approach (from left to right: lasso, group lasso, Elitist lasso, and sparse group lasso). The panels referring to the sparse group lasso are with varying values for the lasso tuning parameter and with the group lasso tuning parameter fixed at $f_G = 10$. In general, the results confirm the expected relation between the structure of the variable selection and the different approaches to sparseness: The best recovery for selection within blocks is by Elitist lasso with a value of 0.5 for the tuning parameter f_E , for selection between blocks is by the group lasso with $f_G = 10$, and for selection between and within blocks the sparse group lasso ($f_L = 0.1$ for the lasso). Deviations from the expected behavior occur for the sparse group lasso when selection is both within and between blocks in case of many zeros (see Figure 4): the lasso and Elitist lasso then outperform the group lasso. This can be attributed to the fact that the group lasso is less aggressive than the lasso and Elitist lasso [11]. On the other hand, the lasso and Elitist lasso perform less well when selection is within

blocks and the true structure is not so sparse (50 percent of zeros, see the top row of Figure 3) because of their aggressive behavior. Note that a penalty that selects between groups in a more aggressive way was proposed by [11]. The same pattern of results is obtained when the error amounts to 5 percent (though shifted upwards as in these conditions the status of the variables is better recovered) or when the tuning parameter of the ridge penalty takes higher values. In case the ridge equals zero, the box plots show worse results for the lasso and Elitist tuning parameters equal to zero (because there are more variables than objects thus at most 28 non-zero values are obtained for the approach based on sparse weights).

A second point of interest, is the influence of the model used to generate and analyze the data. Figure 5 displays four panels of boxplots for the proportion of correctly classified variables. Within panels, the boxplots are displayed in function of the block structure present in the variable selection. The upper panels refer to data generated under a model with sparse loadings, the lower panels to data generated under a model with sparse weights. The panels at the left were obtained when analyzing the data with a model based on sparse weights and at the right with sparse loadings. In general, analyzing the data with the sparse weights model yields less misclassifications than using the sparse loadings model. However, generating the (underlying) data under a model with sparse weights, in general, results in more misclassifications than generating under a sparse loadings model. These results can be explained by the more direct relation between the loadings and generated or modeled data: Generating the data with sparse loadings imposes a clearer structure than generating them with sparse weights; analyzing/modeling the data with sparse loadings imposes a stronger structure on the modeled data than modeling them with sparse weights. This is because 1) unlike a zero loading, a zero weight for a variable does not necessarily imply a modeled score of zero, because a zero weight for one variable can be compensated by non-zero weights for other variables, and 2) unlike shrinking the weights, shrinking the loadings results more directly in shrunken modeled scores. The latter can be explained by the dependence of the scale of the data, as modeled by PCA model (1), on the scale of the loadings (the model has orthonormal component scores).

Discussion

We proposed an extension of sparse PCA to the context of several data blocks, relying on a generic modeling framework that allows either for sparse component weights or for sparse component loadings and that incorporates several approaches that were taken to sparsity in the regression literature (including the lasso, elastic net, group lasso, Elitist lasso, and sparse group lasso). A very flexible algorithm was developed that allows to analyze the data under a variety of approaches that take the structure of the data

in different ways into account. It also allows for combinations of penalties that were not yet considered in the regression literature.

The flexibility of the approach is important as often a particular kind of structure is needed from data integration methods. The group lasso is a popular tool to find structures that only involve one data block. This is for example relevant in comparative genomics when the focus is on divergence [35] or on tissue-specificity [36]. Elitist lasso, on the other hand, finds sparse structures that involve each of the data blocks. Not only is this of relevance in the aforementioned case of comparative genomics to find conserved processes, but also in a top-down systems biology approach. For example, to integrate microarray gene expression data and interaction data with the aim of finding transcription factors and their target genes [37].

Although the model and algorithm were proposed in the context of simultaneous component analysis, it can be easily translated to the context of principal component analysis and also of regression analysis. In fact the algorithm can be used as it is for PCA and the adaptation to regression analysis is a minor one. In the context of simultaneous component analysis, adaptations of the model (and algorithm) to a context that allows for different values of the tuning parameter for each component and/or each block would be valuable. However, such an extension is not trivial. Moreover, the problem of selecting an optimal model becomes more difficult in that more parameters need to be tuned and this would make the choice of selecting appropriate values for the tuning parameter even more difficult than it already is. A major theoretic challenge for many sparse methods is to find a good method to select the value of the tuning parameters.

Conclusions

We offered a flexible and sparse framework for data integration based on simultaneous component methods. The method is flexible both with respect to the component model and with respect to the sparse structure imposed: Sparsity can be imposed either on the component weights or on the loadings, and can be imposed either within data blocks, across data blocks, or both within and across data blocks. As such, it allows to find structures exclusively tied to one data platform as well as structures that involve all data platforms. A penalty based approach is used that includes the lasso, the ridge penalty, the group lasso, and Elitist lasso. The method includes principal component analysis, sparse principal component analysis, and ordinary simultaneous component analysis as special cases. Real and simulated data were used to validate the method. We believe the method offers a very flexible and versatile tool for many data integration problems.

Methods

Here we derive the estimates used in the alternating least squares and iterative majorization algorithm. First, it is shown how the conditional estimates for the objective function relying on sparse component weights can be obtained and then for the objective function relying on sparse loadings.

Sparse component weights

The generic objective function that we rely on to find a simultaneous component solution with sparse component weights is to minimize

$$\begin{aligned}
L(\mathbf{W}_k, \mathbf{P}_k) &= \\
&\sum_k \left(\|\mathbf{X}_k - \mathbf{X}_k \mathbf{W}_k \mathbf{P}_k^T\|^2 + \lambda_L \|\mathbf{W}_k\|_1 + \lambda_R \|\mathbf{W}_k\|_2^2 \right) \\
&+ \sum_k \left(\lambda_G \sqrt{J_k} \|\mathbf{W}_k\|_2 + \lambda_E \|\mathbf{W}_k\|_{1,2} \right) \\
&= \|\mathbf{X}_c - \mathbf{X}_c \mathbf{W}_c \mathbf{P}_c^T\|^2 + \lambda_L \|\mathbf{W}_c\|_1 + \lambda_R \|\mathbf{W}_c\|_2^2 \\
&+ \sum_k \left(\lambda_G \sqrt{J_k} \|\mathbf{W}_k\|_2 + \lambda_E \|\mathbf{W}_k\|_{1,2} \right) \\
&= \text{tr} [(\mathbf{X}_c - \mathbf{X}_c \mathbf{W}_c \mathbf{P}_c^T)^T (\mathbf{X}_c - \mathbf{X}_c \mathbf{W}_c \mathbf{P}_c^T)] \\
&+ \lambda_L \|\mathbf{W}_c\|_1 + \lambda_R \|\mathbf{W}_c\|_2^2 \\
&+ \sum_k \left(\lambda_G \sqrt{J_k} \|\mathbf{W}_k\|_2 + \lambda_E \|\mathbf{W}_k\|_{1,2} \right) \\
&= \text{tr} [\mathbf{X}_c^T \mathbf{X}_c - 2\mathbf{X}_c^T \mathbf{X}_c \mathbf{W}_c \mathbf{P}_c^T + \mathbf{P}_c \mathbf{W}_c^T \mathbf{X}_c^T \mathbf{X}_c \mathbf{W}_c \mathbf{P}_c^T] \\
&+ \lambda_L \|\mathbf{W}_c\|_1 + \lambda_R \|\mathbf{W}_c\|_2^2 \\
&+ \sum_k \left(\lambda_G \sqrt{J_k} \|\mathbf{W}_k\|_2 + \lambda_E \|\mathbf{W}_k\|_{1,2} \right) \\
&= \text{tr} \mathbf{X}_c^T \mathbf{X}_c - 2\text{tr} \mathbf{X}_c^T \mathbf{X}_c \mathbf{W}_c \mathbf{P}_c^T + \text{tr} \mathbf{P}_c \mathbf{W}_c^T \mathbf{X}_c^T \mathbf{X}_c \mathbf{W}_c \mathbf{P}_c^T \\
&+ \lambda_L \|\mathbf{W}_c\|_1 + \lambda_R \|\mathbf{W}_c\|_2^2 \\
&+ \sum_k \left(\lambda_G \sqrt{J_k} \|\mathbf{W}_k\|_2 + \lambda_E \|\mathbf{W}_k\|_{1,2} \right),
\end{aligned} \tag{13}$$

with respect to \mathbf{W}_c and \mathbf{P}_c and under the constraint $\mathbf{P}_c^T \mathbf{P}_c = \mathbf{I}$. λ_L , λ_R , λ_G , and λ_E are considered to be known non negative constants. We use an alternating approach in which each set of parameters is updated in turn while keeping the remaining sets fixed on their last update. Let \mathbf{P}_c be the first set to be updated,

conditionally upon fixed values for \mathbf{W}_c . Rewriting (13) gives

$$\begin{aligned} L(\mathbf{W}_c, \mathbf{P}_c) &= k_1 - 2\text{tr}\mathbf{W}_c^T \mathbf{X}_c^T \mathbf{X}_c \mathbf{P}_c \\ &\quad + \text{tr}\mathbf{P}_c^T \mathbf{P}_c \mathbf{W}_c^T \mathbf{X}_c^T \mathbf{X}_c \mathbf{W}_c, \end{aligned} \tag{14}$$

with $k_1 = \mathbf{X}_c^T \mathbf{X}_c + \lambda_L \|\mathbf{W}_c\|_1 + \lambda_R \|\mathbf{W}_c\|_2^2 + \sum_k (\lambda_G \sqrt{J_k} \|\mathbf{W}_k\|_2 + \lambda_E \|\mathbf{W}_k\|_{1,2})$ the terms that are constant with respect to \mathbf{P}_c . Using $\mathbf{P}_c^T \mathbf{P}_c = \mathbf{I}$ yields

$$L(\mathbf{W}_c, \mathbf{P}_c) = k_2 - 2\text{tr}\mathbf{W}_c^T \mathbf{X}_c^T \mathbf{X}_c \mathbf{P}_c, \tag{15}$$

with $k_2 = k_1 + \text{tr}\mathbf{W}_c^T \mathbf{X}_c^T \mathbf{X}_c \mathbf{W}_c$. The minimization of (15) under the constraint of orthogonal loadings is equivalent to the maximization of $\text{tr}\mathbf{W}_c^T \mathbf{X}_c^T \mathbf{X}_c \mathbf{P}_c$ under the same constraint. This is a problem with known closed form solution [22]

$$\mathbf{P}_c = \mathbf{V}\mathbf{U}^T \tag{16}$$

with \mathbf{U} and \mathbf{V} the left and right singular vectors of $\mathbf{W}_c^T \mathbf{X}_c^T \mathbf{X}_c$.

The minimization of (13) with respect to \mathbf{W}_c is not a standard problem due to the Lasso, Group Lasso, and Elitist Lasso penalties on \mathbf{W}_c . We will make use of a numerical procedure, known as Majorization Minimization (MM) or also Iterative Majorization, which has been proven to be a superior algorithmic strategy in regularization problems [25, 38]. Briefly stated, MM replaces functions that are complicated to minimize by surrogate functions that are easy to minimize, that lie on/above the original function, and that touch the original function in the so-called supporting point. These properties lead to the sandwich inequality [23].

A useful property of majorizing functions is that a sum of terms can be majorized by majorizing the terms [39]. Therefore, a majorizing function can be obtained for (13) by finding a linear or quadratic majorizing function for each of the penalty terms except the ridge. First we consider the Lasso penalty: $\lambda_L \|\mathbf{W}_c\|_1 = \sum_{j_k, r, k} \lambda_L |w_{j_k r k}|$. Applying the additivity property again, we need to find a majorizing function for $|w_{j_k r k}|$. Such a function is [40]

$$|w_{j_k r k}| \leq \frac{1}{2} \frac{w_{j_k r k}^2}{|w_{j_k r k}^o|} + \frac{1}{2} |w_{j_k r k}^o|, \tag{17}$$

with $w_{j_k r k}^o$ the current estimate of $w_{j_k r k}$ that was obtained in the previous iteration. This yields

$$\lambda \sum_{j_k, r, k} |w_{j_k r k}| \leq \lambda \sum_{j_k, r, k} \left(\frac{1}{2} \frac{w_{j_k r k}^2}{|w_{j_k r k}^o|} + \frac{1}{2} |w_{j_k r k}^o| \right)$$

$$= \frac{\lambda}{2} \text{Vec}(\mathbf{W}_c)^T \mathbf{D}_1 \text{Vec}(\mathbf{W}_c) + k_3, \quad (18)$$

with the Vec notation indicating that the matrix is vectorized, with $k_3 = \sum_{j_k, r, k} \frac{\lambda}{2} |w_{j_k r k}^o|$, and with \mathbf{D}_1 a diagonal matrix containing the $|w_{j_k r k}^o|^{-1}$ on its diagonal. Second, we consider the k Group Lasso penalty terms $\lambda_G \|\mathbf{W}_k\|_2 = \lambda_G \left(\sum_{j_k, r} w_{j_k r}^2 \right)^{1/2}$. A majorizing function for the root is (see [39])

$$\begin{aligned} \lambda_G \sum_k \left(\sum_{j_k, r} w_{j_k r}^2 \right)^{1/2} &\leq \\ &\frac{\lambda_G}{2} \sum_k \left(\sum_{j_k, r} (w_{j_k r}^o)^2 \right)^{1/2} \\ &\quad + \frac{\lambda_G}{2} \sum_k \left(\sum_{j_k, r} (w_{j_k r}^o)^2 \right)^{-1/2} \left(\sum_{j_k, r} w_{j_k r}^2 \right) \\ &= k_4 + \frac{\lambda_G}{2} \text{Vec}(\mathbf{W}_c)^T \mathbf{D}_2 \text{Vec}(\mathbf{W}_c), \end{aligned} \quad (19)$$

with $k_4 = \frac{\lambda_G}{2} \sum_k \left(\sum_{j_k, r} (w_{j_k r}^o)^2 \right)^{1/2}$, and with \mathbf{D}_2 a diagonal matrix containing the $\left(\sum_{j_k, r} (w_{j_k r}^o)^2 \right)^{-1/2}$ on its diagonal. Third, we majorize the Elitist Lasso penalty term $\lambda_E \|\mathbf{W}_k\|_{1,2} = \lambda_E \left(\sum_{j_k, r} |w_{j_k r}| \right)^2$ with the following quadratic function (see [39]),

$$\begin{aligned} \lambda_E \sum_k \left(\sum_{j_k, r} w_{j_k r}^2 \right)^{1/2} &\leq \\ &\lambda_E \sum_k \left(\left(\sum_{j_k, r} |w_{j_k r}^o| \right) \sum_{j_k, r} \frac{w_{j_k r}^2}{|w_{j_k r}^o|} \right) \\ &= k_5 + \lambda_E \text{Vec}(\mathbf{W}_c)^T \mathbf{D}_3 \text{Vec}(\mathbf{W}_c), \end{aligned} \quad (20)$$

with \mathbf{D}_3 a diagonal matrix containing the $\left(\sum_{j_k, r} |w_{j_k r}^o| \right) (|w_{j_k r}^o|)^{-1}$ on its diagonal.

Combining (13) with the results (18), (19), and (20), we obtain the following majorizing function for (13):

$$\begin{aligned} L(\mathbf{W}_c, \mathbf{P}_c) &= \\ &\|\mathbf{X}_c - \mathbf{X}_c \mathbf{W}_c \mathbf{P}_c^T\|^2 + \lambda_L \|\mathbf{W}_c\|_1 + \lambda_R \|\mathbf{W}_c\|_2^2 \\ &\quad + \sum_k (\lambda_G \|\mathbf{W}_k\|_2 + \lambda_E \|\mathbf{W}_k\|_{1,2}) \\ &= \|\text{Vec}(\mathbf{X}_c) - \text{Vec}(\mathbf{X}_c \mathbf{W}_c \mathbf{P}_c^T)\|^2 \\ &\quad + \lambda_L \|\mathbf{W}_c\|_1 + \lambda_R \|\mathbf{W}_c\|_2^2 \end{aligned}$$

$$\begin{aligned}
& + \sum_k (\lambda_G \|\mathbf{W}_k\|_2 + \lambda_E \|\mathbf{W}_k\|_{1,2}) \\
\leq & \|\text{Vec}(\mathbf{X}_c) - (\mathbf{P}_c \otimes \mathbf{X}_c) \text{Vec}(\mathbf{W}_c)\|^2 \\
& + \text{Vec}(\mathbf{W}_c)^T (\mathbf{D}_{sup}) \text{Vec}(\mathbf{W}_c) + k \\
= & Q(\mathbf{W}_c, \mathbf{P}_c), \tag{21}
\end{aligned}$$

with $\mathbf{D}_{sup} = \frac{\lambda_L}{2} \mathbf{D}_1 + \frac{\lambda_G}{2} \mathbf{D}_2 + \lambda_E \mathbf{D}_3 + \lambda_R \mathbf{I}$, \mathbf{I} an identity matrix, and $k = k_3 + k_4 + k_5$. This function can be minimized with respect to \mathbf{W}_c by finding the value for which the partial derivative of (21) is zero. The partial derivative equals

$$\begin{aligned}
\frac{\partial Q}{\partial \mathbf{W}_c} = & \\
& -2(\mathbf{P}_c \otimes \mathbf{X}_c)^T [\text{Vec}(\mathbf{X}_c) - (\mathbf{P}_c \otimes \mathbf{X}_c) \text{Vec}(\mathbf{W}_c)] \\
& + 2(\mathbf{D}_{sup}) \text{Vec}(\mathbf{W}_c), \tag{22}
\end{aligned}$$

and is equal to zero for

$$\begin{aligned}
\text{Vec}(\mathbf{W}_c) = & \\
& [\mathbf{D}_{sup} + (\mathbf{P}_c \otimes \mathbf{X}_c)^T (\mathbf{P}_c \otimes \mathbf{X}_c)]^{-1} (\mathbf{P}_c^T \otimes \mathbf{X}_c)^T \text{Vec}(\mathbf{X}_c) \\
= & [\mathbf{D}_{sup} + \mathbf{I} \otimes (\mathbf{X}_c^T \mathbf{X}_c)]^{-1} \text{Vec}(\mathbf{X}_c^T \mathbf{X}_c \mathbf{P}_c), \tag{23}
\end{aligned}$$

where the inverse is taken of a block-diagonal matrix. \mathbf{W}_c is obtained by rearranging $\text{Vec}(\mathbf{W}_c)$. Note that the second derivative is positive so (23) is a minimum of (21). In this equation, the penalty terms occur as diagonal matrices that are summed together in the matrix \mathbf{D}_{sup} and with the variance-covariance matrix of the data; the resulting matrix is inverted and will be dominated by large values on the diagonal (yielding small values after inversion). This shows the behavior of the penalties: increasing the tuning parameters results in such large diagonal values; furthermore, the diagonal matrices themselves are inverse functions of the weights in the previous iteration of the algorithm such that small weights further enhance the shrinkage or selection. Note that the matrix to be inverted in equation (23) is of the form $\mathbf{D} + \mathbf{A}^T \mathbf{A}$ with \mathbf{D} a diagonal matrix; then, the following holds [41],

$$(\mathbf{D} + \mathbf{A}^T \mathbf{A})^{-1} = \mathbf{D}^{-1} - \mathbf{D}^{-1} \mathbf{A}^T (\mathbf{I} + \mathbf{X} \mathbf{D}^{-1} \mathbf{X}^T)^{-1} \mathbf{A} \mathbf{D}^{-1} \tag{24}$$

which may be useful when $J_k > I$.

Sparse loadings

The generic objective function that we rely on to find a simultaneous component solution with sparse component weights is to minimize

$$\begin{aligned}
L(\mathbf{T}, \mathbf{P}_k) &= \\
&\sum_k \left(\|\mathbf{X}_k - \mathbf{T}\mathbf{P}_k^T\|^2 + \lambda_L \|\mathbf{P}_k\|_1 + \lambda_R \|\mathbf{P}_k\|_2^2 \right) \\
&+ \sum_k \left(\lambda_G \sqrt{J_k} \|\mathbf{P}_k\|_2 + \lambda_E \|\mathbf{P}_k\|_{1,2} \right) \\
&= \|\mathbf{X}_c - \mathbf{T}\mathbf{P}_c^T\|^2 + \lambda_L \|\mathbf{P}_c\|_1 + \lambda_R \|\mathbf{P}_c\|_2^2 \\
&+ \sum_k \left(\lambda_G \sqrt{J_k} \|\mathbf{P}_k\|_2 + \lambda_E \|\mathbf{P}_k\|_{1,2} \right) \\
&= \text{tr} [(\mathbf{X}_c - \mathbf{T}\mathbf{P}_c^T)^T (\mathbf{X}_c - \mathbf{T}\mathbf{P}_c^T)] \\
&+ \lambda_L \|\mathbf{P}_c\|_1 + \lambda_R \|\mathbf{P}_c\|_2^2 \\
&+ \sum_k \left(\lambda_G \sqrt{J_k} \|\mathbf{P}_k\|_2 + \lambda_E \|\mathbf{P}_k\|_{1,2} \right) \\
&= \text{tr} [\mathbf{X}_c^T \mathbf{X}_c - 2\mathbf{X}_c^T \mathbf{T}\mathbf{P}_c^T + \mathbf{P}_c \mathbf{T}^T \mathbf{T}\mathbf{P}_c^T] \\
&+ \lambda_L \|\mathbf{P}_c\|_1 + \lambda_R \|\mathbf{P}_c\|_2^2 \\
&+ \sum_k \left(\lambda_G \sqrt{J_k} \|\mathbf{P}_k\|_2 + \lambda_E \|\mathbf{P}_k\|_{1,2} \right) \\
&= \text{tr} \mathbf{X}_c^T \mathbf{X}_c - 2\text{tr} \mathbf{X}_c^T \mathbf{T}\mathbf{P}_c^T + \text{tr} \mathbf{P}_c \mathbf{P}_c^T \\
&+ \lambda_L \|\mathbf{P}_c\|_1 + \lambda_R \|\mathbf{P}_c\|_2^2 \\
&+ \sum_k \left(\lambda_G \sqrt{J_k} \|\mathbf{P}_k\|_2 + \lambda_E \|\mathbf{P}_k\|_{1,2} \right),
\end{aligned} \tag{25}$$

with respect to \mathbf{T} and \mathbf{P}_k under the constraint that $\mathbf{T}^T \mathbf{T} = \mathbf{I}$. λ_L , λ_R , λ_G , and λ_E are considered to be known non negative constants. In case all tuning parameters are equal to zero, a regular simultaneous component analysis results and in that case the algorithm should be based on SVD of the concatenated data. We use an alternating approach in which each set of parameters is updated in turn while keeping the remaining sets fixed on their last update. Let \mathbf{T} be the first set to be updated, conditionally upon fixed values for \mathbf{P}_c . Rewriting (25) gives

$$L(\mathbf{T}, \mathbf{P}_c) = k_6 - 2\text{tr} \mathbf{X}_c^T \mathbf{T}\mathbf{P}_c^T \tag{26}$$

with $k_6 = \text{tr} [\mathbf{X}_c^T \mathbf{X}_c + \mathbf{P}_c \mathbf{P}_c^T] + \lambda_L \|\mathbf{P}_c\|_1 + \lambda_R \|\mathbf{P}_c\|_2^2 + \sum_k (\lambda_G \sqrt{J_k} \|\mathbf{P}_k\|_2 + \lambda_E \|\mathbf{P}_k\|_{1,2})$ the terms that are constant with respect to \mathbf{T} . Minimizing function (26) is equivalent to maximizing $\text{tr} \mathbf{P}_c^T \mathbf{X}_c^T \mathbf{T}$ with known closed form solution [22]

$$\mathbf{T} = \mathbf{V} \mathbf{U}^T \quad (27)$$

with \mathbf{U} and \mathbf{V} the left and right singular vectors of $\mathbf{P}_c^T \mathbf{X}_c^T$.

Combining (25) with the results (18), (19), and (20) adapted to the case of loadings, we obtain the following majorizing function for (25):

$$\begin{aligned} L(\mathbf{T}, \mathbf{P}_c) &= \\ & \|\mathbf{X}_c - \mathbf{T} \mathbf{P}_c^T\|^2 + \lambda_L \|\mathbf{P}_c\|_1 \\ & + \lambda_R \|\mathbf{P}_c\|_2^2 + \sum_k (\lambda_G \|\mathbf{P}_k\|_2 + \lambda_E \|\mathbf{P}_k\|_{1,2}) \\ & = \|\text{Vec}(\mathbf{X}_c) - \text{Vec}(\mathbf{T} \mathbf{P}_c^T)\|^2 + \lambda_L \|\mathbf{P}_c\|_1 \\ & + \lambda_R \|\mathbf{P}_c\|_2^2 + \sum_k (\lambda_G \|\mathbf{P}_k\|_2 + \lambda_E \|\mathbf{P}_k\|_{1,2}) \\ & \leq \|\text{Vec}(\mathbf{X}_c) - (\mathbf{I} \otimes \mathbf{T}) \text{Vec}(\mathbf{P}_c^T)\|^2 \\ & + \text{Vec}(\mathbf{P}_c)^T (\mathbf{D}_{sup}) \text{Vec}(\mathbf{P}_c) + k \\ & = Q(\mathbf{T}, \mathbf{P}_c), \end{aligned} \quad (28)$$

and the first derivative of $Q(\mathbf{T}, \mathbf{P}_c)$ with respect to \mathbf{P}_c is equal to zero for

$$\begin{aligned} \text{Vec}(\mathbf{P}_c^T) &= \\ & [\mathbf{D}_{sup} + (\mathbf{I}^T \otimes \mathbf{T})^T (\mathbf{I}^T \otimes \mathbf{T}_c)]^{-1} (\mathbf{I}^T \otimes \mathbf{T})^T \text{Vec}(\mathbf{X}_c) \\ & = [\mathbf{D}_{sup} + \mathbf{I}]^{-1} \text{Vec}(\mathbf{T}^T \mathbf{X}_c). \end{aligned} \quad (29)$$

List of abbreviations

E. coli: *Escherichia coli* GC: Gas Chromatography; LC: Liquid Chromatography; MM: Majorization Minimization; MS: Mass Spectrometry; PCA: Principal Component Analysis; SCA: Simultaneous Component Analysis; SVD: Singular Value Decomposition

Author's contributions

KVD derived and implemented the algorithms, performed the data analysis, and drafted the manuscript. TFW participated in the data analysis, model selection, and simulation study. RvdB carried out the interpretation of the results and helped to draft the manuscript. AA and IVM conceived of the study. All authors read and approved the final manuscript.

Acknowledgements

This work was supported by the Research Fund of Katholieke Universiteit Leuven (SymBioSys: CoE EF/05/007, GOA/2005/04, PDM: Tom Wilderjans); by IWT-Flanders (IWT/060045/SBO Bioframe); and by the Belgian Federal Science Policy Office (IUAP P6/03 and P6/04). We would like to thank TNO, Quality of Life, Zeist, The Netherlands, for making the data available. The authors also wish to thank the reviewers for their valuable comments and suggestions.

References

1. van der Werf MJ, Overkamp KM, Muilwijk B, Coulier L, Hankemeier T: **Microbial metabolomics: Toward a platform with full metabolome coverage.** *Analytical Biochemistry* 2007, **370**:17 – 25.
2. Le Cao KA, Martin P, Robert-Granie C, Besse P: **Sparse canonical methods for biological data integration: application to a cross-platform study.** *BMC Bioinformatics* 2009, **10**:34, [<http://www.biomedcentral.com/1471-2105/10/34>].
3. Ishii N, Nakahigashi K, Baba T, Robert M, Soga T, Kanai A, Hirasawa T, Naba M, Hirai K, Hoque A, Ho PY, Kakazu Y, Sugawara K, Igarashi S, Harada S, Masuda T, Sugiyama N, Togashi T, Hasegawa M, Takai Y, Yugi K, Arakawa K, Iwata N, Toya Y, Nakayama Y, Nishioka T, Shimizu K, Mori H, Tomita M: **Multiple High-Throughput Analyses Monitor the Response of E. coli to Perturbations.** *Science* 2007, **316**(5824):593–597, [<http://www.sciencemag.org/cgi/content/abstract/316/5824/593>].
4. de Tayrac M, Le S, Aubry M, Mosser J, Husson F: **Simultaneous analysis of distinct Omics data sets with integration of biological knowledge: Multiple Factor Analysis approach.** *BMC Genomics* 2009, **10**:32.
5. van den Berg R, Van Mechelen I, Wilderjans T, Van Deun K, Kiers H, Smilde A: **Integrating functional genomics data using maximum likelihood based simultaneous component analysis.** *BMC Bioinformatics* 2009, **10**:340, [<http://www.biomedcentral.com/1471-2105/10/340>].
6. Van Deun K, Smilde A, van der Werf M, Kiers H, Van Mechelen I: **A structured overview of simultaneous component based data integration.** *BMC Bioinformatics* 2009, **10**:246, [<http://www.biomedcentral.com/1471-2105/10/246>].
7. Wilderjans TF, Ceulemans E, Van Mechelen I, van den Berg RA: **Simultaneous analysis of coupled data matrices subject to different amounts of noise.** *British Journal of Mathematical and Statistical Psychology* 2011, **64**(2):277–290, [<http://dx.doi.org/10.1348/000711010X513263>].
8. Lee D, Lee W, Lee Y, Pawitan Y: **Super-sparse principal component analyses for high-throughput genomic data.** *BMC Bioinformatics* 2010, **11**:296, [<http://www.biomedcentral.com/1471-2105/11/296>].
9. Tibshirani R: **Regression Shrinkage and Selection via the Lasso.** *Journal of the Royal Statistical Society, Series B* 1996, **58**:267–288.
10. Zou H, Hastie T: **Regularization and variable selection via the elastic net.** *Journal of the Royal Statistical Society, Series B* 2005, **67**:301–320.
11. Jenatton R, Obozinski G, Bach F: **Structured sparse principal component analysis.** *Journal of Machine Learning Research* 2010, **9**:366–373.
12. Jolliffe I, Trendafilov N, Uddin M: **A Modified Principal Component Technique Based on the LASSO.** *Journal of Computational & Graphical Statistics* 2003, **12**(3):531–547.
13. Witten DM, Tibshirani R, Hastie T: **A penalized matrix decomposition, with applications to sparse principal components and canonical correlation analysis.** *Biostatistics* 2009, **10**(3):515–534, [<http://biostatistics.oxfordjournals.org/content/10/3/515.abstract>].
14. Zou H, Hastie T, Tibshirani R: **Sparse principal component analysis.** *Journal of Computational and Graphical Statistics* 2006, **15**(2):265–286.
15. Kiers H: **Towards a Standardized Notation and Terminology in Multiway Analysis.** *Journal of Chemometrics* 2000, **14**:105–122.
16. Gabriel KR: **The biplot graphic display of matrices with application to principal component analysis.** *Biometrika* 1971, **58**:453–467.
17. Jolliffe IT: *Principal component analysis*. New York: Springer 2002.
18. Hoerl AE, Kennard RW: **Ridge Regression: Biased Estimation for Nonorthogonal Problems.** *Technometrics* 1970, **12**:pp. 55–67, [<http://www.jstor.org/stable/1267351>].
19. Kowalski M, Torr sani B: **Structured sparsity: from mixed norms to structured shrinkage.** *SPARS09-Signal Processing with Adaptive Sparse Structured Representations* 2009, **53**:814–861.
20. Yuan M, Lin Y: **Model selection and estimation in regression with grouped variables.** *Journal of the Royal Statistical Society: Series B* 2006, **68**:49–67.

21. Friedman J, Hastie T, Tibshirani R: **A note on the group lasso and a sparse group lasso**. Tech. rep., Statistics Department, Stanford University 2010.
22. Ten Berge JMF: *Least squares optimization in multivariate analysis*. Leiden: DSWO 1993.
23. de Leeuw J: **Block relaxation algorithms in statistics**. In *Information Systems and Data Analysis*. Edited by Bock HH, Lenski W, Richter MM, Berlin: Springer-Verlag 1994:308–325.
24. Heiser WJ: **Convergent computation by iterative majorization: theory and applications in multidimensional data analysis**. In *Recent advances in descriptive multivariate analysis*. Edited by Krzanowski WJ, Oxford: Oxford University Press 1995:157–189.
25. Lange K, Hunter DR, Yang I: **Optimization transfer using surrogate objective functions**. *Journal of computational and graphical statistics* 2000, **9**:1–20.
26. Van Deun K, Groenen PJF: **Majorization algorithms for inspecting circles, ellipses, squares, rectangles, and rhombi**. *Operations Research* 2005, **53**:957–967.
27. Barabasi AL, Oltvai ZN: **Network biology: understanding the cell’s functional organization**. *Nature Review Genetics* 2004, **5**:101–113.
28. Hochreiter S, Bodenhofer U, Heusel M, Mayr A, Mitterecker A, Kasim A, Khamiakova T, Van Sanden S, Lin D, Talloen W, Bijmens L, Gohlmann HWH, Shkedy Z, Clevert DA: **FABIA: factor analysis for bicluster acquisition**. *Bioinformatics* 2010, **26**(12):1520–1527.
29. Huang J, Ma S, Xie H, Zhang CH: **A group bridge approach for variable selection**. *Biometrika* 2009, **96**(2):339–355.
30. Zhao P, Rocha G, Yu B: **Grouped and Hierarchical Model Selection through Composite Absolute Penalties**. Tech. rep., Department of Statistics, University of California, Berkeley 2006.
31. Ma S, Song X, Huang J: **Supervised group Lasso with applications to microarray data analysis**. *BMC Bioinformatics* 2007, **8**:60.
32. Meier L, Van De Geer S, Bühlmann P: **The group lasso for logistic regression**. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 2008, **70**.
33. Kim Y, Kim J, Kim Y: **Blockwise sparse regression**. *Statistica Sinica* 2006, **16**:375–390.
34. Wilderjans T, Ceulemans E, Van Mechelen I: **Simultaneous analysis of coupled data blocks differing in size: A comparison of two weighting schemes**. *Comput. Stat. Data Anal.* 2009, **53**:1086–1098, [<http://dl.acm.org/citation.cfm?id=1497631.1497740>].
35. Alter O, Brown PO, Botstein D: **Generalized singular value decomposition for comparative analysis of genome-scale expression data sets of two different organisms**. *Proceedings of the National Academy of Sciences* 2003, **100**:3351–3356.
36. Van Deun K, Hoijtink H, Thorrez L, Van Lommel L, Schuit F, Van Mechelen I: **Testing the hypothesis of tissue selectivity: the intersection–union test and a Bayesian approach**. *Bioinformatics* 2009, **25**(19):2588–2594.
37. Lemmens K, De Bie T, Dhollander T, De Keersmaecker S, Thijs I, Schoofs G, De Weerd A, De Moor B, Vanderleyden J, Collado-Vides J, Engelen K, Marchal K: **DISTILLER: a data integration framework to reveal condition dependency of complex regulons in Escherichia coli**. *Genome Biology* 2009, **10**(3):R27, [<http://genomebiology.com/2009/10/3/R27>].
38. Kiers HAL: **Setting up alternating least squares and iterative majorization algorithms for solving various matrix optimization problems**. *Computational Statistics and Data Analysis* 2002, **41**:157–170.
39. Groenen PJF: **Iterative majorization algorithms for minimizing loss functions in classification** 2002. [Working paper presented at the 8th conference of the IFCS, Krakow, Poland].
40. Borg I, Groenen PJF: *Modern Multidimensional Scaling: Theory and Applications*. Springer series in statistics, New York: Springer-Verlag, 2nd edition 2005.
41. McLachlan GJ, Peel D: *Finite mixture models / Geoffrey McLachlan, David Peel*. Wiley, New York ; Chichester : 2000, [<http://www.loc.gov/catdir/toc/onix07/00043324.html>].

Figures

Figure 1 - Adding the ridge panel to the (Elitist) lasso

Left panel: Elastic net; Right panel: Elitist lasso with ridge penalty. Fit (full lines) and proportion of zeros (dashed lines) in function of the lasso tuning parameter (left panel) and in function of the Elitist lasso tuning parameter (right panel). The different lines refer to different values of the ridge parameter.

Figure 2 - Sparse group lasso

Fit (full lines) and proportion of zeros (dashed lines) for the sparse group lasso. The different lines refer to different values of the group lasso penalty.

Figure 3 - Boxplots of the proportion of recovered variables: 50 percent of true zeros

Boxplots of the proportion of variables correctly classified (selected versus dropped) in function of the value of the tuning parameters. Case with 50 percent of the variables dropped. The different panels refer to the different combinations of structure in the variable selection (from top to bottom: within blocks, between blocks, within and between blocks) and of sparseness approach (from left to right: lasso, group lasso, Elitist lasso, and sparse group lasso). The panels referring to the sparse group lasso are with varying values for the lasso tuning parameter and with the group lasso tuning parameter fixed at $f = 10$.

Figure 4 - Boxplots of the proportion of recovered variables: 90 percent of true zeros

Boxplots of the proportion of variables correctly classified (selected versus dropped) in function of the value of the tuning parameters. Case with 90 percent of the variables dropped. The different panels refer to the different combinations of structure in the variable selection (from top to bottom: within blocks, between blocks, within and between blocks) and of sparseness approach (from left to right: lasso, group lasso, Elitist lasso, and sparse group lasso). The panels referring to the sparse group lasso are with varying values for the lasso tuning parameter and with the group lasso tuning parameter fixed at $f = 10$.

Figure 5 - Comparing sparse weights versus sparse loadings

Boxplots for the proportion of correctly classified variables. Within panels, the boxplots are displayed in function of the block structure present in the variable selection. The upper panels refer to data generated under a model with sparse loadings, the lower panels to data generated under a model with sparse weights. The panels at the left were obtained when analyzing the data with a model based on sparse weights; the panels at the right with a model based on sparse loadings.

Tables

Table 1 - Sparse approaches

Norm	Properties	Sparse approach				
		Lasso	Elastic net	Group lasso	Sparse group lasso	Elitist lasso
l_1	selection and shrinkage at the level of the concatenated data	YES	YES	NO	YES	NO
l_2^2	shrinkage, groups correlated variables	NO	YES	NO	NO	NO
$l_{2,1}$	selection and shrinkage of entire blocks	NO	NO	YES	YES	NO
$l_{1,2}$	selection and shrinkage within each block	NO	NO	NO	NO	YES

Different norms used in the context of sparse approaches, their properties, and specific sparse approaches based on particular combinations of the penalties. A ‘YES’ indicates that the norm is active in the approach.

Table 2 - Summary results for the different simultaneous component analyses with sparse weights

f	Lasso		GroupLasso		ElitistLasso	
	Fit	% zeros	Fit	% zeros	Fit	% zeros
0	0.57	0	0.57	0	0.57	0
0.0001	0.57	86	0.57	0	0.57	88
0.001	0.57	87	0.57	9	0.56	92
0.01	0.57	88	0.57	9	0.52	96
0.1	0.56	92	0.56	9	0.26	99
0.2	0.55	94	0.55	25	0.16	97
0.5	0.52	97	0.47	45	0.08	98
1	0.43	99	0.23	50	0.04	99

Different panels correspond to different approaches: The lasso in the left panel, the group lasso in the middle panel, and Elitist lasso in the right panel. Within each panel, both the fit of the model to the data and the percentage of zero weights are reported. The different rows correspond to different values of the tuning parameter.

Additional Files

Additional file 1 — MATLAB code

The zip file SparseSCA.zip contains four MATLAB functions and a script (test_script.m) to illustrate the use of the main function (sparsesca_weights.m). The main functions sparsesca_weights.m and sparsescaloadings.m implement the proposed sparse simultaneous component algorithms.

Table 3 - Summary results for the different simultaneous component analyses with sparse loadings

f	Lasso		GroupLasso		ElitistLasso	
	Fit	% zeros	Fit	% zeros	Fit	% zeros
0	0.57	0	0.57	0	0.57	0
0.0001	0.57	0	0.57	0	0.57	0
0.001	0.57	0	0.57	0	0.57	4
0.01	0.57	0	0.57	0	0.54	19
0.1	0.57	7	0.57	0	0.36	37
0.2	0.56	10	0.56	0	0.28	41
0.5	0.53	20	0.54	0	0.17	47
1	0.46	28	0.46	0	0.10	47

Different panels correspond to different approaches: The lasso in the first panel, the group lasso in the second panel, and Elitist lasso in the last panel. Within each panel, both the fit of the model to the data and the number of zero loadings are reported. The different rows correspond to different values of the tuning parameter.

Table 4 - Overview of fit and sparseness for retained sparse group lasso models

f_L	f_R	f_G	Fit	Number of zeros in				
				C1	C2	C3	C4	C5
0.5	0.0001	0.01	0.52	178	176	176	178	179
0.5	0.0001	0.1	0.49	166	156	167	161	159
0.5	0.0001	0.2	0.44	150	173	145	143	169
0.5	0.001	0.1	0.49	158	167	161	166	156
0.5	0.001	0.2	0.44	169	150	146	145	173
0.5	0.01	0.1	0.48	154	154	166	165	160
1	0.0001	0.01	0.43	184	181	183	184	182
1	0.001	0.001	0.43	181	185	185	185	186
1	0.001	0.01	0.43	181	182	184	183	180
1	0.01	0.0001	0.42	180	183	180	179	174
1	0.01	0.001	0.42	182	179	174	180	179
1	0.01	0.01	0.41	177	180	173	178	181

Solutions with five components that have non-zero component weights in only one data block, a fit $> .40$, and more than 50 percent of zero weights in the remaining block. The strength of the different tuning parameters is indicated in the first three columns, the fit is displayed in the fourth column, and the 5 remaining columns show for each component (C1-C5) how many of the 188 metabolites received a zero-weight.

Table 5 - Metabolites with non-zero weights in the two selected solutions

	metabolite	$f_L = 0.5$	$f_L = 1$
C1	3,5-dihydroxypentanoate :	0.68	0.88
C1	valine :	0.58	0.20
C1	3-phenyllactate or isomer :	0.55	1.21
C1	isoleucine :	0.48	
C1	tyrosine :	0.41	0.03
C1	phenylalanine :	0.40	
C1	unknown mass 304, 319 and 406 :	0.01	
C1	spectrum not complete6 :	-0.06	
C1	mixed spectrum3 :	-0.43	0.36
C1	keto-gluconate (?) :	-0.46	0.25
C2	fumarate :	1.40	1.99
C2	malate :	0.96	1.06
C2	aspartate :	0.42	
C2	monomethylphosphate :	0.39	
C2	C18:1 fatty acid3 :	0.37	0.19
C2	unknown1 :	0.37	
C2	spectrum not complete4 :	0.20	
C2	mixed spectrum2 :	0.19	
C2	glycerate :	0.14	
C2	unknown20 :	0.02	
C3	lactate :	1.23	2.18
C3	pyruvate :	0.71	0.39
C3	disaccharide12 :	0.49	0.11
C3	3-dehydroquininate :	0.38	
C3	disaccharide8 :	0.33	
C3	citrate :	0.29	
C3	disaccharide9 :	0.27	
C3	unknown mass 318 and 420 :	0.17	
C3	unknown mass 217 and 191 :	0.17	
C3	disaccharide13 :	0.11	
C3	2-hydroxybutanoate :	0.09	
C4	ADP :	1.16	1.01
C4	GDP :	0.96	1.21
C4	UDP-glucose :	0.71	0.14
C4	UTP :	0.34	
C4	unknown27 :	0.20	
C4	GMP :	0.20	
C4	FBP :	0.09	
C5	spectrum not found7 :	1.41	2.04
C5	guanine :	0.73	
C5	orotate :	0.51	0.34
C5	spectrum not complete5 :	0.31	
C5	mixed spectrum6 :	0.24	
C5	N-acetylaspartate		
C5	+ beta-phenylpyruvate :	0.23	
C5	thymine :	0.12	

Metabolites with non-zero component weights for each of the five components (C1 to C5). The component weights of two selected models are shown that differ in the degree of sparsity ($f_L = 0.5$ and $f_L = 1$).

Table 6 - Component scores for the selected solution

Condition	Ferm. time	C1	C2	C3	C4	C5
Reference	16	-0.42	-0.11	-0.06	-0.30	-0.27
	24	-0.26	-0.14	0.00	0.29	-0.09
	32	0.30	-0.09	-0.26	0.07	0.05
	40	0.40	-0.15	-0.27	-0.24	0.03
pH +	48	0.38	-0.06	-0.06	0.34	0.09
	16	-0.35	-0.13	-0.28	0.99	-0.25
	24	0.08	-0.22	0.14	-0.35	-0.10
	40	0.46	-0.20	-0.35	-0.30	-0.13
oxygen + oxygen ?	48	0.54	-0.26	-0.38	-0.10	-0.12
	40	-0.21	0.05	0.51	-0.02	-0.13
	16	-0.44	-0.24	0.00	-0.24	-0.21
	24	-0.22	-0.03	0.42	0.32	-0.15
phosphate +	40	0.34	0.10	1.05	0.24	-0.03
	64	0.59	0.05	0.50	0.24	-0.08
	16	-0.54	-0.23	-0.08	-0.23	-0.23
	24	-0.53	-0.26	0.18	-0.27	-0.17
phosphate -	40	-0.09	0.06	0.59	0.26	-0.10
	48	0.14	-0.01	-0.02	0.13	-0.13
	16	-0.27	-0.25	-0.03	0.04	-0.14
	24	0.26	-0.21	0.19	-0.35	0.01
succinate	40	0.53	-0.21	-0.33	-0.56	-0.14
	24	-0.10	1.03	-0.09	-0.13	-0.19
	40	0.06	1.21	-0.13	-0.05	-0.08
Wild type	48	0.12	1.07	-0.11	-0.02	0.19
	16	-0.42	-0.27	-0.34	-0.20	-0.05
	24	-0.23	-0.14	-0.31	0.38	0.44
	40	-0.11	-0.17	-0.22	0.22	0.94
	48	-0.04	-0.19	-0.26	-0.14	1.06

Component scores for each of the five components (C1-C5). The samples were obtained in a specific environmental condition (first column) and at a particular fermentation time (second column).

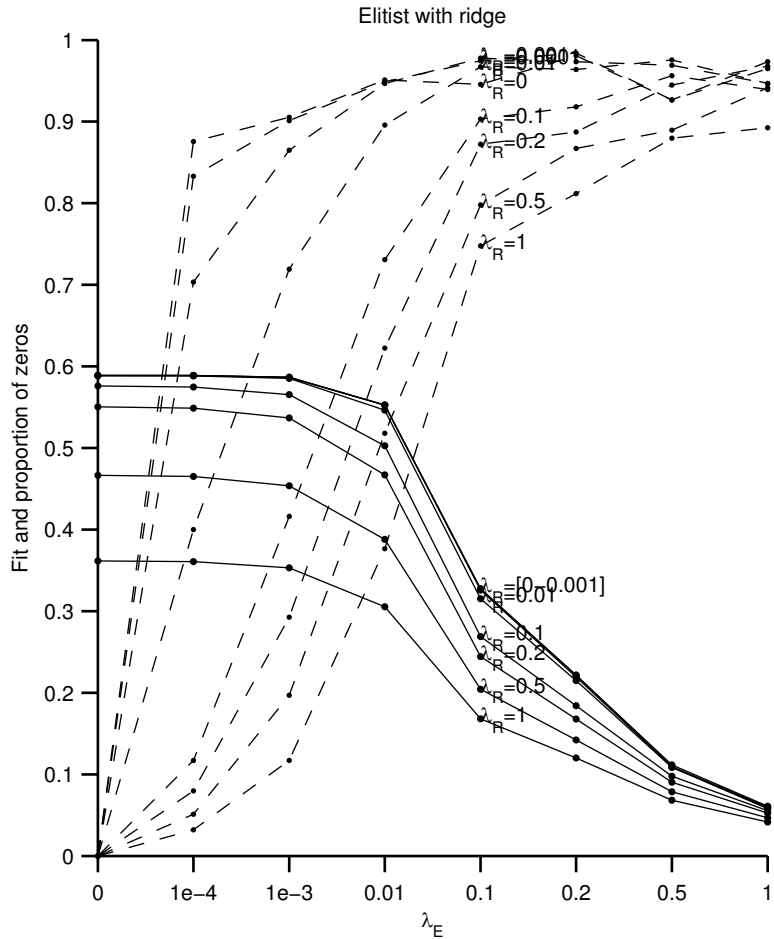
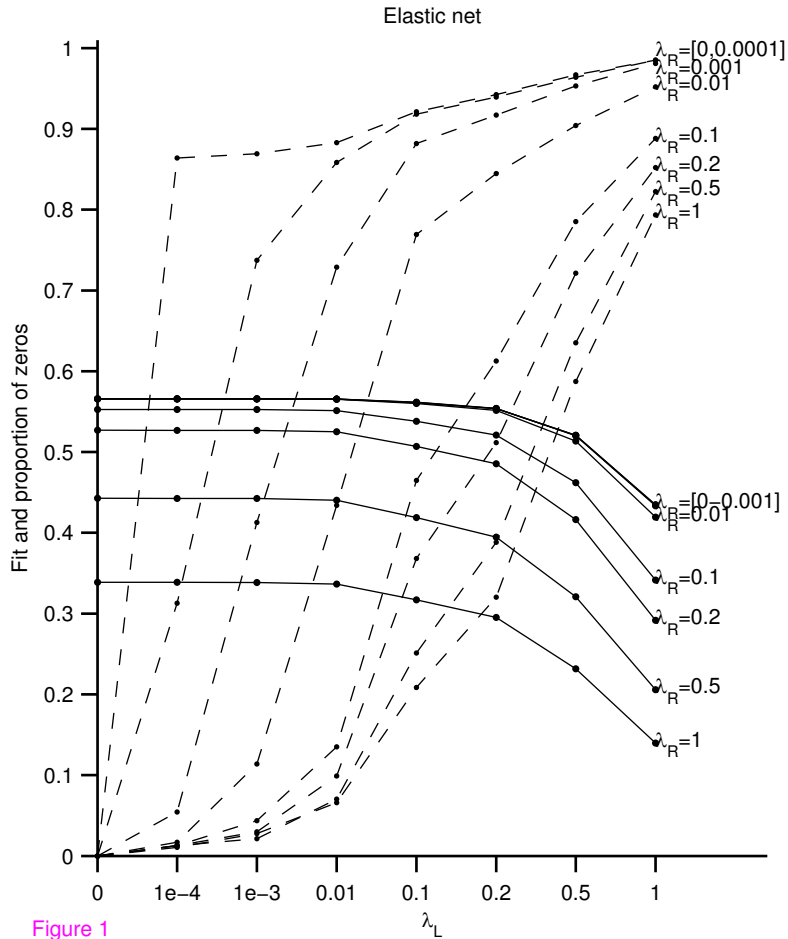


Figure 1

Sparse group lasso

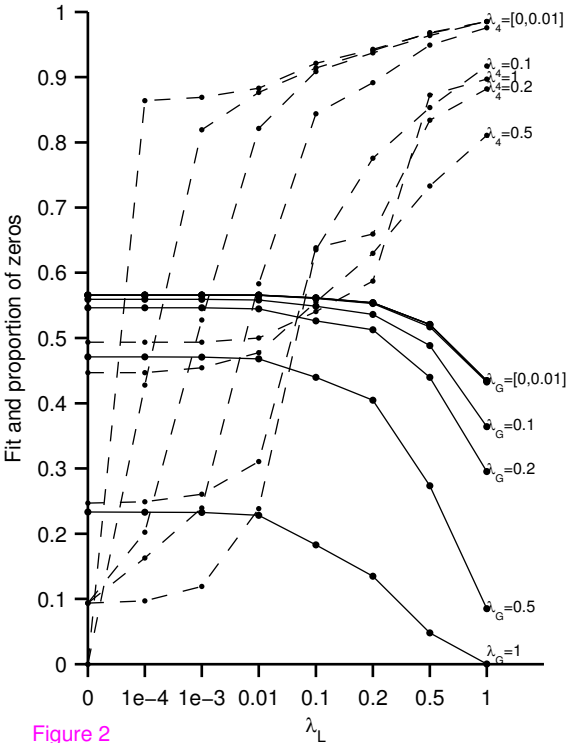


Figure 2

50% zeros

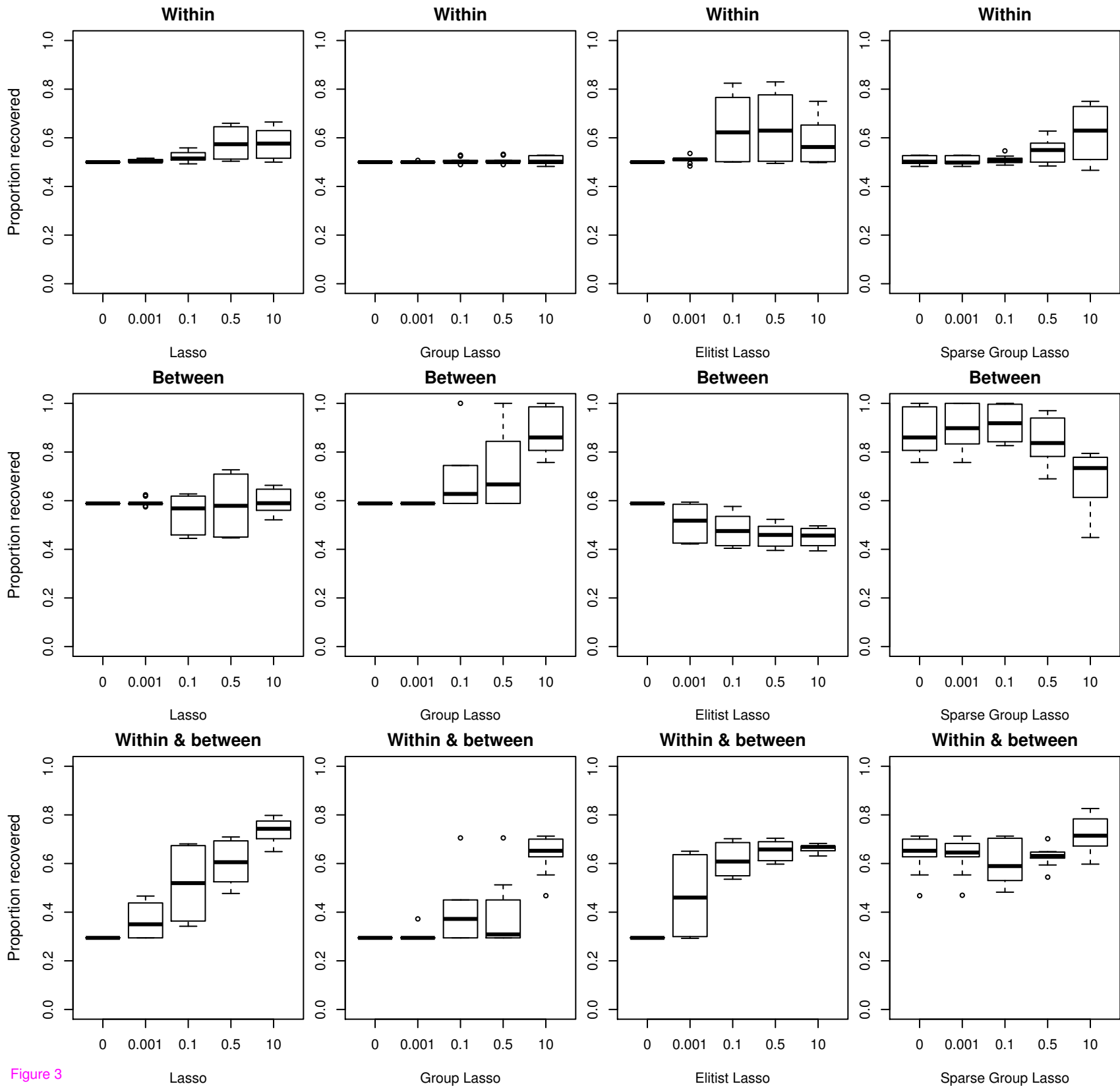


Figure 3

90% zeros

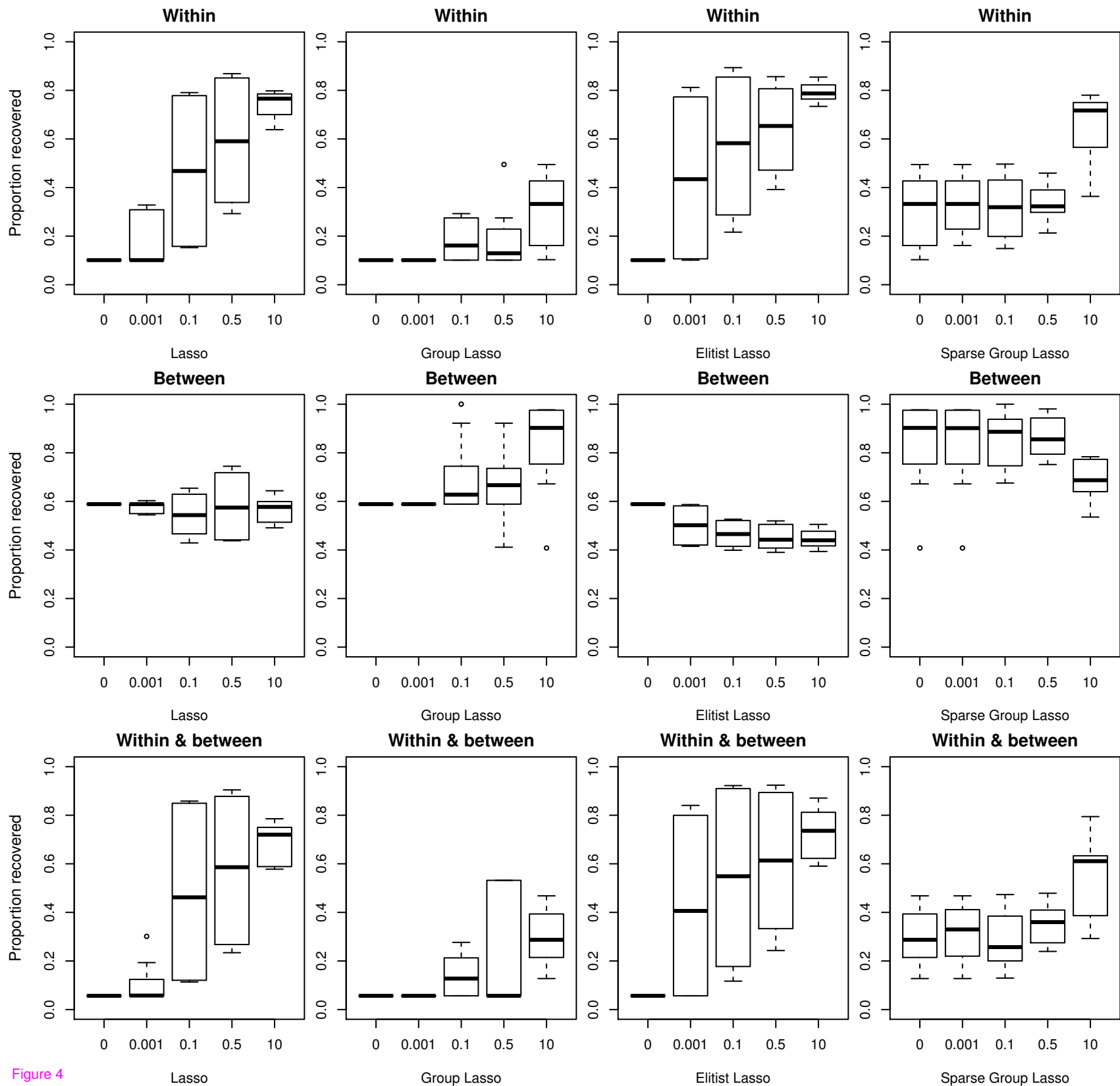


Figure 4

Proportion correctly dropped/selected

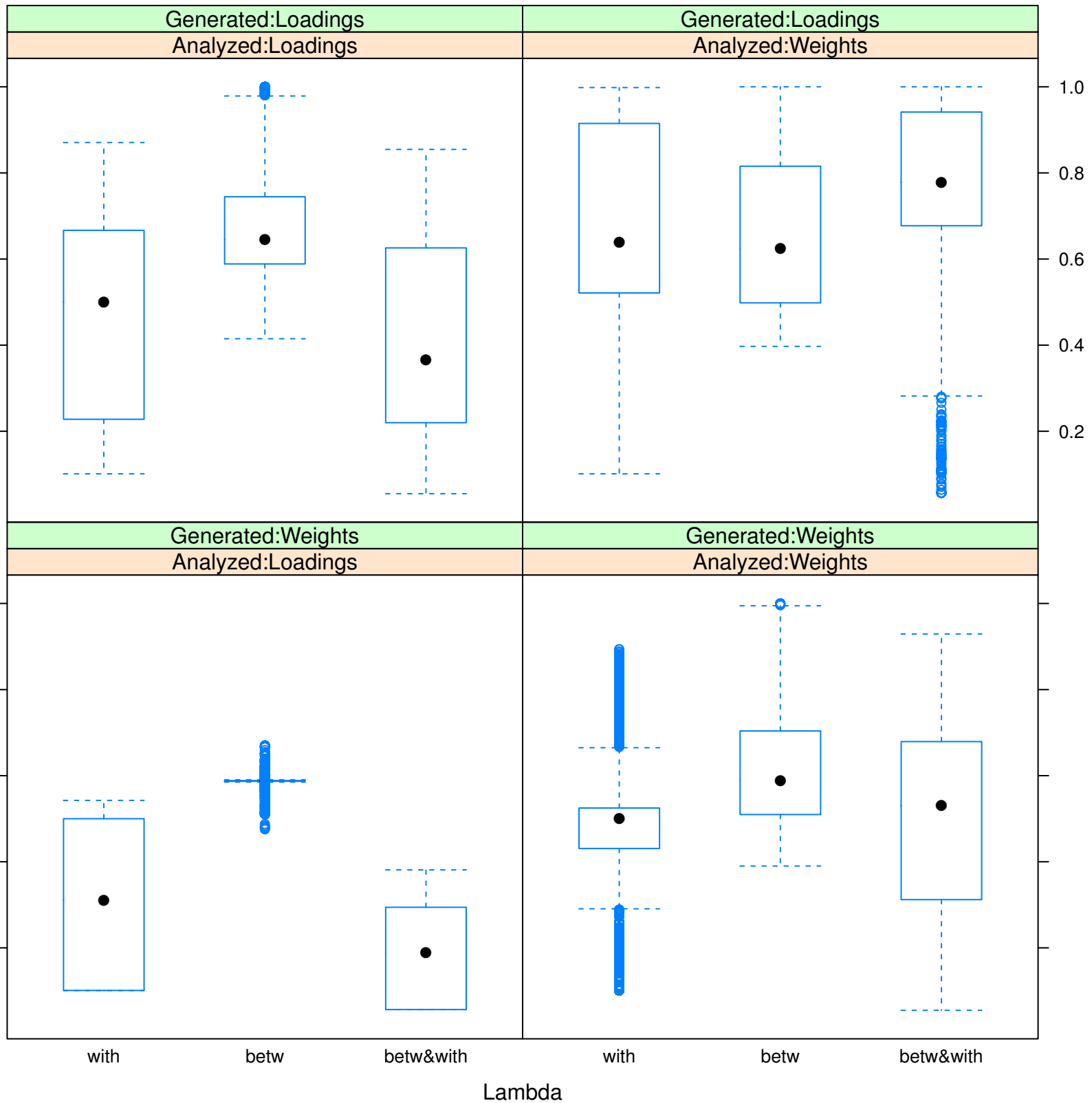


Figure 5

Additional files provided with this submission:

Additional file 1: SparseSCA.zip, 4K

<http://www.biomedcentral.com/imedia/1014234599618365/supp1.zip>