

Performance of methods that separate common and distinct variation in multiple data blocks

Ingrid Måge¹, Age K. Smilde², Frans M. van der Kloet³

1. Nofima, Osloveien 1, N-1430 Ås, Norway
2. Biosystems Data Analysis, Swammerdam Institute for Life Sciences, University of Amsterdam, Science Park 904, 1098 XH, Amsterdam, The Netherlands
3. Department of women and child, Academic Medical Center, Meibergdreef 9, 1105 AZ, Amsterdam, The Netherlands

Abstract

In many areas of science, multiple sets of data are collected from the samples. Such data sets can be analysed by data fusion (or multi-block) methods. The aim is usually to get a holistic understanding of the system or better prediction of some response. Lately, several scientific groups have developed methods for separating *common* and *distinct* variation between multiple data blocks. Although the objective is the same, the strategies and algorithms are completely different for these methods.

In this paper, we investigate the practical aspects of the four most popular methods for separating common and distinct variation: JIVE, DISCO, PCA-GCA and OnPLS. The main barrier complicating the use of any of these methods is model selection and validation. Especially when the numbers of blocks is more than two. By the use of extensive simulations we have elucidated the three properties that are important for assessing the validity of the results: The ability to identify the correct model, the ability to estimate the true, underlying subspaces, and the robustness towards misspecification of the model.

The simulated datasets mimic a range of “real life” data, with different dimensionalities and variance structures. We are thus able to identify which methods work best for different types of data structures, and pinpoint weak spots for each method. The results show that PCA-GCA works best for model selection, while JIVE and DISCO give the best estimates of the subspaces and are most robust towards model misspecification.

Keywords: Data fusion, DISCO, JIVE, OnPLS, PCA-GCA

1 Introduction

Multi-block methods are a family of data fusion methods designed to analyse and interpret systems where the same samples are characterized by several blocks of variables. Typical examples are found in food science, where the same products may be characterized by sensory attributes, instrumental measurements and consumer acceptance, or in medical science, where the blocks might correspond to different –omics platforms as well as clinical measurements and lifestyle variables of the same

patients. Multi-block methods have existed for a long time¹⁻³, but the interest in the field is renewed lately due to increased data generation in many scientific fields⁴⁻⁶. Multi-set analysis is a different type of data fusion, where the data are linked in the variables mode instead of the samples. In this paper we focus on the multi-block case, although several of the methods and results also apply for multi-set data.

The motivation for analysing several data blocks simultaneously can be either a better understanding of the system at hand or an improved prediction of some response variable. So-called *asymmetric* fusion methods are used when there is a natural path or predictive direction between the data blocks, whereas *symmetric* methods treat all blocks on equal footing and are more suited for explorative analysis. In this paper, we focus on symmetric data fusion but the methods and conclusions are also relevant for asymmetric data fusion.

The most straightforward way to fuse data is simply to combine all data blocks into one matrix and apply standard data analysis tools on the merged data. Simultaneous Component Analysis (SCA)⁷, also called SUM-PCA, is an example of such a method. The advantage of this approach is that it can easily handle any number of blocks. The drawback however, is that the individual block contribution and their relationship to each other is hard to interpret. To overcome this drawback, a subgroup of data fusion methods has emerged lately. These methods aim at identifying and separating *common* and *distinct* variation across data blocks. Several scientific groups have developed such methods in parallel. In this paper we compare four of the most used methods, called DISCO, JIVE, PCA-GCA and OnPLS⁸⁻¹¹. Although the objective of all these methods is the same, all these methods rely on latent variables for data compression, but the strategies and algorithms are completely different. A barrier for practical use of these methods is the model validation, especially when the numbers of blocks is more than two. A better understanding of the properties of these methods is therefore needed, in order to assess the validity of the resulting models.

Common variation (also called joint or overlapping) refers to underlying phenomena that are captured in several of the data blocks, while distinct variation (also called individual or unique) correspond to phenomena that are only found in one block. The separation of common and distinct variation in cases with two, three and four data blocks is illustrated schematically in Figure 1. The problem is quite straightforward with two blocks, resembling a bi-directional regression problem with multivariate input and output where the common part is the predictive/predicted part of each block. The complexity increases dramatically with increasing number of data blocks, since the common variation can be either global (across *all* blocks) or local (across subsets of blocks). In most real cases however, several of the subspaces are likely to be empty, and the final model is not necessarily that complex. Not knowing the true underlying model however, makes it a challenge to define which subspaces to include in or to keep out of the model.

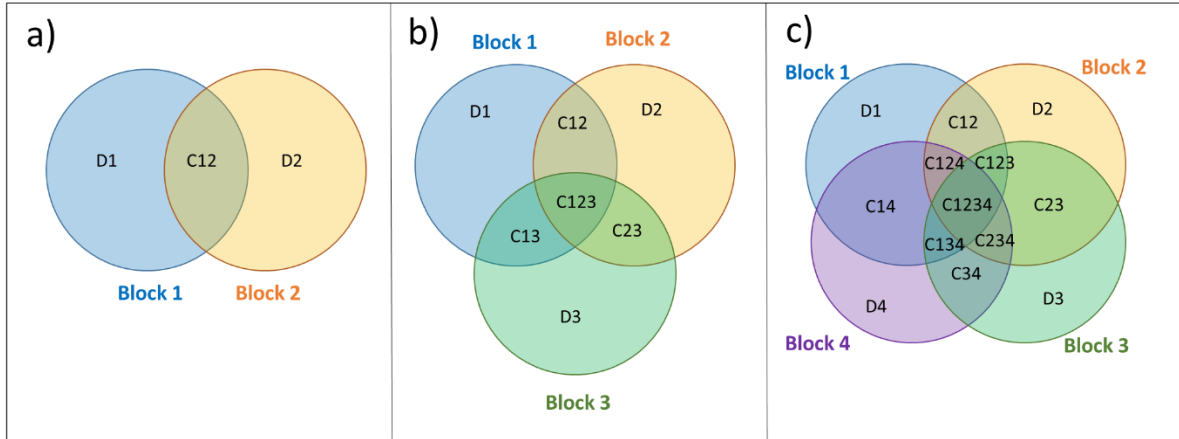


Figure 1. Schematic illustration of the decomposition of common and distinct subspaces in the case of a) two, b) three and c) four data blocks. Common and distinct subspaces are denoted by the letters C and D respectively, followed by the block numbers they belong to. Subspaces that are common for all blocks are called “globally common” and subspaces across subsets of blocks are called “locally common”.

A unifying framework for the aforementioned methods has been published lately, describing the scientific problem itself and the existing methods in linear algebraic terms¹². The methods JIVE, DISCO and OnPLS have been compared previously in a two block scenario¹³. Simulations were used to demonstrate the strength and weaknesses of the different methods and real data was used to demonstrate the applicability of the methods. Not knowing the true underlying sources of variation in the real data however, a true performance of the methods could not be assessed. To complete the comparisons between the methods from a theoretical to a more practical point of view, in this paper we compare the true performances of the methods.

In contrast to the previous work, we have done a broader range of simulations of a three-block scenario. We investigate the properties of four of the methods (JIVE, DISCO, PCA-GCA and OnPLS) with regard to three properties that are important for practical use of the methods:

1. **Model selection:** The ability to identify the correct model, i.e. number of common and distinct components.
2. **Subspace recovery:** The ability to estimate the true subspaces, given the correct model.
3. **Robustness:** How the subspace recovery is affected by fitting a “wrong” model.

The investigations are based on a series of simulations, where data sets with different variance structures and dimensions are simulated according to an experimental design. In this way, we try to mimic the diversity of “real life” multi-block situations as closely as possible, and identify how the different methods perform under varying conditions. No real data sets are analysed in this work, as the objective is to compare how each method recovers the *true* data structure (which is unknown for real data). By doing this, we are able to identify how the methods perform on different types of data structures, and point out weak spots for each method. We also suggest improvements for further method development.

2 Data fusion methods

Consider K data matrices $\mathbf{X}_1, \dots, \mathbf{X}_K$, with the same l samples and different J_k variables in each matrix. The most straightforward method for fusing these data sets is then to do a PCA of the concatenated

matrices. This method is known under several names, for instance SUM-PCA⁷, which is identical to Simultaneous Component Analysis (SCA¹⁴; applied in the multi-block setting instead of the multi-set setting) for the case where data is linked in the sample mode. The solution is found by calculating the singular value decomposition of the concatenated matrix

$$\mathbf{X} = \left[\frac{\mathbf{X}_1}{m_1} \mid \dots \mid \frac{\mathbf{X}_K}{m_K} \right], \quad (1)$$

where m_k is some scaling factor to correct for differences in variance. The concatenated \mathbf{X} is decomposed into scores \mathbf{T} and loadings \mathbf{P} :

$$\mathbf{X} = \mathbf{TP}' + \mathbf{E} \quad (2)$$

By splitting the loadings \mathbf{P} into block-specific parts, each data block \mathbf{X}_k can be represented by:

$$\mathbf{X}_k = \mathbf{TP}'_k + \mathbf{E}_k \quad (3)$$

Note that the scores \mathbf{T} are the same for all blocks, and the model does not separate common and distinct components explicitly.

The decomposition of common and distinct subspaces in three data blocks is illustrated schematically in Figure 1b). Each subspace is represented by a set of basis vectors (scores \mathbf{T}), and the decomposition of each data block can be written as

$$\begin{aligned} \mathbf{X}_1 &= \mathbf{T}_1^{(C123)} \mathbf{P}_1^{(C123)'} + \mathbf{T}_1^{(C12)} \mathbf{P}_1^{(C12)'} + \mathbf{T}_1^{(C13)} \mathbf{P}_1^{(C13)'} + \mathbf{T}_1^{(D1)} \mathbf{P}_1^{(D1)'} + \mathbf{E}_1 \\ \mathbf{X}_2 &= \mathbf{T}_2^{(C123)} \mathbf{P}_2^{(C123)'} + \mathbf{T}_2^{(C12)} \mathbf{P}_2^{(C12)'} + \mathbf{T}_2^{(C23)} \mathbf{P}_2^{(C23)'} + \mathbf{T}_2^{(D2)} \mathbf{P}_2^{(D2)'} + \mathbf{E}_2 \\ \mathbf{X}_3 &= \mathbf{T}_3^{(C123)} \mathbf{P}_3^{(C123)'} + \mathbf{T}_3^{(C23)} \mathbf{P}_3^{(C23)'} + \mathbf{T}_3^{(C13)} \mathbf{P}_3^{(C13)'} + \mathbf{T}_3^{(D3)} \mathbf{P}_3^{(D3)'} + \mathbf{E}_3 \end{aligned} \quad (4)$$

Where the subscript represents the block index and the superscript shows if a subspace is common (C) or distinct (D), and which blocks that are included in the subspace. Each of the subspaces can be split into individual basis vectors, represented by lowercase letters and an additional subscript (in parenthesis) representing the index of the basis vector. For instance, the two-dimensional subspace $\mathbf{T}_1^{(C123)} \mathbf{P}_1^{(C123)'}$ can be split into $\mathbf{t}_{1(1)}^{(C123)} \mathbf{p}_{1(1)}^{(C123)'}$ + $\mathbf{t}_{1(2)}^{(C123)} \mathbf{p}_{1(2)}^{(C123)'}$.

The assumption is that there exists some underlying phenomenon that is captured by several data blocks, yielding a subspace that is common between these blocks. This means that the basis vectors in $\mathbf{T}_1^{(C123)}$, $\mathbf{T}_2^{(C123)}$ and $\mathbf{T}_3^{(C123)}$ should span the same column space, representing the globally common variation. The calculation of scores and loadings differ for the different methods, as well as orthogonality constraints between scores. This will be explained in more detail in the following sections, but for a thorough description of the mathematical framework we refer to reference¹² Note also that often, only a subset of all the possible common subspaces are included in the final model, especially if the number of blocks is large.

2.1 DISCO

The DISCO method is mostly used in behavioural sciences¹⁵, and can also be used in multi-set data fusion, i.e. when the blocks are linked in the variables mode. Simultaneous Component Analysis

(SCA) acts as a starting point for the DISCO method. The SCA is followed by an orthogonal rotation towards a user-defined target loading matrix \mathbf{P}^* that defines the common and distinct parts. The loading matrix \mathbf{P} from equation 2 is rotated orthogonally towards \mathbf{P}^* with rotation matrix \mathbf{B} such that the squared sum $\sum((\mathbf{1} - \mathbf{P}^*) * \mathbf{P}\mathbf{B})^2$ is minimized. The resulting rotation matrix is then used to calculate the rotated scores and loadings, representing either common or distinct variation. See original papers for further details^{10,16}.

In DISCO, the common scores are exactly the same for all blocks, so $\mathbf{T}_1^{(C123)} = \mathbf{T}_2^{(C123)} = \mathbf{T}_3^{(C123)}$, $\mathbf{T}_1^{(C12)} = \mathbf{T}_2^{(C12)}$, $\mathbf{T}_1^{(C13)} = \mathbf{T}_3^{(C13)}$ and $\mathbf{T}_2^{(C23)} = \mathbf{T}_3^{(C23)}$. The scores are therefore in the column space of the concatenated \mathbf{X} , but not necessarily in the columnspaces of each individual \mathbf{X}_k . The rotation is orthogonal, meaning that all (both common and distinct) score vectors are orthogonal to each other. While the orthogonality definitely has some advantages, there is little reason to expect all distinct phenomena to be orthogonal in real life data. These constraints might therefore be too strict and give a suboptimal representation of the common and distinct subspaces.

The model selection consists of two steps: first, the total number of components is decided upon. Next, the best target rotation matrix (\mathbf{P}^*) is sought. Usually, the rotations to all possible targets for a pre-defined total number of components are evaluated in order to find the best model. The goodness of fit is evaluated by determining the sum of squared deviation of the normalized rotated score matrix with the target rotation matrix that was used. The best rotation matrix is the one that shows the smallest deviation. This process is very computationally intensive, as the number of possible targets quickly becomes large. In addition, several targets often have an approximately equal fit, and there is no clear global minimum. In such cases, it is a good idea to inspect all the models that are not significantly different from the global minimum.

2.2 JIVE

The JIVE method has gained popularity mainly in biomedical applications^{17–21}. JIVE is, similar to DISCO, also an extension of the regular SCA decomposition and can be used for both multi-block and multi-set applications. In contrast to the other methods we address in this paper, the JIVE method only facilitates decomposition in global common and distinct subspaces and does not allow for local common subspaces.

The optimisation criterion of JIVE is to minimize the squared residuals, $\|\mathbf{E}\|^2$, where \mathbf{E} is the concatenated residuals for all data blocks. This is obtained through an iterative algorithm that starts by estimating the common components by SCA on the concatenated matrix \mathbf{X} . Then, the distinct components for each block are found by applying SVD on what remains after deflating the common part. The original \mathbf{X} is then updated by deflating the distinct components, and the procedure is repeated until convergence of the residuals. For details, see the original JIVE paper⁸.

As for DISCO, the common scores are equal for all blocks. The orthogonality constraints in JIVE however are a bit more flexible, as the distinct parts of the different blocks are not necessarily orthogonal to each other.

The model complexity is estimated in a procedure where first a significant number of components for the distinct parts is estimated using permutation tests followed by the estimation of the number of common components in a similar manner. Details can be found in the supplementary material of the original JIVE paper⁸. In this work we have used a significance level of 5% for the permutation tests.

2.3 PCA-GCA

This method is a combination of PCA and Generalized canonical Correlation Analysis (GCA). Some applications of this approach have been published^{22–24}, and a similar method for asymmetric data fusion has also been developed^{11,25}. GCA is a generalized version of the two-block method Canonical Correlation Analysis (CCA), and can be applied to any number of blocks²⁶.

The PCA-GCA algorithm starts by decomposing each block individually by PCA, keeping a relevant number of scores from each block. Then, GCA is used to find common components between these scores. The common components are removed from the original blocks by orthogonalisation, and the distinct components are found by applying SVD on the remainders. It can also be used with multi-set data by applying GCA on the PCA-loadings instead of the scores.

A major difference between PCA-GCA and the other methods is that it operates on the individual data blocks \mathbf{X}_k , *not* on the concatenated data. This means that the common components are in the column spaces of each block, not of the concatenated \mathbf{X} . Because of this, the method is invariant to between-block scaling, meaning that scaling by the factor m_k in Equation (1) is unnecessary.

The common scores are not identical for the blocks, but the correlation between them should be high, meaning that $\mathbf{T}_1^{(C123)} \approx \mathbf{T}_2^{(C123)} \approx \mathbf{T}_3^{(C123)}$. The distinct components are not orthogonal to distinct components from other blocks, and neither to common components among the other blocks. For instance, $\mathbf{T}_1^{(D1)}$ is not orthogonal to $\mathbf{T}_2^{(D2)}$, $\mathbf{T}_3^{(D3)}$, $\mathbf{T}_2^{(C23)}$ and $\mathbf{T}_3^{(C23)}$, but it is orthogonal to all components which include block 1.

Like for DISCO, the model selection is a two-step procedure. First, the numbers of components in the initial PCA's has to be decided. Then, the GCA provides a number of *candidates* for common components, and the actual number is selected by evaluating the canonical correlation coefficient together with the explained variances for each of these candidates. The cut-offs should be set based on knowledge about the noise level in the data, and in this paper we have defined common components as those with a correlation >0.9 and explained variance >5% in all of the involved blocks. The model selection strategy may seem complicated, but we have seen from experience that the choice of initial PCA components is not crucial, as long as the number is not too low. If too many initial components are included, the GCA will yield some spurious noise components with high correlation, but these will be filtered out by the explained variance cut-off.

2.4 OnPLS

OnPLS is the multiblock extension of the O2PLS algorithm, and the method is often used for integrating –omics data^{27–30}. O2PLS is based on analysis of the covariance matrix of two blocks, $\mathbf{X}_1^t \mathbf{X}_2$. If multiple blocks are involved, the different covariance matrices $\mathbf{X}_k^t \mathbf{X}_l$ ($k, l = 1, \dots, K, k \neq l$) are concatenated and analysed similar to an SCA approach. In this way, a global common direction is estimated. Anything orthogonal to this direction is either distinct or partially common and is determined by repetitions of the OnPLS procedure on these orthogonal parts. After the global and local variation is determined they are removed from the original data and the global common variation is (re-)calculated. For a full description of the algorithm used here, see appendix and reference⁹.

The orthogonality characteristics are equal to those of the PCA-GCA method: The common scores are not identical for all the blocks, but the correlation between them should be high. The distinct components are *not* orthogonal to distinct components from other blocks neither to the common components among the other blocks.

Similar to PCA-GCA, the number of common components is decided by evaluating the correlation between score vectors from different blocks, in addition to the explained variances. This correlation is however not the same as in PCA-GCA, and the threshold is usually set much lower. Here we have used a threshold of 0.5 for the correlation, as suggested in reference⁹, and 5% for the explained variance. No clear model selection strategy has been published for OnPLS, at least not for more than two blocks. We have therefore developed a strategy based on some cross-validation principles, see appendix.

3 Simulation study

3.1 Simulated data

The simulation study is based on a situation with three data blocks, with varying dimensions and variance structures. In every case, each block has three underlying dimensions, according to Eq. (5). The number of variables is also kept constant at 40, 50 and 60 for the three blocks respectively.

$$\mathbf{X}_k = \mathbf{t}_{k(1)}\mathbf{p}'_{k(1)} + \mathbf{t}_{k(2)}\mathbf{p}'_{k(2)} + \mathbf{t}_{k(3)}\mathbf{p}'_{k(3)} + \mathbf{E}_k, \quad k = 1,2,3 \quad (5)$$

All the methods have slightly different orthogonality constraints, and the simulated data is not intended to match the constraints of any specific method. The idea is rather to mimic the assumption that a common subspace is defined by the same underlying phenomenon, represented by identical score vectors in the simulated data. The score vectors (\mathbf{t}) are therefore normally distributed random numbers (i.e. non-orthogonal), while the loadings (\mathbf{p}) within each block are orthogonal vectors of normally distributed random numbers. Three different allocations of common and distinct components were investigated as described in Table 1.

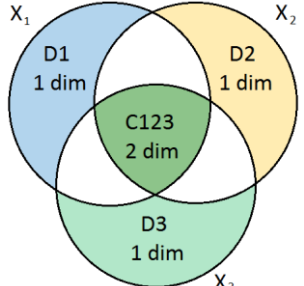
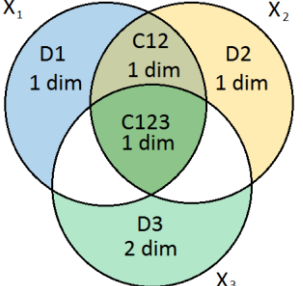
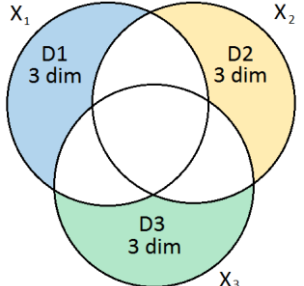
	MODEL 1 Two global common components	MODEL 2 One global and one local common component	MODEL 3 No common components
Model subspaces	 $\mathbf{X}_1 = \mathbf{X}_1^{(C123)} + \mathbf{X}_1^{(D1)} + \mathbf{E}_1$ $\mathbf{X}_2 = \mathbf{X}_2^{(C123)} + \mathbf{X}_2^{(D2)} + \mathbf{E}_2$ $\mathbf{X}_3 = \mathbf{X}_3^{(C123)} + \mathbf{X}_3^{(D3)} + \mathbf{E}_3$	 $\mathbf{X}_1 = \mathbf{X}_1^{(C123)} + \mathbf{X}_1^{(C12)} + \mathbf{X}_1^{(D1)} + \mathbf{E}_1$ $\mathbf{X}_2 = \mathbf{X}_2^{(C123)} + \mathbf{X}_2^{(C12)} + \mathbf{X}_2^{(D2)} + \mathbf{E}_2$ $\mathbf{X}_3 = \mathbf{X}_3^{(C123)} + \mathbf{X}_3^{(D3)} + \mathbf{E}_3$	 $\mathbf{X}_1 = \mathbf{X}_1^{(D1)} + \mathbf{E}_1$ $\mathbf{X}_2 = \mathbf{X}_2^{(D2)} + \mathbf{E}_2$ $\mathbf{X}_3 = \mathbf{X}_3^{(D3)} + \mathbf{E}_3$
Equality of scores in Eq. 5	$\mathbf{t}_{1(1)} = \mathbf{t}_{2(1)} = \mathbf{t}_{3(1)}$ $\mathbf{t}_{1(2)} = \mathbf{t}_{2(2)} = \mathbf{t}_{3(2)}$ $\mathbf{t}_{1(3)} \neq \mathbf{t}_{2(3)} \neq \mathbf{t}_{3(3)}$	$\mathbf{t}_{1(1)} = \mathbf{t}_{2(1)} = \mathbf{t}_{3(1)}$ $\mathbf{t}_{1(2)} = \mathbf{t}_{2(2)}$ $\mathbf{t}_{1(3)} \neq \mathbf{t}_{2(3)} \neq \mathbf{t}_{3(2)} \neq \mathbf{t}_{3(3)}$	$\mathbf{t}_{1(1)} \neq \mathbf{t}_{1(2)} \neq \mathbf{t}_{1(3)} \neq \mathbf{t}_{2(1)} \neq \mathbf{t}_{2(2)} \neq \mathbf{t}_{2(3)} \neq \mathbf{t}_{3(1)} \neq \mathbf{t}_{3(2)} \neq \mathbf{t}_{3(3)}$

Table 1 Overview of the three simulated models. The common components have identical scores in the simulations, but the estimates are not necessarily identical across blocks. MODEL 3 does not have any common components, and is included in the study only to evaluate false positives in the model selection.

For each of these models, data sets with different sample sizes, variance distributions and noise levels were simulated according to a full factorial design, giving eighteen different combinations of the design factors. The design factors and levels are detailed in Table 2. Fifty data sets were simulated for each model type and each combination of design factors.

Table 2 Overview of the design factors and their levels. For MODEL 3 (no common components), only design factors F1 and F3 are relevant. Note also that $\mathbf{t}_{k(1)}$ is globally common and $\mathbf{t}_{k(3)}$ is distinct in both MODEL 1 and MODEL 2. $\mathbf{t}_{k(2)}$, on the other hand, is globally common in MODEL 1 and locally common (between \mathbf{X}_1 and \mathbf{X}_2) in MODEL 2. The percentages explained variance for F2 refer to noise-free data.

Design factor	Levels
F1 sample size	<ol style="list-style-type: none"> 20 samples. Number of samples \ll number of variables 60 samples. Number of samples \approx number of variables 200 samples. Number of samples \gg number of variables
F2 variance distribution	<ol style="list-style-type: none"> Common components dominate all blocks <ul style="list-style-type: none"> $\mathbf{t}_{k(1)}$ explain 50%, $\mathbf{t}_{k(2)}$ explain 35% and $\mathbf{t}_{k(3)}$ explain 15% Unequal variance distribution <ul style="list-style-type: none"> \mathbf{X}_1: $\mathbf{t}_{1(1)}$ explain 50%, $\mathbf{t}_{1(2)}$ explain 35% and $\mathbf{t}_{1(3)}$ explain 15% \mathbf{X}_2: $\mathbf{t}_{2(1)}$ explain 30%, $\mathbf{t}_{2(2)}$ explain 20% and $\mathbf{t}_{2(3)}$ explain 50% \mathbf{X}_3: $\mathbf{t}_{3(1)}$ explain 15%, $\mathbf{t}_{3(2)}$ explain 15% and $\mathbf{t}_{3(3)}$ explain 70% Distinct components dominate all blocks <ul style="list-style-type: none"> $\mathbf{t}_{k(1)}$ explain 15%, $\mathbf{t}_{k(2)}$ explain 10% and $\mathbf{t}_{k(3)}$ explain 75%
F3 noise	<ol style="list-style-type: none"> 5% (of average signal) homoscedastic noise in all three blocks, mimicking precise data 20% (of average signal) homoscedastic noise in all three blocks, mimicking noisy data
Full factorial design = $3 \times 3 \times 2 = 18$ different combinations	

3.2 Analysis of the simulated data

All datasets were analysed by the four methods JIVE, DISCO, PCA-GCA and OnPLS. The analysis was done in three steps:

- Model selection.** Each method has a procedure for estimating the dimension of the common and distinct subspaces, as explained in sections 2.1-2.4. We have put main emphasis on the numbers of common components, as this affects all data-blocks and consequently is the most critical for model interpretation. Furthermore, all methods but DISCO (where there is no order) start by estimating/determining the common part, indicating its significance.
- Subspace recovery.** Models were fitted using the correct numbers of common and distinct components, and the estimated scores were compared to the true (noise-free) simulated score vectors. For each subspace, the recovery was calculated as the explained variance after regressing the estimated score vectors on the true scores, in the same manner as was done in reference¹³.
- Robustness.** Models were fitted using *incorrect* numbers of components for one or several subspaces. In each case, at least one subspace was also modelled with the correct number of components. The list of misspecified models is given in Table 3. The recovery of the correctly specified subspace(s) was then calculated and compared to the recovery using the correct model

(step 2). In this way, we are able to evaluate to what extent the estimation of a given subspace is affected by the other subspaces in the fitted model.

MODEL 3 does not have any common components, and was only used for comparing the model selection strategies (i.e. identifying false common components). All the simulations and analyses were done in MATLAB (R2016a, The MathWorks Inc.). An implementation of JIVE was downloaded from the University of North Carolina⁸. A graphical user interface of DISCO was downloaded from KU Leuven³¹, and converted into a command line version for this paper. The OnPLS implementation is our interpretation of the algorithm as described by the original authors^{9,32}. MATLAB code for OnPLS, PCA-GCA as well as the code to generate the simulation data can be downloaded from Nofima³³.

Table 3. Overview of the misspecified models used to evaluate robustness. The shaded cells represent the incorrectly specified subspaces. See Table 1 for description of the subspaces and MODEL1/2.

Case	True model	Fitted model: Number of components in each subspace						
		C123	C12	C13	C23	D1	D2	D3
Mis-1	MODEL 1	2				2	2	2
Mis-2	MODEL 1	2	1			1	1	1
Mis-3	MODEL 2	1	1			2	2	2
Mis-4	MODEL 2	1				2	2	2
Mis-5	MODEL 2	1	2			1	1	2
Mis-6	MODEL 2	2	1			1	1	2
Mis-7	MODEL 2	1	1	1		1	1	2
Mis-8	MODEL 1	1	1			1	1	1
Mis-9	MODEL 2	2				1	1	2
Mis-10	MODEL 1	3				1	1	1

4 Results

In this section, the results for model selection and subspace recovery are described separately for the three models in Table 1 (subsections 4.1-4.3). Then, the robustness with regard to model misspecification is described in subsection 4.4.

4.1 Two global common components (MODEL 1)

The simulations revealed that the four methods handle the data sets with different characteristics quite differently, with regard to both model selection and recovery of the true subspaces. When it comes to model selection, PCA-GCA performs best on average, finding the correct number of common components for 79% of the data sets. OnPLS, DISCO and JIVE follow with 59%, 53% and 43% respectively. The actual estimated numbers of common components are visualized in Figure 2, and we clearly see that:

- JIVE performs well for the case where the common components dominate, but fails completely when the variance distribution is unequal or when the distinct components dominate. The performance is better with larger sample sizes, but it is not affected by noise.

- PCA-GCA always finds the correct number when the noise is low, but fails in some cases when the noise level is high and common components are not dominating all blocks. Surprisingly, the performance decreases as the sample size increases for these instances. The reason might be that the correlation threshold should be set lower than 0.9 for noisy data, and the correlation is simply overestimated when the sample size is low.
- DISCO has problems with low sample size, and the performance is very poor for noisy data. The estimated number of common components is often five, which is the maximum number set in the analysis for this model. This means that it defines all components as common. DISCO performs best when the distinct components dominate the data, which is expected since the rotation criterion is designed to find distinct subspaces. As explained in section 2.1, there are often several models with approximately equal fit. If all these are taken into account, the percentage of correct models increases from 53% to 75%. However, the models with equally good fit are often very different, and there is no easy way to single out the correct one.
- OnPLS is quite variable across all combinations of design factors. The best results are obtained for the “easiest” data set (low noise and dominating common components), and the poorest results are obtained when the distinct components dominate the data.

The same conclusions hold for the estimated numbers of distinct components (results not shown). When it comes to recovery of the true common and distinct subspaces (given the correct model), the overall average values are 94% (JIVE), 92% (PCA-GCA), 92% (DISCO) and 88% (OnPLS). Multiple comparisons (Tukey-Kramer, 5% significance level) show that there are no statistically significant differences between JIVE, DISCO and PCA-GCA, while OnPLS is significantly lower than all the other methods. The average recoveries for each setting of the design factors are plotted in Figure 3. All methods have lower recoveries for the distinct subspaces when sample size is low, and data sets where the distinct components dominate are generally the most challenging. Although these effects can be seen for all methods, they are larger for OnPLS, causing the lower overall average value.

OnPLS is based on concatenated pairs of covariance matrices. As the number of blocks increase, this concatenated data matrix becomes very wide. It is therefore reasonable to assume that the estimation depends on the number of data blocks. In order to investigate this further, we did additional simulations with two and four data blocks. The results showed no differences in subspace recovery related to the number of blocks, for any of the four methods.

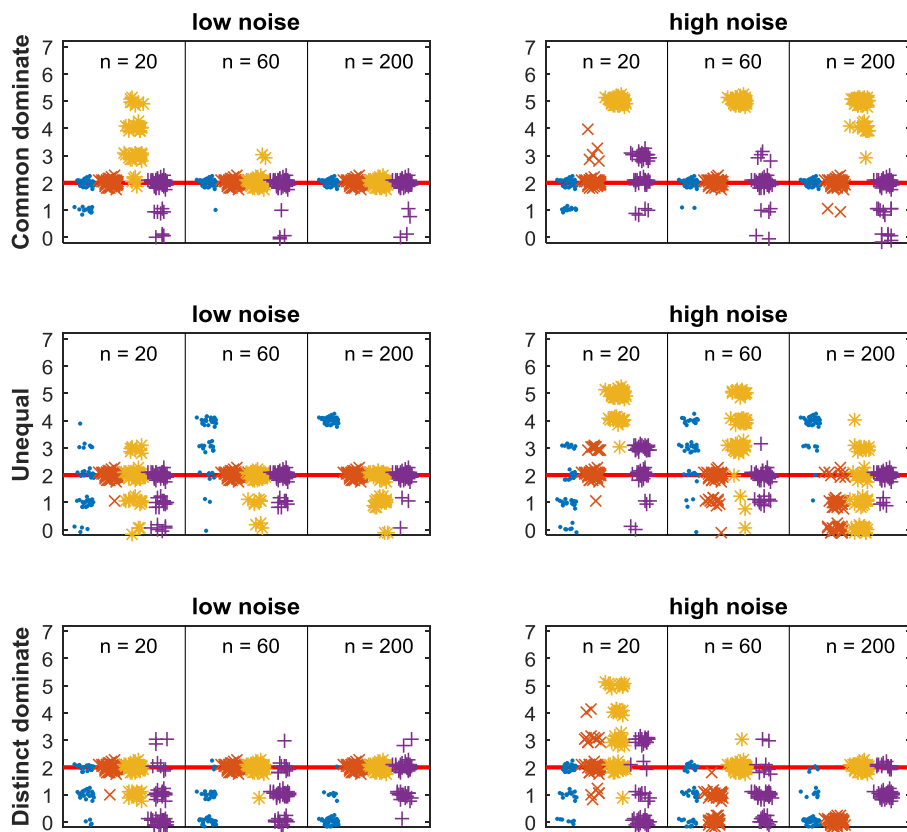


Figure 2 MODEL 1: Estimated numbers of common components from JIVE (.), PCA-GCA (x), DISCO (*) and OnPLS (+) for all combinations of design factors. Each symbol represents one of the fifty repetitions of each combination. The numbers are integers, but some random jitter is added in order to see the amount of symbols at each location. The solid line represents the true number of common components.

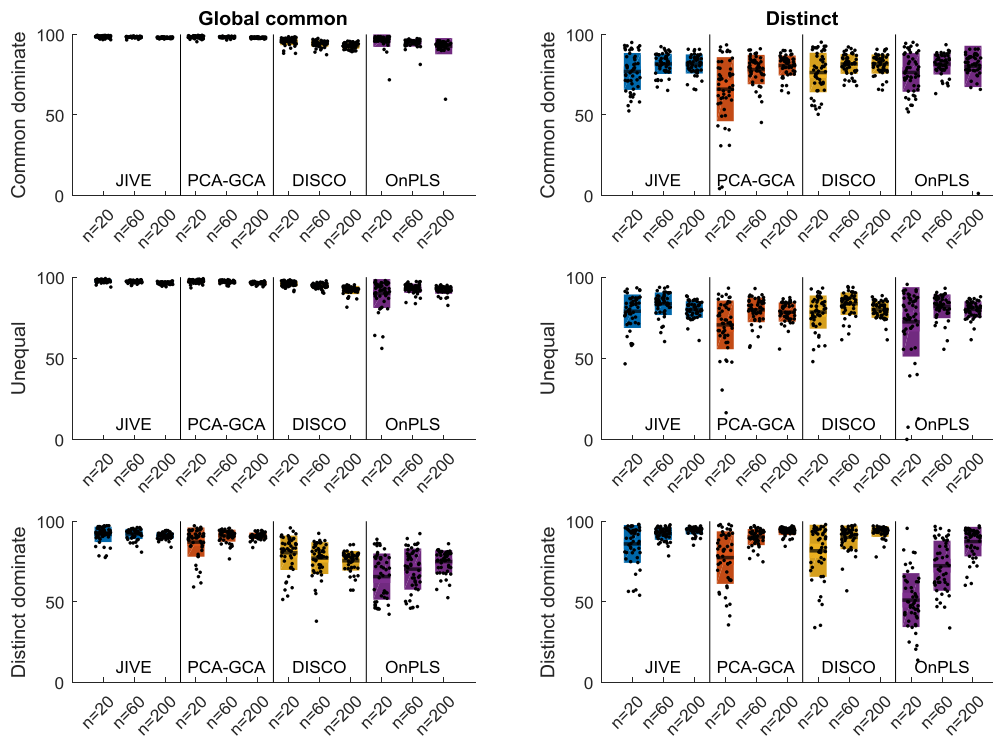


Figure 3 Subspace recoveries (%) of MODEL 1 for each combination of the design factors “F1: sample size” and “F2: variance distribution”. Recoveries for the distinct subspaces are averaged over the three blocks. Only results for the high noise level is shown, as results for the low noise level show a similar pattern.

4.2 One global and one local common component (MODEL 2)

The selected numbers of global and local common components are illustrated in Figure 4 and Figure 5 respectively, and the recovery of subspaces are shown in Figure 6. On average, PCA-GCA finds the correct model for 73% of the data sets, followed by OnPLS (58%) and DISCO (34%). Compared to MODEL 1, OnPLS performs slightly better while DISCO performs poorer. Aside from that, the methods handle different types of data structures in the same way as described for MODEL 1.

JIVE does not distinguish between global and common components. In these simulations, JIVE identified one common component in 50% of the data sets, and two common components in 11% of the data sets. This suggests that local common components are usually defined as distinct in the model selection procedure. However, we should not put too much emphasis on these results, given the shortcomings of the JIVE model selection procedure illustrated in the previous section. In order to investigate further how JIVE handles local components, models with either one or two common components were fitted. When one common component was fitted, the true local component was recovered in the individual subspaces of blocks one and two. When the fitted model had two common components, one of these corresponded to the true local component. This component explained very little variance (<5%) in block three, which indirectly identifies it as a local component.

The average recovery over all common and distinct subspaces was very similar to the MODEL 1 results: 93% (JIVE), 92% (PCA-GCA), 91% (DISCO) and 84% (OnPLS). Again, OnPLS is significantly lower than the other three methods. The recovery for all combinations of design factors are shown

in Figure 6. It is clear that data sets where the distinct components dominate are the most challenging for all methods, and that it is necessary to have a large sample size in such cases. This effect is most profound for OnPLS, which completely fails to recover the global and local common subspaces for data sets with only 20 samples and where the distinct components dominate.

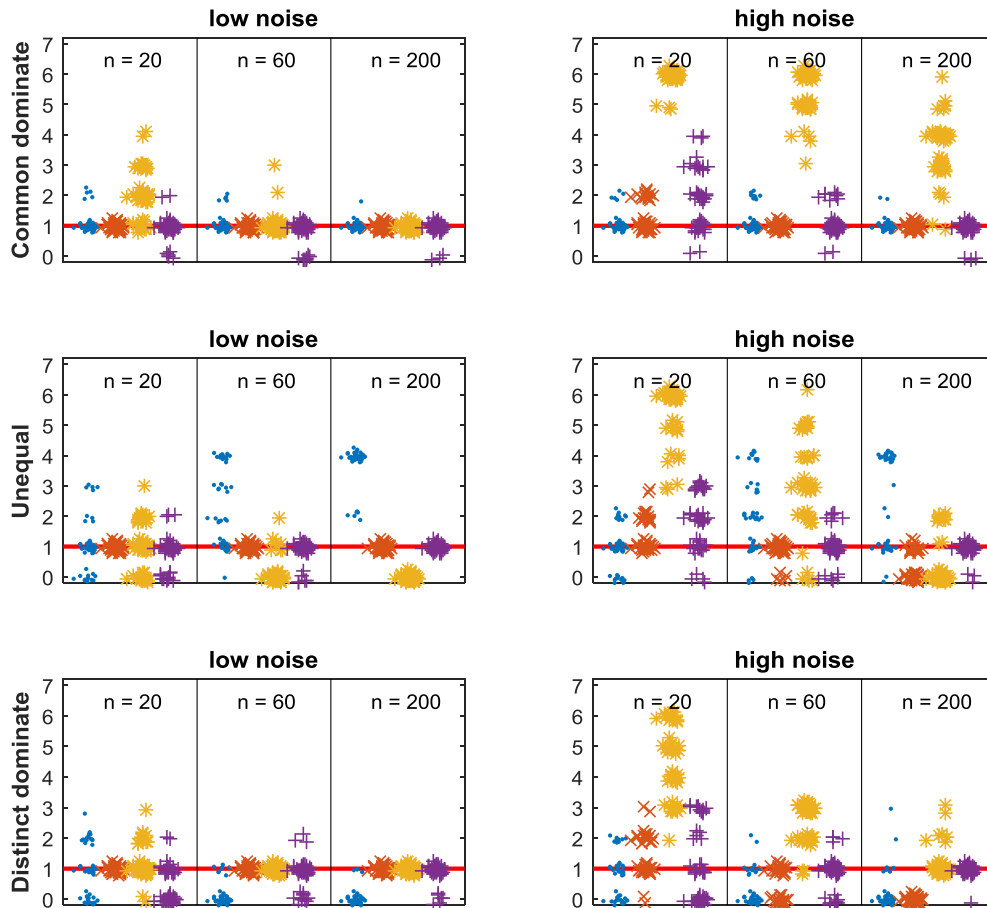


Figure 4 MODEL 2: Estimated numbers of global common components from JIVE (\cdot), PCA-GCA (\times), DISCO ($*$) and OnPLS ($+$) for all combinations of design factors. Each symbol represents one of the fifty repetitions of each combination. The numbers are integers, but some random jitter is added in order to see the amount of symbols at each location. The solid line represents the true number of components.

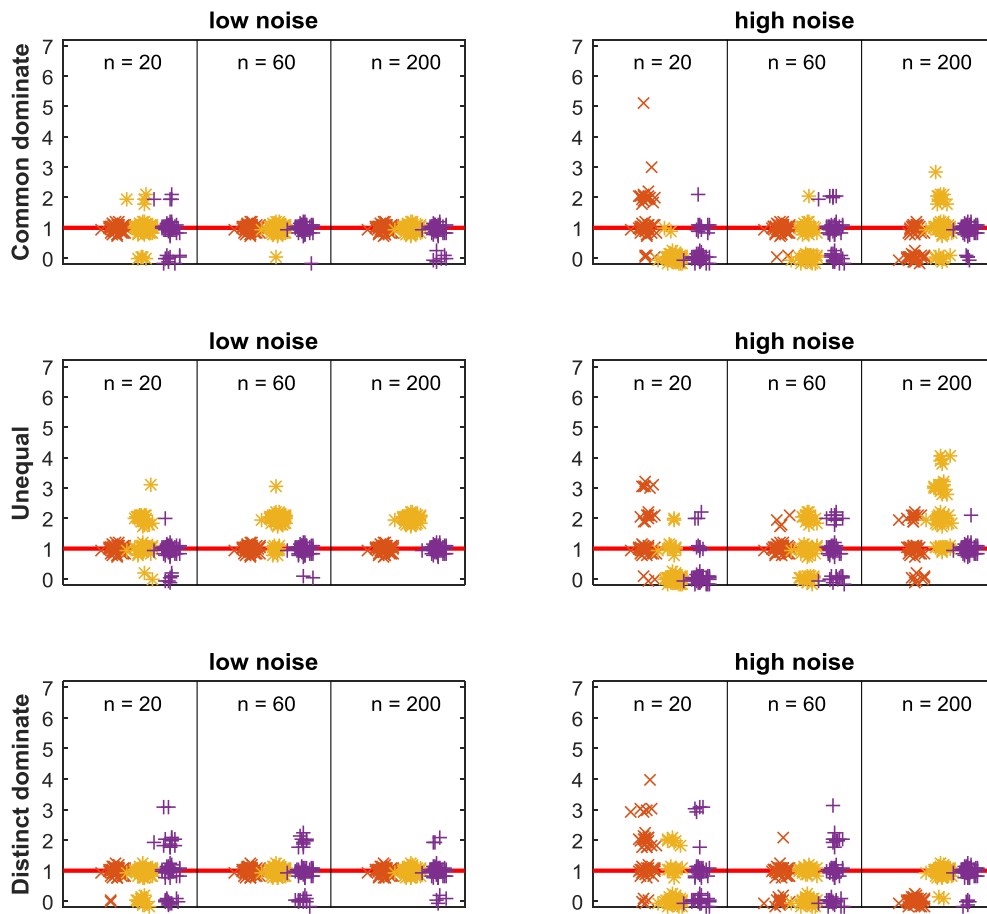


Figure 5 MODEL 2: Estimated numbers of local common components from PCA-GCA (x), DISCO (*) and OnPLS (+) for all combinations of design factors. Each symbol represents one of the fifty repetitions of each combination. The numbers are integers, but some random jitter is added in order to see the amount of symbols at each location. The solid line represents the true number of components.

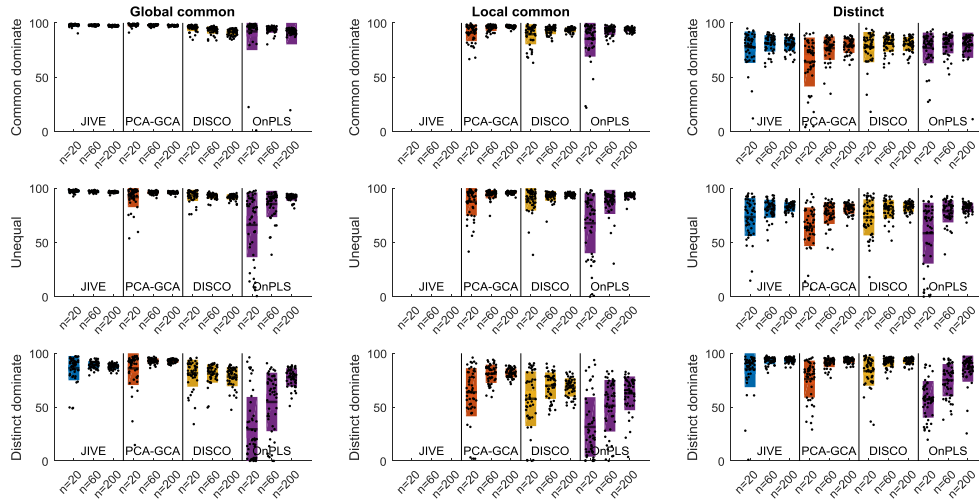


Figure 6. Subspace recoveries (%) of MODEL 2 for each combination of the design factors “F1: sample size” and “F2: variance distribution”. Recoveries for the distinct subspaces are averaged over the three blocks. JIVE does not support local common components, so the local common is fitted as a global common for JIVE. Only results for the high noise level is shown, as results for the low noise level show a similar pattern.

4.3 Zero common components (MODEL 3)

Datasets with no common components were simulated in order to assess the type I error for finding common components (false positives). The PCA-GCA method was exceptional in this case, with no false positives at all. DISCO and OnPLS had 66% and 4% false positives when the sample size was low (20 samples), but not for the higher sample sizes. JIVE, on the other hand, had substantial amounts of false positives regardless of samples size (14%, 20% and 6% for sample size 20, 60 and 200 respectively). These results agree with the previous assertion that there is a problem with the model selection procedure for JIVE.

4.4 Robustness towards model misspecification

The robustness of each method was investigated by changing the dimensions of at least one of the fitted subspaces, while other subspace(s) were correctly specified. In this way, we could evaluate how the recovery of a correctly defined subspace was affected by changing other subspaces in the model. The misspecified models are listed in Table 3, and the reductions in recovery compared to those obtained with correct model settings (calculated as $\text{Recovery}_{\text{correct}} - \text{Recovery}_{\text{misspecified}}$) are shown in Figure 7. The results were similar for both noise levels, so the figure shows data sets with high noise level only. Note that some reductions are slightly above zero, meaning that the recovery is *better* for the misspecified than the true model. This occurs by chance in some cases where the recovery by the true model is low.

Only two of the models can be fitted by JIVE (models *Mis-1* and *Mis-10*), since it does not fit local subspaces. These models show good robustness for both the global and distinct subspaces, with a maximum reduction in subspace recover of approximately thirteen percentage points. Small sample size makes the recovery of subspaces less robust, especially for the distinct components.

PCA-GCA is a sequential method, where the global components are extracted first. The global part is therefore never affected by changing the local and distinct parts. Similarly, the local components are not affected by changing the distinct components. If we increase the number of global components, as in model *Mis-6*, the local component is modelled as global and the recovery of the local component therefore drops towards zero. The huge impact however diminishes after closer

inspection which reveals that one of the global components explains little variance in one of the blocks and therefore should be interpreted as local. The distinct components are extracted in succession to the local and global components and are therefore affected by changes in both global and local components.

DISCO is remarkably stable for most of the misspecified models. The largest reduction in subspace recovery is seen for the global common subspace when fitting *Mis-4* (Figure 7c), except when distinct components dominate all blocks. This is logical, since the rotation in DISCO is defined by the distinct components. We also notice that the robustness is often poorer for data sets with small sample size (small dots), especially for the distinct subspaces.

For OnPLS, the robustness of the common parts is stable though subpar compared to DISCO and PCA-GCA. The distinct parts however are relatively robust and are comparable to DISCO for most of the misspecified models. Like DISCO, it has problems with recovering the global subspace for *Mis-4* except from when distinct components dominate all data blocks (black dots). For the local subspaces in Figure 7h), it has big problems with *Mis-6*, but not as severe as PCA-GCA. *Mis-3* is problematic for data sets where the common components dominate (black dots).

Overall, DISCO and JIVE are the most robust methods. OnPLS is quite variable, but it is poorer or approximately equal to DISCO in all cases. PCA-GCA is able to recover the global common variation in all cases of misspecification of the subspaces. The local and distinct subspaces however, are very sensitive to misspecification of the global part. To some extent this is also true for OnPLS. Both these methods determine the subspaces in successive order, first global then local and finally the distinct. In cases where the dimension of the global subspace is uncertain, there is a high risk of misinterpretation when using PCA-GCA and OnPLS.

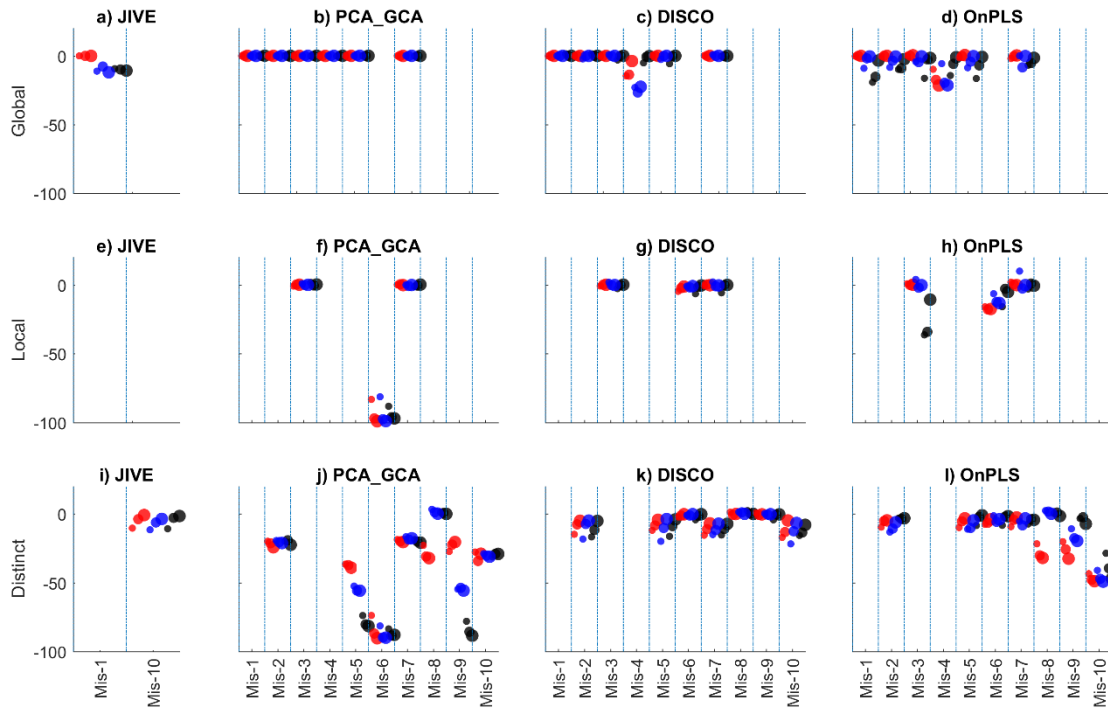


Figure 7. Robustness towards model misspecification. Each subplot displays the reduction in subspace recovery for various misspecified models (defined in Table 3). Each dot represents one of the nine combinations of sample size and variance distribution (defined in Table 2), averaged over fifty repetitions. The dot size represents sample size and colour represents variance distribution (red: common dominates all blocks, blue: unequal variance distribution, black: distinct dominates all blocks). Note that several of the models cannot be fitted by JIVE, since the method does not allow for local common components.

5 Discussion

JIVE has a great potential and should be extended to handle local common components. The estimation of the model complexity however fails terribly. This might be due to the permutation procedure that is used to determine the number of significant components with. Recently it was described that the determination of the number components based on permutation tests has to be corrected for the reduced ranks of the lower components^{34,35}. This is currently not done in the JIVE method. Incorporation of such a correction might lead to a better estimation of the number of components.

DISCO is also a good method to fit a known model. This might seem counterintuitive, since it imposes full orthogonality of components between all data blocks. The rotation towards the orthogonal solution however, is never perfect for real data (i.e. the rotated loadings are not exactly equal to the target matrix (\mathbf{P}^*)). The distinct components are therefore not truly distinct and if the fit is poor they might explain a substantial amount of variation in other data blocks. Because of this “imperfect” rotation towards the target, the residuals with (normalized) target matrices \mathbf{P}^* with more common components are often much lower than those with only few common components. Consequently, the ‘congruence’ approach to select the best model with is biased towards models with (too) many common components. This is exactly what was observed in the simulations.

The results of OnPLS show a high susceptibility towards unequal distributions of variation across the different data-blocks. If the common variation is large however, it seems to decompose the original sources of variation well. One reason might be that OnPLS never is a *true* combination of common

components but rather a concatenation of pairs of common blocks (i.e. covariance matrices between the different pairs of blocks). It can be argued that these concatenations are skewed with respect to the original common direction, especially if the common direction is weak. OnPLS is also more sensitive to sample size than the other methods. This is logical since the estimation of covariance matrices are unstable when samples are few. Furthermore, the OnPLS (and O2PLS) algorithm has an unclear step in defining the parts orthogonal to the common direction, which forms the basis for the resulting local and/or distinct parts. The subsequent deflation steps are not necessarily in the direction of most explained variation of the residual and could therefore limit the interpretability of these parts. It was observed that the recovery of the local and distinct parts is the lowest when using OnPLS, and the robustness towards model misspecification was also variable.

PCA-GCA is not a new method, but rather a combination of two good old workhorses in multivariate data analysis. The results show that it works well for both model selection and recovery of the subspaces. Due to its sequential approach, PCA-GCA is however vulnerable for misspecification of the subspaces that are extracted early in the sequence. PCA-GCA is therefore not recommended for fitting models where the dimensionalities of the first subspaces are doubtful. Both in PCA-GCA and OnPLS, the results depend on the order of extraction of the common subspaces that are on the same "level". This means that in a three-block case, the order of subspaces $C12$, $C23$ and $C13$ is not arbitrary (e.g. $C12$, if removed first from block1 and block2 can be different from $C12$ when $C23$ is removed first from block2 and block3). In practise, however, changing the order will not have a significant effect on the results if the selected model fits the data well.

In real applications, the model selection is usually done manually by inspecting some model characteristics. This can be a quite complicated task, especially if there are many blocks. In PCA-GCA and OnPLS one needs to evaluate both the correlations and the explained variances for every potential component in all subspaces. Although cumbersome, the procedure can also be seen as an advantage for the experienced data analyst since it is transparent and interactive. The same goes for DISCO, where it is advisable to inspect several alternative models thoroughly with regard to explained variances and the interpretability of scores/loadings. JIVE, on the other hand, has a less transparent model selection procedure where the user only has to decide on a confidence level. This approach is easy-to-use and very appealing to a less experienced analyst, but may be seen as a black box by others. In its current state, however, this method does not work properly.

6 Conclusion

The idea is that separation of common and distinct subspaces will give better interpretation of complex systems. However, the applicability of these methods is limited by difficulties in assessing the validity of the separation. The separation in each of these methods is a very intricate process. To prevent these methods of being used as a black box however, we have compared JIVE, DISCO, PCA-GCA and OnPLS with respect to their capability in selecting the right model complexity, their ability to estimate common and distinct variation in different datasets, and their robustness towards fitting a slightly "wrong" model. The results provide some clear guidelines on how to use the methods in practise.

Most of the complexity concerns the model selection procedure, i.e. deciding the dimensionalities of the common and distinct subspaces. This is not a trivial task, and the interpretation of the system will depend heavily of the selected model. Our simulations showed that PCA-GCA works best for model selection in most of the cases.

The simulations showed that JIVE, PCA-GCA and DISCO have good recovery of the underlying common and distinct subspaces in most conditions. It is also clear that OnPLS performs the worst in recovering the real underlying components. DISCO and JIVE are most robust to misspecifications of the fitted model but in its current implementation JIVE is not able to deal with local common components.

Recommended strategy for the current methods:

- Use PCA-GCA for model selection.
- If PCA-GCA finds that there are no local common components, use JIVE for model fitting.
- If there *are* local common components, but the dimensions of the common subspaces are not well defined, use DISCO for model fitting. Alternatively, if the dimensions of the common subspaces are clear use PCA-GCA

It is always a good idea to fit models with both DISCO, PCA-GCA and JIVE, and compare results. Since the goal of all methods is the same, similar scores and loadings imply that the fitted models are valid.

Acknowledgements

We would like to thank Dr. Johan Westerhuis for his helpful and critical comments. Furthermore, we would like to thank the Norwegian Levy on Agricultural Products (FFL; Project no. 262308) for financial support.

References

1. Kettenring JR. Canonical analysis of several sets of variables. *Biometrika*. 1971;58:433-460.
2. Geer van de. Linear relations among k sets of variables. *Psychometrika*. 1984;49(1):79-94.
3. Westerhuis JA, Kourti T, MacGregor JF. Analysis of multiblock and hierarchical PCA and PLS models. *J Chemom*. 1998;12(5):301-321.
4. Lahat D, Adali T, Jutten C. Multimodal Data Fusion: An Overview of Methods, Challenges, and Prospects. *Proc IEEE*. 2015;103(9):1449-1477. doi:10.1109/JPROC.2015.2460697.
5. Blanchet L, Smolinska A. Data Fusion in Metabolomics and Proteomics for Biomarker Discovery. In: Jung K, ed. *Statistical Analysis in Proteomics SE - 14*. Vol 1362. Methods in Molecular Biology. Springer New York; 2016:209-223. doi:10.1007/978-1-4939-3106-4_14.
6. Borràs E, Ferré J, Boqué R, Mestres M, Aceña L, Busto O. Data fusion methodologies for food and beverage authentication and quality assessment – A review. *Anal Chim Acta*. 2015;891:1-14. doi:10.1016/j.aca.2015.04.042.
7. Smilde AK, Westerhuis JA, de Jong S. A framework for sequential multiblock component methods. *J Chemom*. 2003;17(6):323-337. doi:10.1002/cem.811.
8. Lock EF, Hoadley K a., Marron JS, Nobel AB. Joint and individual variation explained (JIVE) for integrated analysis of multiple data types. *Ann Appl Stat*. 2013;7(1):523-542. doi:10.1214/12-AOAS597.
9. Löfstedt T, Hoffman D, Trygg J. Global, local and unique decompositions in OnPLS for multiblock data analysis. *Anal Chim Acta*. 2013;791(June 2012):13-24. doi:10.1016/j.aca.2013.06.026.

10. Schouteden M, Van Deun K, Wilderjans TF, Van Mechelen I. Performing DISCO-SCA to search for distinctive and common information in linked data. *Behav Res Methods*. 2014;46(2):576-587. doi:10.3758/s13428-013-0374-6.
11. Måge I, Menichelli E, Næs T. Preference mapping by PO-PLS: Separating common and unique information in several data blocks. *Food Qual Prefer*. 2012;24(1):8-16.
12. Smilde AK, Måge I, Næs T, et al. Common and Distinct Components in Data Fusion. *J Chemom*. 2017;31(7). doi:10.1002/cem.2900.
13. Van Der Kloet FM, Sebastián-León P, Conesa A, Smilde AK, Westerhuis JA. Separating common from distinctive variation. *BMC Bioinformatics*. 2016;17. doi:10.1186/s12859-016-1037-2.
14. ten Berge JMF, Kiers HAL, van der Stel V. Simultaneous Component Analysis. *Stat Appl*. 1992;4:277-392.
15. Van Mechelen I, Ceulemans E. Component- and Factor-Based Models for Data Fusion in the Behavioral Sciences. *Proc IEEE*. 2015;103(9, SI):1621-1634. doi:10.1109/JPROC.2015.2442652.
16. Van Deun K, Van Mechelen I, Thorrez L, et al. DISCO-SCA and properly applied GSVD as swinging methods to find common and distinctive processes. *PLoS One*. 2012. doi:10.1371/journal.pone.0037840.
17. Kaplan A, Lock EF. Prediction With Dimension Reduction of Multiple Molecular Data Sources for Patient Survival. *Cancer Inform*. 2017;16:1-11. doi:10.1177/1176935117718517.
18. Yu Q, Risk BB, Zhang K, Marron JS. JIVE integration of imaging and behavioral data. *Neuroimage*. 2017;152:38-49. doi:10.1016/j.neuroimage.2017.02.072.
19. Hellton KH, Thoresen M. Integrative clustering of high-dimensional data with joint and individual clusters. *BIOSTATISTICS*. 2016;17(3):537-548. doi:10.1093/biostatistics/kxw005.
20. Kuligowski J, Perez-Guaita D, Sanchez-Illana A, et al. Analysis of multi-source metabolomic data using joint and individual variation explained (JIVE). *Analyst*. 2015;140(13):4521-4529. doi:10.1039/c5an00706b.
21. Lacroix S, Lauria M, Scott-Boyer M-P, Marchetti L, Priami C, Caberlotto L. Systems biology approaches to study the molecular effects of caloric restriction and polyphenols on aging processes. *GENES Nutr*. 2015;10(6). doi:10.1007/s12263-015-0508-9.
22. van den Berg R a, Rubingh CM, Westerhuis J a, van der Werf MJ, Smilde AK. Metabolomics data exploration guided by prior knowledge. *Anal Chim Acta*. 2009;651(2):173-181. doi:10.1016/j.aca.2009.08.029.
23. Smilde AK, Måge I, Næs T, et al. Common and Distinct Components in Data Fusion. *ArXiv*. 2016;1607.02328.
24. Levin-Schwartz Y, Song Y, Schreier PJ, Calhoun VD, Adali T. Sample-poor estimation of order and common signal subspace with application to fusion of medical imaging data. *Neuroimage*. 2016;134:486-493. doi:10.1016/j.neuroimage.2016.03.058.
25. Måge I, Mevik BH, Næs T. Regression models with process variables and parallel blocks of raw material measurements. *J Chemom*. 2008;22(8):443-456.
26. Carroll JD. A generalization of canonical correlation analysis to three or more sets of variables. *Proc 76th Ann Conv Am Psychol Assoc*. 1968:227-228.
27. Copley TR, Aliferis KA, Kliebenstein DJ, Jabaji SH. An integrated RNAseq-H-1 NMR metabolomics approach to understand soybean primary metabolism regulation in response

- to *Rhizoctonia foliar blight disease*. *BMC Plant Biol.* 2017;17. doi:10.1186/s12870-017-1020-8.
28. el Bouhaddani S, Houwing-Duistermaat J, Salo P, Perola M, Jongbloed G, Uh H-W. Evaluation of O2PLS in Omics data integration. *BMC Bioinformatics.* 2016;17(2). doi:10.1186/s12859-015-0854-z.
 29. Srivastava V, Obudulu O, Bygdell J, et al. OnPLS integration of transcriptomic, proteomic and metabolomic data shows multi-level oxidative stress responses in the cambium of transgenic hipl- superoxide dismutase *Populus* plants. *BMC Genomics.* 2013;14. doi:10.1186/1471-2164-14-893.
 30. Szymanski J, Brotman Y, Willmitzer L, Cuadros-Inostroza A. Linking Gene Expression and Membrane Lipid Composition of Arabidopsis. *Plant Cell.* 2014;26(3):915-928. doi:10.1105/tpc.113.118919.
 31. Schouteden M, Van Deun K, Wilderjans T, Van Mechelen I. DISCO-SCA. 2011. <https://ppw.kuleuven.be/okp/software/disco-sca/>. Accessed May 19, 2017.
 32. Löfstedt T, Trygg J. OnPLS-a novel multiblock method for the modelling of predictive and orthogonal variation. *J Chemom.* 2011;25(January):441-455. doi:10.1002/cem.1388.
 33. Nofima modelling downloads. <http://nofimamodeling.org/software-downloads-list/>.
 34. Endrizzi I, Gasperi F, Rødbotten M, Næs T. Interpretation, validation and segmentation of preference mapping models. *Food Qual Prefer.* 2014;32(PA):198-209. doi:10.1016/j.foodqual.2013.10.002.
 35. Vitale R, Westerhuis JA, Næs T, Smilde AK, Noord OE De, Ferrer A. Selecting the number of factors in Principal Component Analysis by permutation testing - Numerical and practical aspects. *J Chemom.* 2017;in review.