Chemometrics and Intelligent Laboratory Systems 195 (2019) 103875

Contents lists available at ScienceDirect

Chemometrics and Intelligent Laboratory Systems

journal homepage: www.elsevier.com/locate/chemometrics

# Weighted sparse principal component analysis

Katrijn Van Deun [a,b,*], Lieven Thorrez [c], Margherita Coccia [d], Dicle Hasdemir [e], Johan A. Westerhuis [e], Age K. Smilde [e], Iven Van Mechelen [a]

[a] Quantitative Psychology and Individual Differences, KU Leuven, Leuven, B-3000, Belgium
[b] Methodology & Statistics, Tilburg University, Tilburg, 5000 LE, the Netherlands
[c] Department of Development and Regeneration, KULAK, Kortrijk, B-8500, Belgium
[d] GSK, Rixensart, Belgium
[e] Biosystems Data Analysis, University of Amsterdam, Amsterdam, 1098 XH, the Netherlands

ARTICLE INFO

*Keywords:*
Sparse principal component analysis
Elementwise weighted least squares
Multiplicative-additive error

ABSTRACT

Sparse principal component analysis (SPCA) has been shown to be a fruitful method for the analysis of high-dimensional data. So far, however, no method has been proposed that allows to assign elementwise weights to the matrix of residuals, although this may have several useful applications. We propose a novel SPCA method that includes the flexibility to weight at the level of the elements of the data matrix. The superior performance of the weighted SPCA approach compared to unweighted SPCA is shown for data simulated according to the prevailing multiplicative-additive error model. In addition, applying weighted SPCA to genomewide transcription rates obtained soon after vaccination, resulted in a biologically meaningful selection of variables with components that are associated to the measured vaccine efficacy. The MATLAB implementation of the weighted sparse PCA method is freely available from https://github.com/katrijnvandeun/WSPCA.

## 1. Introduction

The use of high-throughput experimentation is becoming more popular in both research and practice. Examples include the use of spectroscopic techniques (visible/near-infrared spectroscopy in food processing [1], nuclear magnetic resonance (NMR)); spectrometric techniques (e.g., mass-spectrometry of urine samples for diagnosis [2]); imaging techniques (e.g., functional magnetic resonance in brain research [3]) and of high throughput screening (e.g., genomewide screening of blood samples of patients [4]). These advanced measurement technologies result in vast amounts of data which characteristically contain several thousands or even millions of different variables measured on a limited number of samples. In the general statistical literature such very wide data are called high-dimensional [5]. In many disciplines, including chemistry and biology, dimension reduction techniques such as singular value decomposition and principal component analysis are very popular tools for the analysis of multivariate data [6]. Yet, with the increasing number of measurements, more advanced dimension reduction techniques are often needed for several reasons. First, often there is a need to automatically select the most important variables within the large set for further investigation [7] because such *sparseness* in the number of features at play corresponds to biological reality. Examples of this include transcriptional regulation - which is steered by a few transcription factors only - and the limited number of interacting elements in biological networks [8]. Furthermore, as shown in Ref. [9] PCA performs poorly in case of many more variables than samples (in terms of yielding inconsistent estimates of the loadings), an issue that can also be addressed by selecting a subset of variables having non-zero loadings on the components. Yet another reason for imposing sparseness is to address issues related to the interpretation of the estimates. In principal component analysis (PCA) for example, this would be typically done by inspecting the loadings of each of the variables. With thousands of variables such inspection is an infeasible task, and as shown in the important paper of [10], the common practice of neglecting small loadings is flawed. Hence, sparse PCA (SPCA) techniques [11,12] have been proposed that perform variable selection by making the components dependent on a limited number of variables in the sense that many of the loadings are constrained to be zero. Furthermore, SPCA methods have better statistical properties in terms of consistency of the estimates than ordinary PCA in the high-dimensional setting [13]. Hence, not surprisingly, SPCA techniques and other sparse dimension reduction techniques

* Corresponding author. Quantitative Psychology and Individual Differences, KU Leuven, Leuven, B-3000, Belgium
*E-mail address:* k.vandeun@uvt.nl (K. Van Deun).

https://doi.org/10.1016/j.chemolab.2019.103875
Received 19 April 2019; Received in revised form 12 October 2019; Accepted 21 October 2019
Available online 28 October 2019
0169-7439/© 2019 The Authors. Published by Elsevier B.V. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).

have become a very popular tool for the analysis of high-dimensional data. Yet, this is a field of statistical research that is still growing and subject to improvements.

Like classical PCA, the majority of SPCA methods have several drawbacks. One problem of both PCA and SPCA is the implicit assumption of independent and identically distributed noise [14,15]. Homoscedastic error is often not a realistic assumption: in analytical chemistry, for example, the error increases proportionally with the concentration of the analyte [16]; the same holds for microarray fluorescence intensities, which have a variance that increases with their mean [17,18]. This kind of error variance that increases with the strength of the signal is known as multiplicative error and implies heteroscedasticity (yet not the other way around). Also for fMRI data the assumption of homoscedastic additive noise is unrealistic given the strong structural dependencies in such data [14]. In these situations, methods that do not account for the heteroscedasticity may yield incorrect estimates of the standard errors but may stil be *un*biased: As long as the data (signal with noise) scatter symmetrically around the true model scores (signal only), PCA will yield unbiased estimates. However, for analytical chemistry and micro-array expression data usually a model is assumed that contains both additive and multiplicative error [19]; this is a model with heteroscedastic noise that is no longer symmetrically scattered around the true score and hence implies *biased* estimates in ordinary least squares approaches. An elementwise weighted least squares approach with weights that are inversely proportional to the noise can be used to resolve this issue [19,20]. For classical PCA, Wentzell et al. [18] developed such an approach that incorporates differential weighting of the residuals with inversely proportional weights (see also [21]). No such approach that can weight at the level of single measurements has been proposed yet for sparse PCA.

In summary, researchers and others working with high-dimensional data in disciplines like chemometrics, bioinformatics, and neuroinformatics, are in need of sparse dimension reduction techniques that allow for a differential weighting of each of the elements in the matrix of residuals. Beyond the possibility of incorporating general noise structures, this approach has several other uses. One is that it naturally accommodates for missing data (by introducing zero weights). Furthermore, weighted PCA has also been used for differential weighting of the variables and of the observations. One family of examples that calls for such differential weighting are problems of chemical process monitoring and process control in which one may wish to form principal components for process forecasting. The components in question are based on variables or observations measured at different time points, so more weight being needed for more recent information. A second group of examples may be found in standard atmospheric science. When variables correspond to spatial locations, and weights are needed to account for an uneven spacing between locations (see further [22,23]). Also in these cases, an extension of SPCA to a weighted SPCA (WSPCA) method is useful. Hence, we propose a sparse PCA method that allows to weight each of the elements of the data matrix individually in this paper.

The remainder of this paper is structured as follows: First we introduce the sparse PCA model and the elementwise weighted approach, including a discussion of the estimation of the model parameters and the algorithm. The performance of the novel WSPCA method, compared to state-of-the-art SPCA methods, is assessed in a simulation study. Next, we show how the inclusion of weights that account for the heterogeneity of the measurement error improves prediction for publicly available data from a study on genomewide transcription in samples obtained from subjects vaccinated against influenza [24].

## 2. Methods

We will make use of the following formal notation: matrices will be denoted using bold uppercase letters, the transpose by the superscript $^T$, vectors using bold lowercase, and scalars using lowercase italics. Furthermore, we will use the convention to indicate the cardinality of a running index by the capital of the letter used to run the index (e.g., this paper deals with $J$ variables with $j$ running from 1 to $J$), see Ref. [25].

### 2.1. Formal model

We consider the following low rank approximation of the data matrix $\mathbf{X}$ ,

$$\mathbf{X} = \mathbf{T}\mathbf{P}^T + \mathbf{E}, \tag{1}$$

with $\mathbf{T}$ of size $I \times R$ a matrix of component scores of the $I$ observational units on $R$ underlying components, with $\mathbf{P}$ of size $J \times R$ a matrix with loadings of the $J$ variables on the $R$ components, and with $\mathbf{E}$ an $I \times J$ matrix of residuals. The number of components, $R$, is supposed to be small and $\mathbf{T}$ is subject to an identification constraint; here we will impose $\mathbf{T}^T\mathbf{T} = \mathbf{I}$, which implies orthonormality of the component scores. Furthermore, we will consider model (1) to be subject to sparseness constraints on the loadings: that is, many of the $p_{jr}$ are assumed to be zero but it is unknown which ones these are. Related sparse models have been proposed in the work of [11,14,26–28]. Except for [14] all these papers implicitly rely on the assumption of independent and identically distributed noise.

Standard estimation of the classical PCA model in (1) is based on minimizing the least squares criterion

$$\left\|\mathbf{X} - \mathbf{T}\mathbf{P}^T\right\|^2 \tag{2}$$

with respect to $\mathbf{T}$ and $\mathbf{P}$ and such that $\mathbf{T}^T\mathbf{T} = \mathbf{I}$. This comes down to minimizing the squared Frobenius norm of the residuals: $||\mathbf{E}||^2 = \sum_{ij} e_{ij}^2$. To obtain sparse loadings, the state-of-the-art approach is to add a sparsity inducing penalty such as the lasso, that is, to add $||\mathbf{P}||_1 = \sum_{j,r} \left|p_{jr}\right|$ to the least squares criterion (see, e.g. Refs. [11,27,29,30]),

$$\min_{\mathbf{T},\mathbf{P}} \left\|\mathbf{X} - \mathbf{T}\mathbf{P}^T\right\|^2 + \lambda\|\mathbf{P}\|_1. \tag{3}$$

The lasso penalty is tuned by the metaparameter $\lambda \geq 0$ with larger values implying more sparseness and shrinkage to zero. When $\lambda = 0$, a non-sparse PCA approach is obtained. Note that the lasso is a shrinkage and selection operator meaning that it shrinks all loadings to zero and some exactly to zero [31].

Instead of adding the penalty to the ordinary least squares criterion in (2), however, we will rely on penalizing an *elementwise weighted* least squares criterion. The resulting objective function hence becomes

$$\min_{\mathbf{T},\mathbf{P}} \left\|\mathbf{W} \circ (\mathbf{X} - \mathbf{T}\mathbf{P}^T)\right\|^2 + \lambda\|\mathbf{P}\|_1 \tag{4}$$

with $\mathbf{W}$ a given matrix of nonnegative weights (e.g., reciprocal to the standard error of the noise or 0 in case of missing data) and $\circ$ denoting the elementwise or Hadamard product: $(\mathbf{W} \circ \mathbf{X})_{ij} = w_{ij}x_{ij}$. (If all weights are put equal to 1, the first term in (4) becomes an ordinary least squares objective function, which goes with the implicit assumption that the residuals $e_{ij} = x_{ij} - \sum_r t_{ir}p_{jr}$ are independent and identically distributed.)

To allow for more flexibility and the use of stability selection as a tool for model selection [32] (see the model selection section further on in this paper), we finally propose a slightly more general objective function:

$$\min_{\mathbf{T},\mathbf{P}} \left\|\mathbf{W} \circ (\mathbf{X} - \mathbf{T}\mathbf{P}^T)\right\|^2 + \lambda\left|\mathbf{B} \circ \mathbf{P}\right|_1. \tag{5}$$

The penalty term now includes a matrix $\mathbf{B}$, which contains sets of prespecified, nonnegative weights that allow to tune the strength of the penalty for the individual loadings. Through these matrices we can accommodate the adaptive [30] or randomized lasso [32]; these may be preferred over the ordinary lasso (which is obtained if all entries of $\mathbf{B}$ are put equal to one). The ordinary lasso may be biased in the sense that the

non-zero coefficients are shrunken too much [33,34], and this issue is addressed by the adaptive lasso that penalizes coefficients that have more information content less, for example, by using weights that are inversely proportional to the coefficients resulting from a non-sparse analysis. The randomized lasso, used in combination with a resampling scheme, has been put forward as a powerful and simple procedure that is consistent in terms of variable selection in conditions where the ordinary lasso is not; see Ref. [32] for details.

### 2.2. Algorithm

To find optimizers of the objective function (5), we will make use of an alternating procedure in which each of the arguments (**T** and **P**) is updated conditional upon fixed values for $R$, $\lambda$, **W** and **B** (how to obtain $R$, $\lambda$, and **B** will be discussed in the Section on tuning and model selection). The non-standard elementwise weighted least squares problem is solved by relying on a majorize minimize (MM) procedure [35]. This is a numerical technique that introduces surrogate functions to solve complicated optimization problems. Kiers et al. [36] discusses how to set up a majorizing function for the weighted least squares problem that takes the form of an ordinary least squares problem. Because MM is closed under summation (that is, the sum of majorizing functions is a majorizing function) the problem posed in (5) can be reformulated as a standard SPCA problem. Several procedures have been proposed to solve the SPCA problem, these include both procedures for sparse component weights [12] and for sparse component loadings [11,27,37] (see Ref. [29] for a discussion of the two approaches). All these methods rely on an alternating scheme where **T** and **P** are updated in turn. In our procedure, for the estimation of the sparse loadings **P**, we follow the univariate soft thresholding approach taken by Ref. [27] because this is a closed form and computationally efficient solution to the conditional optimization problem. Furthermore, it allows to solve the problem for a prespecified number of zero-coefficients by adapting the lasso tuning parameter throughout the iterative procedure (see Refs. [26,27] for further details). For the estimation of the component score matrix **T**, we make use of a standard -though not so generally known - matrix algebra result [38] to estimate all components simultaneously ([27] uses a sequential deflation approach and thereby looses control over the orthogonality of the component scores).

The following iterative scheme underlies our algorithm (assuming that **W**, **B**, $R$, and $\lambda$ are known):

1. **T** and **P** are initialized with $\mathbf{T}^{(t)}$ and $\mathbf{P}^{(t)}$, respectively, and with $\mathbf{T}^{(t)}$ subject to the constraint: $(\mathbf{T}^{(t)})^T\mathbf{T}^{(t)} = \mathbf{I}$.
2. Calculate the update $\mathbf{T}^{(t+1)}$ of **T** conditional upon $\mathbf{T}^{(t)}, \mathbf{P}^{(t)}$ under the constraint. Set $\mathbf{T}^{(t)} = \mathbf{T}^{(t+1)}$.
3. Calculate the update $\mathbf{P}^{(t+1)}$ of **P** conditional upon $\mathbf{T}^{(t)}, \mathbf{P}^{(t)}$. Set $\mathbf{P}^{(t)} = \mathbf{P}^{(t+1)}$.
4. Check the stop criteria: If met, terminate, else return to step 2.

The details of the procedure, including a derivation of the parameter estimates, are given in the appendix. Here, we discuss a few important properties of the procedure. A first one is convergence. In each step, it is guaranteed that the loss is non-increasing. Hence, given that the loss is bounded from below by zero, the procedure converges to a fixed point (for suitable starting values, unsuitable values including, e.g., $\mathbf{P} = \mathbf{0}$). Termination of the algorithm is based on two criteria: A first one is convergence of the loss function values (when the difference in loss between subsequent iterations is smaller than some threshold, the procedure terminates) and a second one is a maximum number of iterations. A second important property of the procedure is that it is a local optimization heuristic. Therefore we use a multistart approach: The algorithm is run multiple times with different initializations and the solution with the lowest value of (5) is retained as the final solution.

The algorithm avoids costly operations in terms of computation time or memory. Specifically, because of 1) the separability of the loss function in the variables and 2) the orthogonality of the component scores, the estimation of the loadings becomes a univariate soft thresholding problem [31]. This means that all loadings can be calculated in parallel similar to performing a number of simple regressions (this is regression with only one predictor).

### 2.3. Tuning and model selection

The derivation of the parameter estimates is conditional upon fixed values for the weights **W**, **B**, the number of components $R$, and the lasso tuning parameter $\lambda$. Here we present a model selection strategy for the latter three; the discussion of how to obtain the weights $w_{ij}$ is deferred to the Results section.

Several strategies have been proposed for selecting the number of components, including the use of scree plots [22,39] and cross-validation [40]. Also for the tuning of the lasso parameter, cross-validation is a highly popular model selection tool [31]. Yet, this approach is known to result in too many non-zero coefficients (i.e., it results in a superset of the relevant variables [41]). An alternative that allows to control the false positive error rate is stability selection [32]. Here, we propose to first select the number of components based on a scree plot and to subsequently tune the lasso with stability selection; see Ref. [42] for a similar proposition. Support for a sequential approach which involves the scree test as a first step was recently found by Vervloet et al. [39]: using simulated data they showed that the sequential approach outperformed a simultaneous cross-validation strategy (in a non-sparse setting). Nevertheless, more research into model selection for SPCA methods is needed.

The first step in our model selection approach is to determine the number of components via a scree test. The scree test is based on a visual inspection of the proportion of variance accounted (VAF) for by the components resulting from the non-sparse weighted PCA analysis of the data; we refer to the Results section for the details of the calculation of the VAF by each component.

Given a fixed number of components $R$, a second step is to tune the lasso using stability selection in combination with a randomized lasso (i.e., the elements of **B** are set equal to one or to a constant value $\alpha \in [0.2, 0.8]$, which comes down to either $b_{ij} = 1$ or $b_{ij} = \alpha$ determined in a random fashion; see Ref. [32] for more details). Stability selection is a resampling-based method where different subsamples are generated (by sampling from the observations with replacement) in order to estimate the selection probability of each loading (i.e., the probability that the loading in question is nonzero): Given some level of sparsity, each subsample is analyzed with WSPCA and for each loading the proportion of samples in which it is non-zero is recorded as an estimate of the selection probability. Only loadings with a high selection probability (values between 0.6 and 0.9 are recommended in Ref. [32]) are considered truly non-zero. Meinshausen and Bühlmann [32] propose to tune the level of sparsity by controlling the expected number of false positives (this is, coefficients that are wrongly estimated to be non-zero). This can be done by making use of the following theorem. Let $\pi_{thr}$ denote the set probability threshold (e.g., $\pi_{thr} = 0.9$), $E(V)$ the expected number of false positives, and $q_\Lambda$ the number of coefficients with selection probability equal to or larger than $\pi_{thr}$. Under the assumption of exchangeability and that the selection by the weighted sparse PCA procedure is not worse than random guessing, Meinshausen and Bühlmann [32] have shown that

$$E(V) \leq \frac{1}{2\pi_{thr} - 1} \frac{q_\Lambda^2}{J}, \tag{6}$$

with $J$ denoting the number of variables. From this inequality it follows that for a given value for the probability threshold (e.g., $\pi_{thr} = 0.9$) and upperbound on the expected number of false positives (e.g., $E(V) = 1$)

$$q_\Lambda \leq \sqrt{J(2\pi_{thr} - 1)E(V)}. \tag{7}$$

The latter equation (7) allows to tune λ: λ must be chosen such that the number of coefficients with estimated selection probability exceeding $\pi_{thr}$ is equal to $q_\Lambda$. For further details on how to find this λ we refer to the original publication.

The result of the stability selection procedure, is that the set of nonzero loadings is determined for each component. However, estimates of the actual values of the non-zero loadings are not obtained and, if needed, have to be estimated in an additional analysis that constraints loadings not belonging to the stable set to zero and yields non-zero estimates for the stable set. This can be accomplished, for example, by a non-sparse constrained PCA analysis or a constrained sparse PCA analysis that is tuned such that no additional zeros are introduced. Variables with zero coefficients on all components can be removed from the data to speed up the calculations. In the appendix, we give more detail on how such a constrained analysis can be performed and the accompanying MATLAB code is online available: https://github.com/katrijnvandeun/WSPCA.

## 3. Results and discussion

In this section we will illustrate the added value of the weighted sparse PCA approach in two ways. First, in a simulation study, we show that weighted sparse PCA outperforms non-sparse and/or non-weighted approaches in recovering the loadings for data generated under a model which has both additive and multiplicative noise. Second, in an empirical analysis of transcriptomics data we show that a weighted sparse PCA approach (with weights accounting for differences in measurement quality of the intensities) gives better predictions than a non-weighted sparse PCA approach.

### 3.1. Simulation study

In analytical chemistry and also for microarray gene expression data, a hybrid model containing both additive and multiplicative noise is assumed to underlie the data [16,19]:

$$x_{ij} = \alpha + \mu_{ij} e^{\eta_{ij}} + \varepsilon_{ij} \tag{8}$$

with $\mu_{ij}$ the signal or true score, $\eta_{ij} \sim \mathcal{N}(0,\ \sigma_\eta^2)$ the multiplicative error, and $\varepsilon \sim \mathcal{N}(0,\ \sigma_\varepsilon^2)$ the additive error. Here, we will assume for the offset $\alpha = 0$. With microarray expression data, the convention is to log transform the data so our interest is mainly in $\ln(x_{ij})$. Furthermore, we assume that a PCA model underlies the log transformed data. This means that in absence of noise, $\ln(x_{ij}) = \ln(\mu_{ij}) = \sum_r t_{ir} p_{jr}$ or $x_{ij} = \exp(\sum_r t_{ir} p_{jr})$. Hence, in presence of noise the following model including both additive and multiplicative noise is assumed:

$$x_{ij} = \exp\left(\sum_r t_{ir} p_{jr}\right) e^{\eta_{ij}} + \varepsilon_{ij}. \tag{9}$$

Of crucial importance for the weighted SPCA approach proposed in this paper, are the weights. As noted before, these should be proportional to the reciprocals of the error variance of the log-transformed observed data,

$$\mathrm{var}\left(\ln\left(x_{ij}\right)\right) \approx \frac{\mu_{ij}^2 \sigma_\eta^2 + \sigma_\varepsilon^2}{\mu_{ij}^2}; \tag{10}$$

see Ref. [16] for the derivation of this result. In the simulation study, we will use the model parameters to calculate these weights. With real data this is not possible. One option is to use the expressions derived in Section 3.4 of Rocke and Durbin [19], another may be to calculate the variances over replicate data if these are available.

To obtain some intuitions on the additive-multiplicative error model in (8) for log-transformed data and how the analysis of such data may benefit from a differential weighting, we plotted data with and without noise, both for the transformed and untransformed scores. In Fig. 1 this is done for nine different levels of additive and multiplicative noise. We created a vector of equally spaced values between −3 and 3 representing the log transformed true scores $\ln(\mu_{ij})$; these are taken as the reference values on the x-axis. Against these values, several other scores have been plotted for each of the nine cases. On the one hand this is $\ln(\mu_{ij})$ itself (the red dots on the first bisector), but also the true scores before log transform are shown ($\mu_{ij}$, the green line), in addition to perturbed scores before ($x_{ij}$, the orange dots) and after log transform ($\ln(x_{ij})$, the blue dots). Finally, the black line represents the weights $w_{ij}$ as expressed by (12). The case of data with no noise at all is depicted in the upperleft corner of Fig. 1: Because the true and perturbed scores are the same, some of the lines overlay and the weights are not defined because of the zero variances in (10). The leftmost column contains data with multiplicative noise only while the top row contains data with additive noise only (which corresponds to the homoscedastic case that is implicitly assumed in typical SPCA methods). Of primary interest are the transformed scores (which are the blue dots representing the observed data that will be the input for the SPCA analysis) and the red line representing the true model scores that -ideally- are recovered by the weighted analysis. As long as the blue dots scatter symmetrically around the red line, the unweighted analysis should be able to recover the model scores $\mu_{ij}$. However, when the scatter is no longer symmetric, as in the case with multiplicative noise $= 0.01$ and additive noise $= 0.05$ depicted in the middle row, rightmost column of Fig. 1, there will be bias unless focus is on the part of the data showing symmetric scatter. In this particular panel ($\sigma_\varepsilon^2 = 0.05$ and $\sigma_\eta^2 = 0.01$), this is precisely what the weights suggested by Rocke and Durbin do: The data corresponding to $\ln(\mu_{ij}) < 0$ on the abscissa get a zero weight and this is the region that introduces bias. Using the same kind of reasoning, it can also be seen that the approximation by Rocke and Durbin [19] is problematic when there is no multiplicative noise and only additive noise (middle and rightmost column in the top row). There almost all weight is put on a very limited number of observations; hence, when using the weighted approach, with weights calculated as suggested by Ref. [19], poor performance can be expected in case of homoscedastic noise. This was also pointed out in Ref. [43].

In the simulation study, we manipulated the level of additive and multiplicative noise using the same levels as for the illustrative example, namely 0 (no noise), 0.01, and 0.05. We also manipulated the number of observations units ($I = 100$ and $I = 1000$) and the number of variables ($J = 100$ and $J = 1000$) in addition to the level of sparsity (with two levels: no sparsity or 50% meaning that half of the loadings are set equal to zero). The number of components $R$ was fixed to two. Using a fully crossed design, this leads to $3 \times 3 \times 2 \times 2 \times 2 = 72$ conditions. In each condition we generated ten replicated data sets, so overall we generated 720 data sets.

To generate true scores $\ln(\mu_{ij})$ in accordance with a (sparse) PCA model, we created a matrix of component scores $\mathbf{T}$ and a matrix of sparse loadings $\mathbf{P}$ as follows: An initial matrix of size $I \times J$ was created by sampling from the standard normal distribution. From the singular value decomposition of this matrix, orthonormal component scores $\mathbf{T}$ were obtained by taking the $R$ left singular vectors corresponding to the largest singular values. Loadings were obtained by generating values from a uniform distribution. In the condition with 50% sparsity, half of the loadings, selected in a random way, were set equal to zero yielding $\mathbf{P}$. On average the GINI index was equal to 0.69 for the loading matrices generated with sparseness and 0.36 for those generated without sparseness. True scores were then calculated as $\mu_{ij} = \exp(\sum_r t_{ir} p_{jr})$. To these multiplicative and/or additive noise was added under model equation (8). The final step was to take the log transform. Yet, in case of relatively large additive noise, negative $x_{ij}$ may result in a number of cases for which the log transform is not

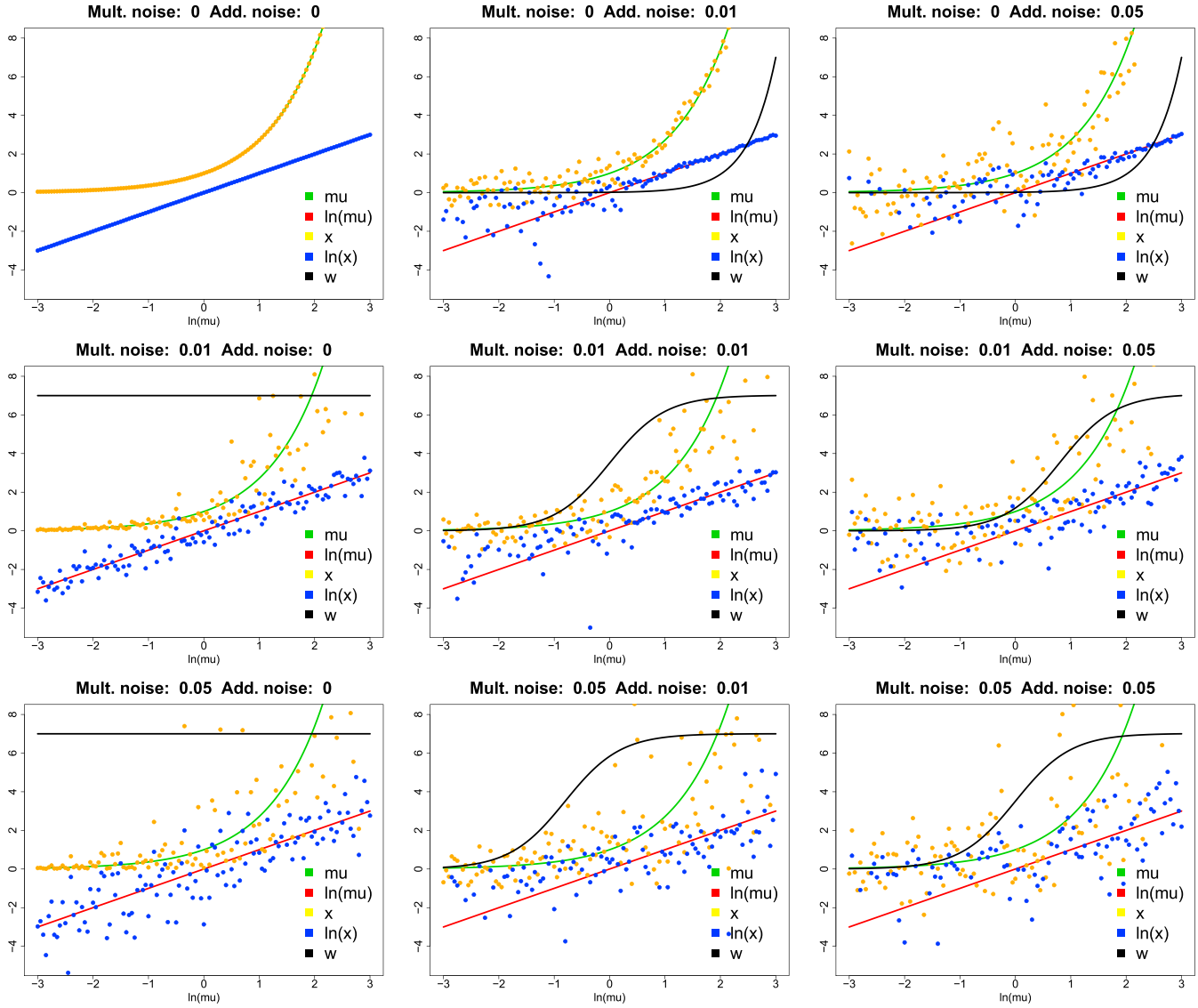



**Fig. 1.** Multiplicative-additive error model: Data were generated under a model that is linear in the log-transformed domain and that contains both multiplicative and additive error. The true and observed scores in the untransformed and transformed domain are plotted against the log transformed true scores.

defined. Such cases were treated as missing data, this is with $w_{ij} = 0$. Weights as defined by Rocke and Durbin were obtained by calculating

$$w_{ij} = \left[\mathrm{var}\left(\ln\left(x_{ij}\right)\right)\right]^{-1} \tag{11}$$

$$= \frac{\mu_{ij}^2}{\mu_{ij}^2\sigma_\eta^2 + \sigma_\varepsilon^2}. \tag{12}$$

Note that in those cases without additive noise (i.e., $\sigma_\varepsilon^2 = 0$) all $w_{ij} = (\sigma_\eta^2)^{-1}$, meaning that all weights are equal and, hence, that the weighted PCA approach is equivalent to an unweighted approach. In case of no noise at all (both $\sigma_\eta^2 = 0$ and $\sigma_\varepsilon^2 = 0$) the weights are not defined. In this condition of the simulation study, we set all weights equal to one. When there is no multiplicative noise, $\sigma_\eta^2 = 0$, then $w_{ij} = (\sigma_\varepsilon^2)^{-1}\mu_{ij}^2$, which means that more weight is given to data with higher true scores.

To check the added value of imposing sparseness and weighting, four different analyses were performed to cover all possible combinations (weighted vs. unweighted combined with sparse vs. non-sparse). Thus each of the data sets was analyzed with classical PCA, weighted PCA, sparse PCA (using the PMA R package [11]) and with our newly developed weighted sparse PCA method. For the weighted approaches we used the weights defined by (12) making use of the true values of $\mu_{ij}$ and the error variances $\sigma_\varepsilon^2$ and $\sigma_\eta^2$. Note that each of the four methods used could deal with missing data; for the weighted approaches this was possible by setting $w_{ij} = 0$; PMA allows for missing values in the data matrix by excluding them from the computations (see Ref. [11]), and classical PCA was performed using our own implementation of weighted sparse PCA (setting $\lambda = 0$, i.e., no sparseness, and $w_{ij} = 0$ for the missing values and $w_{ij} = 1$ elsewhere). Both sparse analyses were run using information on the true number of non-zero loadings. As the data are generated with orthogonal $\mathbf{T}$, the PMA package was run with the option of orthogonal component scores.

The performance of the methods was assessed at two levels: first, at the level of the loadings to assess configurational similarity, and, second, at the level of the reconstructed scores to measure fit. First, the recovery of the loadings was assessed by calculating Tucker's coefficient of

congruence [44]. Let $\widehat{p}_{jr}$ be the estimated loading of variable $j$ on component $r$, then Tucker's congruence between the estimated and true loadings for component $r$ is

$$\varphi_r = \frac{\sum_j p_{jr}\widehat{p}_{jr}}{\sum_j p_{jr}^2 \sum_j \widehat{p}_{jr}^2}, \tag{13}$$

with $\varphi_r = 1$ indicating perfect congruence and values closer to zero indicating lower congruence. We calculated the average congruence over the two components, taking into account that sparse PCA is invariant under reflection and permutation; this means that in case of a negative congruence value the estimated loadings were reflected and that φ was calculated as the maximum over all possible permutations of the components. PCA and WPCA have rotational freedom; this was taken into account by rotating towards the true loading matrix. Note that rotational freedom - in general - does not apply to sparse PCA techniques as optimality is defined by a penalized loss and rotation will result in higher values of the penalty term. MATLAB code implementing this procedure is available online:https://github.com/katrijnvandeun/WSPCA. Fig. 2 displays boxplots of the congruence coefficients obtained with the four different PCA analyses in the conditions with 50% of sparseness in the generated loadings, $I = 100$, and $J = 1000$ (this setting corresponds to the case of high-dimensional data; plots for the other values of $I$ and $J$ can be found online: https://github.com/katrijnvandeun/WSPCA). The leftmost panel ($MULT = 0$) represents the conditions where there is no multiplicative noise. Within that panel, the results at the left were obtained in the condition with no noise at all (this is, also no additive noise $ADD = 0$) while the results in the middle and right subpanel were obtained for the conditions with additive noise. In absence of any noise, all methods should perform extremely well. The non-sparse PCA methods and WSPCA indeed have almost perfect congruence (i.e., $\varphi = 1$). SPCA, on the other hand, has low congruence in this noise-free condition. The reason for this is the higher sensitivity of the SPCA implementation by Witten et al. [11] to local optima: WSPCA relies on a multistart procedure including one rational start (namely initializing with the singular vectors) and ten random starts while the SPC function in the PMA R package of Witten et al. [11] relies on a single SVD-based rational start. When there is no multiplicative noise but some additive noise, WSPCA does not perform as well as the other methods as could be expected (see the top row of Fig. 1). When there is no additive noise but only multiplicative noise (the subpanels at the left in the middle panel $MULT = 0.01$ and right panel $MULT = 0.05$), WSPCA performs equally well as PCA and WPCA as it correctly imposes sparseness and outperforms SPCA because it uses a multistart procedure. Note that PCA and WPCA have been rotated to the true structure while WSPCA does not rely on information on the true structure. In practical data analysis situations, such information is not available: without the additional rotation, PCA and WPCA perform worse than WSPCA (see Fig. 3). When there is both additive and multiplicative noise, WSPCA clearly outperforms the other methods as it is able to downweight the observations that introduce bias. Summarizing, in the majority of the conditions, weighted sparse PCA outperforms the three other methods in recovering the true loadings.

A second measure of performance that we assessed, is the deviation between the true data scores $\ln(\mu_{ij}) = \sum_r t_{ir}p_{jr}$ and the fitted data $\widehat{x}_{ij} = \sum_r \widehat{t}_{ir}\widehat{p}_{jr}$, expressed by the *badness-of-recovery* (BOR) statistic,

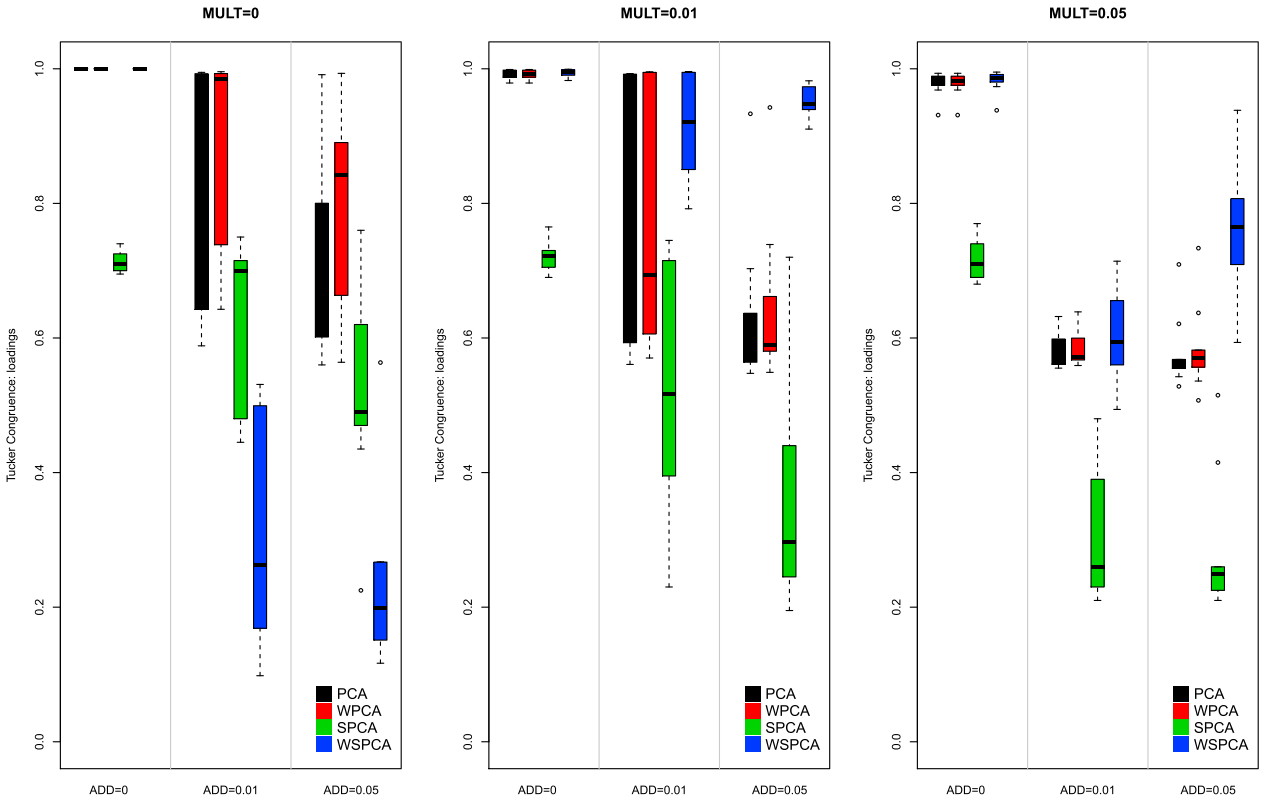



**Fig. 2.** Tucker congruence of estimated loadings with true loadings for the conditions with sparseness in the model, $I = 100$, and $J = 1000$. PCA and WPCA have been rotated towards the true loading matrix. The left panel displays the results obtained in the condition with no multiplicative noise (MULT = 0), the middle with an intermediate level of multiplicative noise, and the right panel with the high level of multiplicative noise. Within each panel, the additive noise level is varied from no additive noise (ADD = 0) to the highest level (ADD = 0.05).

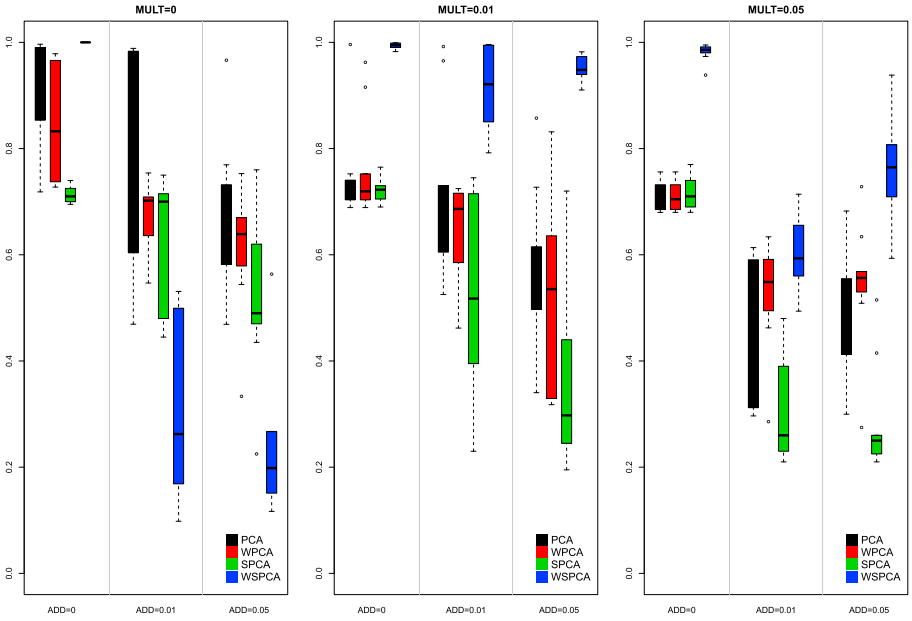



**Fig. 3.** Tucker congruence of estimated loadings with true loadings for the conditions with sparseness in the model, $I = 100$, and $J = 1000$. The left panel displays the results obtained in the condition with no multiplicative noise (MULT = 0), the middle with an intermediate level of multiplicative noise, and the right panel with the high level of multiplicative noise. Within each panel, the additive noise level is varied from no additive noise (ADD = 0) to the highest level (ADD = 0.05).

$$BOR = \frac{\sum_{i,j}\left(\ln\left(\mu_{ij}\right) - \widehat{x}_{ij}\right)^2}{\sum_{i,j}\left[\ln\left(\mu_{ij}\right)\right]^2}. \tag{14}$$

When $BOR = 0$ this means that the data are perfectly fitted by the model, with higher values meaning worse fit. The log transformed values of the *BOR* statistics plus one are summarized by the boxplots in Fig. 4;

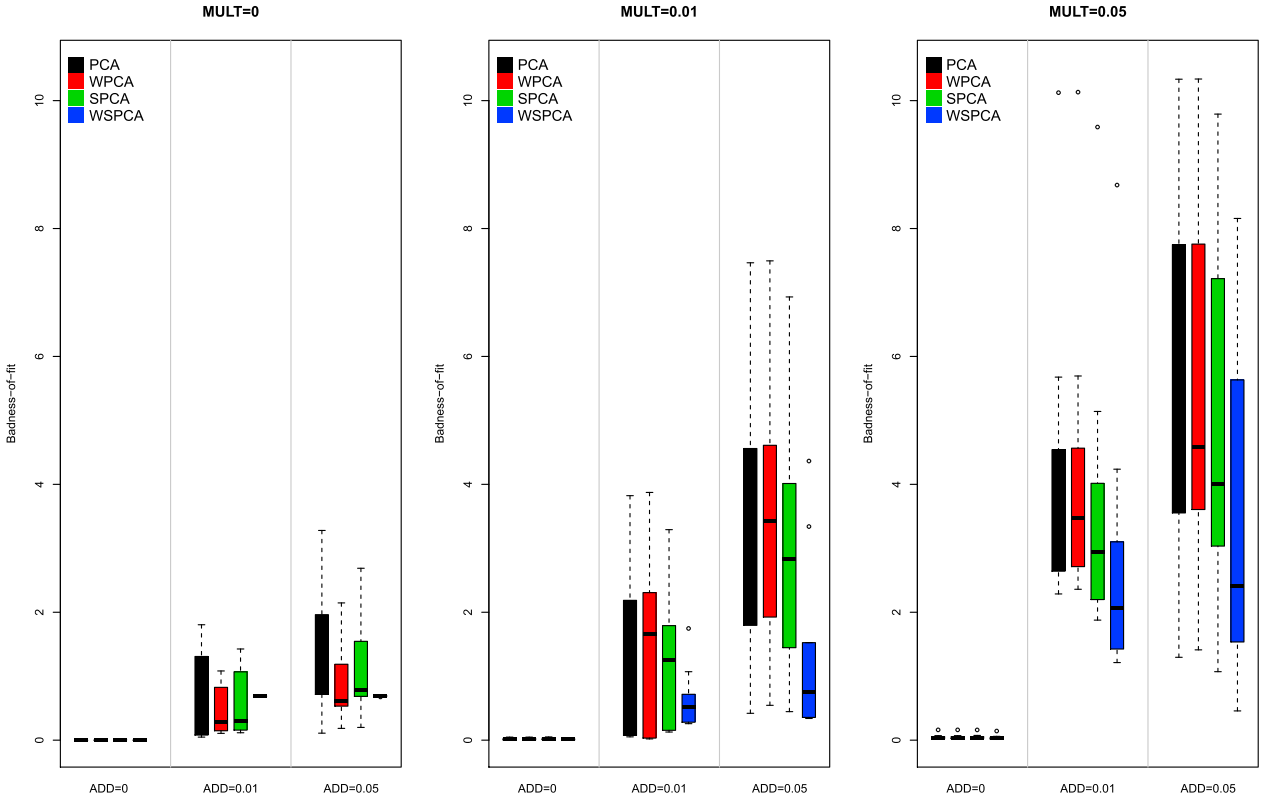



**Fig. 4.** Badness-of-fit of the estimated data to the true data for the case with sparseness, $I = 100$, and $J = 1000$. The left panel displays the results obtained in the condition with no multiplicative noise (MULT = 0), the middle with an intermediate level of multiplicative noise, and the right panel with the high level of multiplicative noise. Within each panel, the additive noise level is varied from no additive noise (ADD = 0) to the highest level (ADD = 0.05).

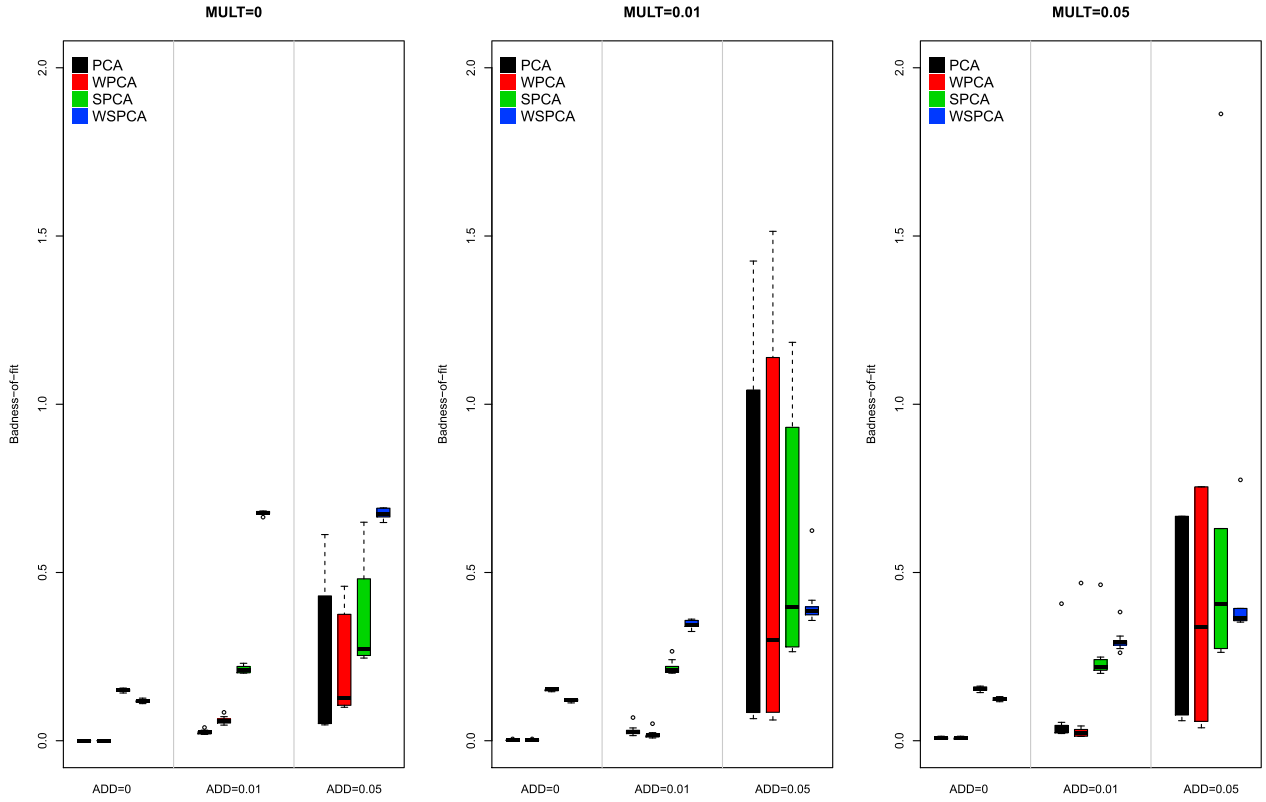



**Fig. 5.** Badness-of-fit for the case without sparseness, $I = 100$, and $J = 1000$. The left panel displays the results obtained in the condition with no multiplicative noise (MULT = 0), the middle with an intermediate level of multiplicative noise, and the right panel with the high level of multiplicative noise. Within each panel, the additive noise level is varied from no additive noise (ADD = 0) to the highest level (ADD = 0.05).

also in this case a zero means perfect fit. The reason for displaying the log-transformed values is the occurrence of extreme outliers of the BOR statistic in (14). The boxplots confirm the results observed for the recovery of the loadings: In presence of hybrid (i.e., mixed additive-multiplicative) noise WSPCA outperforms the other methods, when there is no noise at all, all methods have (close to) perfect fit, and when there is only additive noise WSPCA does not outperform the other methods. WSPCA outperforms SPCA in the conditions with both additive and multiplicative noise because this is a situation that results in biased estimates when ordinary (penalized) least squares is used (as previously discussed in the introduction). It may seem surprising that WSPCA also outperforms WPCA - which is a more flexible model - in these conditions with hybrid noise. This is because here performance is measured as the deviation from the true scores (and *not* from the observed data) with worse performance of WPCA because of overfitting. Note that (not visible in the Figure) in the conditions without any noise, perfect fit is only obtained for the non-sparse methods while SPCA and WSPCA have BOR values that differ -although barely-from zero. This is due to the penalty in the optimization criterion resulting in lower values of the optimization criterion (5) for solutions that do not have perfect fit. For this optimization criterion, as discussed above in relation to the presence of local optima, WSPCA has a lower value than SPCA.

Fig. 5 displays boxplots of the *badness-of-fit* for the four different PCA analyses in the conditions with data generated without sparseness, $I = 100$, and $J = 1000$ (this setting corresponds to the case of high-dimensional data; plots for the other values of $I$ and $J$ can be found online: https://github.com/katrijnvandeun/WSPCA). Note that the sparse PCA methods were tuned to yield zeros for 50% of the loadings although the data were generated with non-sparse loading matrices. Hence, a reasonable expectation is that PCA and WPCA outperform SPCA and WSPCA.[1] Furthermore, WPCA may be expected to perform better than all other methods when there is both additive and multiplicative noise present in the data. The *BOR* statistics in Fig. 5 support this except for the condition with large additive noise ($ADD = 0.05$) and medium multiplicative noise ($MULT = 0.01$). In this condition PCA seems to perform slightly better than WPCA. Note that when there is no multiplicative noise, PCA outperforms WPCA. This confirms the observation made before (see the discussion of Fig. 1).

### 3.2. Empirical data

#### 3.2.1. Description of the data

Nakaya et al. studied the response to vaccination against influenza at several levels of the cellular organization. Microarray gene expression data and antibody titers are publicly available for two seasons (see the GSE29617 and GSE29614 series in the Gene Expression Omnibus at http://www.ncbi.nlm.nih.gov/geo/). The titers are included as a dependent variable that we will try to predict on the basis of the component scores. Here we concentrate mainly on the data resulting from the collection of peripheral blood mononuclear cells (PBMC) three days after vaccination in 26 subjects[2] vaccinated with trivalent inactivated influenza vaccine (TIV) in the year 2008; the data from the 2007 season will

[1] In this case, we do not inspect tucker congruence because the measure is not invariant under rotation while PCA and WPCA have rotational freedom.

[2] These are the 26 subjects with micro-array expression data at all time points (day 0, day 3, and day 7).

be used for external validation, yet, given their very limited sample size (9 subjects only), will not be used to build the weighted sparse PCA model. Hybridization was to the Affymetrix Human Genome U133 Plus 2.0 and U133 + PM arrays for the 2007 and 2008 seasons, respectively; these arrays share 54 675 probesets which form the variables in our analyses. Note that the data are ultra-high-dimensional (over 50 000 variables for only 26 observations).

### 3.2.2. Data preprocessing

To obtain the expression values, we used the affyPLM package [45] with default settings yielding Robust Multichip Average (RMA) values. The baseline corrected values were obtained by subtracting the RMA values at baseline from the RMA values three days after vaccination. This pre-processing procedure was applied both to the 2007 and 2008 data. Weights were obtained as follows: From the GESTr R package [46], estimates of the variance components were obtained; the true expression scores $\mu_{ij}$ in equation (8) were estimated by the observed expression scores. In Fig. 6, a histogram of the weights is shown for the 2008 season: almost all weights lie in the range $[0.6 - 1.0]$. Hence, we do not expect the results of the weighted analysis to differ a lot from those of the unweighted analysis.

Plasma hemagglutination-inhibition (HAI) antibody titers were obtained as described in the original paper [24]. First, we calculated the difference in antibody titers at day 28 with the titers at baseline for each of the three influenza strains that compose TIV. The maximal difference was retained and this score was log 2 transformed.

### 3.2.3. PCA analyses

To determine the number of components we performed a weighted PCA analysis of the data and plotted the variance accounted for (VAF) by each component in Fig. 7; following the recommendations by Willett and Singer [47], this is

$$VAFr = 1 - \frac{\sum_{ij}\left(x_{ij} - \hat{t}_{ir}\hat{p}_{jr}\right)^2}{\sum_{ij}x_{ij}^2}. \quad (15)$$

Note that with this criterion, the VAF accounted for by the first $R$ components is equal to the sum of the VAF by each of the $r = 1, \ldots, R$ components. The first three components stand out compared to the remaining components. Hence, we select three components for further analysis. We also performed an unweighted analysis; this resulted in *VAFr* values that only differ in the fourth decimal from those obtained with the weighted PCA. Next, we determine the level of sparsity using

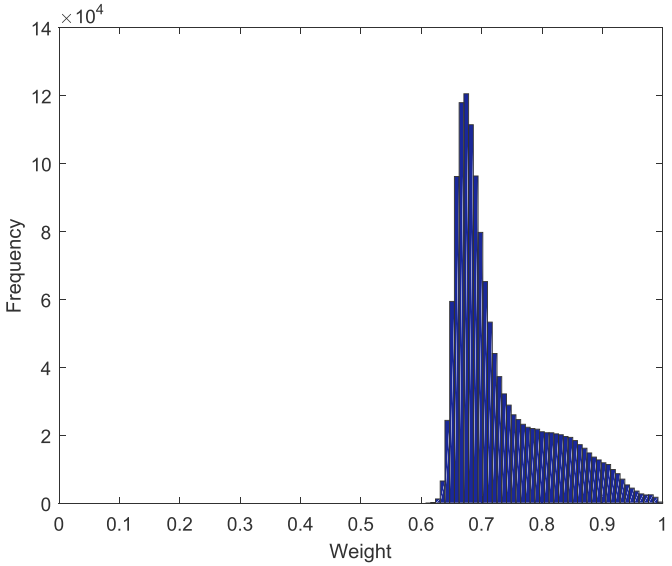



**Fig. 6.** Histogram of weights used in the weighted PCA analysis.

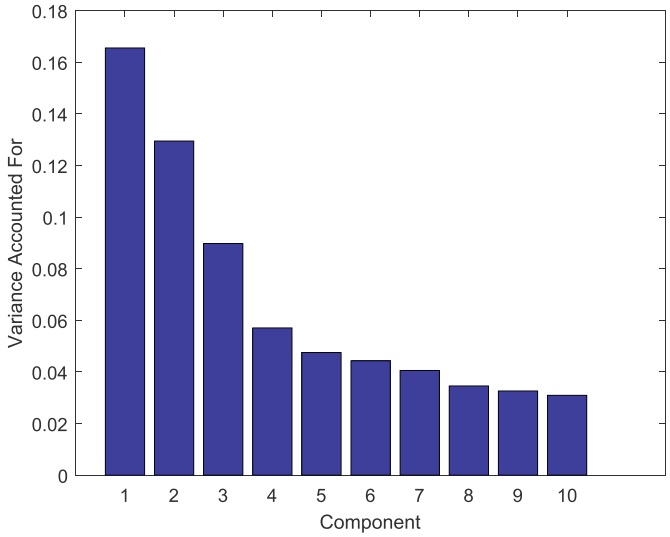



**Fig. 7.** Scree plot: The bars display the variance accounted for by the first ten components.

expression (7): Taking the default values $E(V) = 1$ and $\pi_{thr} = 0.90$ (see Meinshausen and Bühlmann [32]) yields $q_\lambda = 209$ non-zero coefficients for a single component or 627 in total. We ran two different types of SPCA analyses: an unweighted SPCA analysis using the PMA package and a weighted sparse PCA analysis using the WSPCA method that was proposed in the present paper. Both analyses were tuned such that 627 non-zero loadings over the three components were obtained. For the PMA analysis a single rational start based on the singular value decomposition was used; for the stability selection part of the WSPCA analysis, 250 subsamples were used for each level of sparsity with the level of sparsity being gradually decreased until the desired number of 627 non-zero loadings was attained. Each of the subsamples was subjected to a WSPCA with a single rational start.

The performance of the unweighted and weighted SPCA in terms of how well their component scores fit the antibody titers is summarized in Table 1. First we trained the model by obtaining loadings and component scores for the 2008 data as well as regression weights (by regressing the antibody titers for the vaccinations in 2008 on the component scores derived from the transcriptomics data for 2008). These model parameters were then used for the prediction of the 2007 antibody titers (the independent test set): Component scores were derived from the 2007 data using the loadings obtained for the 2008 data; these components scores where then used as the predictor scores in the regression model with intercept and regression weights as obtained for the 2008 data. As shown in Table 1, the squared correlation between the observed and fitted antibody titers for the training set were 0.15 and 0.20 for the unweighted and weighted solutions respectively. For the 2007 test set, the squared correlation between predicted and observed scores obtained for the unweighted and weighted SPCA analysis were 0.26 and 0.43 respectively. Hence, although the same number of components and level of sparsity was used and the same type of initial configuration, the components resulting from our weighted SPCA procedure had better predictive quality than those obtained from the unweighted SPCA analysis

**Table 1**
Fit of modeled to observed data for the two unweighted and weighted SPCA analyses. Displayed are the squared correlations between the modeled and observed data for the 2008 season and between predicted and observed outcome for the 2007 season. The model was constructed using the 2008 data.

| Method | $r(\hat{y}_{2008}, y_{2008})^2$ | $r(\hat{y}_{2007}, y_{2007})^2$ |
|---|---|---|
| unweighted | 0.15 | 0.26 |
| weighted | 0.20 | 0.43 |

performed with the PMA package. Different variables were selected by the two procedures which is likely the result of weighting.

The results of the WSPCA analysis with stability selection were further checked for a biologically meaningful selection. A list was generated containing the 627 probesets with non-zero loadings. Probesets not annotated to gene symbols were eliminated and those mapping to the same gene were summarized using the median score. A hypergeometric test was used to analyse the enrichment in blood transcriptional module (BTM), using a false discovery rate (FDR) threshold of 0.001 (a high FDR threshold is suggested for the use of hypergeometric tests in module enrichments [48]). The BTM are sets of manually annotated correlated gene sets which were obtained from meta-analysis of over 500 gene expression studies obtained from human blood. These are widely used for the interpretation of human blood gene expression data as they are more tissue-adapted compared to general pathways and are generally regarded as well annotated. The enrichment analysis was performed using the tmod R package [49]. The script for the analysis is available from https://github.com/katrijnvandeun/WSPCA.

The enriched modules are presented in Fig. 8. Overall, the genes identified by WSPCA are mainly enriched in pathways of the subcompartment of the immune system responsible for the first rapid response to pathogens (innate immunity), as expected for gene expression data collected soon after vaccination and as originally reported. Mainly, the genes identified were found to be enriched in innate immune cells such as dendritic cells, monocytes and neutrophils, and in essential functions of the innate immune system such as the recognition of pathogens via pattern recognition receptors and the induction of interferon responses. Additionally, the genes were also enriched in functions associated with cell cycle and transcription, which in addition to being general processes involved in both active responses to interventions and housekeeping activities, also underlie the immunological activities referred to above. Overall, the profile presented here is representative of the early transcriptional response to vaccination.

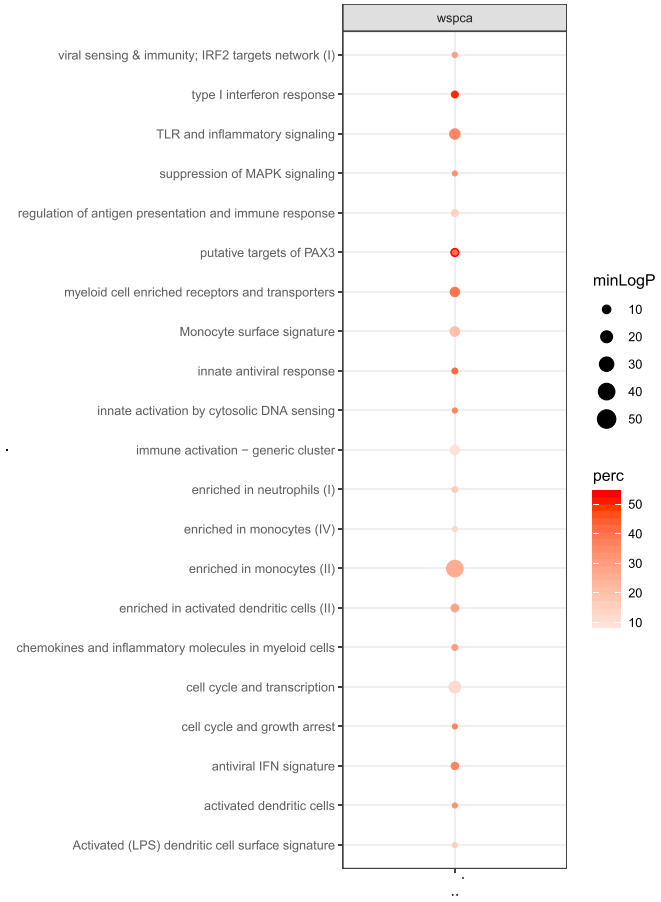



**Fig. 8.** Modules enriched in the genes selected with WSPCA. Each row represents a blood transcriptional module (BTM). The size of the bubbles is proportional to the inverse of the log 10 FDR for the enrichment. The colour of the bubble is more intense when the proportion of genes in the BTM present in the input gene list is high. (For interpretation of the references to colour in this figure legend, the reader is referred to the Web version of this article.)

### 3.3. Discussion of the results

In this section we assessed the performance of weighted sparse PCA compared to unweighted and/or nonsparse PCA under various conditions in a simulation study and for real data in an empirical application. The main factors of interest in the simulation study were presence versus absence of additive and multiplicative noise and of sparseness. When both additive and multiplicative noise are present, weighted PCA clearly outperforms unweighted PCA. When only multiplicative noise is present, the weighted and unweighted approaches perform equally well. However, when only additive noise is present, unweighted PCA outperforms weighted PCA. Hence, in this situation we advise against using a weighted sparse PCA approach. In analytical chemistry usually both additive and multiplicative noise are present but this should in general be checked, for example, by means of the maximum likelihood estimates of the additive and multiplicative error variance parameters [16]. With respect to sparseness, the results obtained in the simulation study show that it is important to impose the correct amount of sparseness: The sparse PCA methods outperformed the nonsparse methods when there was sparseness in the loadings while the nonsparse methods outperformed the sparse methods in case of no sparseness. Various model selection tools have been proposed to tune the proper amount of sparseness in sparse PCA, including the case of no sparseness at all. These include cross-validation [12], stability selection [32], and the BIC [50].

## 4. Conclusion

In this paper, we have introduced an approach to sparse PCA that allows to weight each of the elements of the data matrix individually. Such a weighted sparse approach was illustrated to be useful in terms of an improved recovery of the loadings in a simulation study and an improved prediction of antibody titers on the basis of components derived from a weighted sparse PCA analysis of early gene signatures from a study of vaccination against influenza. Overall the simulation results support the use of a weighted sparse PCA approach when the data are subject to both additive and multiplicative noise.

To account for dependence, for example between observations related in time, the weighted sparse PCA method may be extended in a similar way as presented in Ref. [51]. Another useful extension may be the development of an iteratively reweighted least squares procedure, as such a procedure does not rely on prior knowledge for the weighting. The algorithm presented here already contains the key ingredients to implement such an extension.

## Author contributions

KVD developed the WSPCA procedure and performed the analyses. MC was involved in the analysis conception and design; All authors were involved in drafting the manuscript or critically revising it for important intellectual content. All authors had full access to the data and approved the manuscript before it was submitted by the corresponding author.

## Declaration of competing interest

MC is an employee of the GSK group of companies and reports owning shares in GSK.

**Acknowledgements**

KVD was funded by the Netherlands Organisation for Scientific Research: NWO-VIDI 452.16.012. MC received funding from the Bill & Melinda Gates Foundation grant OPP1220977.

## Appendix A. Derivation of an algorithm for weighted L1 penalized PCA

The objective of the WSPCA method is to minimize the function

$$\left|\left|\mathbf{W} \circ (\mathbf{X} - \mathbf{T}\mathbf{P}^T)\right|\right|^2 + \lambda\left|\mathbf{B} \circ \mathbf{P}\right|_1 \tag{A.1}$$

with respect to $\mathbf{T}, \mathbf{P}$ and such that $\mathbf{T}^T\mathbf{T} = \mathbf{I}$; the number of components $R$ and the value of the tuning parameter $\lambda$ are assumed to be given. Also, all weights are assumed to be nonnegative: $w_{ij} \geq 0$ for all $i,j$. Here, we will show how to solve the weighted and penalized least squares problem by making use of an alternating procedure in which each of the structures ($\mathbf{T}$ and $\mathbf{P}$) is updated in turn conditional upon fixed values for the other structure.

### *Appendix A.1. . A majorizing function for weighted least-squares*

The weighted least-squares problem is not a standard problem. However, the problem can be solved by a majorize minimize or iterative majorization procedure [35,36]. The weighted least-squares problem (the left term in (A.1)) is replaced by a standard least-squares problem of a particular form [36, 52]:

$$||\mathbf{W} \circ (\mathbf{Y} - \mathbf{M})||^2 \leq k + w_m^2||\mathbf{Y}^* - \mathbf{M}||^2 \tag{A.2}$$

with $k$ a term that is constant with respect to the optimization problem (i.e., $k$ summarizes the terms that do not include parameters that have to be estimated), $w_m$ the largest value of $\mathbf{W}$ and $\mathbf{Y}^* = \mathbf{M}^{(0)} + w_m^{(-2)}(\mathbf{W} \circ \mathbf{W} \circ (\mathbf{Y} - \mathbf{M}^{(0)}))$. Here, $\mathbf{Y}$ represents the fixed part and $\mathbf{M}$ the part that has to be estimated (with $\mathbf{M}^{(0)}$ denoting the current estimate). This results in the following majorizing function:

$$\left|\left|\mathbf{W} \circ (\mathbf{X} - \mathbf{T}\mathbf{P}^T)\right|\right|^2 \leq k + w_m^2\left|\left|\mathbf{Y}_2^* - \mathbf{T}\mathbf{P}^T\right|\right|^2 \tag{A.3}$$

with $\mathbf{Y}_2^* = \mathbf{T}^{(0)}(\mathbf{P}^{(0)})^T + w_m^{(-2)}(\mathbf{W} \circ \mathbf{W} \circ (\mathbf{X} - \mathbf{T}^{(0)}(\mathbf{P}^{(0)})^T))$.

### *Appendix A.2. . Update of the component scores in the unweighted majorizing function*

The update of the component scores $\mathbf{T}$ with restriction $\mathbf{T}^T\mathbf{T} = \mathbf{I}$ conditional upon fixed values for $\mathbf{P}$ is found by minimizing

$$\begin{aligned} L(\mathbf{T}) = &\left|\left|\mathbf{W} \circ \left(\mathbf{X} - \mathbf{T}\left(\mathbf{P}^{(0)}\right)^T\right)\right|\right|^2 + \\ &\lambda\left|\mathbf{B} \circ \mathbf{P}^{(0)}\right|_1, \end{aligned} \tag{A.4}$$

which can be majorized by (A.3):

$$L(\mathbf{T}) \leq k_1 + w_m^2\left|\left|\mathbf{Y}_2^* - \mathbf{T}\left(\mathbf{P}^{(0)}\right)^T\right|\right|^2, \tag{A.5}$$

with $k_1 = k + \lambda\left|\mathbf{P}^{(0)}\right|_1$. Hence, a known regression problem is obtained with solution $\mathbf{T} = \mathbf{V}\mathbf{U}^T$ with $\mathbf{U}$ and $\mathbf{U}$ from the SVD of $\mathbf{P}^{(0)T}\mathbf{Y}_2^{*T} = \mathbf{U}\mathbf{S}\mathbf{V}^T$ (see, for example, [38].

### *Appendix A.3. . Derivation of the update* $\mathbf{P}^+$

An update of the loadings is derived conditional upon fixed values for $\mathbf{T}$. The weighted least squares function

$$\begin{aligned} L(\mathbf{P}) = &\left|\left|\mathbf{W} \circ \left(\mathbf{X} - \mathbf{T}^{(0)}\mathbf{P}^T\right)\right|\right|^2 \\ &+\lambda\left|\mathbf{B} \circ \mathbf{P}\right|_1, \end{aligned} \tag{A.6}$$

can be separated for each of the variables,

$$L(\mathbf{P}) = \sum_j \left\{\left|\left|\mathbf{w}_j \circ \left(\mathbf{x}_j - \mathbf{T}^{(0)}\mathbf{p}_j^T\right)\right|\right|^2 + \lambda\left|\mathbf{b}_j \circ \mathbf{p}_j\right|_1\right\}. \tag{A.7}$$

This implies that the loss function can be optimized by optimizing for each variable $j$,

$$\begin{aligned} L(\mathbf{p}_j) &= \left\| \mathbf{w}_j \circ \left( \mathbf{x}_j - \mathbf{T}^{(0)} \mathbf{p}_j^T \right) \right\|^2 + \lambda \left| \mathbf{b}_j \circ \mathbf{p}_j \right|_1 \\ &= \sum_i \left( w_{ij} x_{ij} - \sum_r w_{ij} t_{ir} p_{jr} \right)^2 \\ &+ \sum_r \lambda b_{jr} \left| p_{jr} \right| \\ &= \sum_i \left( x_{ij}^* - \sum_r t_{ir}^* p_{jr} \right)^2 + \sum_r \lambda_r^* \left| p_{jr} \right|, \end{aligned} \tag{A.8}$$

with $x_{ij}^* = w_{ij} x_{ij}$, $t_{ir}^* = w_{ij} t_{ir}$, and $\lambda_r^* = \lambda b_{jr}$. The latter objective function is in the form of a lasso regression problem with $R$ uncorrelated predictors. This can be solved by univariate soft thresholding [53,54], this is calculating the update for each $p_{jr}$ ($j = 1 \ldots J$, $r = 1 \ldots R$):

$$\begin{aligned} L(p_{jr^*}) &= \sum_i \left( x_{ij}^* - \sum_{r \neq r^*} t_{ir}^* p_{jr} - t_{ir^*}^* p_{jr^*} \right)^2 \\ &+ \sum_{r \neq r^*} \lambda_r^* \left| p_{jr} \right| + \lambda_{r^*}^* \left| p_{jr^*} \right| \\ &= \sum_i \left( e_{ij}^{r^*} - t_{ir^*}^* p_{jr^*} \right)^2 + \sum_{r \neq r^*} \lambda_r^* \left| p_{jr} \right| + \lambda_{r^*}^* \left| p_{jr^*} \right|, \end{aligned} \tag{A.9}$$

which is minimized by the soft thresholding operator

$$p_{jr}^+ = S\left( \sum_i x_{ij}^* t_{ir^*}^*, \frac{1}{2} \lambda_{r^*} \right), \tag{A.10}$$

with

$$S(p_{jr}, \lambda) = \begin{cases} p_{jr}^+ = p_{jr} + \lambda, & \text{if } p_{jr} < 0 \text{ and } \left| p_{jr} \right| > \lambda \\ p_{jr}^+ = p_{jr} - \lambda, & \text{if } p_{jr} > 0 \text{ and } \left| p_{jr} \right| > \lambda \\ p_{jr}^+ = 0, & \text{if } \left| p_{jr} \right| < \lambda, \end{cases} \tag{A.11}$$

where we use $p_{jr} = \sum_i x_{ij}^* t_{ir^*}^*$ to denote the estimate that would be obtained in the ordinary least squares case (this is, without the $L1$ penalty). Because the update can be calculated for all $p_{jr}^+$ simultaneously using simple matrix operations, this yields a very efficient procedure also for the multivariate PCA problem we are dealing with here; note, in particular, that we treat the problem as a regression problem with $R$ independent predictors [27]. proposed to adaptively tune λ throughout the iterations such that a given number of $j^* \leq J$ non-zero coefficients is obtained. This option is included in the MATLAB implementation that is available online; see also Appendix B.

### *Appendix A.4. . Constrained optimization of* **T** *or* **P**

The expression (A.10) allows for a targeted estimation in which only some coefficients are updated while the other coefficients have fixed values. In this way, loadings or scores may be constrained to a particular value, for example to zero.

## Appendix B. Derivation of an algorithm for weighted cardinality constrained PCA

The sparse PCA problem as presented here, corresponds to the problem proposed by Ref. [37]. In particular, as the solution to the estimation problem of the loadings for the majorizing function can be written in the form of a regression problem with independent predictors for each variable, the following cardinality constrained problem can be solved,

$$\min_{\mathbf{T},\mathbf{P}} \left|\left| \mathbf{W} \circ (\mathbf{X} - \mathbf{T}\mathbf{P}^T) \right|\right|^2, \tag{B.1}$$

such that $\mathbf{T}^T\mathbf{T} = \mathbf{I}$ and $Card(\mathbf{P}) = j^*$ with $0 \leq j^* \leq JR$. The cardinality constraint $Card(\mathbf{P}) = j^*$ means that $\mathbf{P}$ is subject to having exactly $j^*$ non-zero values (over the $R$ components). As shown above, the weighted least squares problem can be solved by solving the surrogate criterion

$$w_m^2 \left\| \mathbf{Y}^* - \mathbf{T}\mathbf{P}^T \right\|^2, \tag{B.2}$$

subject to the constraint $Card(\mathbf{P}) = j^*$.

## Appendix C. Implementation

See online: https://github.com/katrijnvandeun/WSPCA.

## Appendix D. Supplementary data

Supplementary data to this article can be found online at https://doi.org/10.1016/j.chemolab.2019.103875.

## References


[1] H.-J. He, D. Wu, D.-W. Sun, Nondestructive spectroscopic and imaging techniques for quality evaluation and assessment of fish and fish products, Crit. Rev. Food Sci. Nutr. 55 (2015) 864–886.

[2] M.M. Khamis, D.J. Adamko, A. El-Aneed, Mass spectrometric based approaches in urine metabolomics and biomarker discovery, Mass Spectrom. Rev. 36 (2017) 115–134.

[3] B. Mwangi, T.S. Tian, J.C. Soares, A review of feature reduction techniques in neuroimaging, Neuroinformatics 12 (2014) 229–244.

[4] T.K. Karakach, R.M. Flight, S.E. Douglas, P.D. Wentzell, An introduction to dna microarrays for gene expression analysis, Chemometr. Intell. Lab. Syst. 104 (2010) 28–52 (OMICS).

[5] P. Buhlmann, S. van de Geer, Statistics for High-Dimensional Data: Methods, Theory and Applications, first ed., Springer Publishing Company, Incorporated, 2011.

[6] B.-H. Mevik, R. Wehrens, The pls package: principal component and partial least squares regression in r, Journal of Statistical Software, Articles 18 (2007) 1–23.

[7] M. Rasmussen, R. Bro, A tutorial on the lasso approach to sparse modeling, Chemometr. Intell. Lab. Syst. 119 (2012) 21–31.

[8] T. Friedlander, A. Mayo, T. Tlusty, U. Alon, Mutation rules and the evolution of sparseness and modularity in biological systems, PLoS One 10 (3) (2013), e0118129.

[9] I.M. Johnstone, A.Y. Lu, On consistency and sparsity for principal components analysis in high dimensions, J. Am. Stat. Assoc. 104 (2009) 682–703.

[10] J. Cadima, I. Jolliffe, Loadings and correlations in the interpretation of principal components, J. Appl. Stat. 22 (1995) 203–214.

[11] D.M. Witten, R. Tibshirani, T. Hastie, A penalized matrix decomposition, with applications to sparse principal components and canonical correlation analysis, Biostatistics 10 (2009) 515–534.

[12] H. Zou, T. Hastie, R. Tibshirani, Sparse principal component analysis, J. Comput. Graph. Stat. 15 (2006) 265–286.

[13] T.T. Cai, Z. Ma, Y. Wu, Sparse pca: optimal rates and adaptive estimation, Ann. Stat. 41 (2013) 3074–3110.

[14] G.I. Allen, L. Grosenick, J. Taylor, A generalized least-square matrix decomposition, J. Am. Stat. Assoc. 109 (2014) 145–159.

[15] P.D. Wentzell, D.T. Andrews, D.C. Hamilton, K. Faber, B.R. Kowalski, Maximum likelihood principal component analysis, J. Chemom. 11 (1997) 339–366.

[16] D.M. Rocke, S. Lorenzato, A two-component model for measurement error in analytical chemistry, Technometrics 37 (1995) 176–184.

[17] W. Huber, A. von Heydebreck, H. Sültmann, A. Poustka, M. Vingron, Variance stabilization applied to microarray data calibration and to the quantification of differential expression, Bioinformatics 18 (2002) S96–S104.

[18] P. Wentzell, T. Karakach, S. Roy, M.J. Martinez, C. Allen, M. Werner-Washburne, Multivariate curve resolution of time course microarray data, BMC Bioinf. 7 (2006) 343.

[19] D.M. Rocke, B. Durbin, A model for measurement error for gene expression arrays, J. Comput. Biol. 8 (2001) 557–569.

[20] S. Waaijenborg, O. Korobko, K. Willems van Dijk, M. Lips, T. Hankemeier, T. Wilderjans, J.A. Westerhuis, Fusing metabolomics data sets with heterogeneous measurement errors, PLoS One 13 (4) (2018), e0195939.

[21] K.R. Gabriel, S. Zamir, Lower rank approximation of matrices by least squares with any choice of weights, Technometrics 21 (1979) 489–498.

[22] I.T. Jolliffe, Principal Components Analysis, Springer Series in Statistics, 2 edition, Springer Us, New York, 2002.

[23] S. Wold, Exponentially weighted moving principal components analysis and projections to latent structures, Chemometr. Intell. Lab. Syst. 23 (1994) 149–161. Proceedings of the 3rd Scandinavian Symposium on Chemometrics (SSC3).

[24] H.I. Nakaya, J. Wrammert, E.K. Lee, L. Racioppi, S. Marie-Kunze, W.N. Haining, A.R. Means, S.P. Kasturi, N. Khan, G.-M.M. Li, M. McCausland, V. Kanchan, K.E. Kokko, S. Li, R. Elbein, A.K. Mehta, A. Aderem, K. Subbarao, R. Ahmed, B. Pulendran, Systems biology of vaccination for seasonal influenza in humans, Nat. Immunol. 12 (2011) 786–795.

[25] H.A.L. Kiers, Towards a standardized notation and terminology in multiway analysis, J. Chemom. 14 (2000) 105–122.

[26] M. Lee, H. Shen, J.Z. Huang, J.S. Marron, Biclustering via sparse singular value decomposition, Biometrics 66 (2010) 1087–1095.

[27] H. Shen, J.Z. Huang, Sparse principal component analysis via regularized low rank matrix approximation, J. Multivar. Anal. 99 (2008) 1015–1034.

[28] M. Sill, S. Kaiser, A. Benner, A. Kopp-Schneider, Robust biclustering by sparse singular value decomposition incorporating stability selection, Bioinformatics 27 (2011) 2089–2097.

[29] K. Van Deun, T.F. Wilderjans, R.A. van den Berg, A. Antoniadis, I. Van Mechelen, A flexible framework for sparse simultaneous component based data integration, BMC Bioinf. 12 (2011) 448.

[30] H. Zou, The adaptive lasso and its oracle properties, J. Am. Stat. Assoc. 101 (2006) 1418–1429.

[31] R. Tibshirani, Regression shrinkage and selection via the lasso, J. R. Stat. Soc. 58 (1996) 267–288.

[32] N. Meinshausen, P. Bühlmann, Stability selection, J. R. Stat. Soc. Ser. B 72 (2010) 417–473.

[33] B. Efron, T. Hastie, I. Johnstone, R. Tibshirani, Least angle regression, Ann. Stat. 32 (2004) 407–451.

[34] N. Meinshausen, Relaxed lasso, Computational Statistics and Data Analysis 52 (2007) 374–393.

[35] K. Lange, D.R. Hunter, I. Yang, Optimization transfer using surrogate objective functions, J. Comput. Graph. Stat. 9 (2000) 1–20, https://doi.org/10.2307/1390605.

[36] H.A.L. Kiers, Weighted least squares fitting using ordinary least squares algorithms, Psychometrika 62 (1997) 251–266, https://doi.org/10.1007/BF02295279.

[37] K. Adachi, N.T. Trendafilov, Sparse principal component analysis subject to prespecified cardinality of loadings, Comput. Stat. 31 (2016) 1403–1427.

[38] G.H. Golub, C. Van Loan, Matrix Computations, third ed., Johns Hopkins University Press, Baltimore, 1996.

[39] M. Vervloet, K.V. Deun, W.V. den Noortgate, E. Ceulemans, Model selection in principal covariates regression, Chemometr. Intell. Lab. Syst. 151 (2016) 26–33.

[40] R. Bro, K. Kjeldahl, A.K. Smilde, H.A.L. Kiers, Cross-validation of component models: a critical look at current methods, Anal. Bioanal. Chem. 390 (2008) 1241–1251.

[41] R. Tibshirani, Regression shrinkage and selection via the lasso: a retrospective, J. R. Stat. Soc. Ser. B 73 (2011) 273–282.

[42] K. Van Deun, E.A.V. Crompvoets, E. Ceulemans, Obtaining insights from high-dimensional data: sparse principal covariates regression, BMC Bioinf. 19 (2018) 104.

[43] M.F. Van Batenburg, L. Coulier, F. van Eeuwijk, A.K. Smilde, J.A. Westerhuis, New figures of merit for comprehensive functional genomics data: the metabolomics case, Anal. Chem. 83 (2011) 3267–3274.

[44] U. Lorenzo-Seva, J.M. ten Berge, Tucker's congruence coefficient as a meaningful index of factor similarity, Methodology 2 (2006) 57–64.

[45] J. Brettschneider, F. Collin, B.M. Bolstad, T.P. Speed, Quality assessment for short oligonucleotide microarray data, Technometrics 50 (2008) 241–264.

[46] E. Curry, GESTr: Gene Expression State Transformation, 2013. R package version 0.1.

[47] J.B. Willett, J.D. Singer, Another cautionary note about $R^2$: Its use in weighted least-squares regression analysis, Am. Stat. 42 (1988) 236–238.

[48] S. Li, N. Rouphael, S. Duraisingham, S. Romero-Steiner, S. Presnell, C. Davis, D.S. Schmidt, S.E. Johnson, A. Milton, G. Rajam, S. Kasturi, G.M. Carlone, C. Quinn, D. Chaussabel, A.K. Palucka, M.J. Mulligan, R. Ahmed, D.S. Stephens, H.I. Nakaya, B. Pulendran, Molecular signatures of antibody responses derived from a systems biology study of five human vaccines, Nat. Immunol. 15 (2014) 195–204.

[49] J. Weiner, Tmod: Feature Set Enrichment Analysis for Metabolomics and Transcriptomics, 2018. R package version 0.40.

[50] J. Guo, G. James, E. Levina, G. Michailidis, J. Zhu, Principal component analysis with sparse fused loadings, J. Comput. Graph. Stat. 19 (2010) 930–946.

[51] G.I. Allen, M. Maletić-Savatić, Sparse non-negative generalized pca with applications to metabolomics, Bioinformatics 27 (2011) 3029–3035.

[52] H.A.L. Kiers, Setting up alternating least squares and iterative majorization algorithms for solving various matrix optimization problems, Comput. Stat. Data Anal. 40 (2002) 157–170, https://doi.org/10.1016/S0167–9473(02)00142–1.

[53] J. Friedman, T. Hastie, H. Hofling, R. Tibshirani, Pathwise coordinate optimization, Ann. Appl. Stat. 2 (2007) 302–332.

[54] J. Friedman, T. Hastie, R. Tibshirani, Regularization paths for generalized linear models via coordinate descent, J. Stat. Softw. 33 (2010) 1–22.