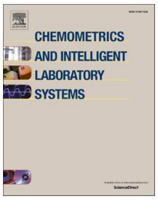
Chemometrics and Intelligent Laboratory Systems 248 (2024) 105118

Contents lists available at ScienceDirect

Chemometrics and Intelligent Laboratory Systems

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



Coupling randomisation and sparse modelling for the exploratory analysis of large hyperspectral datasets

Rosalba Calvini [a], José Manuel Amigo [b,c,*]

[a] *Department of Life Sciences, University of Modena and Reggio Emilia, Pad. Besta, Via Amendola, 2, 42122, Reggio Emilia, Italy*
[b] *Ikerbasque, Basque Foundation for Sciences, Plaza Euskadi, 5 Bilbao, 48009, Spain*
[c] *Department of Analytical Chemistry, University of the Basque Country, Barrio Sarriena S/N, Leioa, 48940, Spain*

ARTICLE INFO

Keywords:
Hyperspectral imaging
Randomisation
Sparse methods
Principal component analysis
Data reduction
sPCA
rsPCA

ABSTRACT

Sparse-based models are a powerful tools for data compression, variable reduction, and model complexity reduction. Nevertheless, their major issue is the high computational time needed in large matrices. This manuscript proposes, for the first time, to couple randomised decomposition as a first step before sparsity calculations, followed by a projection of the full data onto a reduced-sparse set of loadings that will drastically reduce the time needed for calculations and built models that are equally reliable as their sparse-based homologous. While this new approach might be valid for several scenarios (exploration, regression and classification), we will focus on exploration methods (like Principal Component Analysis – PCA) applied to large datasets of hyperspectral images. Two datasets of different complexity have been tested, and the benefits of the coupled randomisation and sparse PCA (rsPCA) are extensively studied.

1. Introduction

Nowadays, modern analytical tools such as spectroscopic techniques or -*omics* platforms allow to easily acquire high dimensional data arrays, which can be composed of an immense number of samples, variables, or both. Therefore, one of the main challenges of data analytics consists in the development and application of data mining methods able to deal with such high dimensional data arrays with reasonable computational efforts [1–3].

In this context, sparse-based methods have been demonstrated to be a powerful approach allowing to perform data compression and variable reduction at the same time. Sparse methods are extensions of traditional multivariate methods in which sparsity is induced on the estimated parameter vector of the considered model. In this manner, the parameter vector is forced to contain many zeros and few non-zero entries: the zero entries correspond to irrelevant or noisy variables that are discarded, while the non-zero entries correspond to the more informative variables [4]. For example, in the context of data exploration by means of Principal Component Analysis (PCA), a sparse PCA (sPCA) model will provide sparse loading vectors where only the more relevant variables have non-zero loading coefficients and contribute to the trends observed in the score plots [5]. Sparse versions of classical multivariate methods have been developed not only for data exploration but also for regression and classification problems. For regression purposes, different versions of sparse Partial Least Squares (sPLS) were proposed, considering different penalisation methods to achieve sparsity [6,7]. In the context of classification, sparse versions of common classification algorithms such as Partial Least Squares Discriminant Analysis (PLS-DA) [8], Linear Discriminant Analysis (LDA) [9] and Support Vector Classification (SVC) [10] were developed.

Due to their advantages, sparse methods have been broadly applied in different data analytics frameworks including signal processing [11], analysis of massive biomedical and biological data [12–15] or extraction of relevant features from imaging data [16,17]. Despite their great potential, the high amount of time needed for the computations is one of the major issues and it becomes even more evident when the analysed matrix has a large number of rows.

As a possible solution, this manuscript proposes for the first time to couple randomisation algorithms for a low-rank approximation of the data as a previous step before sparsity is induced in the model. Randomisation methods aim at generating a low-rank approximation of the original high dimensional data matrix that maintain the main sources of variance of the original high dimensional data matrix. In other words, the new low-rank matrix is a nearly accurate approximation of the

[*] Corresponding author. Ikerbasque, Basque Foundation for Sciences, Plaza Euskadi, 5 Bilbao, 48009, Spain.
E-mail address: josemanuel.amigo@ehu.eus (J.M. Amigo).

https://doi.org/10.1016/j.chemolab.2024.105118
Received 26 December 2023; Received in revised form 21 March 2024; Accepted 21 March 2024
Available online 28 March 2024

original matrix [18]. A common example of this kind of approach consists in performing a random subsampling of the original data matrix using for instance Monte Carlo method or its modifications [19,20]. In this manuscript we considered an alternative approach to achieve dimension reduction based on random projection, which consists in the projection of original data to a set of randomly taken vectors followed by a factorisation step [21]. This approach demonstrated to be faster and more robust than deterministic methods, and it has been used to solve clustering and classification issues [22].

Therefore, we used randomisation as a previous step before sparse model calculation to drastically reduce the time needed for calculations and to ease the model optimisation. While this new approach could be valid when all types of sparse models are applied, in this paper we will focus on exploration methods like PCA applied to large datasets of hyperspectral images.

Hyperspectral imaging (HSI) [23] is a powerful non-destructive technique able to merge spectroscopic and imaging technologies in order to obtain both spatial and spectral information from a sample. Spectral information allows retrieving the chemical composition of the investigated sample while spatial information allows evaluating the variation of the chemical composition within the sample surface, obtaining the so-called chemical maps [24]. Thanks to these advantages, HSI has found a wide application in different fields, including remote sensing [25,26], food science [27,28], pharmaceutical and medical analyses [29,30], and forensic science [31,32] among others.

Each pixel of a hyperspectral image contains a full spectrum acquired in a defined wavelength range of the electromagnetic spectrum based on the acquisition device. Therefore, hyperspectral images are three-dimensional data arrays with two spatial dimensions, related to pixel rows and columns, and one spectral dimension. As an example, an HSI system with a spatial resolution of 250 × 250 pixels will provide hyperspectral images with 62 500 pixel spectra, and each spectrum can contain more than 100 spectral points; consequently, a single image can be composed by more than 6.25×10^6 data points. This toy example makes evident that hyperspectral images are high dimensional data arrays, and their analysis involves issues related to data handling and storage [33]. These problems become even more relevant considering that for practical applications it is necessary to acquire a large number of images and analyse them altogether to compare samples in different images or to evaluate changes over time of sample composition [34,35].

Sparse methods, and in particular sparse PCA in an exploratory framework, have been successfully used for the analysis of hyperspectral images in order to increase model interpretability by selecting only the relevant spectral features for the problem at hand [17,36–38]. However, when dealing with the analysis of a large number of images altogether sparse methods require high computational times for model calculation, which limits their possible application.

Common approaches to perform data dimensionality reduction of hyperspectral images to speed up the computation time are generally based on reducing the spatial dimension by means of binning or computation of the average spectrum of each image. The main limitation of these methods is the loss of relevant information when spatial resolution is important. Conversely, the proposed approach allows to preserve both spatial and spectral information since the original high dimensional data matrix is converted into a low-rank approximation that maintain the relevant sources of data variance.

This manuscript will apply randomisation as a preliminary step of applying sparse PCA (generating a methodology called randomised sparse PCA – rsPCA) to demonstrate that sparse models can be performed, first on a reduced, well selected, subspace of the spectral matrix and then project the rest of the matrix into the sparse solution with accuracy, reducing the time of analysis drastically. For this purpose, two well-known hyperspectral datasets will be used. Benefits, drawbacks and future possibilities will be discussed.

2. Theory

2.1. Sparse Principal Component Analysis

Sparse PCA (sPCA) calculates a PCA model inducing sparsity on the model parameters: scores, loadings, or both. Focusing our interest on the sparsity induced on the loading vectors, this induction means converting a certain number of loading points to zero to eliminate all loading points that are not giving essential information. Therefore, variables that are not important are set to zero, increasing the explainability and interpretability of the model.

Sparsity is achieved by adding a penalty term to the objective function of the PCA model [4]. Different penalisation methods are proposed in the literature [39,40], being the Least Absolute Shrinkage and Selection Operator (LASSO) one of the most helpful in spectroscopy [41,42]. LASSO applies a constraint to the sum of absolute values of a vector (or L_1 norm), forcing several coefficients of the vector to be equal to zero [43] and, therefore, keeping the natural behaviour of the spectra composed of peaks and bands.

Let **X** be the unfolded hyperspectral image data matrix with size $\{r \times c, s\}$, where r and c correspond to the number of row and column pixels of the original hyperspectral image, respectively, and s corresponds to the number of spectral channels. The calculation of the sPCA model with A PCs and sparsity induced on the loadings can be formulated as:

$$min\left(\left|\mathbf{X} - \mathrm{TP}^T\right|_F^2\right) \tag{1}$$

subject to:

$$|\mathbf{p}_i|_1 \leq c, for\ i = 1, \dots, A \tag{2}$$

where **T** is the scores matrix with size $\{r \times c, A\}$, **P** is the loadings matrix with size $\{s, A\}$, $|\bullet|_F^2$ is the sum of squared elements (Frobenius norm), while c is a scalar corresponding to the L_1 norm constraint applied to each column of matrix **P** ($\mathbf{p_i}$) and from here onwards it will be referred to as sparsity constraint. The sparsity constraint is a positive tuning parameter that controls the sparsity level of the model, i.e., the number of variables forced to be equal to zero. In particular, the value of c may range between 1 and the square root of the number of variables. When c is equal to 1, the sPCA model has a high sparsity level with only one variable selected for each PC. In contrast, a value of c equal to the square root of the number of variables gives the same loadings as those obtained with standard PCA. Therefore, the lower the value of the sparsity constraint, the higher the sparsity induced on the loadings [17].

To calculate sPCA models, two different algorithms are normally used. The first algorithm is based on an iterative alternating procedure similar to Nonlinear Iterative Partial Least Squares (NIPALS) in which a soft thresholding is applied on the loadings according to the sparsity constraint. This algorithm initially finds the solution for the first sparse PC, and then the subsequent PCs are calculated after a deflation step [44]. The second algorithm, named Alternating Shrunken Least Squares (ASLS), calculates all the PCs simultaneously by iterating between scores and loadings until convergence. Sparsity is applied PC-wise on the loadings during the iterations [45]. Since the PCs in sPCA are not orthogonal, calculating the PCs based on current residuals using deflation does not aim at solving Eq. (1) and Eq. (2) directly [46]. Therefore, ASLS represents a more direct approach to calculating sPCA models. For a more detailed description of the sPCA algorithms used in this study, the reader is referred to Refs. [44,45].

2.2. Randomised PCA

sPCA can be directly applied to **X**. Nevertheless, when the dimensionality of **X** is very large, sPCA models tend to be computationally expensive and time-consuming. Therefore, we propose to couple sPCA with randomisation methods [47].

The aim of randomisation is simple. Randomisation aims at obtaining a representative subset of **X**, namely $\mathbf{X_R}$, containing all the essential information of the original dataset, but with significantly smaller number of rows compared to **X**. Then, PCA is applied to this subset.

$$\mathbf{X_R} = \mathbf{T_R}\mathbf{P_R^T} + \mathbf{E_R} \tag{3}$$

If the first assumption is fulfilled, then it is easy to assume that the loadings of the reduced model will be similar (or even the same) to those obtained with the full model ($\boldsymbol{P_R^T} = \boldsymbol{P^T}$). Therefore, the calculation of the scores of the full matrix is straightforward:

$$\widehat{\mathbf{T}} = \mathbf{X}\mathbf{P_R} \tag{4}$$

The main issue of randomisation methods is to achieve a subset of the full data that is representative enough to have the warranty that all the essential information of the full data is contained in the subset sample. Therefore, performing a pure randomisation step (e.g. random selection of a small subset of samples) might produce under-fitted models or models that only represent some of the variance in the full data. One possibility that has been demonstrated to be very optimal is what is called randomised PCA (rPCA) [18,47]. This approach does not calculate a subset of **X** but a low-rank alternative matrix called **B**, which is a good approximation of **X** but containing much fewer rows.

B is found by projecting **X** onto its orthonormal basis **Q** ($\mathbf{X} = \mathbf{Q}\mathbf{Q^T}\mathbf{X}$). Ideally, the number of columns in **Q** should correspond to the number of PCs (A) on **X**. Nevertheless, since the exact number of PCs is unknown, the number of columns calculated in **Q** is slightly larger (A + p), where p is an oversampling parameter. Following an iterative procedure based on q iterations (further details in the supplied references), **B** can then be calculated and, therefore, a set of scores and loadings can be calculated as follows:

$$\mathbf{B} = \mathbf{T_R}\mathbf{P_R^T} + \mathbf{E} \tag{5}$$

Once this **B** is calculated, the new scores can be easily predicted following eq. (4).

2.3. Randomised sparse PCA (rsPCA)

Following this idea, rPCA can be easily merged to sparse models by, first, calculating **B** and then applying sparsity to **B** instead of **X**.

1) Calculate **B** on **X.** For this, the oversampling parameter (p) and the number of iterations (q) must be optimised. This might look cumbersome. Nevertheless, as demonstrated in the supplied references [18,22] and in further sections of this manuscript, this optimisation is only somewhat critical.
2) Apply sPCA on **B.**
3) Calculate the new set of scores using the original full matrix **X** and the loadings from the previous step.

3. Materials and methods

Two well-known datasets have been evaluated: a benchmark dataset of a Raman hyperspectral image of an oil-in-water emulsion and a large time-series dataset of NIR hyperspectral images of bread samples.

3.1. Oil-in-water emulsion Raman hyperspectral image

The first dataset (emulsion dataset) was used for an initial comparison between sPCA and rsPCA models to evaluate the effect of sparsity level, sparse PCA algorithm, number of iterations, and the oversampling value in the randomisation step. The model consistency between sPCA and rsPCA was evaluated component-wise considering the correlation coefficients between the loading vectors and the percentage of mean squared difference between the score vectors (scores MSD%) of the two models (i.e., full model and reduced model) with respect to the variance of the corresponding score vectors of the full model:

$$scores\ MSD\% = \frac{SSQ_D}{SSQ_T} \times 100 \tag{6}$$

where $\mathrm{SSQ_D}$ is sum of squares of the differences between the score vectors of a considered component for sPCA and rsPCA while $\mathrm{SSQ_T}$ is the sum of squares of the corresponding score vector of full sPCA model.

Table 1
Summary information about the considered datasets and the model parameters used to calculate the sPCA and rsPCA models.

	Emulsion image	Bread staling dataset
Image size	60 × 60	2592 × 636
# pixels	3600	1 063 583 after background removal
# spectral variables	253	142
Preprocessing	Mean center	2nd derivative +mean center
# sPCs	6	From 1 to 6
sPCA algorithms	ASLS and deflation	ASLS and deflation
Sparsity levels	High sparsity (3.18) Mid-High sparsity (6.36) Mid-Low sparsity (9.54) Low sparsity (12.72)	High sparsity (2.30) Mid-High sparsity (4.60) Mid-Low sparsity (6.90) Low sparsity (9.19)
# iterations	0 and 1	0, 1 and 2
oversample	5 and 100	5, 10, 20, 50 and 100

This dataset consists of a Raman hyperspectral image of an oil-in-water emulsion sample. The hyperspectral image is composed of 60 × 60 pixels and 253 spectral variables from 950 $\mathrm{cm^{-1}}$ to 1800 $\mathrm{cm^{-1}}$. Details about the experimental setup are described in Andrew et al., 1998 [48]. This dataset is rather small (4 MB). Nevertheless, it is an excellent choice because it has been extensively analysed, and based on previous studies, the sampled area has four main components: a structural phase, two droplet phases and an aqueous phase [49–51]. Before calculating sPCA and rsPCA models, the hyperspectral image was unfolded into a 3600 × 253 data matrix and mean centring was applied.

Different sPCA models were calculated considering 6 sPCs, ASLS and deflation-based algorithms were applied to imply sparsity in the loadings direction, testing 4 sparsity levels corresponding to the following values of the sparsity constraint (c): 3.18, 6.36, 9.54 and 12.72, corresponding to high, medium-high, medium-low and low sparsity level, respectively. For this dataset, the maximum value allowed for the sparsity constraint is equal to 15.90, which corresponds to the squared root of the number of variables. In this latter case (c = 15.90) no sparsity is induced on the model, and the results are the same as PCA (see Section 2.1).

Considering the rsPCA models, the following parameters were tested: number of iterations (q) equal to 0 or 1 and oversample value (p) equal to 5 and 100. A detailed description of the model parameters tested for this dataset is reported in the second column of Table 1.

3.2. Bread dataset of NIR hyperspectral images

The second dataset is a large dataset of time series NIR hyperspectral images to assess, in this scenario, the effect of the different model parameters on the consistency of the results obtained with rsPCA compared to a sPCA model. In this case, we considered the correlation coefficient between the loadings and the scores MSD% to check the consistency between rsPCA models and the corresponding sPCA model. In addition, given the nature of the dataset, we also compared the computation time.

The bread dataset consists of 108 NIR-HSI images (938–1630 nm) of white bread slices of three different types measured in six subsequent times (1, 4, 7, 10, 14 and 21 days). The three bread types were prepared with different doughs: one was made following a standard recipe

		ASLS				Deflation			
		$q = 0$		$q = 1$		$q = 0$		$q = 1$	
		$p = 5$	$p = 100$	$p = 5$	$p = 100$	$p = 5$	$p = 100$	$p = 5$	$p = 100$
HIGH	sPC1	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
	sPC2	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
	sPC3	0.999	1.000	1.000	1.000	1.000	1.000	1.000	1.000
	sPC4	0.995	1.000	1.000	1.000	1.000	1.000	1.000	1.000
	sPC5	0.999	1.000	1.000	1.000	0.995	1.000	1.000	1.000
	sPC6	0.985	1.000	1.000	1.000	0.999	1.000	1.000	1.000
MID-HIGH	sPC1	0.999	1.000	1.000	1.000	1.000	1.000	1.000	1.000
	sPC2	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
	sPC3	0.995	1.000	1.000	1.000	0.999	1.000	1.000	1.000
	sPC4	0.103	0.999	0.992	1.000	0.860	1.000	1.000	1.000
	sPC5	0.817	1.000	0.998	1.000	0.100	1.000	1.000	1.000
	sPC6	0.778	1.000	0.997	1.000	0.429	0.999	0.999	1.000
MID-LOW	sPC1	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
	sPC2	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
	sPC3	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
	sPC4	0.974	1.000	1.000	1.000	0.975	1.000	1.000	1.000
	sPC5	0.984	1.000	1.000	1.000	0.974	1.000	1.000	1.000
	sPC6	0.916	1.000	1.000	1.000	0.811	0.999	0.999	1.000
LOW	sPC1	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
	sPC2	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
	sPC3	0.999	1.000	1.000	1.000	1.000	1.000	1.000	1.000
	sPC4	0.756	1.000	1.000	1.000	0.969	1.000	1.000	1.000
	sPC5	0.897	1.000	1.000	1.000	0.958	1.000	1.000	1.000
	sPC6	0.841	1.000	1.000	1.000	0.913	1.000	1.000	1.000

Fig. 1. Heat map of the correlation coefficients of the loadings obtained with the sPCA models and those obtained with the corresponding reduced models calculated with different iterations (q) and oversample values (p).

without any modifications and used as control bread (CR). In contrast, the other two doughs were added with different maltogenic α-amylases enzymes (EZ1 and EZ2). This dataset was designed to study how bread staling affects the behaviour of the whole crumb surface and how these phenomena are modified by adding maltogenic α-amylases in the dough. A detailed description of this dataset, including sample preparation and image acquisition setup, is reported in Ref. [52].

Each hyperspectral image had dimensions of 144 row pixels × 106 column pixels × 142 spectral channels. All 108 images were merged into a single augmented image with dimensions of 2592 × 636 × 142, corresponding to 234 088 704 data points, as shown in Fig. 2 of reference [52]. The background was removed from the augmented image using a thresholding procedure at 1400 nm, resulting in 1 063 583 pixels kept after background masking. The size of this array is, approximately, 5 GB.

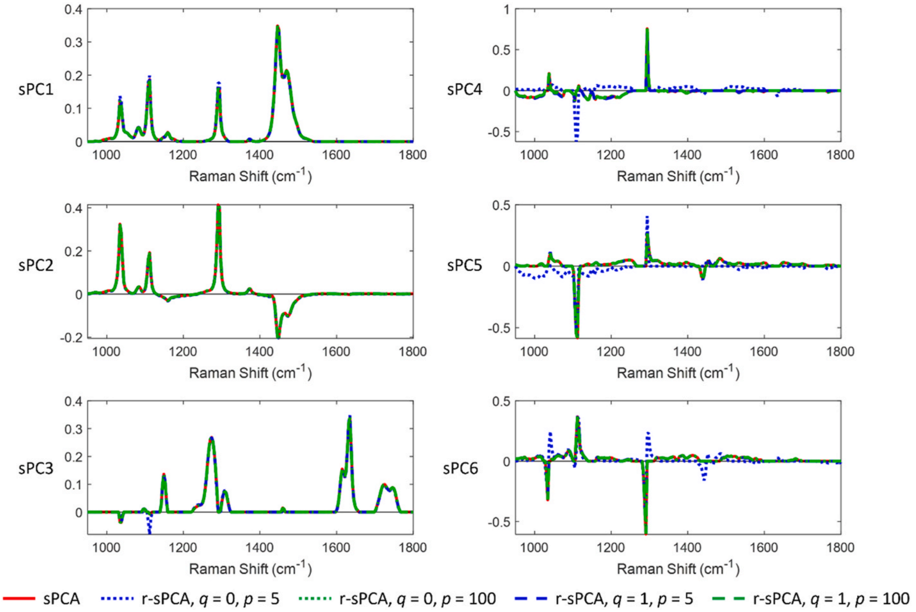


Fig. 2. sPCA and rsPCA loading vectors of the models calculated with ASLS algorithm considering a mid-high sparsity level. Solid red lines correspond to the loadings of the full sPCA model, blue lines correspond to the loadings of rsPCA models calculated with $p = 5$, green lines correspond to the loadings of rsPCA models calculated with $p = 100$, dotted lines correspond to the loadings of rsPCA models calculated with $q = 0$ and dashed lines correspond to the loadings of rsPCA models calculated with $q = 1$. (For interpretation of the references to colour in this figure legend, the reader is referred to the Web version of this article.)

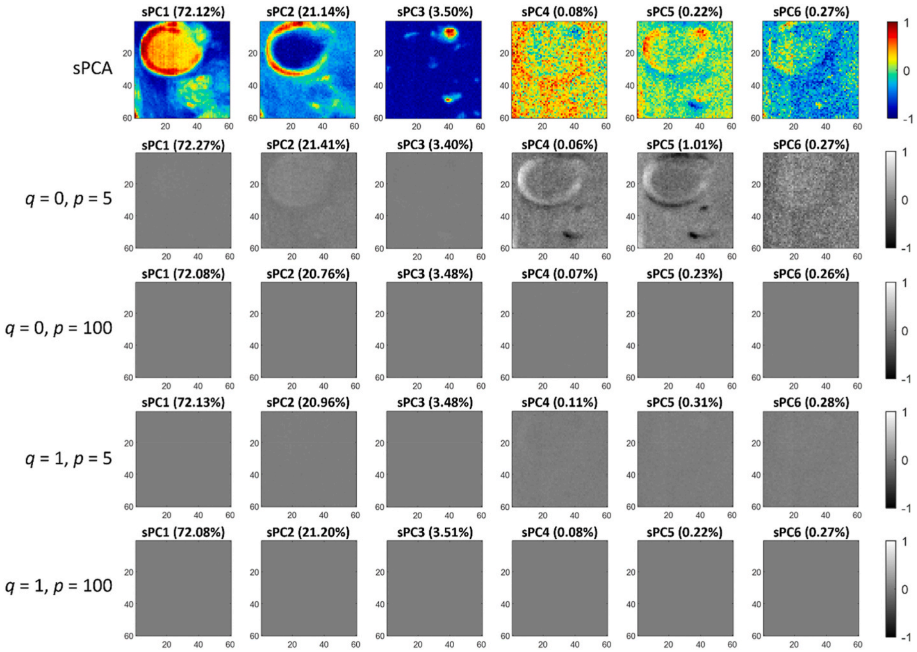


Fig. 3. Normalised score images obtained from the sPCA model calculated considering a mid-high sparsity level and ASLS algorithm, and difference images between the normalised sPCA scores (mid-high sparsity and ASLS) and the corresponding normalised scores of the rsPCA models calculated with different iterations ($q = 0$ and $q = 1$) and different oversampling values ($p = 5$ and $p = 100$).

Before calculating the sPCA and rsPCA models, the spectra were preprocessed with Savitzky-Golay second derivative (window size of 11 points and second polynomial degree) and mean center.

Different sPCA models were calculated considering a varying number of sPCs from 1 to 6. ASLS and deflation-based algorithms were used to induce sparsity. Four sparsity levels corresponding to a sparsity constraint equal to 2.30 (high sparsity), 4.60 (mid-high sparsity), 6.90 (mid-low sparsity) and 9.19 (low sparsity) were used.

Each sPCA model was compared with the corresponding rsPCA models calculated considering a number of iterations equal to 0, 1 and 2 and oversampling values equal to 5, 10, 20, 50 and 100. A detailed description of the model parameters tested for this dataset is reported in the last column of Table 1.

For both emulsion and bread datasets, the selection of q and p values to be tested were based on previous studies where the randomisation algorithm was evaluated in the frame of PCA-based data exploration [18,22,47].

3.3. Software

In this study, sPCA was performed using the MATLAB (The MathWorks) function freely downloadable from https://ucphchemometrics.com/186-2/algorithms/[45].

The rsPCA was implemented following the nomenclature in Cruz-Tirado et al. [47]. Moreover, it has been implemented in the newest version of HYPER-Tools [53] (freely available at https://www.hypertools.org/).

All the calculations were carried out in MATLAB 2020a (Mathworks, Natick, USA) environment on an HP EliteBook laptop, core i7 CPU 2 GHz, 16 GB RAM and 1 TB storage. Both datasets used in this study are available at https://www.hypertools.org/.

4. Results

4.1. Emulsion dataset

The correlation coefficients between the loadings obtained with the sPCA model and the ones obtained with rsPCA are reported in Fig. 1. The first observation that must be made is the extraordinary similarity obtained in all the cases studied.

For the models with high sparsity, it is possible to observe that rsPCA performs with an outstanding similarity to the corresponding sPCA model for both sparse algorithms and a correlation coefficient between the loading vectors equal to one is obtained considering either 0 or 1 iterations and an oversample value (p) equal to 5 or 100. Interestingly, there is a lower consistency between rsPCA and sPCA when rsPCA is calculated with 0 iterations and a low oversample value ($p = 5$) at lower sparsity levels. The decrease in the correlation coefficient between the loading vectors does not affect the first three sPCs. This decrease, though, mainly involves the PCs from PC4 onwards. This effect is particularly evident considering the mid-high sparsity level, while it can be considered negligible with the low sparsity level.

Obviously, the correlation between the loadings is reflected in the projected scores of the models, affecting the consistency between the scores of rsPCA and sPCA. Supplementary Table S1 reports the scores MSD% to compare the scores of each tested rsPCA model and the corresponding sPCA model. The main discrepancies between rsPCA and sPCA were found for the models calculated considering mid-high sparsity. Fig. 2 reports the loading vectors of the rsPCA models calculated with ASLS algorithm, mid-high sparsity and the different model parameters (i.e., number of iterations and oversampling value) together with the loading vectors of the corresponding sPCA model.

As already observed, for the first 3 sPCs, the loading vectors obtained with rsPCA models are essentially the same as those obtained with sPCA. The main differences are evident for sPC4, sPC5 and sPC6 when the rsPCA model is calculated with no iterations ($q = 0$) and oversample equal to 5 ($p = 5$). Also, considering the deflation-based algorithm to calculate the PCA model, the main differences between sPCA and rsPCA loading vectors are found in sPC4, sPC5 and sPC6 with the rsPCA model calculated considering $q = 0$ and $p = 5$ (Supplementary Fig. S1). It must be highlighted that when the loading vectors do not match exactly, the spectral regions selected by the sparse algorithms generally correspond in the major peaks, being the differences allocated in very small, even noisy, spectral regions.

Fig. 3 shows the normalised score images of the sPCA model calculated with mid-high sparsity and ASLS algorithm together with the

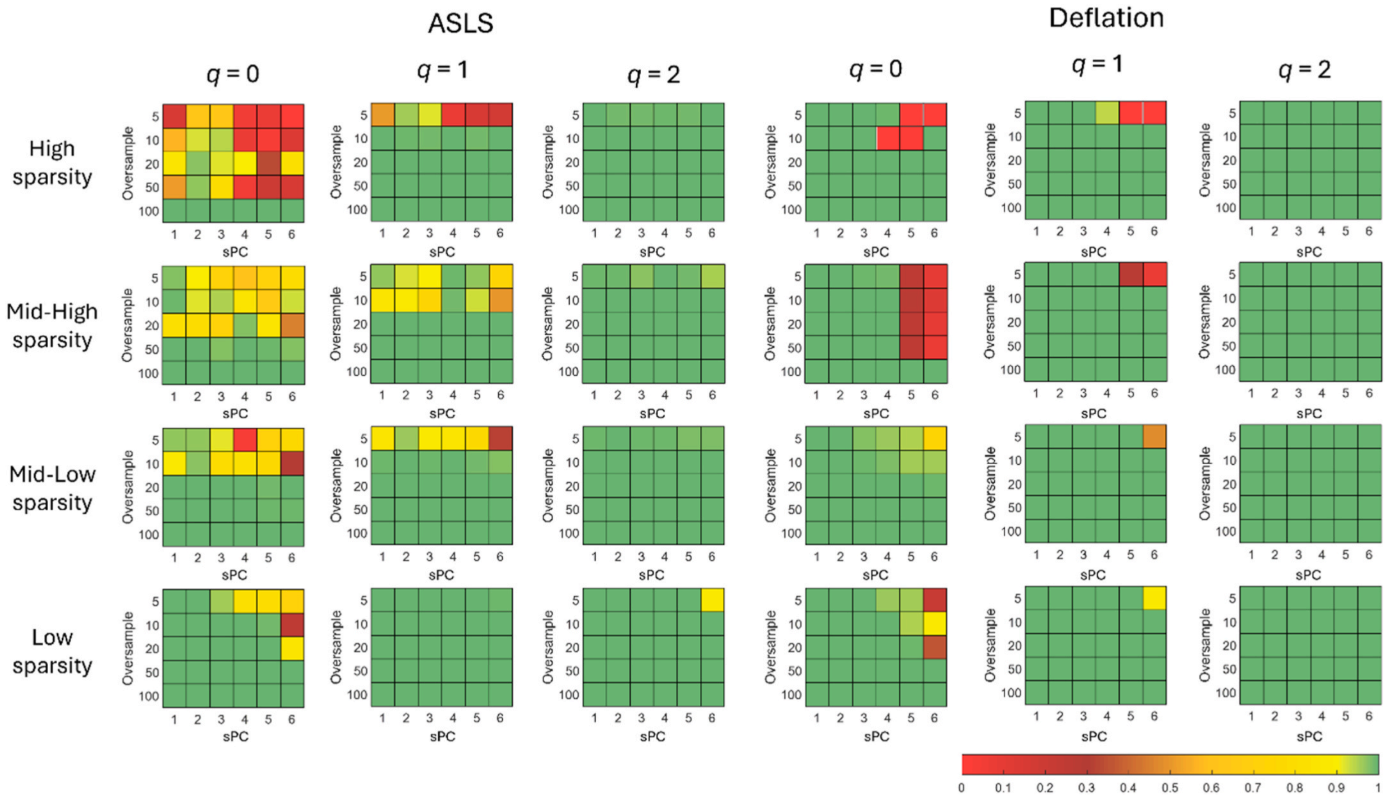


Fig. 4. Heatmaps of the correlation coefficients between the loading vectors of rsPCA and sPCA models; the models were calculated considering 6 PCs, different sparsity levels, ASLS and deflation-based algorithms and, for rsPCA, varying iterations (q) and oversampling values.

images of the difference between the sPCA scores and the rsPCA scores calculated with a varying number of iterations (q) and oversample values (p). Each sPCA score image was normalised between -1 and $+1$ for a more direct comparison. The same normalisation was also applied to the calculated score images of the different rsPCA models.

Except for the rsPCA model calculated with $q = 0$ and $p = 5$, in the other rsPCA models, the score images for all 6 sPCs correspond to those obtained with the sPCA model. Similar results were also observed considering the deflation-based algorithm to achieve sparsity (Supplementary Fig. S2).

For both ASLS and deflation-based models, the main differences between sPCA and rsPCA models are particularly evident from sPC4 to the subsequent sPCs. However, observing the corresponding score images, it is possible to highlight that these sPCs do not fully retain the structural information of the image. Instead, they retain spurious effects promoted by the noisy background of the sample. Therefore, when retaining relevant information in the sPCs, rsPCA and sPCA are more likely to provide consistent results even if considering a low number of iterations and oversampling values in rsPCA. When more sPCs have to be retained in the model, e.g., for an initial exploratory evaluation of the image dataset, it is necessary to increase either the number of iterations or the oversampling value to expect that rsPCA provides the same results as the corresponding sPCA model.

4.2. Bread dataset

The correlation coefficients between the loading vectors of the sPCA models and the corresponding rsPCA models are displayed in Fig. 4. As for the emulsion dataset, we evaluated the effect of considering different sparsity levels to achieve sparsity and the two algorithmic approaches to calculate sPCA. In addition, we also evaluated the effect of calculating the models considering a different number of sPCs since, in sPCA, the components are not orthogonal. Therefore, the number of sPCs may also influence the model consistency between sPCA and rsPCA. For ease of visualisation, Fig. 4 reports the correlation coefficients between sPCA and rsPCA loading vectors of the models calculated considering 6 sPCs, while the complete results are reported in Supplementary Figs. S3–S8.

Generally, more consistent results are obtained between rsPCA and sPCA when using the deflation-based algorithm, also considering low iterations and oversampling values. However, retaining fewer sPCs in model computation also allows consistent results with the ASLS algorithm (see Figs. S3–S5 of the Supplementary Material). This fact can be explained considering that the ASLS algorithm computes all the sPCs altogether, and thus retaining in the model an excessive number of sPCs determines that not only the useful information is considered but also the noise is modelled if the number of sPCs is elevated. Therefore, all sPCs are influenced by the presence of noise, and this, in turn, affects the consistency between the results obtained with sPCA and rsPCA. Conversely, when using the deflation-based algorithm, the main differences between sPCA and rsPCA are found in the last sPCs since the successive calculation of the sPCs using deflation determines that the first sPCs are less influenced by non-relevant information.

Differently from the emulsion dataset, the higher the sparsity of the model is, the less consistent the results between sPCA and rsPCA are, particularly for the ASLS algorithm. This fact can be explained by considering the different nature of Raman and NIR spectra. Raman spectra are composed of sharper and well-defined peaks. Therefore, considering higher sparsity levels, the model tends to select only the relevant peaks, while moving to lower sparsity levels implies the explanation of baseline shifts or smaller peaks with an increase in noise. Conversely, NIR spectra are composed of wider and broader bands. Therefore, sparse models with too high sparsity levels may need to include more information about the investigated dataset.

However, similarly to what is observed for the emulsion dataset, to have consistent results between sPCA and rsPCA, it is possible to increase the number of iterations or the oversampling value. Fig. 5 shows

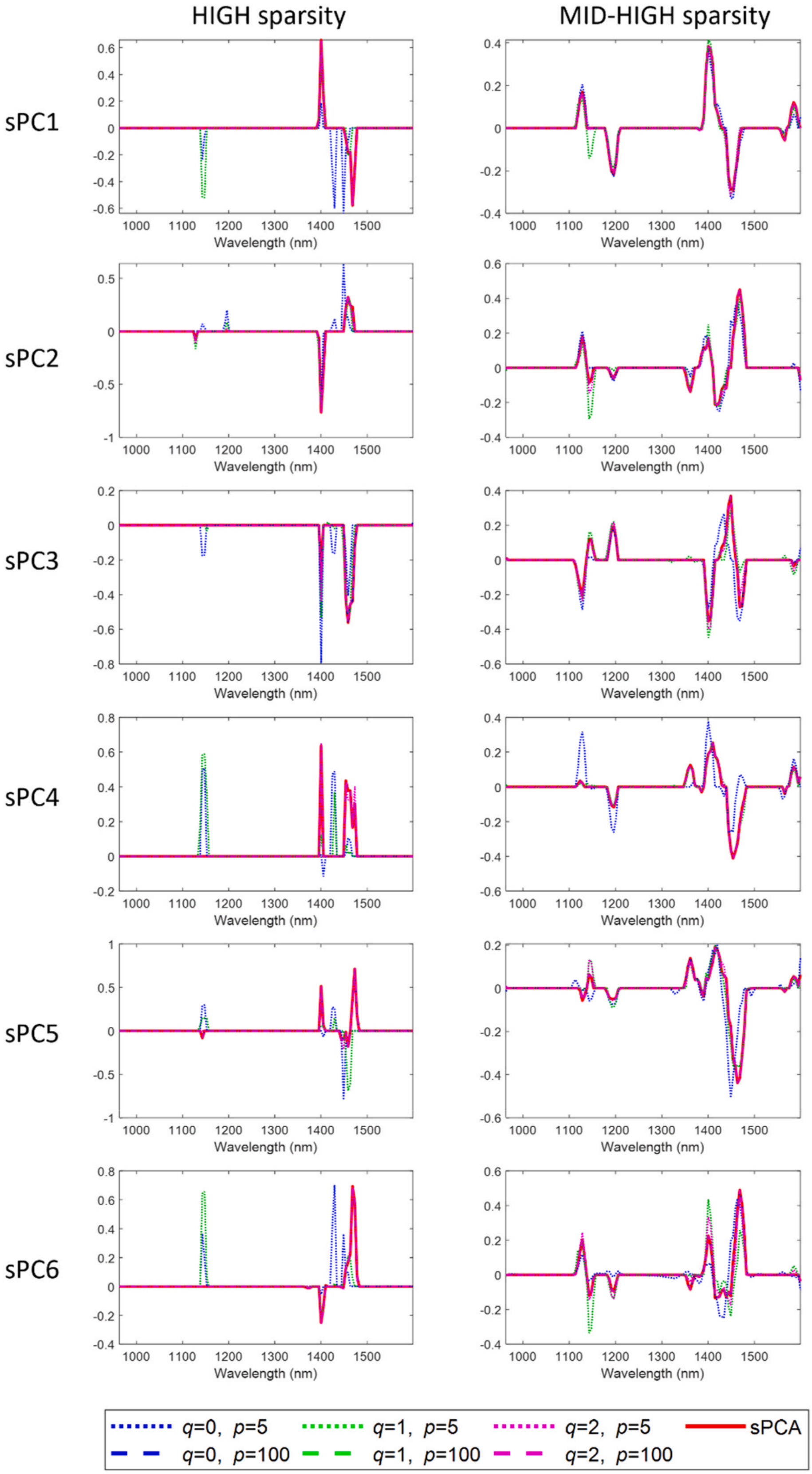


Fig. 5. Loading vectors of the sPCA and rsPCA models calculated considering 6 sPCs, ASLS algorithm, high and mid-high sparsity levels, and for rsPCA different combinations of a number of iterations ($q = 0$, $q = 1$ and $q = 2$) and oversampling values ($p = 5$ and $p = 100$). Solid red lines correspond to the loadings of the full sPCA models, blue lines correspond to the loadings of rsPCA models calculated with $q = 0$, green lines correspond to the loadings of rsPCA models calculated with $q = 1$, magenta lines correspond to the loadings of rsPCA models calculated with $q = 2$, dotted lines correspond to the loadings of rsPCA models calculated with $p = 5$ and dashed lines correspond to the loadings of rsPCA models calculated with $p = 100$. (For interpretation of the references to colour in this figure legend, the reader is referred to the Web version of this article.)

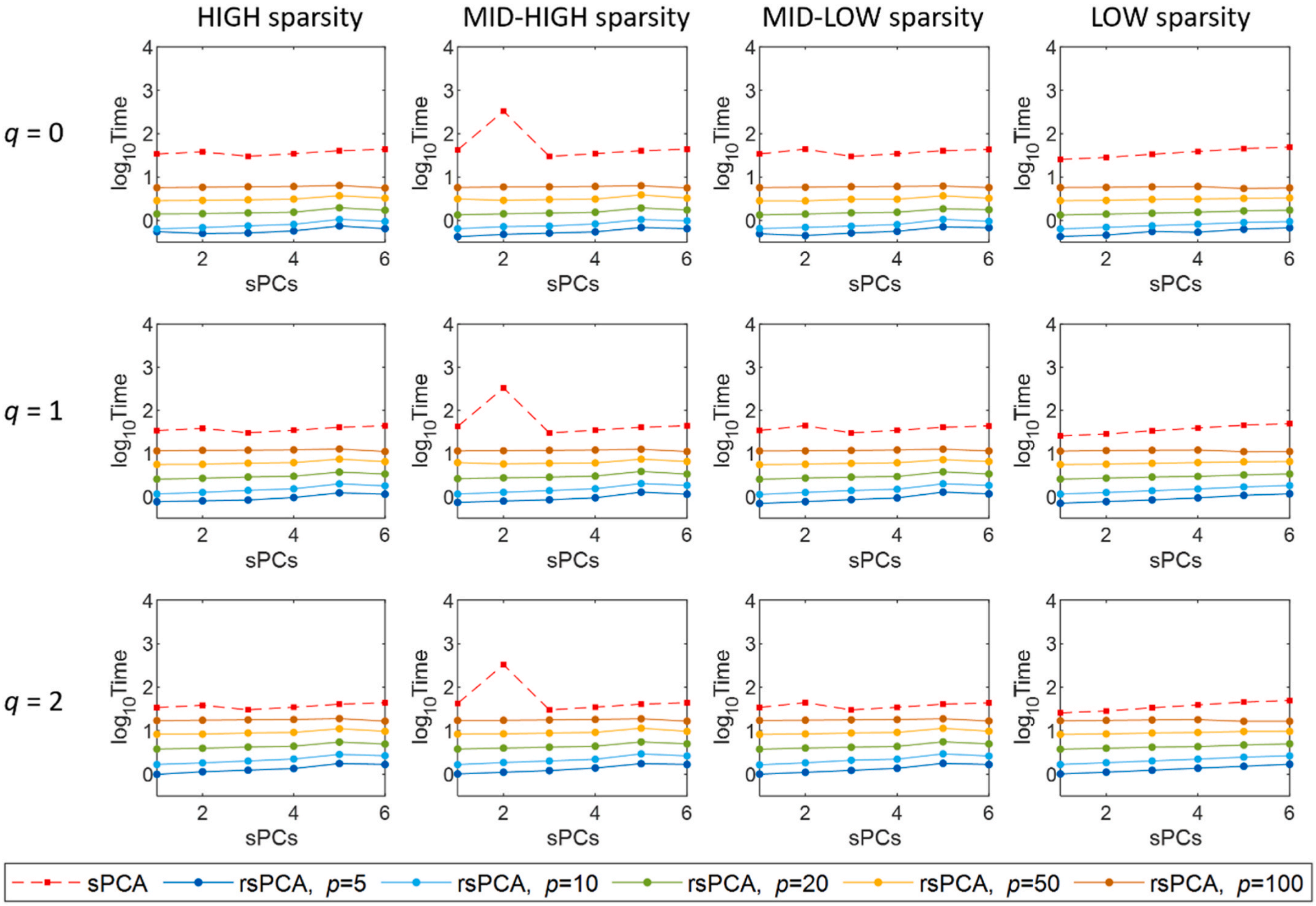


Fig. 6. Computation time (seconds, expressed in logarithmic scale) for the sPCA and rsPCA models calculated considering the ASLS algorithm. Red colour refers to the full sPCA models while the other colours refer to the rsPCA models calculated with varying values of p: blue colour for $p = 5$, cyan colour for $p = 10$, green colour for $p = 20$, orange colour for $p = 50$ and brown colour for $p = 100$. (For interpretation of the references to colour in this figure legend, the reader is referred to the Web version of this article.)

the loading vectors for the sPCA and rsPCA models calculated with 6 sPCs, ASLS algorithm, high and mid-high sparsity levels, a number of iterations from 0 to 2, and oversampling values equal to 5 or 100. The same comparison between the loading vectors of sPCA and rsPCA models calculated considering the deflation-based algorithm to achieve sparsity is reported in Fig. S9 of Supplementary Material.

Fig. 5 clearly shows that the differences in the loading vectors are mainly present when the oversampling value (p) is equal to 5, and q is equal to 0 or 1 (blue and green dotted lines in Fig. 5). Using the deflation-based algorithm, the loading vectors obtained with sPCA and rsPCA are essentially the same for the first 4 sPCs (Fig. S9). Conversely, for the ASLS algorithm, if p and q values are not correctly set, the differences between sPCA and rsPCA models are evident also in the first sPCs in particular for high sparsity, while for mid-high sparsity even if the loading vectors do not exactly match there is convergence on the selected spectral regions.

One of the most important reasons for using randomisation in the calculation of sparse models is the drastic decrease in the computation time. This is especially relevant in very large datasets like this one. Figs. 6 and 7 report the computation time for the sPCA and rsPCA models calculated considering ASLS and deflation-based algorithms. In both figures, the computation time is expressed in seconds and reported using a logarithmic scale.

The first observation is that the ASLS algorithm is generally faster than the deflation-based algorithm. Indeed, when the sPCA model is calculated using ASLS, the computation time is between about 50 and 150 s (red dashed line in Fig. 6), and the number of sPCs does not influence it since this algorithm computes all the sPCs simultaneously. Conversely, for deflation-based algorithm, the computation time of sPCA models strongly depends on the number of sPCs and the sparsity level. In particular, when more than 3 or 4 sPCs are considered in the model, the computation time may reach 3000 s with high and mid-high sparsity.

The most important observation is that the rsPCA models drastically reduced the computation time while keeping the same results as in sPCA. As discussed before, rsPCA requires defining the number of iterations and the oversampling value for model computation. Considering the effect of these parameters on computation time, it is possible to observe that setting higher oversample values determines a stronger increase in the computation time compared to performing more iterations. Therefore, to gain an advantage from randomness and maintain consistency with sPCA results, it is advisable to increase the number of iterations rather than the oversample value.

Given these considerations, for the bread dataset, it is possible to identify the best compromise between lower computation time and higher consistency between sPCA and rsPCA outcomes. Since when applying sPCA, it is interesting to gain a better interpretability of the dataset selecting only relevant variables, from now on, we will focus on the models calculated considering high sparsity.

In this situation, for the ASLS algorithm considering the models calculated with 6 sPCs, the optimal situation, i.e., the best compromise between computation time and consistency between sPCA and rsPCA

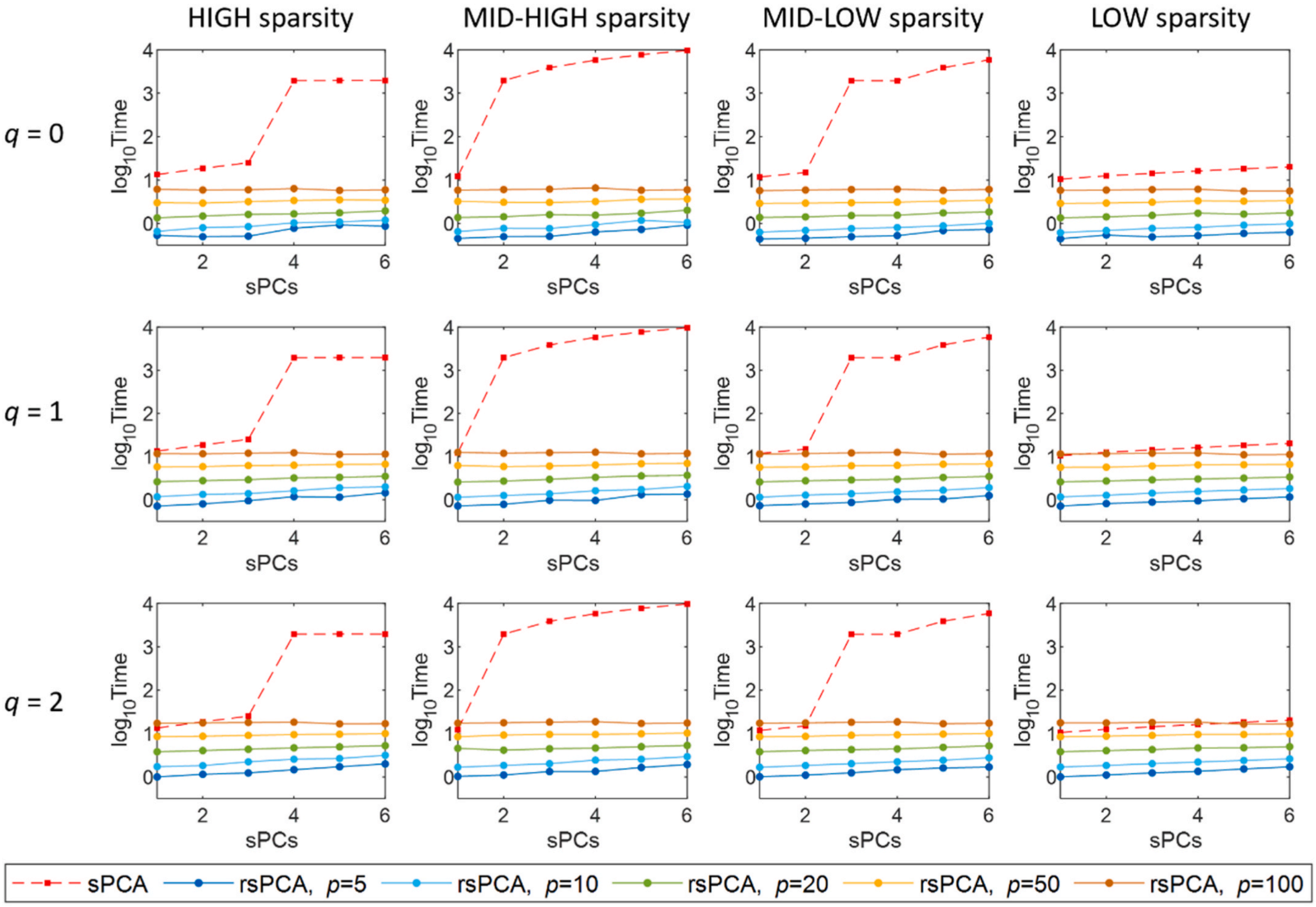


Fig. 7. Computation time (expressed in logarithmic scale) for the sPCA and rsPCA models calculated considering the deflation-based algorithm. Red colour refers to the full sPCA models while the other colours refer to the rsPCA models calculated with varying values of p: blue colour for $p = 5$, cyan colour for $p = 10$, green colour for $p = 20$, orange colour for $p = 50$ and brown colour for $p = 100$. (For interpretation of the references to colour in this figure legend, the reader is referred to the Web version of this article.)

Table 2
Summary information about sPCA and the selected rsPCA models considering both ASLS and deflation based algorithms to achieve sparsity.

		ASLS	Deflation
sPCA	# PCs	6	6
	Sparsity level	High sparsity (c = 2.30)	High sparsity (c = 2.30)
	Computation time (s)	43.72	1955.37
rsPCA	# PCs	6	6
	Sparsity level	High sparsity (2.30)	High sparsity (2.30)
	# iterations (q)	2	2
	Oversample (p)	10	5
	Computation time (s)	2.85	2.00

results, corresponds to the rsPCA model calculated with $q = 2$ and $p = 10$. The computation time of this model is equal to 2.85 s, while the computation time of the corresponding sPCA model is 43.72 s (Table 2). Even if the computation time of the sPCA model is not excessively high, it has to be considered that rsPCA allowed to gain the same results with a computation time reduction of about 93.5%. This suggests that it could be possible to simultaneously analyse a higher hyperspectral image dataset than the one considered at this time and obtain the results much faster using the randomisation method.

Moving to the deflation-based algorithm, the benefits of rsPCA on computation time are even more evident. In this case, the optimal rsPCA model was calculated with $q = 2$ and $p = 5$, corresponding to a computation time of 2.00 s. The corresponding sPCA model has a computation time of 1955.37 s (≈32 h), as shown in Table 2. In this case, randomisation reduces the computation time by about 99.9%.

For both ASLS and deflation-based algorithms, Table 3 reports the percentage of explained variance for each component, the correlation coefficients between the loading vectors of sPCA and the selected rsPCA model as well as the scores MSD% values to also compare the scores of the two models. It is possible to observe that the correlation coefficients between the loading vectors are always equal to 1.000 except for sPC3 and sPC4 for ASLS and deflation-based algorithms, respectively, where this value is equal to 0.999.

Furthermore, the scores MSD% values reported in Table 3 suggest that the differences in the score values between sPCA and rsPCA can be considered negligible for both algorithms to achieve sparsity, further highlighting the benefits of the randomised approach for data reduction.

In addition, it is possible to observe that considering both ASLS and deflation-based algorithms the percentage of explained variance of each sPC is comparable between sPCA and rsPCA. As previously highlighted, ASLS algorithm computes all the components simultaneously and the sPCS are not orthogonal, and this is also reflected in the explained variance.

The normalised score images of the sPCA model calculated with high sparsity, 6 sPCs and ASLS algorithm are reported in Fig. S10 of Supplementary Material, together with the difference images between the

Table 3
Comparison between sPCA and the selected rsPCA models considering both ASLS and deflation-based algorithms to achieve sparsity. *% E.V. is the percentage of explained variance; **Loadings CC is the correlation coefficient between sPCA and rsPCA loading vectors.

	ASLS				Deflation			
	sPCA	rsPCA			sPCA	rsPCA		
	% E.V.*	% E. V.*	Loadings CC**	Scores MSD%	% E.V.*	% E. V.*	Loadings CC**	Scores MSD%
sPC1	22.37	23.24	1.000	2.22E-03	28.73	28.39	1.000	7.18E-06
sPC2	0.64	0.64	1.000	6.67E-03	15.54	15.40	1.000	1.16E-04
sPC3	0.21	0.19	0.999	5.37E-02	12.90	12.71	1.000	1.30E-03
sPC4	1.33	1.09	1.000	1.80E-03	8.58	8.71	0.999	5.27E-03
sPC5	1.34	1.40	1.000	4.19E-02	6.11	6.17	1.000	7.11E-03
sPC6	1.97	2.02	1.000	1.69E-02	5.48	5.41	1.000	7.17E-03

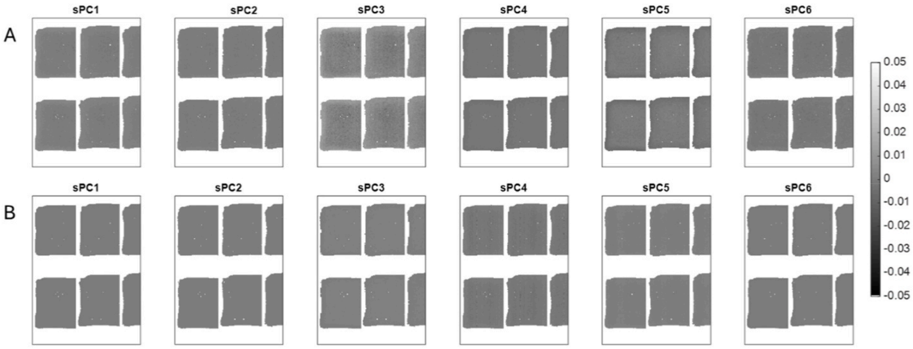


Fig. 8. Magnification of the difference images between normalised score images of sPCA and rsPCA models calculated considering ASLS (A) and deflation based (B) algorithms to achieve sparsity, reported in Fig. S9 and Fig. S10 of Supplementary Material, respectively. The magnification reports the bread slices placed in the top left corner of the original hyperspectral image.

normalised sPCA scores and the corresponding normalised scores of the selected reduced model calculated with $q = 2$ and $p = 10$. Considering the difference images, it is possible to observe that there are not appreciable variations between the score images of the two models and only sPC3 shows some minor differences that are mainly due to the noise caused by the irregular surface of the bread slices (Fig. 8A).

Similar results can be retrieved considering the comparison between sPCA and rsPCA ($q = 2$, $p = 5$) score images obtained using deflation-based algorithm (Fig. S11 of Supplementary Material). Also in this case, the reduced model provides essentially the same results as the corresponding full model and, interestingly, the little variations due to the irregular surface of the samples are not clearly visible (Fig. 8B).

5. Conclusions

This manuscript opens up the possibility of using randomisation methods as a previous step in sparse-based modelling for reducing drastically the computational time while keeping the stability of the sparse models. In particular, the benefits of coupling randomisation and sparse methods were evaluated in the context of unsupervised data analysis with sPCA of hyperspectral images of different nature and complexity.

One of the most important achievements obtained with rsPCA is the drastic reduction in computation time during sparse modelling. Indeed, rsPCA allowed obtaining results comparable to sPCA in few seconds of computation time, where the corresponding full sPCA models may take up to tens of hours to get to the final outcomes.

Nevertheless, since the randomisation step might compromise the variance on the full dataset, the consistency between rsPCA and the original sPCA models must be evaluated regarding the similarity of the loadings obtained with both methods.

In this paper we extensively assessed the effect of different model parameters in the consistency of the results between sPCA and rsPCA models. Indeed, when using sPCA it is necessary to define the algorithm type (e.g., ASLS or deflation-based), the number of sPCs and the sparsity level, while the randomisation method used in this study requires to tune the number of iterations and the oversampling value.

The effect of the sparsity level depends on the nature of the dataset and on the spectral information associated with the hyperspectral images, while, considering the algorithm used to achieve sparsity, ASLS resulted to be less stable than deflation when coupled with the randomisation step. Moving to the parameters specific of the randomisation method, increasing both the number of iterations and the oversampling value allows to obtain essentially the same results as sPCA from rsPCA. However, a higher oversampling value has a much stronger impact on the computation time than performing more iterations. Therefore, to ensure consistent results from rsPCA with the corresponding full model, it is advisable to increase the number of iterations. Based on the outcomes of this study, it is generally advisable to use at least one iteration, but moving to two iterations allowed to keep oversample values extremely low (i.e., 5 or 10). These findings confirm the results of previous studies where randomisation was used in the context of PCA data exploration [18,22,47].

In this paper, we have assessed the effects of coupling randomisation and sparse modelling in a limited scenario of unsupervised exploration of large hyperspectral datasets. However, the outcomes obtained in this study pave the way to the combined use of randomisation and sparse modelling in other scenarios.

- Inducing sparsity in the spatial dimension of hyperspectral images with fast computation times.
- Using the same approach also for different unsupervised and supervised models.
- Using datasets of different nature (e.g. with mass spectrometry imaging).

All these aspects have to be taken into account in further studies in order to gain a comprehensive understanding of the benefits and possible limitations of the approach presented in this paper.

CRediT authorship contribution statement

Rosalba Calvini: Writing – original draft, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **José Manuel Amigo:** Writing – original draft, Validation, Supervision, Methodology, Investigation, Formal analysis, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgments

Rosalba Calvini would like to thank the Italian funding programme Fondo sociale europeO REACT-EU - PON "Ricerca e Innovazione" 2014–2020 – Azione IV.6 Contratti di ricerca su tematiche Green (D.M. 1062 del 10/08/2021) for supporting her research (CUP: E95F21002330001; contract number 17-G-13884–4).

José Manuel Amigo thanks the Innovation Fund Denmark for supporting this research under "HyperSort - Hyper-spectral imaging for advanced textile sorting and food control" project. Besides, this research has been supported by the Elkartek Programme, Basque Government (Spain) (SMART-EYE, KK-2023/00021).

Appendix A. Supplementary data

Supplementary data to this article can be found online at https://doi.org/10.1016/j.chemolab.2024.105118.

References

[1] I.M. Johnstone, D.M. Titterington, Statistical challenges of high-dimensional data, Phil. Trans. Math. Phys. Eng. Sci. 367 (2009) 4237–4253, https://doi.org/10.1098/rsta.2009.0159.

[2] R. Gautam, S. Vanga, F. Ariese, S. Umapathy, Review of multidimensional data processing approaches for Raman and infrared spectroscopy, EPJ. Tech. Instrum. 2 (2015) 8, https://doi.org/10.1140/epjti/s40485-015-0018-6.

[3] A. Csala, A.H. Zwinderman, Multivariate statistical methods for high-dimensional multiset omics data analysis, in: Computational Biology, Codon Publications, 2019, pp. 71–83, https://doi.org/10.15586/computationalbiology.2019.ch5.

[4] P. Filzmoser, M. Gschwandtner, V. Todorov, Review of sparse methods in regression and classification with application to chemometrics, J. Chemom. 26 (2012) 42–51, https://doi.org/10.1002/cem.1418.

[5] C. Croux, P. Filzmoser, H. Fritz, Robust sparse principal component analysis, Technometrics 55 (2013) 202–214, https://doi.org/10.1080/00401706.2012.727746.

[6] H. Chun, S. Keleş, Sparse partial least squares regression for simultaneous dimension reduction and variable selection, J R Stat. Soc. Series B Stat. Methodol. 72 (2010) 3–25, https://doi.org/10.1111/j.1467-9868.2009.00723.x.

[7] K.-A. Lê Cao, D. Rossouw, C. Robert-Granié, P. Besse, A sparse PLS for variable selection when integrating omics data, Stat. Appl. Genet. Mol. Biol. 7 (2008), https://doi.org/10.2202/1544-6115.1390.

[8] K.-A. Lê Cao, S. Boitard, P. Besse, Sparse PLS discriminant analysis: biologically relevant feature selection and graphical displays for multiclass problems, BMC Bioinf. 12 (2011) 253, https://doi.org/10.1186/1471-2105-12-253.

[9] L. Clemmensen, T. Hastie, D. Witten, B. Ersbøll, Sparse discriminant analysis, Technometrics 53 (2011) 406–413, https://doi.org/10.1198/TECH.2011.08118.

[10] K. Huang, D. Zheng, J. Sun, Y. Hotta, K. Fujimoto, S. Naoi, Sparse learning for support vector classification, Pattern Recogn. Lett. 31 (2010) 1944–1951, https://doi.org/10.1016/j.patrec.2010.06.017.

[11] N. Subrahmanya, Y.C. Shin, Sparse multiple kernel learning for signal processing applications, IEEE Trans. Pattern Anal. Mach. Intell. 32 (2010) 788–798, https://doi.org/10.1109/TPAMI.2009.98.

[12] Y. Li, A. Ngom, Sparse representation approaches for the classification of high-dimensional biological data, BMC Syst. Biol. 7 (2013) S6, https://doi.org/10.1186/1752-0509-7-S4-S6.

[13] P. Boileau, N.S. Hejazi, S. Dudoit, Exploring high-dimensional biological data with sparse contrastive principal component analysis, Bioinformatics 36 (2020) 3422–3430, https://doi.org/10.1093/bioinformatics/btaa176.

[14] J. Shi, Q. Jiang, Q. Zhang, Q. Huang, X. Li, Sparse kernel entropy component analysis for dimensionality reduction of biomedical data, Neurocomputing 168 (2015) 930–940, https://doi.org/10.1016/j.neucom.2015.05.032.

[15] J. Ye, J. Liu, Sparse methods for biomedical data, ACM SIGKDD Explorat. Newslet. 14 (2012) 4–15, https://doi.org/10.1145/2408736.2408739.

[16] D. Lin, H. Cao, V.D. Calhoun, Y.-P. Wang, Sparse models for correlative and integrative analysis of imaging and genetic data, J. Neurosci. Methods 237 (2014) 69–78, https://doi.org/10.1016/j.jneumeth.2014.09.001.

[17] R. Calvini, A. Ulrici, J.M. Amigo, Practical comparison of sparse methods for classification of Arabica and Robusta coffee species using near infrared hyperspectral imaging, Chemometr. Intell. Lab. Syst. 146 (2015) 503–511, https://doi.org/10.1016/j.chemolab.2015.07.010.

[18] S. Kucheryavskiy, Blessing of randomness against the curse of dimensionality, J. Chemom. 32 (2018), https://doi.org/10.1002/cem.2966.

[19] P. Drineas, R. Kannan, M.W. Mahoney, Fast Monte Carlo algorithms for matrices III: computing a compressed approximate matrix decomposition, SIAM J. Comput. 36 (2006) 184–206, https://doi.org/10.1137/S0097539704442702.

[20] J.P. Cruz-Tirado, M. Oliveira, M. de Jesus Filho, H.T. Godoy, J.M. Amigo, D. F. Barbin, Shelf life estimation and kinetic degradation modeling of chia seeds (Salvia hispanica) using principal component analysis based on NIR-hyperspectral imaging, Food Control 123 (2021) 107777, https://doi.org/10.1016/j.foodcont.2020.107777.

[21] N. Halko, P.G. Martinsson, J.A. Tropp, Finding structure with randomness: probabilistic algorithms for constructing approximate matrix decompositions, SIAM Rev. 53 (2011) 217–288, https://doi.org/10.1137/090771806.

[22] K. Varmuza, P. Filzmoser, B. Liebmann, Random projection experiments with chemometric data, J. Chemom. 24 (2010) 209–217, https://doi.org/10.1002/cem.1295.

[23] J.M. Amigo (Ed.), Hyperspectral Imaging, Elsevier, 2019.

[24] H.F. Grahn, P. Geladi (Eds.), Techniques and Applications of Hyperspectral Image Analysis, Wiley, 2007, https://doi.org/10.1002/9780470010884.

[25] T. Adão, J. Hruška, L. Pádua, J. Bessa, E. Peres, R. Morais, J. Sousa, Hyperspectral imaging: a review on UAV-based sensors, data processing and applications for agriculture and forestry, Rem. Sens. 9 (2017) 1110, https://doi.org/10.3390/rs9111110.

[26] B. Lu, P. Dao, J. Liu, Y. He, J. Shang, Recent advances of hyperspectral imaging technology and applications in agriculture, Rem. Sens. 12 (2020) 2659, https://doi.org/10.3390/rs12162659.

[27] H. Huang, L. Liu, M. Ngadi, Recent developments in hyperspectral imaging for assessment of food quality and safety, Sensors 14 (2014) 7248–7276, https://doi.org/10.3390/s140407248.

[28] Y. Lu, W. Saeys, M. Kim, Y. Peng, R. Lu, Hyperspectral imaging technology for quality and safety evaluation of horticultural products: a review and celebration of the past 20-year progress, Postharvest Biol. Technol. 170 (2020) 111318, https://doi.org/10.1016/j.postharvbio.2020.111318.

[29] P.-Y. Sacré, C. De Bleye, P.-F. Chavez, L. Netchacovitch, Ph Hubert, E. Ziemons, Data processing of vibrational chemical imaging for pharmaceutical applications, J. Pharm. Biomed. Anal. 101 (2014) 123–140, https://doi.org/10.1016/j.jpba.2014.04.012.

[30] M.A. Calin, S.V. Parasca, D. Savastru, D. Manea, Hyperspectral imaging in the medical field: present and future, Appl. Spectrosc. Rev. 49 (2014) 435–447, https://doi.org/10.1080/05704928.2013.838678.

[31] G.J. Edelman, E. Gaston, T.G. van Leeuwen, P.J. Cullen, M.C.G. Aalders, Hyperspectral imaging for non-contact analysis of forensic traces, Forensic Sci. Int. 223 (2012) 28–39, https://doi.org/10.1016/j.forsciint.2012.09.012.

[32] R. Calvini, A. Ulrici, J.M. Amigo, Growing applications of hyperspectral and multispectral imaging, in: J.M. Amigo (Ed.), Hyperspectral Imaging, Data Handling in Science and Technology, vol. 32, Elsevier, 2020, pp. 605–629, https://doi.org/10.1016/B978-0-444-63977-6.00024-9.

[33] J. Burger, A. Gowen, Data handling in hyperspectral image analysis, Chemometr. Intell. Lab. Syst. 108 (2011) 13–22, https://doi.org/10.1016/j.chemolab.2011.04.001.

[34] A.A. Gowen, F. Marini, C. Esquerre, C. O'Donnell, G. Downey, J. Burger, Time series hyperspectral chemical imaging data: challenges, solutions and applications, Anal. Chim. Acta 705 (2011) 272–282, https://doi.org/10.1016/j.aca.2011.06.031.

[35] C. Ferrari, G. Foca, A. Ulrici, Handling large datasets of hyperspectral images: reducing data size without loss of useful information, Anal. Chim. Acta 802 (2013) 29–39, https://doi.org/10.1016/j.aca.2013.10.009.

[36] D. Yang, A. Lu, D. Ren, J. Wang, Rapid determination of biogenic amines in cooked beef using hyperspectral imaging with sparse representation algorithm, Infrared Phys. Technol. 86 (2017) 23–34, https://doi.org/10.1016/j.infrared.2017.08.013.

[37] B. Yousefi, S. Sojasi, C.I. Castanedo, X.P.V. Maldague, G. Beaudoin, M. Chamberland, Comparison assessment of low rank sparse-PCA based-clustering/classification for automatic mineral identification in long wave infrared hyperspectral imagery, Infrared Phys. Technol. 93 (2018) 103–111, https://doi.org/10.1016/j.infrared.2018.06.026.

[38] J.F.Q. Pereira, M.F. Pimentel, J.M. Amigo, R.S. Honorato, Detection and identification of Cannabis sativa L. using near infrared hyperspectral imaging and machine learning methods. A feasibility study, Spectrochim. Acta Mol. Biomol. Spectrosc. 237 (2020) 118385, https://doi.org/10.1016/j.saa.2020.118385.

[39] A.E. Hoerl, R.W. Kennard, Ridge regression: biased estimation for nonorthogonal problems, Technometrics 12 (1970) 55–67, https://doi.org/10.1080/00401706.1970.10488634.

[40] H. Zou, T. Hastie, Regularization and variable selection via the elastic net, J. R Stat. Soc. Series B Stat. Methodol. 67 (2005) 301–320, https://doi.org/10.1111/j.1467-9868.2005.00503.x.

[41] E. Andries, S. Martin, Sparse methods in spectroscopy: an introduction, overview, and perspective, Appl. Spectrosc. 67 (2013) 579–593, https://doi.org/10.1366/13-07021.

[42] R. Calvini, J.M. Amigo, A. Ulrici, Transferring results from NIR-hyperspectral to NIR-multispectral imaging systems: a filter-based simulation applied to the classification of Arabica and Robusta green coffee, Anal. Chim. Acta 967 (2017), https://doi.org/10.1016/j.aca.2017.03.011.

[43] R. Tibshirani, Regression shrinkage and selection via the lasso, J. Roy. Stat. Soc. B 58 (1996) 267–288, https://doi.org/10.1111/j.2517-6161.1996.tb02080.x.

[44] 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, https://doi.org/10.1093/biostatistics/kxp008.

[45] M.A. Rasmussen, R. Bro, A tutorial on the Lasso approach to sparse modeling, Chemometr. Intell. Lab. Syst. 119 (2012) 21–31, https://doi.org/10.1016/j.chemolab.2012.10.003.

[46] J. Camacho, A.K. Smilde, E. Saccenti, J.A. Westerhuis, All sparse PCA models are wrong, but some are useful. Part I: computation of scores, residuals and explained variance, Chemometr. Intell. Lab. Syst. 196 (2020) 103907, https://doi.org/10.1016/j.chemolab.2019.103907.

[47] J.P. Cruz-Tirado, J.M. Amigo, D.F. Barbin, S. Kucheryavskiy, Data reduction by randomization subsampling for the study of large hyperspectral datasets, Anal. Chim. Acta 1209 (2022) 339793, https://doi.org/10.1016/j.aca.2022.339793.

[48] J.J. Andrew, M.A. Browne, I.E. Clark, T.M. Hancewicz, A.J. Millichope, Raman imaging of emulsion systems, Appl. Spectrosc. 52 (1998) 790–796. https://opg.optica.org/as/abstract.cfm?URI=as-52-6-790.

[49] L. Duponchel, Exploring hyperspectral imaging data sets with topological data analysis, Anal. Chim. Acta 1000 (2018) 123–131, https://doi.org/10.1016/j.aca.2017.11.029.

[50] A. de Juan, M. Maeder, T. Hancewicz, R. Tauler, Use of local rank-based spatial information for resolution of spectroscopic images, J. Chemom. 22 (2008) 291–298, https://doi.org/10.1002/cem.1099.

[51] F. Marini, J.M. Amigo, Unsupervised exploration of hyperspectral and multispectral images, in: J.M. Amigo (Ed.), Hyperspectral Imaging, Data Handling in Science and Technology, vol. 32, Elsevier, 2019, pp. 93–114, https://doi.org/10.1016/B978-0-444-63977-6.00006-7.

[52] J.M. Amigo, A. del Olmo, M.M. Engelsen, H. Lundkvist, S.B. Engelsen, Staling of white wheat bread crumb and effect of maltogenic α-amylases. Part 3: spatial evolution of bread staling with time by near infrared hyperspectral imaging, Food Chem. 353 (2021) 129478, https://doi.org/10.1016/j.foodchem.2021.129478.

[53] N. Mobaraki, J.M. Amigo, HYPER-Tools. A graphical user-friendly interface for hyperspectral image analysis, Chemometr. Intell. Lab. Syst. 172 (2018) 174–187, https://doi.org/10.1016/j.chemolab.2017.11.003.