# Data reduction by randomization subsampling for the study of large hyperspectral datasets

Cruz-Tirado, J. P.; Amigo, José Manuel; Barbin, Douglas Fernandes; Kucheryavskiy, Sergey

*Published in:*
Analytica Chimica Acta

*DOI (link to publication from Publisher):*
10.1016/j.aca.2022.339793



*Publication date:*
2022

*Document Version*
Publisher's PDF, also known as Version of record

Analytica Chimica Acta 1209 (2022) 339793

Contents lists available at ScienceDirect

Analytica Chimica Acta

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

# Data reduction by randomization subsampling for the study of large hyperspectral datasets

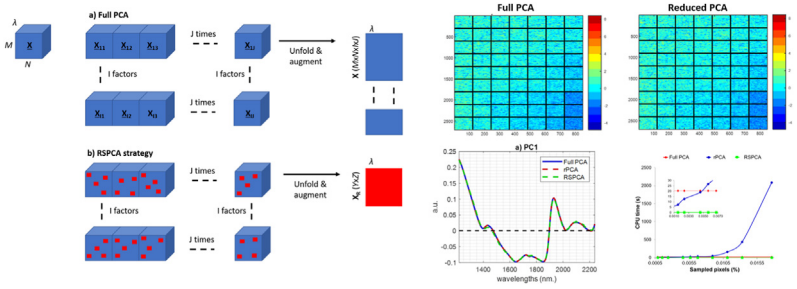
J.P. Cruz-Tirado [a], José Manuel Amigo [b, c, *], Douglas Fernandes Barbin [a], Sergey Kucheryavskiy [d]

[a] Department of Food Engineering, University of Campinas, Cidade Universitária, Rua Monteiro Lobato, 80, Campinas, SP, 13083-862, Brazil
[b] Ikerbasque, Basque Foundation for Sciences, María Díaz de Haro, 3, Bilbao, 48013, Spain
[c] Department of Analytical Chemistry, University of the Basque Country, Barrio Sarriena S/N, Leioa, 48940, Spain
[d] Department of Chemistry and Bioscience, Aalborg University, Denmark

## HIGHLIGHTS

- Reduced PCA by randomization saves computing time and RAM memory.
- The numerical accuracy of reduced models is as reliable as the full models.
- Hyperspectral time series analysis studied in a fraction of computing time and effort.
- Two reduced models tested in this manuscript with outstanding results.

## GRAPHICAL ABSTRACT




## ARTICLE INFO

*Article history:*
Received 31 January 2022
Received in revised form
28 March 2022
Accepted 30 March 2022
Available online 01 April 2022

*Keywords:*
Randomization
Sub-sampling
Data reduction
Principal component analysis
Hyperspectral imaging
Time series

## ABSTRACT

Large amount of information in hyperspectral images (HSI) generally makes their analysis (e.g., principal component analysis, PCA) time consuming and often requires a lot of random access memory (RAM) and high computing power. This is particularly problematic for analysis of large images, containing millions of pixels, which can be created by augmenting series of single images (e.g., in time series analysis). This tutorial explores how data reduction can be used to analyze time series hyperspectral images much faster without losing crucial analytical information. Two of the most common data reduction methods have been chosen from the recent research. The first one uses a simple randomization method called randomized sub-sampling PCA (RSPCA). The second implies a more robust randomization method based on local-rank approximations (rPCA). This manuscript exposes the major benefits and drawbacks of both methods with the spirit of being as didactical as possible for a reader. A comprehensive comparison is made considering the amount of information retained by the PCA models at different compression degrees and the performance time. Extrapolation is also made to the case where the effect of time and any other factor are to be studied simultaneously.



* Corresponding author. Ikerbasque, Basque Foundation for Sciences, María Díaz de Haro, 3, Bilbao, 48013, Spain.
*E-mail address:* josemanuel.amigo@ehu.eus (J.M. Amigo).

https://doi.org/10.1016/j.aca.2022.339793

**Contents**



## 1. Introduction

### 1.1. Series of hyperspectral images

Since its appearance in the early '70s in remote sensing [1–3], hyperspectral imaging (HSI) has spread to various fields of application, including, among others, pharmaceutical research and production [4,5], food science and safety [6], forensic science [7,8] or biomedical science [9]. Regardless of the spectroscopic sensor used (Near Infrared, Middle Infrared, Raman, among others), the rapid dissemination of HSI is mostly due to the feasibility of acquiring an entire spectrum for every single pixel in which the depicted area of a sample is divided and the availability of Chemometric methods to analyze such amount of information [10].

A hyperspectral image is generally represented by a three-way data array $\underline{\mathbf{X}}$ ($M$ x $N$ x $\lambda$), where each slice of the three-way data array corresponds to an image with the spatial dimensions being ($M$ x $N$) at each wavelength ($\lambda$) (Fig. 1). The true potential of such datasets is normally exploited with dedicated algorithms [11], where unsupervised and supervised methodologies are implemented for exploration, regression, resolution or classification purposes [12].

There are situations where the sample under study changes over time (or over another factor like, e.g., pH) or where many images need to be analyzed together to compare them with the scientific aim of unraveling patterns occurring between the different nature of the samples. In order to perform this analysis, a common action is the concatenation/augmentation of the three-way data arrays, as shown in Fig. 1a, to generate an enormous three-way data array. The augmentation procedure and further analysis of all the samples under the umbrella of the same model allow obtaining surface patterns related to the spectral evolution considering all the variance sources of the augmented three-way data array simultaneously.

One of the most common methodologies to analyze the augmented three-way data array is the well-known unsupervised/exploratory analytical method Principal Component Analysis (PCA). PCA requires a previous step of unfolding the three-way data array into a matrix $\mathbf{X}$ ($MNIJx\lambda$), where each entry in $\mathbf{X}$ is the reflection of one pixel at the corresponding wavelength [13] (Fig. 1a). Here, $MNIJ$ is the total number of pixels in the augmented three-way data array (product of spatial dimension of one image to the number of factors, $I$, and to the number of time points, $J$, in the series). The unfolded final matrix $\mathbf{X}$ can contain from hudreds of thousdands to several millions of rows (which correspond to the pixels in the three-way data array). Operating with such big matrices might become a big issue since their analysis normally leads to a lack of memory problems, depending on the computer used, or too long calculation times [14]. In order to overcome the abovementioned limitations, several strategies can be applied.

One of the most powerful strategies, yet not commonly used, is the use of randomized algorithms to construct an approximate matrix factorization that, using a small fraction of the information of **X,** can provide the same results.

### 1.2. Randomized algorithms for data reduction

The data reduction is, basically, the compression of $\mathbf{X}$ in the $MNIJ$ direction into $\mathbf{X_R}$, which is then factorized to generate a low-rank matrix approximation of the original $\mathbf{X}$ [23,25]. That is, the major aim and, at the same time, the major difficulty in the data reduction operations is that the space represented by $\mathbf{X_R}$ must retain all the relevant information (variance) from $\mathbf{X}$ [15]. The most important applications of data reduction are 1) the decrease of the total memory needed to store a sample [24], and 2) the ability to select a representative subset of pixels to perform a model with the assurance that the model performed in the subset sample is as representative as possible of the information in the whole three-way data array.

To exemplify how data reduction can benefit the analysis of large datasets, we will use the most popular deterministic method to explain the major sources of variance in the samples, principal

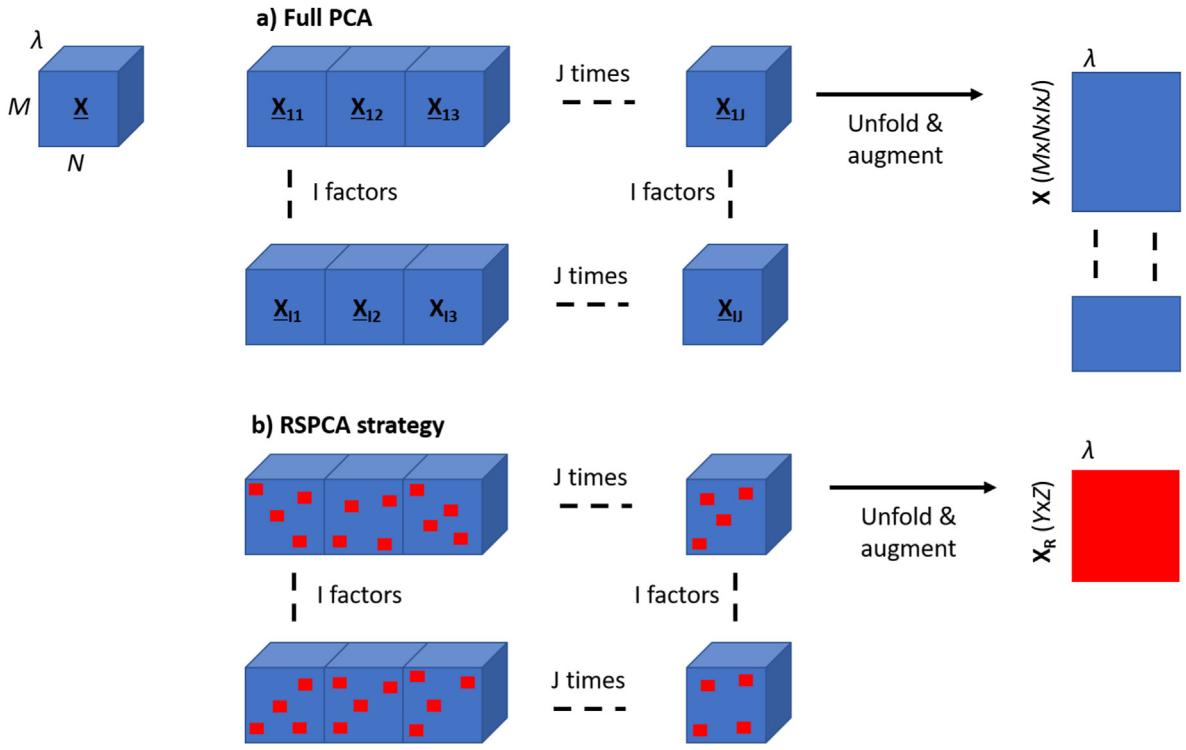



**Fig. 1.** Graphical representation of a) Full (conventional) PCA model and a b) RSPCA model retaining important representative information from the original dataset (red squares representing the selected pixels and being compiled to form $\mathbf{X_R}$). (For interpretation of the references to colour in this figure legend, the reader is referred to the Web version of this article.)

component analysis (PCA) [16]. As we will explain later, the most straightforward way of defining PCA is that it factorizes **X** into the product of two sub-matrices, the so-called scores (**T**) and loadings (**P**):

$$\mathbf{X} = \mathbf{T}\mathbf{P}^{\mathrm{T}} + \mathbf{E} \quad \text{Eq. 1}$$

Where **E** represents the residual matrix (i.e., a matrix containing variation in **X**, which is not captured by the PCA model). Analogously, the reduced matrix $\mathbf{X_R}$ can also be factorized as follow:

$$\mathbf{X_R} = \mathbf{T_R}\mathbf{P_R^{\mathrm{T}}} + \mathbf{E_R} \quad \text{Eq. 2}$$

As defined earlier, the target of data reduction is to achieve that **X** and $\mathbf{X_R}$ span the same variance space so that the loadings of the full model ($\mathbf{P}^{\mathrm{T}}$) and the reduced model ($\mathbf{P_R^{\mathrm{T}}}$) are as similar as possible. If this is achieved, it will be possible to calculate scores ($\widehat{\mathbf{T}}$) using the original data (**X**) and the loadings from the reduced data ($\mathbf{P_R^{\mathrm{T}}}$):

$$\widehat{\mathbf{T}} = XP_R \quad \text{Eq. 3}$$

Consequently, $\widehat{\mathbf{T}}$ will contain the same information (or as close as possible) as in **T** but calculated from a reduced dataset, saving time and computing effort.

PCA algorithms used in Chemometrics have been optimized for the cases when the number of observations is smaller than the number of variables [17], not being the case of HSI data. In this context, randomness is shown as one simple, fast, efficient and a less complex option to obtain a low-rank approximation matrix of original HSI data [13,17–20]. The literature offers several alternatives to obtain the most accurate $\mathbf{X_R}$ matrix. One popular example is to solve deterministic problems by random sampling with the Monte Carlo method and its adaptations [21,22]. Options like the performance of wavelets transformation for compressing the data before SVD [23], or modelling subspaces of PCA that evolve with time [15] have been also tested with very optimal compression results. Another solution is to perform random projections to obtain a low-rank matrix [24–26]. This probabilistic method begins with the projection of the original data to a set of randomly taken vectors generating a much smaller matrix, with linearly independent columns [17] (equation (2)). Then, the matrix is factorized, and the resultant singular vectors are back-projected to the original data space [13] (Eq. (3)). These methods turn out to be faster and more robust than deterministic methods, and they have been quite useful for solving clustering and classification chemometric problems [27].

### 1.3. Objectives of this tutorial

Whichever strategy to be followed, there is always a fundamental issue to solve, that is how to be sure that the reduced data is as representative as the original data and, still, be able to perform the PCA analysis in a shorter time or with less requirement of computing power. This tutorial explores two randomized approaches [28,29]: (1) The one proposed by Halko et al. [19] (called from now on as *randomized PCA* - rPCA), and (2) the randomized subsampling PCA (RSPCA) reported by Cruz-Tirado et al. [30]. Recently, Kucheryavskiy [17] reported a dedicated work to explore the impact of conventional PCA algorithms (eigenvectors of variance-covariance matrix, singular value decomposition (SVD) and Nonlinear iterative partial least squares (NIPALS)) on the randomization of hyperspectral images. Their results show that the

SVD algorithm is computationally more stable. Therefore, we will use SVD as a basis to calculate the full and reduced PCA models. The results obtained using both randomization algorithms will be compared to corresponding full PCA models using the percentage of variance explained, the correlation coefficients of the loadings, the variance of the residuals between the scores of different models, and the CPU-processing time.

Although both algorithms are compared, regardless of other literature approaches, this manuscript does not intend to conclude the superiority of one of them over the other. On the contrary, the manuscript intends to introduce both methods to a reader, underlining pros and cons of each one for PCA decomposition of large sets of three-way data array. A more detailed theoretical description of the methods, such as proof, mathematical calculations, stability, etc., can be found in the references by Halko et al. [19] and by Cruz-Tirado et al. [30]. Besides, the major intention of this tutorial paper is to encourage researchers to adopt randomization strategies in their analysis. For this purpose, the complete MATLAB code of both approaches is provided in the appendixes, together with a complete demonstration code in the supplementary material.

This paper is organized as follows. First, a brief description of reduced PCA using the SVD algorithm for computational experiments is presented. Next, we describe how the two selected methods for data reduction work. The next section introduces the data sets used for testing the algorithms: (1) Time series Near Infrared hyperspectral images (NIR-HSI) of stored chia seeds and (2) Time series NIR-HSI of stored enzymatically treated bread. The description is followed by a section, where we compare exhaustively the results obtained for the Full PCA and the two randomized alternatives presented here. The manuscript ends with the conclusions and two appendices with the complete Matlab code for RSPCA and rPCA. Moreover, the manuscript includes supplementary material with extra figures and a complete Matlab code to demonstrate the application of the randomized methods.

## 2. Theory

PCA [26] is, arguably, one of the most popular and effective chemometric tool to explore HSI and other high dimensional datasets. PCA aims at extracting the major sources of variance (i.e. latent information), called principal components (PCs), which are orthogonal (not correlated) to each other. The orientation of every PC in the original variable space is defined by a unit vector, which is called *loading*. The coordinates of data points, being projected to such vector are known as *scores*.

Following the nomenclature in Fig. 1, **T** is the *score* matrix with dimensions *MNIJ* x *A*, where *A* is the number of principal components (PCs), and $\mathbf{P}^{\mathrm{T}}$ is the *loading* matrix (A x $\lambda$), where $\lambda$ is the number of variables (wavelength). The superscript T in the loadings denotes the transpose of the matrix. **E** (*MNIJ* x $\lambda$) accounts for the residual variation which is not captured by the PCA model. In HSI, the scores are refolded to the original dimensions of the image (*M* x *N*) to form the corresponding chemical maps at *I* and *J* conditions.

As mentioned before, we used the SVD algorithm for computing PCA, which despite being slower for large data sets, is computationally more stable [17]. SVD algorithm factorizes matrix **X** to a form shown in equation (4):

$$\mathbf{X} = \mathbf{USV}^{\mathrm{T}} \quad \text{Eq. 4}$$

Where **U** (left singular vector) and $\mathbf{V}^{\mathrm{T}}$ (right singular vector) display orthonormal bases for the column and row space of matrix **X**, respectively. **S** is a diagonal matrix containing the singular values. It should be noted that $\mathbf{V}^{\mathrm{T}}$ is the matrix, identical to PCA loadings, $\mathbf{P}^{\mathrm{T}}$. Then, the score matrix, **T**, can be computed as in equation (5):

$$\mathbf{T} = \mathbf{US} \quad \text{Eq. 5}$$

At this point, it is important to keep in mind that the main hypothesis of randomized algorithms is that for the unfolded matrix **X** (*MNIJ* x $\lambda$), where the number of observations, *MNIJ*, is very large, there is a subset of observations $\mathbf{X_R}$ that has the same (or as similar as possible) variance pattern as the original data.

### 2.1. Randomized subsampling PCA (RSPCA)

RSPCA assumes that the values of the augmented three-way data array are randomly distributed and, therefore, selection of a random subsets of pixels will capture the main variability directions of the augmented three-way data array, so the subsets can be used for computation of loadings. Thus the subsampled matrix, $\mathbf{X_R}$, is represented by a random subset of rows of the original matrix **X** (Fig. 1b). The subset selection is made using *q* iterations to assure that all the sources of variance have been captured and the model is properly validated. Given the rank *A* (number of PCs), the number of iterations, *Q*, and the sampled pixels, *p* (%), the algorithm works as follow:

For each *q* iteration:

1. Reshuffle rows of the matrix **X**
2. Select *p*% of subsampled pixels (rows of **X**), to create the matrix $\mathbf{X_{Rq}}$
3. Apply PCA decomposition to $\mathbf{X_{Rq}}$ ($\mathbf{X}_{Rq} = \mathbf{USV}^{T}$, $\mathbf{P_{Rq}} = \mathbf{V}$)
4. Calculate the scores in the original space $\widehat{\mathbf{T}_q} = \mathbf{XP_{Rq}}$

In order to assure that the predicted loadings $\mathbf{P_{Rq}}$ have been calculated a representative number of times to obtain representative values for $\widehat{\mathbf{T}_q}$, the number of *Q* iterations must be carefully chosen. Then, the final scores are calculated as a mean component-wise value:

$$\widehat{\mathbf{T}} = \frac{\sum_{q=1}^{Q} \widehat{\mathbf{T_q}}}{Q} \quad \text{Eq. 6}$$

This procedure is simple, and since the selection is random, it does not require much time or much computing power. Moreover, the sign ambiguity can be easily controlled from each iteration by fixing the starting point in the SVD calculations. Nevertheless, this assumption poses several important concerns like obtaining the most appropriate percentage of pixels to be chosen (*p*) or the number of iterations to be performed to assure that the random selection covers all sources of variance properly (*Q*).

### 2.2. Randomized PCA (rPCA)

As Kucheryavskiy et al. [17] stated, random projections might not be the optimal solution for capturing directions with large variance, which is essential in the exploratory analysis of hyperspectral images. rPCA does not calculate a reduced $\mathbf{X_R}$ *per se*. Instead, it finds a low-rank alternative matrix, **B**, with much fewer rows than original matrix **X**, but being a good approximation. To find **B**, we need to compute an approximate orthonormal basis, **Q**, for the range of input matrix **X**, in such a way that:

$$\mathbf{X} = \mathbf{QQ}^{\mathrm{T}}\mathbf{X} \quad \text{Eq 7}$$

The low-rank matrix **B** can be then calculated as follows:

$$\mathbf{B} = \mathbf{Q}^{T}\mathbf{X} \quad \text{Eq. 8}$$

The low dimensional matrix **B** will have the same number of columns as the original **X**. However, the number of rows will correspond to the number of columns in **Q**. Therefore, it is desired that **Q** contains as few numbers of columns as possible, but being an accurate approximation to input matrix **X**. Ideally, the number of columns in **Q** should correspond to the number of principal components (PCs), *A*. However, in reality the exact rank of the matrix **X** is not known and *A* does not always reflect the effective rank, so the number of columns is taken as a bit larger value $(A + p)$ where *p* is an oversamling parameter.

Then, we need to obtain a set of sample vectors $\mathbf{Y} = \{y_1 \ldots y_{A+p}\}$, which will span the range of the original matrix **X**. This can be achived by $\mathbf{y}_a = \mathbf{X}\mathbf{w}_a$ $(a = 1 \ldots A + p)$, where $\{\mathbf{w}_a\}$ conforms a matrix **W** of a linearly independent set of vectors. Here is the point where randomization can help. If we create the matrix **W** as a set of fully random values, e.g. taken from $N(0, 1)$, then the chance that the columns of **W** will be linearly dependent is negligible small [17]. Finally, **Q**, can be computed by using QR decomposition of matrix **Y**. In brief the algorithm looks as follows:

1. Generate matrix **W** ($\lambda$ x $(A + p)$) using random values from $N(0, 1)$
2. Compute a set of sample vectors $\mathbf{Y} = \mathbf{X}\mathbf{W}$
3. Find **Q** as an orthonormal basis for **Y** using QR decomposition: $\mathbf{Y} = \mathbf{Q}\mathbf{R}$
4. Find matrix **B** using equation 8
5. Apply PCA decomposition method to **B** ($\mathbf{B} = \mathbf{U}\mathbf{S}\mathbf{V}^{T}$, $\mathbf{P}_R = \mathbf{V}$)
6. Calculate the scores of the original pixels $\widehat{\mathbf{T}} = \mathbf{X}\mathbf{P}_R$

This algorithm can be modified by substituting $\mathbf{Y} = \mathbf{X}\mathbf{W}$ with $\mathbf{Y} = (\mathbf{X}\mathbf{X}^{T})^{q}\mathbf{X}\mathbf{W}$, where q is a small number, usually 1 or 2. This will make the computation a bit longer but will increase the numerical stability of the algorithm especially in case when **X** has a large rank with slow decay of singular spectrum.

The literature has shown empirically that even a relatively small value, e.g., $p = 5$ or 10, will work perfectly [27]. However, in this case, we varied the oversampling parameter, *p*, to evaluate its impact on **B** and, consequently, on the results obtained from the PCA. In this paper, the rank *A* was fixed to 5, the parameter *q* to 1, while the oversampling parameter *p* varied from 15 to 400.

### 2.3. Calculations

The results obtained with both algorithms will be compared to the results obtained with the full PCA model considering the percentage of explained Variance, the correlation coefficient between the loadings, the Variance of the scores, and the CPU-processing time. All the calculations were carried out in MATLAB 2020a (Mathworks, Natick, USA) environment on a Lenovo laptop (model ideapad320, China), core i5 7th generation, 2.5 GHz, 4 GB RAM and 1 TB storage.

Hyperspectral Image pre-processing and handling have been performed using HYPER-Tools v.3 [31] (freely available at https://www.hypertools.org/; last accessed March 2022). Both algorithms, RSPCA and rPCA were run using in-house codes written in MATLAB. They are included in the appendices of this manuscript.

## 3. Data sets

The two datasets used for this tutorial are good examples of cases where the spectral evolution of samples is monitored with time and an extra factor. Both datasets are available upon request (josemanuel.amigo@ehu.eus) or, alternatively, freely available at https://www.hypertools.org/(last accessed March 2022).


**Fig. 2.** a) Full experimental design for the chia seeds dataset. b) Full experimental design for the bread staling dataset.

### 3.1. Chia seeds

The chia dataset focuses on studying the evolution of the multivariate accelerated shelf-life (MASLT) of chia seeds stored at different temperatures. The MASLT test was performed seven times (0, 9, 30, 45, 60, 90 and 180 days) and at three different storage temperatures (25, 35 and 45 °C). Moreover, three replicates were measured for each temperature and time (Fig. 2a). Further details about the experimental setup and the final results are published elsewhere [30]. The whole experiment consisted of 63 hyperspectral images, where the changes of chia seeds with time, storage temperature and replicates are to be studied. Each hyperspectral image has 301 × 121 pixels measured at 164 wavelengths in the near infrared spectral range. The final augmented three-way data array had a dimension of 2709 × 847 × 164 (2 294 523 pixels measured at 164 wavelengths, generating a total of 376 301 772 data points). The spectra were pre-processed using standard normal variate (SNV) method.

### 3.2. Bread samples

The bread dataset aims at studying how the staling of white bread affects the behavior of the whole crumb surface and how that mechanism is interrupted/changed by the addition of maltogenic α-amylases. The effect of adding enzymes in the staling with time has been published elsewhere [37–39]. The experiment comprises control bread (CR) and bread containing different enzymes (EZ1 and EZ2) being stored during six periods (1, 4, 7, 10, 14 and 21 days). Two samples were studied for each bread at each storage time, and three replicates per sample were measured (see Fig. 2b for details). The whole experiment resulted in 108 hyperspectral images measured in near infrared. Each image had spatial dimensions of 144 × 106 pixels, measured at 142 wavelengths. The final augmented three-way data array $\underline{\mathbf{X}}$ had dimensions of 2592 × 636 × 142 (1 648 512 pixels measured at 142 wavelengths, generating a total of 234 088 704 data points). The hyperspectral images were pre-processed as follow: (1) background was removed by contrasting at wavelength 1400 nm, (2) spatial binning of a sub-window of 4 × 4 pixels was applied to remove surface defects and minimize the surface roughness and, (3) Savitzky-Golay second derivative (window size of 11 points and second polynomial degree).

## 4. Results

### 4.1. Chia seeds

The loadings plots for the first five PCs obtained from the chia seeds dataset are displayed in Fig. 3. Both randomized methods were performed using $A = 5$, rPCA was carried out with values of $p = 15$ (equivalent to a total of 20 pixels) and $q = 1$ (only one iteration); while RSPCA was performed using $p = 0.0177\%$ (equivalent to a total of 400 pixels) and $q = 1$ (only one iteration). At first glance, the results denote that both methods perform with an outstanding similarity to the Full PCA model for the first three PCs. The explained variance for the first three PCs (Table 1) and the Pearson's correlation coefficients between the loadings of the reduced models and the Full PCA (Fig. 1f and Table 1) were extremely similar.

Instead, more differences were found in the PC4 and PC5 for the RSPCA model. As can be appreciated, they differ greatly from those obtained with the Full PCA and the rPCA model. Despite this observation, the percentage of variance explained by PC4 and PC5 in the three models (Table 1) is very similar. This observation should not be taken as an indication that the explained variance indicates that PC4 and PC5 are explaining the same information of the sample. Instead, a comprehensive observation of the scores surfaces is needed to understand the differences between the three

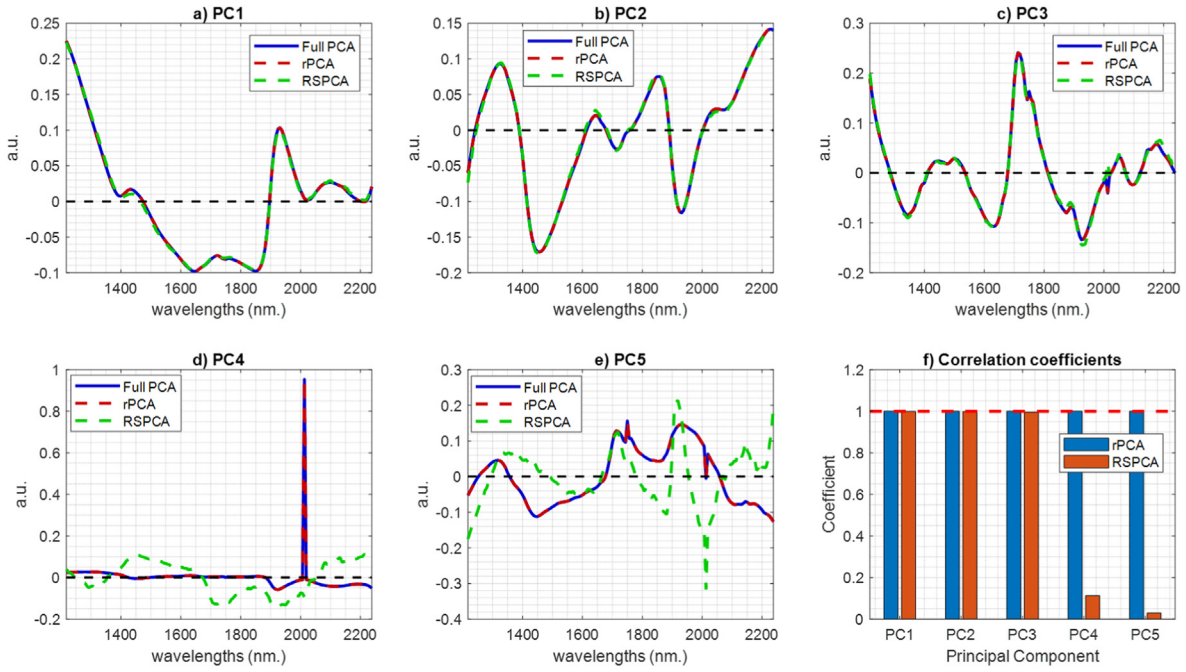



**Fig. 3.** Loading plots for PC1 to PC5 for conventional PCA, rPCA ($p = 15, q = 1, A = 5$) and RSPCA ($p = 0.0177\%, q = 1, A = 5$) obtained from analysis of chia seed samples. f) Correlation coefficients obtained between the loadings of the full PCA model and the loadings obtained for rPCA and RSPCA.

**Table 1**
Figures of merit of comparing the scores and loadings obtained by the Full PCA, rPCA and RSPCA models for both datasets.*Percentage of explained variance for each PC. **Pearson's correlation coefficient between the loading of the reduced model and the Full model. *** mean squared difference between the scores of the reduced model and the Full model.

| | Chia dataset | | | | | | |
|---|---|---|---|---|---|---|---|
| | Full PCA | rPCA | | | RSPCA | | |
| | % var | % var* | Loadings CC** | Scores var*** | % var | Loadings CC | Scores var |
| **PC1** | 42.4 | 42.4 | 1.000 | 6.0E-16 | 42.8 | 0.999 | 4.0E-04 |
| **PC2** | 26.8 | 26.8 | 1.000 | 4.0E-14 | 25.6 | 0.999 | 9.0E-04 |
| **PC3** | 12.3 | 12.3 | 1.000 | 7.0E-12 | 13.1 | 0.996 | 4.0E-04 |
| **PC4** | 11.2 | 11.2 | 1.000 | 4.0E-11 | 11.3 | 0.113 | 1.0E-01 |
| **PC5** | 7.4 | 7.4 | 1.000 | 2.0E-10 | 7.3 | 0.029 | 5.0E-02 |
| | **Bread dataset** | | | | | | |
| | **Full PCA** | **rPCA** | | | **RSPCA** | | |
| | **% var** | **% var** | **Loadings CC** | **Scores var** | **% var** | **Loadings CC** | **Scores var** |
| **PC1** | 44.4 | 44.4 | 1.000 | 2.0E-22 | 43.7 | 0.997 | 1.0E-11 |
| **PC2** | 26.2 | 26.2 | 1.000 | 9.0E-21 | 27.7 | 0.993 | 3.0E-11 |
| **PC3** | 13.8 | 13.8 | 1.000 | 3.0E-18 | 13.3 | 0.991 | 2.0E-11 |
| **PC4** | 8.4 | 8.4 | 1.000 | 4.0E-17 | 7.4 | 0.889 | 4.0E-11 |
| **PC5** | 7.3 | 7.3 | 1.000 | 1.0E-15 | 5.9 | 0.881 | 6.0E-11 |

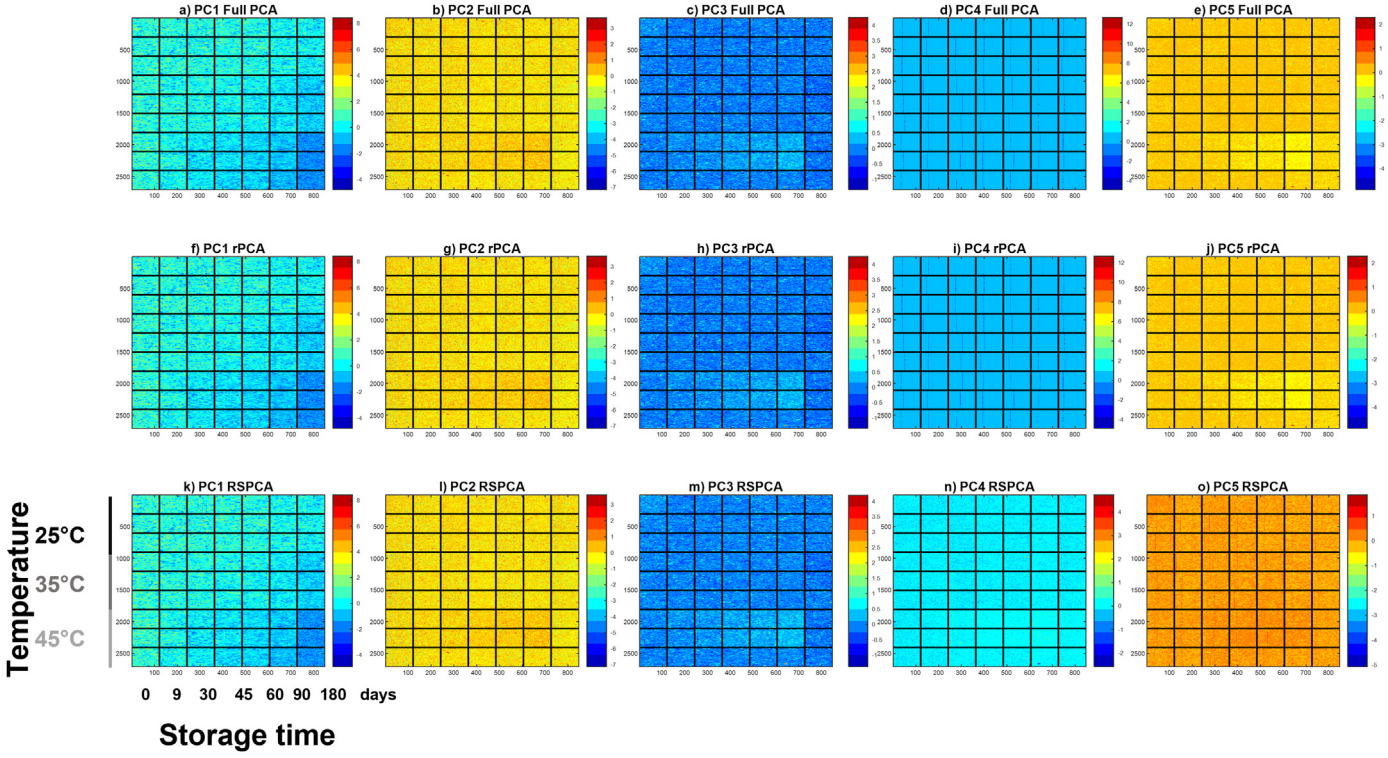



**Fig. 4.** PC1-5 scores sufaces for chia dataset using the full PCA, rPCA ($p = 15$, $q = 1$, $A = 5$) and RSPCA PCA ($p = 0.0177\%$, $q = 1$, $A = 5$).

models in the loadings.

The scores maps for the first five PCs are shown in Fig. 4. Regarding the analytical results, PC1 clearly shows the time and storage-related changes in the Chia seeds, acknowledging the results obtained elsewhere [30]. At this point, it is important to remember that the scores of all the pixels have been calculated using the loadings obtained with the matrix **B** from rPCA (20 pixels) and the subsampled matrix $\mathbf{X_R}$ obtained from RSPCA (400 pixels).

At first sight, the scores surfaces obtained for the reduced models in the first three PCs show similar information compared to those obtained with the full model, while slight differences are seen in PC4 and PC5. This observation indicates the suitability of both methods to capture large sources of variance in the dataset, allowing the extraction of the same analytical answers concerning the variability of the samples with storage time and temperature. Nevertheless, for a comprehensive analysis of the results, the quantification of the small differences between the models is needed. For that reason, Fig. 5 shows the differences obtained between the scores surfaces of the reduced models and the full model for all the pixels. The differences in the surface of the scores are displayed in Fig. S01. Moreover, the variance of the difference is calculated in Table 1.

Numerically speaking, rPCA gives the exact same solution as the solution given by the Full PCA model, where the differences between both models (Table 1) can be ascribed to negligible calculation errors in the precision of the numbers.

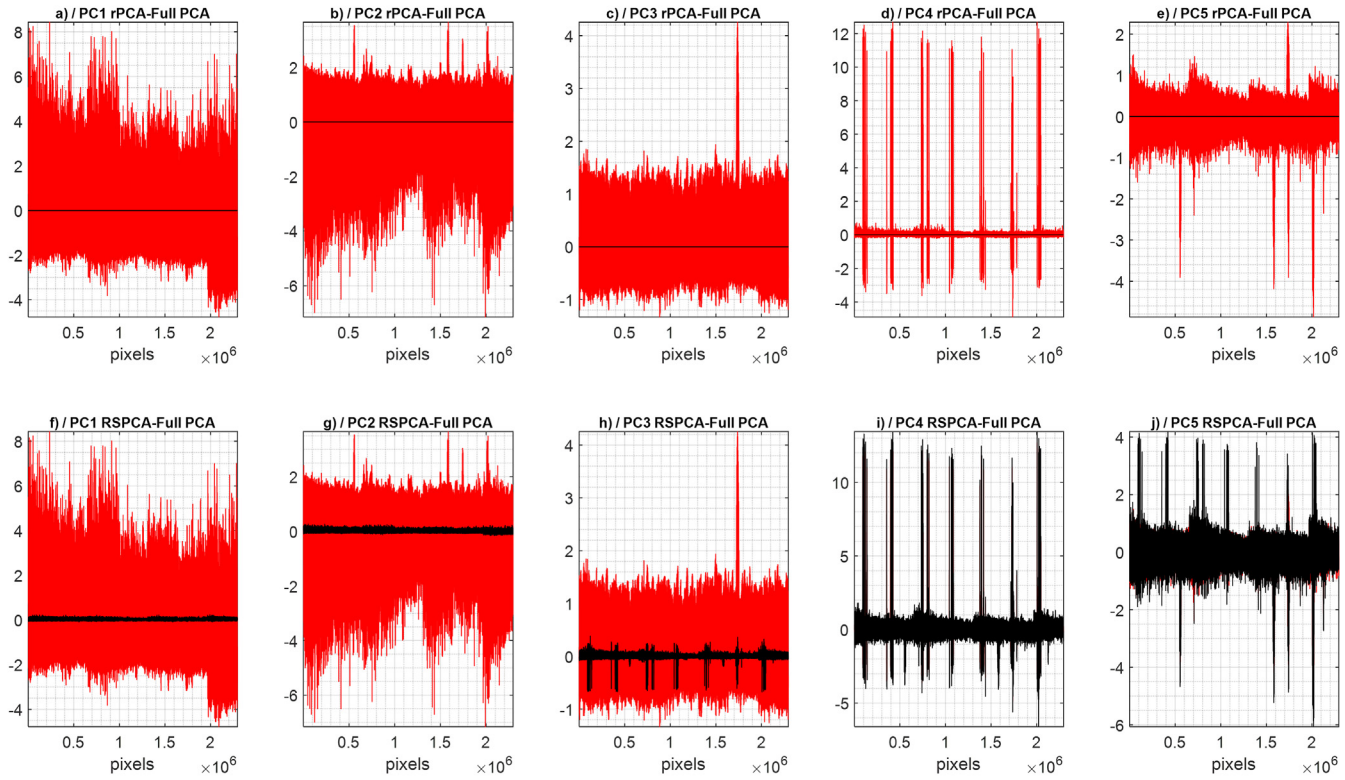



**Fig. 5.** PC scores values for the Full model (red) and the difference between the PC values obtained with the reduced models and the Full models (black). (For interpretation of the references to colour in this figure legend, the reader is referred to the Web version of this article.)

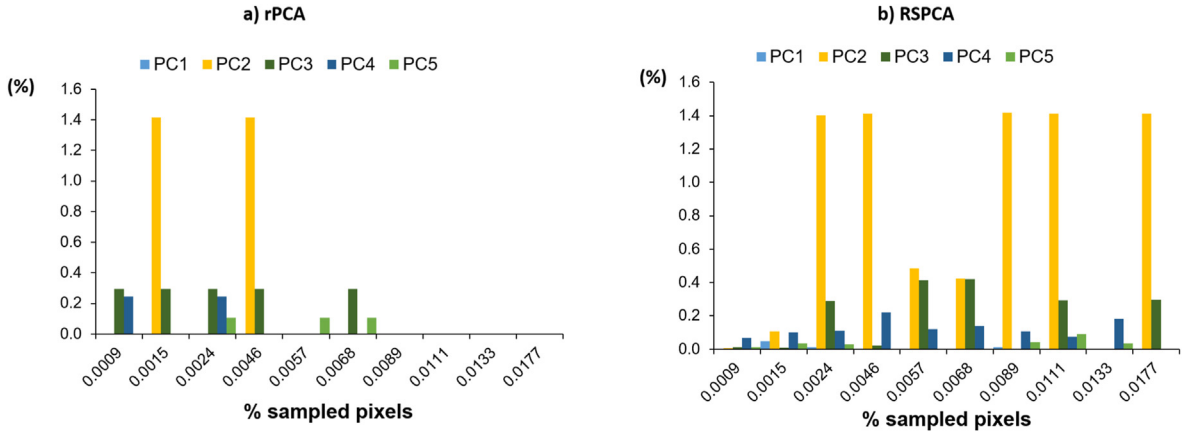



**Fig. 6.** Percentage of the mean squared difference between the scores obtained with the full model and the ones obtained with the reduced models as a function of the percentage of sampled pixels for (B) rPCA and (C) RSPCA in chia seeds hyperspectral images.

As said before, RSPCA model gives similar, yet not the same, results as the Full PCA model for the major sources of variance. Nevertheless, looking at PC4 more concisely (Fig. 3d, i, Fig. 4d and i, Figs. 5d and 9d), it can be seen that the Full PCA model and rPCA explain a small artifact coming from the measurements (vertical lines in the middle of the surface ascribed to a small artifact in the sensor). Instead, the PC4 obtained by RSPCA explains small surface features that might be ascribed to any physicochemical behaviour of the samples and not to the small artifact of the sensor.

Even being so little in error, these differences denote that a pure and simple random method like RSPCA is less sensitive to pixels containing very small portions of variance in the dataset. This fact might be compensated by sampling more XR pixels or increasing the number of iterations. The percentage of the mean squared difference between the scores obtained with the full model and the ones obtained with the reduced models has been calculated with the increase of the amount of sampled pixels (Fig. 6). The first observation is that the numerical stability of the rPCA model increases with the number of sampled pixels. This observation does not hold for the RSPCA model, where PC2 shows certain instability and dependency on the number of sampled pixels. Nevertheless, it is important to remark that the score variance for PC1 is almost zero

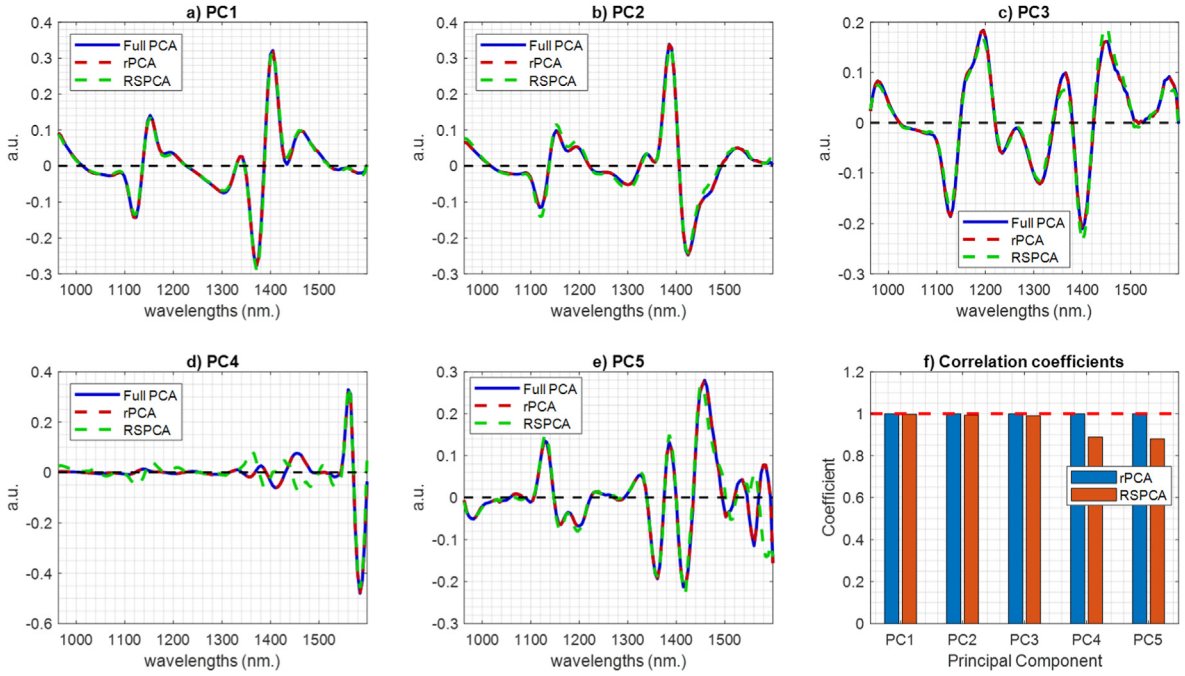



**Fig. 7.** Loading plots for PC1-3 for conventional PCA, rPCA ($p = 15$, $q = 1$, $k = 5$) and RSPCA ($p = 0.0246\%$, $q = 1$, $k = 5$) obtained from analysis of bread samples.

for both reduced models. Moreover, the differences observed in the models with the increasing number of sampled pixels are, in general, very small.

Concerning the number of iterations (Fig. S02 of the supplementary material), neither the increase in the number of iterations nor the increase in the sampled pixels did not improve the results of obtained by RSPCA.

### 4.2. Bread staling

The loadings plot obtained from the PCA analysis for the bread hyperspectral images is shown in Fig. 7 and Table 1. Randomization by rPCA was performed with values of $p = 15$, $q = 1$ and $A = 5$. RSPCA used values $p = 0.0246\%$ (400 pixels), $q = 1$ and $A = 5$. Differently to what it was observed in the previous case, now the loadings obtained for the five first PCs in the two reduced models are extremely similar to the loadings of the Full PCA model, with correlation coefficients of 1 in the case of rPCA and higher than 0.88 in the case of RSPCA.

In this case, the nonexistence of spectral artifacts makes the variation of pixel intensities more homogeneous and, therefore, it is easier in terms of probability to obtain a subspace of $\mathbf{X_R}$ that is much more representative of the variance space of $\mathbf{X}$ than in the previous case. This excellent result is extremely important since,

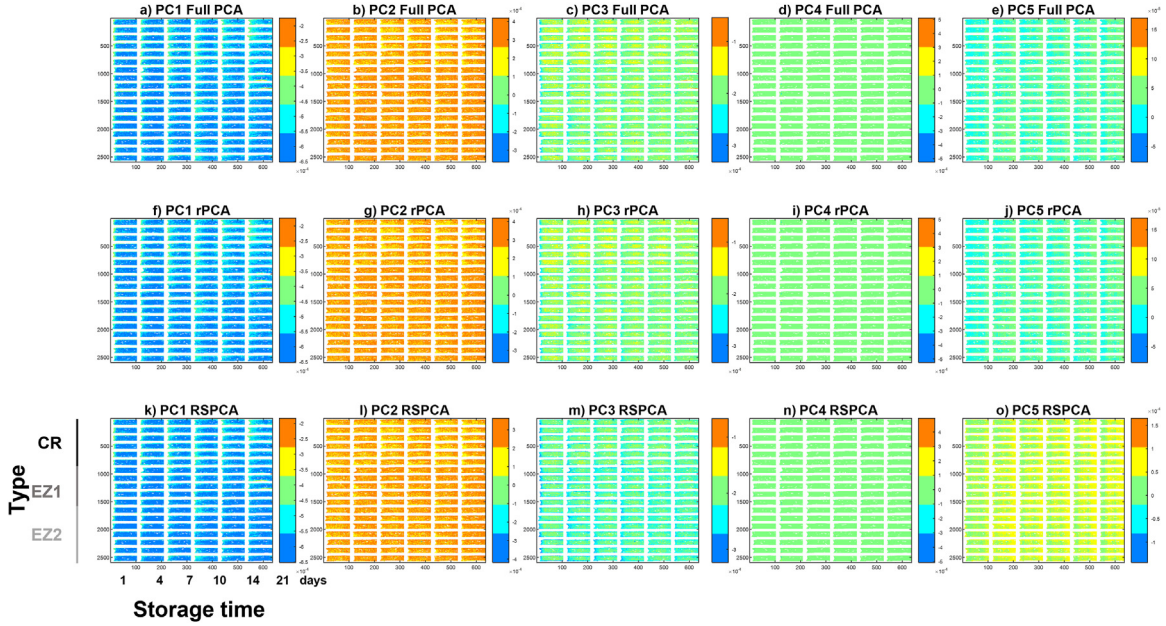



**Fig. 8.** PC1-5 scores sufaces for bread staling samples for the Full PCA model, rPCA ($p = 15$, $q = 1$, $k = 5$) and RSPCA PCA ($p = 0.0246\%$, $q = 1$, $k = 5$).

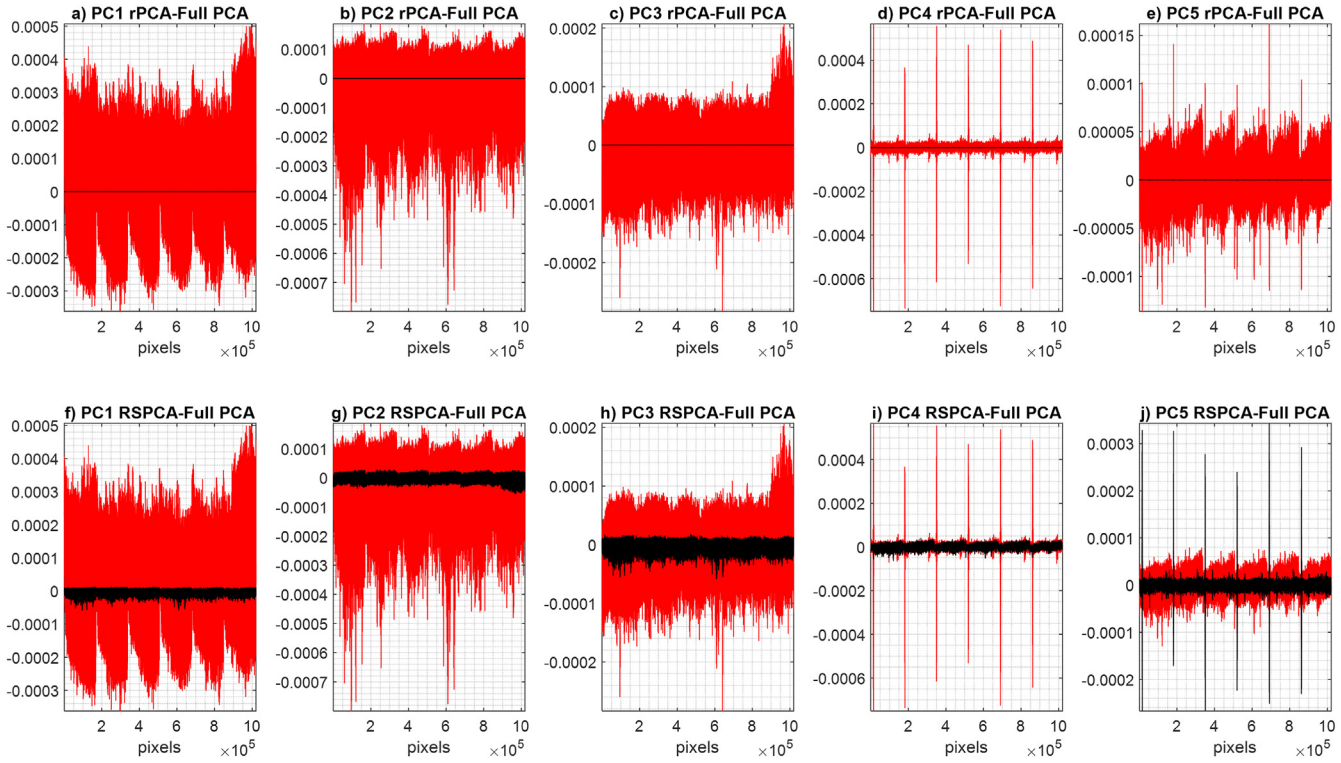



**Fig. 9.** PC scores values for the Full model (red) and the difference between the PC values obtained with the reduced models and the Full models (black). (For interpretation of the references to colour in this figure legend, the reader is referred to the Web version of this article.)

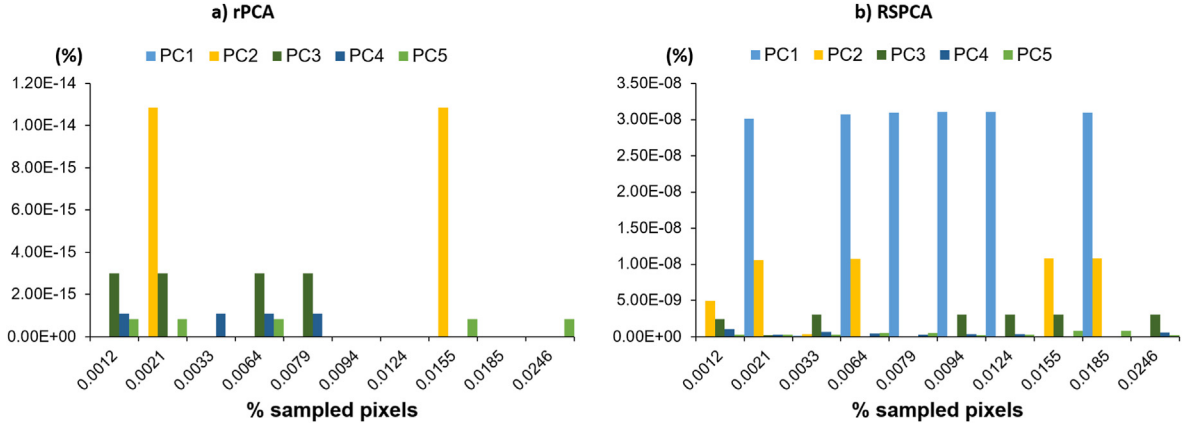



**Fig. 10.** Percentage of the mean squared difference between the scores obtained with the full model and the ones obtained with the reduced models as a function of the percentage of sampled pixels for (B) rPCA and (C) RSPCA in chia seeds hyperspectral images.

according to the analytical results found in the literature [32], the first three PCs contain substantial physico-chemical information. Therefore, all three must be studied to understand the dynamics of the bread being staled.

With the high similarity obtained for the loadings, it is expected that the similarity in the scores surfaces obtained by the three models will be high. The score maps obtained for the three models (Fig. 8) and the differences between the scores obtained with the reduced models and the Full PCA model (Fig. 9 and Fig. S03) demonstrate that assessment to a large extent.

Only very small variations were observed in the scores surfaces obtained by RSPCA in PC5, where a small artifact in the measurement of some lines has not been properly captured. The stability studies by increasing the number of sampled pixels (Fig. 10) and the number of iterations (Fig. S02) show that in this case, the differences in the models can be ascribed to a negligible numeric difference (in the range of $10^{-14}$ for rPCA and $10^{-8}$ for RSPCA), certifying the stability of the reduced models.

### 4.3. Computing performance

Once the performance of the reduced models in terms of differences with the Full PCA model has been thoughtfully evaluated and understood, it is essential to evaluate their computing

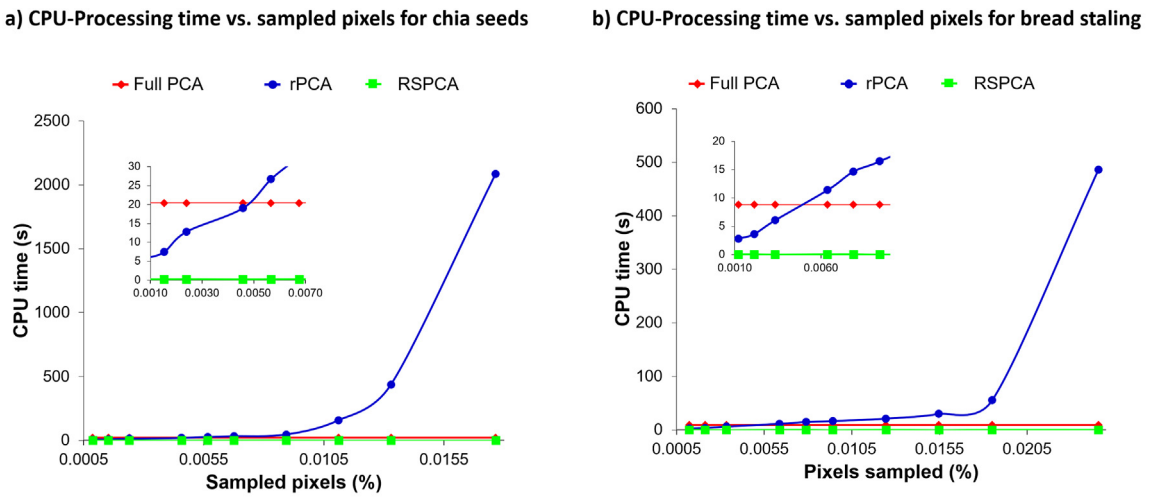



**Fig. 11.** CPU-processing time for PCA decomposition analysis for conventional and randomizes approaches for (a) chia seeds and (b) bread as a function of percentage of sampled pixels.

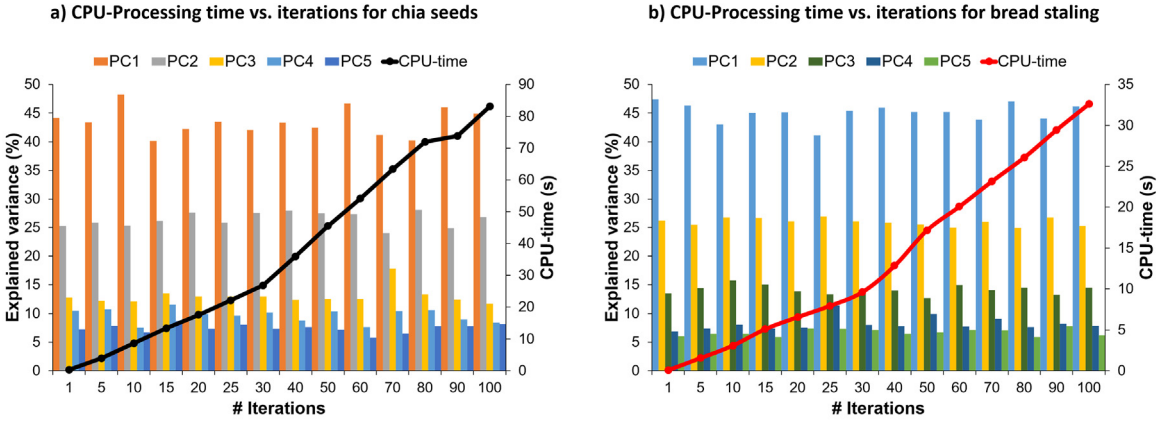



**Fig. 12.** Variation of CPU-processing time and % explained variance as a function of the number of iterations to obtain the subsampled matrix using RSPCA. (a) Chia seeds time series hyperspectral images and (b) bread time series hyperspectral images.

performance. Fig. 11 shows the evolution of the CPU-processing time as a function of the percentage of sampled pixels and the number of iterations.

In both cases, the CPU-processing time using RSPCA was much lower than Full PCA (20 s for Chia seeds and 10 s for bread) and rPCA. This is due to the simplicity of RSPCA. It can be observed that at a certain moment, with the increase in oversampling ($p$), the CPU-processing time using rPCA randomization exceeds the time consumed by the Full PCA models. In contrast to RSPCA, the CPU-processing time in rPCA increases with the number of sampled pixels. Nevertheless, this fact is not very relevant since rPCA does not depend on the number of sampled pixels but on the effective rank of **X**. Therefore, if the effective rank is 5 then 15–20 elements in **B** will be efficient regardless the size of the image.

The CPU-processing time increases linearly with the number of iterations for RSPCA (Fig. 12a and b). This increase of CPU-processing time is linearly correlated to the number of iterations. In the time spent by the conventional PCA model, RSPCA can perform around 30 iterations. This is, indeed, an optimal PCA validation strategy since, as shown elsewhere [30], the repeated predicted values of each pixel give an account of the performance of RSPCA.

## 5. Conclusions

This tutorial paper shows how two randomization methods like rPCA and RSPCA can be extraordinarily powerful to reduce computing time in large hyperspectral datasets. The two alternatives chosen for illustrating the performance of randomization methods clearly represent the wide variety of alternatives that the reader can find in the literature. The two methods reported here assume that the expected/observable changes in the time series hyperspectral images occur with enough importance/variance to capture it by explorative models like PCA easily. rPCA works by matrix triangulation, being quite efficient, using an extremely low number of sampled pixels and only one iteration. RSPCA works by iteratively random selection of a subset of pixels. Despite its simplicity, RSPCA has also demonstrated its usefulness in capturing the major sources of variance in time series hyperspectral images, performing in a much faster manner than rPCA.

Objectively speaking, the scores and loadings correlation with the full model for both data sets was greater than 0.999 for rPCA. The CPU-processing time was significantly lower for RSPCA than the conventional PCA and rPCA. rPCA always showed slightly better performance, even using an extremely small number of pixels (20

in rPCA and 400 in RSPCA). Nevertheless, rPCA only needed one iteration to give a perfectly fixed solution.

Both approaches preserve the orthogonality of the obtained principal components, accounting for inner products with difference at the range of the accuracy of the mathematical operations (i.e. between 1e-16 and 1e-17). This is an important observation since orthogonality is one of th main features of PCA models.

In general, it can be said that when an accurately numeric option is needed, rPCA has clearly demonstrated its extraordinary power. By selecting only a very small portion of the pixels, rPCA was able to capture the exact same variance space as the Full PCA model. This also includes small portions of variance that could be ascribed to unwanted effects, like in the first case presented here. With a little bit more risk, pure random alternatives like RSPCA might be a plausible alternative to filter out small variance artifacts, always bearing in mind that the number of iterations in RSPCA can be increased to obtain a more reliable model with no excessive cost of the computing time implied.

At this point it is important to highlight that the data reduction strategy to be adopted in certain scenarios strictly depends on the objective of the study to be conducted. Also, it must be remarked that it is not only important to select one strategy, but also consider how the data reduction is made in the cases when the samples follow an specific design and a certain number of random pixels at certain designed features want to be used. Randomization and subsampling randomization have been demonstrated to be excellent options when aiming at PCA compression/modelling. Nevertheless, these alternatives might not be optimal for other situations or model application, like in Multivariate Curve Resolution (MCR) scenarios [33].

The two examples shown in this paper focus on hyperspectral time series analysis. Nevertheless, the randomization strategies can also be applied when there is a need to compare different samples.

Observing the outstanding results obtained with the randomization strategies, the reader is strongly encouraged to try them. Therefore, a complete Matlab code for their implementation can be found in the appendices. Moreover, a complete code for a simulated example using both randomization methods and a comparison with a Full PCA model has been included in the supplementary material.

## 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.

## Acknowledges

J.P Cruz-Tirado acknowledges scholarship funding from FAPESP, grant number 2020/09198−1.

## Appendix A. Supplementary data

Supplementary data to this article can be found online at https://doi.org/10.1016/j.aca.2022.339793.

## Appendixes

The following Matlab codes do NOT take into consideration the pre-processing steps. The data is mean centered. This can be changed as normalization step.

### *Appendix 1*

Matlab code (functions) for RSPCA. It includes mean centering as normalization step and study of reproducibility of the ranzomization.

### *Appendix 2*

Matlab code (functions) for rPCA.

## References

[1] C.A. Lee, S.D. Gasster, A. Plaza, C.-I. Chang, B. Huang, Recent developments in high performance computing for remote sensing: a review, IEEE J. Sel. Top. Appl. Earth Obs. Rem. Sens. 4 (2011) 508−527, https://doi.org/10.1109/JSTARS.2011.2162643.
[2] G. Lassalle, Monitoring natural and anthropogenic plant stressors by hyperspectral remote sensing: recommendations and guidelines based on a meta-review, Sci. Total Environ. 788 (2021) 147758, https://doi.org/10.1016/j.scitotenv.2021.147758.
[3] M.T. Eismann, Hyperspectral Remote Sensing, SPIE Bellingham, 2012.
[4] P.-Y. Sacré, C. De Bleye, P.-F. Chavez, L. Netchacovitch, P. 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.
[5] J.M. Amigo, J. Cruz, M. Bautista, S. Maspoch, J. Coello, M. Blanco, Study of pharmaceutical samples by NIR chemical-image and multivariate analysis, TrAC Trends Anal. Chem. (Reference Ed.) 27 (2008) 696−713, https://doi.org/10.1016/j.trac.2008.05.010.
[6] 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.
[7] C.S. Silva, M.F. Pimentel, J.M. Amigo, R.S. Honorato, C. Pasquini, Detecting semen stains on fabrics using near infrared hyperspectral images and multivariate models, TrAC Trends Anal. Chem. (Reference Ed.) 95 (2017) 23−35, https://doi.org/10.1016/j.trac.2017.07.026.
[8] K.B. Ferreira, A.G.G. Oliveira, A.S. Gonçalves, J.A. Gomes, Evaluation of hyperspectral imaging visible/near infrared spectroscopy as a forensic tool for automotive paint distinction, Forensic Chem. 5 (2017) 46−52, https://doi.org/10.1016/j.forc.2017.06.001.
[9] B. Fei, Chapter 3.6 - hyperspectral imaging in medical applications, in: J.M. Amigo (Ed.), Hyperspectral Imaging, Elsevier, 2020, pp. 523−565, https://doi.org/10.1016/B978-0-444-63977.
[10] J.M. Amigo, H. Babamoradi, S. Elcoroaristizabal, Hyperspectral image analysis. A tutorial, Anal. Chim. Acta 896 (2015) 34−51, https://doi.org/10.1016/j.aca.2015.09.030.
[11] J.M. Amigo, Data mining, machine learning, deep learning, chemometrics definitions, common points and trends (spoiler alert: VALIDATE your models!), Braz. J. Anal. Chem. 8 (2021) 22−38.
[12] J.M. Amigo, Hyperspectral Imaging, Elsevier, 2020.
[13] J. Zhang, J. Erway, X. Hu, Q. Zhang, R. Plemmons, Randomized SVD methods in hyperspectral imaging, J. Electr. Comput. Eng. 2012 (2012).
[14] N. Kettaneh, A. Berglund, S. Wold, PCA and PLS with very large data sets, Comput. Stat. Data Anal. 48 (2005) 69−85, https://doi.org/10.1016/j.csda.2003.11.027.
[15] R. Vitale, A. Zhyrova, J.F. Fortuna, O.E. de Noord, A. Ferrer, H. Martens, On-The-Fly Processing of continuous high-dimensional data streams, Chemometr. Intell. Lab. Syst. 161 (2017) 118−129, https://doi.org/10.1016/j.chemolab.2016.11.003.
[16] S. Wold, K. Esbensen, P. Geladi, Principal component analysis, Chemometr. Intell. Lab. Syst. 2 (1−3) (1987) 37−52, https://doi.org/10.1016/0169-7439(87)80084-9.
[17] S. Kucheryavskiy, Blessing of randomness against the curse of dimensionality, J. Chemom. 32 (2018), e2966.
[18] S.I. Smirnov, V. V Mikhailov, V.N. Ostrikov, Application of randomized principal component analysis for compression of hyperspectral data, Curr. Probl. Remote Sens. Earth from Sp. 11 (2014) 9−17.
[19] 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.
[20] R. Vitale, P. Stefansson, F. Marini, C. Ruckebusch, I. Burud, H. Martens, Fast analysis, processing and modeling of hyperspectral videos: challenges and possible solutions, in: Compr. Chemom., 2020, https://doi.org/10.1016/b978-0-12-409547-2.14605-0.
[21] L.M. Chen, Z. Su, B. Jiang, Mathematical Problems in Data Science, Springer, 2015.
[22] 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.
[23] F. Vogt, M. Tacke, Fast principal component analysis of large data sets based

on information extraction, J. Chemom. 16 (2002) 562–575, https://doi.org/10.1002/cem.751.
[24] S.S. Vempala, The Random Projection Method, American Mathematical Soc., 2005.
[25] E. Rabani, S. Toledo, Out-of-core SVD and QR decompositions, in: Proc. 10th SIAM Conf. Parallel Process. Sci. Comput. Portsmouth, SIAM, VA, CD-ROM, 2001, pp. 1–9. Philadelphia, http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.32.7514& rep=rep1& type=pdf.
[26] N. Halko, P.-G. Martinsson, Y. Shkolnisky, M. Tygert, An algorithm for the principal component analysis of large data sets, SIAM J. Sci. Comput. 33 (2011), https://doi.org/10.1137/100804139.
[27] K. Varmuza, P. Filzmoser, B. Liebmann, Random projection experiments with chemometric data, J. Chemom. 24 (2010) 209–217.
[28] R. Dorrepaal, C. Malegori, A. Gowen, Tutorial: time series hyperspectral image analysis, J. Near Infrared Spectrosc. 24 (2016) 89–107, https://doi.org/10.1255/jnirs.1208.
[29] J.-L. Xu, A.A. Gowen, D.-W. Sun, Time series hyperspectral chemical imaging (HCI) for investigation of spectral variations associated with water and plasticizers in casein based biopolymers, J. Food Eng. 218 (2018) 88–105, https://doi.org/10.1016/j.jfoodeng.2017.09.006.
[30] 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), https://doi.org/10.1016/j.foodcont.2020.107777.
[31] 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.
[32] 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. (n.d.).
[33] C. Ruckebusch, R. Vitale, M. Ghaffari, S. Hugelier, N. Omidikia, Perspective on essential information in multivariate curve resolution, TrAC Trends Anal. Chem. (Reference Ed.) (2020), https://doi.org/10.1016/j.trac.2020.116044.

Dr. Douglas Barbin has been employed as Assistant Lecturer at the School of Food Engineering, Department of Food Engineering, at University of Campinas. He is responsible for carrying out teaching and research duties, involved in undergraduation lecturing subjects of refrigeration and unit operations in the food industry. He supervises postgraduate research students and lectures on Food Engineering courses at postgraduate level. He has experience in Process Analytical Technology (PAT) applied to the food industry. His research interests' focus on Process Analytical Technologies (PAT) applied to food processing and preservation, innovative solutions for sustainable food process development. He was granted a project funded by São Paulo Research Foundation (FAPESP), as a grant for Young Researchers in Emerging Centers.

Born in 1978. He obtained his PhD in Chemistry at the Autonomous University of Barcelona, Spain, in 2007. His current position is IKERBASQUE Research Professor and Distinguished Professor at the department of Analytical Chemistry of the University of the Basque Country, Spain. His Current research interests include hyperspectral, multispectral and digital image analysis applied in many different fields (Food Sciences, Environmental Modeling, Forensic Sciences, Pharmaceutical Production, etc.) and teaching chemometrics. He has authored more than 170 publications (130+ peer-reviewed papers, books, book chapters, proceedings, etc.) and given more than 60 conferences at international meetings and given more than 30 courses world-wide. He has supervised or is currently supervising several Masters, Post Docs and PhD students and he is an editorial board member of four scientific journals within chemometrics and analytical chemistry. 2014 - Chemometrics and Intelligent Laboratory Systems Award for his achievements in the field of Chemometrics. 2019 - Thomas Hirschfeld Award for his achievements in the field of Near Infrared Technology.

Luis Jam Pier Cruz Tirado obtained his M.S. degree in Food Engineering in 2020 from the University of Campinas (Brazil). From the same year, he is a PhD student at the Food Engineering Department of the University of Campinas, developing his research work at Laboratory of Innovation in Foods (LINA), inner Process Analytical Technology Research Group. Currently, his research activities are focused on the development of new analytical methods based on Near Infrared Spectroscopy and Hyperspectral Imaging and Machine Learning for food quality and safety control.

Born in 1976 in no more existing country, USSR. Defended PhD in Physics and Mathematics in 2001 and right after that he took a break and was not active as a researcher for several years. In 2004 started rebooting his carrier and decided to change research interests towards Chemometrics. He started from the scratch in 2007 as assistant professor at AAU and has been gradually developing his career as chemometrician since then.In his spare time Sergey works on various pet-projects. Thus, from 2013 he has been developing R package “mdatools” for multivariate data analysis and preprocessing with one-two major releases every year. The results of these activities can be found at https://mda.tools and hist GitHub https://github.com/svkucheryavski