Received: 28 November 2025 | Revised: 3 June 2026 | Accepted: 11 September 2026 | Published online: 16 October 2026

RESEARCH ARTICLE

Journal of Data Science and Intelligent Systems
2026, Vol. 00(00) 1–37
DOI: 10.47852/bonviewJDSIS62028503

BON VIEW PUBLISHING

Dimensionality Reduction Strategy on Four-Dimensional Data Cubes with Independent Slices and Hyper-Slices Focusing on Variability Structure and Components

Zacharenia Kyrana[1,*], Nikolaos Papafilippou[1], Emmanouil D. Pratsinakis[1], Alexandra-Maria Michaelidou[2], Christos Dordas[1], Angelos Markos[3] and George C. Menexes[1]

[1] *Laboratory of Agronomy, Aristotle University of Thessaloniki, Greece*

[2] *Laboratory of Dairy Technology, Aristotle University of Thessaloniki, Greece*

[3] *Department of Primary Education, Democritus University of Thrace, Greece*

Abstract: Data cubes are p-dimensional data structures ($p \geq 3$). In most biological sciences, the key point of statistical data analysis is the study of variability, focusing on its structure and components. The proposed strategy was based on principal component analysis (PCA) and the idea of decomposing the total variability of a four-dimensional cube ($p = 4$) into variability between and within the cube's slices and hyper-slices. The aim of this study was to propose new insights into the dimensionality reduction of four-dimensional cubes with independent slices and hyper-slices and equal or unequal number of experimental units per slice within each hyper-slice. As an example, the simulated dataset "Basil" from the field of agriculture was used. The fourth dimension consisted of three irrigation levels (hyper-slices). The third dimension consisted of five basil varieties (slices). The second dimension consisted of seven quantitative traits-variables, and the first dimension consisted of different plants. The proposed strategy allowed for a detailed examination of the effect of slices and hyper-slices on the overall data cube structure by highlighting the dominant influence of irrigation levels. Also, the study highlighted the controversial case of the variability decomposition on unbalanced four-dimensional cubes. For future research, various clustering strategies could be applied, in terms of the distance metric and the linkage method, either to the initial cubes or to the PCA components' scores from each variability data matrix. The proposed strategy could also be applied to data cubes with greater imbalance, in terms of sample size, between slices, and hyper-slices.

Keywords: multidimensional data, vertical concatenation, principal component analysis, variability decomposition

1. Introduction

In most scientific fields and experimental procedures, data are represented as complex, multidimensional phenomena in the form of data cubes. Specifically, data cubes are p-dimensional data structures, where $p \geq 3$ [1, 2]. p is determined by the number of independent dimensions of the data, and it can be any positive integer greater than or equal to three ($p \geq 3$), depending on the data structure. Specifically, in a three-dimensional data cube ($p = 3$), the 1st dimension consists of objects (e.g., different persons, plants, or animals and, in general, different experimental or sampling units). The 2nd dimension consists of dependent or response variables (e.g., the quantitative characteristics of objects). The 3rd dimension consists of the variable of slices, that is, an independent, usually categorical, variable (e.g., different plants' varieties, different experimental treatments, and, in the field of agricultural experimentation, different blocks of experimental plots or different locations or years of experimentation). If an additional 4th dimension (hyper-slices) is added, that is another independent variable (e.g., different irrigation levels, additional experimental treatments, or other qualitative characteristics of the units), the data cube will become four-dimensional ($p = 4$) and so on [1].

Generally, data cubes appear under different structural characteristics. In the majority of scientific fields, data cubes consist of dependent slices (or else repeated measurements of the same objects per slice) [3]. In this case, the most well-known dimensionality reduction methods are Tucker and CP (CANDECOMP/PARAFAC) decomposition. These methods treat the data cube as a tensor [3].

However, in most biological sciences and agricultural experiments, data cubes consist of independent slices, where the objects are different and either unequal or equal in number per slice [3].

***Corresponding author:** Zacharenia Kyrana, Laboratory of Agronomy, Aristotle University of Thessaloniki, Greece. Email: kyranaze@agro.auth.gr

© The Author(s) 2026. Published by BON VIEW PUBLISHING PTE. LTD. This is an open access article under the CC BY License (https://creativecommons.org/licenses/by/4.0/).

This study focused on four-dimensional data cubes with independent slices and hyper-slices, where the objects were different and either unequal or equal in number per slice within each hyper-slice.

1.1. Mathematical background of balanced four-dimensional data cubes

Regarding the mathematical background of a balanced four-dimensional data cube with independent hyper-slices and slices, and equal number of objects per slice and hyper-slice, let $\underline{\mathbf{X}} = [\mathbf{X}_{11}, \mathbf{X}_{21}, \ldots, \mathbf{X}_{K1}, \mathbf{X}_{12}, \mathbf{X}_{22}, \ldots, \mathbf{X}_{K2}, \ldots, \mathbf{X}_{1G}, \mathbf{X}_{2G}, \ldots, \mathbf{X}_{KG}]$ be a four-dimensional $I \times J \times K \times G$ data cube ($p = 4$), with x_{ijkg} elements, where $i = 1, 2, \ldots, I$, $j = 1, 2, \ldots, J$, $k = 1, 2, \ldots, K$, and $g = 1, 2, \ldots, G$. The 1st dimension I denotes the rows (experimental units or objects) of the cube, which are different and equal in number per slice and hyper-slice. The 2nd dimension J denotes the columns (traits or variables) of the cube. The 3rd dimension K denotes the independent slices, while the 4th dimension G denotes the independent hyper-slices of the cube.

Alternatively, the data cube $\underline{\mathbf{X}}$ can be seen as a collection of either three-dimensional data cubes $\underline{\mathbf{X}}_k$, with dimensions $IK \times J \times G$, or two-dimensional matrices $\mathbf{X}_{kg}$, with dimensions $IKG \times J$, where each matrix represents a slice k within hyper-slice g of a data cube.

The most classic way to manage such a balanced data cube is to transform it into a two-dimensional data matrix through vertical concatenation [4]. Specifically, data cube $\underline{\mathbf{X}}$ ($I \times J \times K \times G$) is transformed into a "stacked" data matrix $\mathbf{X}$ ($IKG \times J$), with x_{ijkg} elements, where $i = 1, 2, \ldots, I$, $j = 1, 2, \ldots, J$, $k = 1, 2, \ldots, K$, and $g = 1, 2, \ldots, G$ (Figure 1).

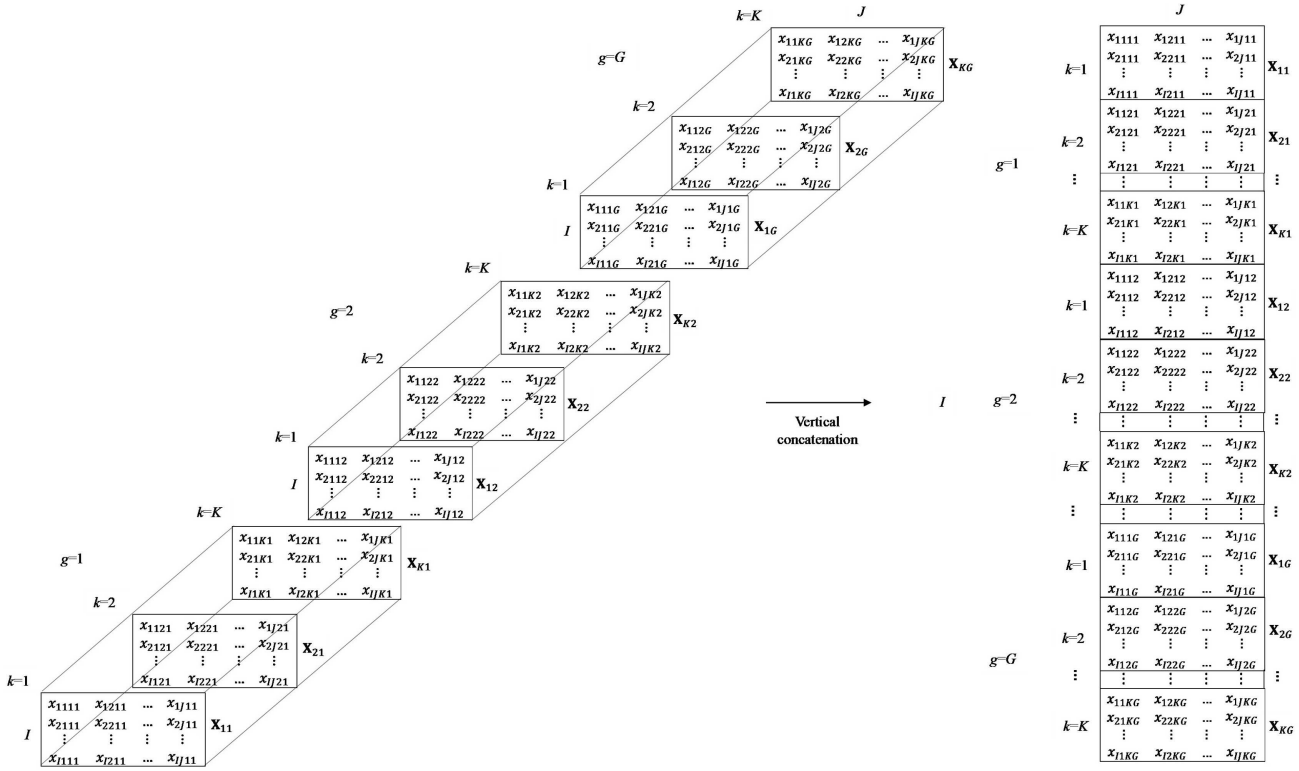
1.2. Mathematical background of unbalanced four-dimensional data cubes

Regarding the mathematical background of an unbalanced four-dimensional data cube with independent hyper-slices and slices, and unequal number of objects per slice and hyper-slice, let $\underline{\mathbf{D}} = [\mathbf{D}_{11}, \mathbf{D}_{21}, \ldots, \mathbf{D}_{K1}, \mathbf{D}_{12}, \mathbf{D}_{22}, \ldots, \mathbf{D}_{K2}, \ldots, \mathbf{D}_{1G}, \mathbf{D}_{2G}, \ldots, \mathbf{D}_{KG}]$ be a generalized four-dimensional $n_{kg} \times J \times K \times G$ data cube ($p = 4$). Each two-dimensional slice $\mathbf{D}_{kg}$ has dimensions $n_{kg} \times J$ and contains d_{ijkg} elements, where $i = 1, 2, \ldots, n_{kg}$, $j = 1, 2, \ldots, J$, $k = 1, 2, \ldots, K$ and $g = 1, 2, \ldots, G$. The 1st dimension n_{kg} denotes the number of objects in the k^{th} slice within g^{th} hyper-slice. The 2nd dimension J denotes the traits of the cube. The 3rd dimension K denotes the independent slices, while the 4th dimension G denotes the independent hyper-slices of the cube.

The most common way to manage an unbalanced four-dimensional data cube is to transform it into a two-dimensional data matrix through vertical concatenation [4]. The generalized data cube $\underline{\mathbf{D}}$ ($n_{kg} \times J \times K \times G$) transformed into a stacked data matrix $\mathbf{D}$ ($N \times J$), with d_{ijkg} elements, where $i = 1, 2, \ldots, n_{kg}$, $j = 1, 2, \ldots, J$, $k = 1, 2, \ldots, K$, and $g = 1, 2, \ldots, G$. Also, $N = \sum_{g=1}^{G} \sum_{k=1}^{K} n_{kg}$ is the total number of objects in the entire

Figure 1

Vertical concatenation of a balanced four-dimensional data cube $\underline{\mathbf{X}}$ ($I \times J \times K \times G$), with G independent hyper-slices, K independent slices, J variables, and I different and equal number of objects per hyper-slice and slice, into a stacked data matrix $\mathbf{X}$ ($IKG \times J$)



data cube, $n_k = \sum_{g=1}^{G} n_{kg}$ is the number of objects in the k^{th} slice across all hyper-slices, and $n_g = \sum_{k=1}^{K} n_{kg}$ is the number of objects in the g^{th} hyper-slice across all slices (Figure 2).

1.3. Proposed methodological approaches

There are two basic methodological approaches for the analysis of data cubes with independent slices. The first is based on analyzing a data cube as a two-dimensional mixed-type dataset (with categorical and scale variables), in which the slices are considered additional (usually categorical) variables [5]. The second is based on analyzing a data cube as a stacked data matrix, in which the slices do not lose their conceptual and structural role [1].

Dimensionality reduction represents a valuable way of analyzing data cubes, providing a compressed representation of the original dataset that is smaller in "volume" but retains critical and informative content [6].

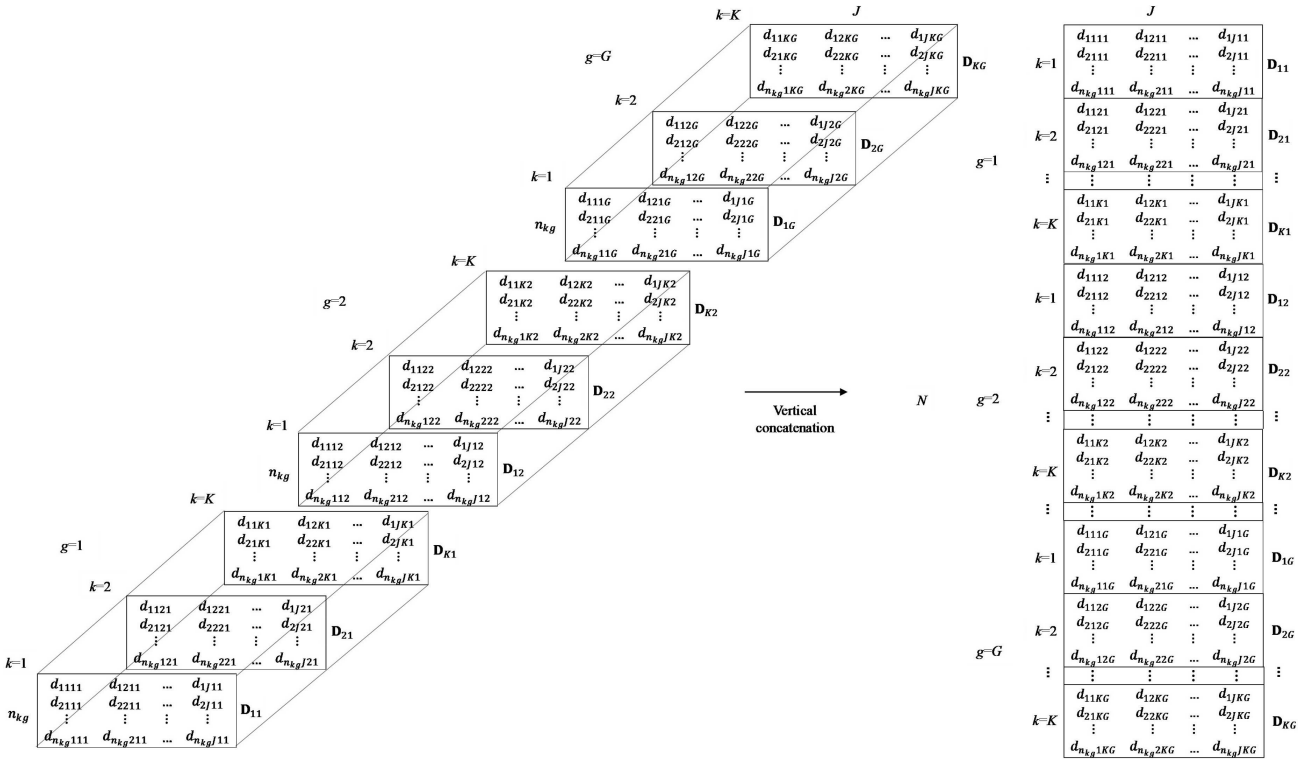
According to Kyrana et al. [5], the first methodological approach involves several dimensionality reduction methods and strategies. Specifically, one way is the application of principal component analysis (PCA) [4, 7–9] and/or factor analysis (FA) [7] to the quantitative variables of the dataset, as well as to the pseudo-variables (dummy coded as 0–1) derived from the categorical variables (slices and hyper-slices). A second way is the application of multiple correspondence analysis (MCA) [10, 11] to the categorical variables, as well as to the quantitative variables after different categorization schemes. A third way is the application of Non-Linear Categorical PCA with optimal scaling (CATPCA) [7, 12, 13] either to the mixed-type data or to the categorical variables, as well as to the quantitative variables after different categorization and coding schemes. Another way is the application of Factor Analysis for Mixed Data (FAMD) [14–17] to the mixed-type dataset.

According to Kyrana et al. [1], the second methodological approach also involves several dimensionality reduction strategies, which are applied to three-dimensional data cubes ($p = 3$) with independent slices. Specifically, these strategies are applied to the stacked data matrix after vertical concatenation. Some strategies are based on the total variability decomposition of the cube into between-slices variability and within-slices variability. Some other strategies either ignore the effect of slices [18] or take it into account on the overall data cube structure [4, 18], while other strategies initially ignore the effect of slices and add it at a second level, based on the French School of Data Analysis [19].

All these strategies can also be applied to four-dimensional data cubes ($p = 4$) with independent slices and hyper-slices and so on. However, the variability decomposition strategies differ and are more complicated. Specifically, in the case of balanced four-dimensional data cubes, the methodology is still the same as in the cases of balanced and unbalanced three-dimensional data cubes. On the other hand, in the case of unbalanced four-dimensional data cubes, the methodology constitutes a controversial scientific issue.

Therefore, this study was based on the second methodological approach by proposing an alternative dimensionality

Figure 2
Vertical concatenation of an unbalanced four-dimensional data cube $\underline{\mathbf{D}}$ ($n_{kg} \times J \times K \times G$), with G independent hyper-slices, K independent slices, J variables, and n_{kg} different and unequal number of objects per hyper-slice and slice, into a stacked data matrix $\mathbf{D}$ ($N \times J$)



reduction strategy for four-dimensional data cubes with independent slices and hyper-slices and equal or unequal number of experimental units per slice within each hyper-slice.

It is worth noting that there are also some well-known dimensionality reduction methods, such as simultaneous components analysis (SCA) [20], multi-group PCA [18, 21], and common PCA [18, 21]. However, these methods are appropriate for multi-group datasets, while their application depends on the objectives of the study, the scientific field, and the structure and composition of the observed data. Therefore, it is wiser and more important to apply different analysis strategies (many of which combine classical analysis methods) in order to increase the range of applications of each method and extract more useful results [1, 5].

The proposed strategy was based on PCA, as it is one of the most well-known linear dimensionality reduction methods of data analysis [8]. Also, many methods are generalizations [3, 21] and/or extensions of PCA [18, 21]. For example, in the case of non-linear data, Kernel PCA (KPCA) transforms the data using a non-linear transformation ("kernel trick"), projecting it into a high-dimensional space where linearity may hold, and then applies PCA to the "new" transformed variable space [22]. Furthermore, according to Kyrana et al. [5], PCA outperforms other well-known methods (FA, MCA, CATPCA, and FAMD) in terms of the combination of satisfactory reduction of mathematical dimensions, faster execution time of results in Python (regardless of the analysis strategy), and its applicability across three different statistical software programs (Python, R, and IBM SPSS Statistics), thus allowing the application of most analysis strategies. Additionally, in the field of agriculture, PCA is an important tool for exploratory data analysis (i.e., the detection of patterns, clusters, extreme values, etc.) through visualizations, in order to identify the key characteristics of a dataset [23, 24]. Also, both PCA and the proposed dimensionality reduction strategy constitute a preprocessing stage for machine learning algorithms [25–28]. Particularly, they constitute feature extraction techniques that contribute to the understanding of data and the interpretation of the results obtained, so that useful conclusions and valid predictions can be extracted, and correct decisions can be made by researchers [25–29].

1.4. Related work

In biological sciences, and especially in agriculture, the "key point" of statistical data analysis is the study of variability, focusing on its structure and components [30]. To study the structure and components of variability, some researchers apply analysis of variance (ANOVA) and/or multivariate ANOVA (MANOVA), which examines differences in mean values between slices and hyper-slices across dependent quantitative variables, as well as their interactions. MANOVA provides valuable information on how the slices and hyper-slices differ across variables, focusing on the comparison of mean values and the detection of overall multivariate differences [31–34].

In international scientific literature, several studies generally follow a similar methodology [35–39]. For example, de Haan et al. [38] focus on the ANOVA-PCA method, that is, the application of ANOVA and then PCA to all variability data matrices, in order to improve the interpretation of ANOVA results and compare main effects. In Reference [39], the focus is on ANOVA with SCA (ASCA), that is, the application of ANOVA and then PCA to the variability data matrices of the main effects and not the variability data matrices of the residuals.

In addition, there are studies that propose improvements to the ASCA method. For example, in Reference [40], the focus is on the weighted-effect ANOVA SCA for unbalanced data and its comparison with the ASCA and ASCA+ methods.

However, these studies primarily focus on the context of ANOVA (or MANOVA), without addressing different analysis strategies or providing an in-depth examination and comparison of the underlying structures of variability components. Furthermore, they concentrate exclusively on multivariate datasets rather than on data cubes. Generally, MANOVA is unable to study the total variability of a data cube in depth [1, 7]. In contrast, the proposed strategy provided a deeper understanding of the components and structure of variability through lower-dimensional representations resulting from the application of the PCA method. In addition, the proposed strategy can be used as a complement to MANOVA.

1.5. Aims of the study

The aim of this study was to propose new insights into the dimensionality reduction of both balanced and unbalanced four-dimensional data cubes, with independent slices and hyper-slices. The proposed strategy focused more deeply on the data cubes' variability structure and components. Specifically, the proposed strategy was based on the idea of decomposing the total variability of the data cube into between hyper-slices variability, between slices variability, variability of interaction between hyper-slices and slices, and within slices and hyper-slices variability.

This study also highlighted the controversial case of the variability decomposition on unbalanced four-dimensional data cubes.

Furthermore, according to the relevant international literature [31–34, 41], the mathematical expressions of the variability data matrices (i.e., matrices where their elements in rows and columns are ready for the application of PCA or other dimensionality reduction methods) are not presented explicitly, analytically, and/or comprehensively, resulting in "gaps" for readers who do not have specialized knowledge on the subject. This study aimed to fill these "gaps" with an explicit and analytical presentation of the relevant mathematical expressions.

It is worth noting that the steps and methods of the proposed strategy, as well as the simulated datasets used, were illustrative. They aimed to demonstrate the methodological approach rather than provide a biological interpretation of the results. Therefore, researchers can choose the strategy's steps and methods that align better with the objectives of their study and the observed data.

1.6. Mathematical background of variability decomposition on balanced four-dimensional data cubes

Based on Williams's study [41], for balanced four-dimensional data cubes, it is true that:

$$\begin{aligned} \textit{Total variability} = {} & \textit{Between hyper-slices variability} \\ & + \textit{Between slices variability} + \textit{Interaction between} \\ & \textit{hyper-slices and slices variability} \\ & + \textit{Within hyper-slices and slices variability} \end{aligned} \tag{1}$$

Let $\mathbf{X}$ ($IKG \times J$) be a balanced stacked data matrix with different objects per slice and hyper-slice, with x_{ijkg} elements, where

$i = 1, 2, \ldots, I$, $j = 1, 2, \ldots, J$, $k = 1, 2, \ldots, K$, and $g = 1, 2, \ldots, G$. Equation (1) is expressed mathematically as follows:

$$\mathbf{X}_{\mathrm{T}} = \mathbf{X}_{\mathrm{G}} + \mathbf{X}_{\mathrm{K}} + \mathbf{X}_{\mathrm{KG}} + \mathbf{X}_{\mathrm{W}} \text{ or } \mathbf{X}_{\mathrm{t}} = \mathbf{X}_{\mathrm{g}} + \mathbf{X}_{\mathrm{k}} + \mathbf{X}_{\mathrm{kg}} + \mathbf{X}_{\mathrm{w}}$$

Also, $\mathbf{x}_{ikg} = \begin{bmatrix} x_{i1kg} \\ x_{i2kg} \\ \vdots \\ x_{iJkg} \end{bmatrix}_{J\times 1}$ is the J-dimensional vector of variables for the object i in the k^{th} slice within g^{th} hyper-slice, $\overline{\mathbf{x}}_{.k.} = \frac{1}{IG}\sum_{g=1}^{G}\sum_{i=1}^{I}\mathbf{x}_{ikg} = \begin{bmatrix} \overline{x}_{.1k.} \\ \overline{x}_{.2k.} \\ \vdots \\ \overline{x}_{.Jk.} \end{bmatrix}_{J\times 1}$ is the J-dimensional vector of means for the k^{th} slice, $\overline{\mathbf{x}}_{..g} = \frac{1}{IK}\sum_{k=1}^{K}\sum_{i=1}^{I}\mathbf{x}_{ikg} = \begin{bmatrix} \overline{x}_{.1.g} \\ \overline{x}_{.2.g} \\ \vdots \\ \overline{x}_{.J.g} \end{bmatrix}_{J\times 1}$ is the J-dimensional vector of means for the g^{th} hyper-slice, $\overline{\mathbf{x}}_{.kg} = \frac{1}{I}\sum_{i=1}^{I}\mathbf{x}_{ikg} = \begin{bmatrix} \overline{x}_{.1kg} \\ \overline{x}_{.2kg} \\ \vdots \\ \overline{x}_{.Jkg} \end{bmatrix}_{J\times 1}$ is the J-dimensional vector of means for the k^{th} slice within the g^{th} hyper-slice, and $\overline{\mathbf{x}}_{...} = \frac{1}{IKG}\sum_{g=1}^{G}\sum_{k=1}^{K}\sum_{i=1}^{I}\mathbf{x}_{ikg} = \begin{bmatrix} \overline{x}_{.1..} \\ \overline{x}_{.2..} \\ \vdots \\ \overline{x}_{.J..} \end{bmatrix}_{J\times 1}$ is the J-dimensional vector of means for the stacked data matrix $\mathbf{X}$.

Then, it is true that the $J \times J$ sum of squares and cross-products (SSCP) matrix of the total variability is defined as $\mathbf{X_t} = \sum_{g=1}^{G}\sum_{k=1}^{K}\sum_{i=1}^{I}(\mathbf{x}_{ikg} - \overline{\mathbf{x}}_{...})(\mathbf{x}_{ikg} - \overline{\mathbf{x}}_{...})^{\mathrm{T}}$. The $J \times J$ SSCP matrix of the variability between hyper-slices is defined as $\mathbf{X_g} = IK\sum_{g=1}^{G}(\overline{\mathbf{x}}_{..g} - \overline{\mathbf{x}}_{...})(\overline{\mathbf{x}}_{..g} - \overline{\mathbf{x}}_{...})^{T}$. The $J \times J$ SSCP matrix of the variability between slices is defined as $\mathbf{X_k} = IG\sum_{k=1}^{K}(\overline{\mathbf{x}}_{.k.} - \overline{\mathbf{x}}_{...})(\overline{\mathbf{x}}_{.k.} - \overline{\mathbf{x}}_{...})^{T}$. The $J \times J$ SSCP matrix of the variability of interaction between hyper-slices and slices is defined as $\mathbf{X_{kg}} = I\sum_{g=1}^{G}\sum_{k=1}^{K}(\overline{\mathbf{x}}_{.kg} - \overline{\mathbf{x}}_{..g} - \overline{\mathbf{x}}_{.k.} + \overline{\mathbf{x}}_{...})(\overline{\mathbf{x}}_{.kg} - \overline{\mathbf{x}}_{..g} - \overline{\mathbf{x}}_{.k.} + \overline{\mathbf{x}}_{...})^{T}$. The $J \times J$ SSCP matrix of the variability within hyper-slices and slices is defined as $\mathbf{X_w} = \sum_{g=1}^{G}\sum_{k=1}^{K}\sum_{i=1}^{I}(\mathbf{x}_{ikg} - \overline{\mathbf{x}}_{.kg})(\mathbf{x}_{ikg} - \overline{\mathbf{x}}_{.kg})^{T}$.

Furthermore, it is true that the $IKG \times J$ data matrix of the total variability is defined as $\mathbf{X}_T = \mathbf{X} - \mathbf{1}_{IKG}\overline{\mathbf{x}}_{...}^{\mathrm{T}}$, where, for example, $\mathbf{1}_n$ is the n-vector of ones. The $IKG \times J$ data matrix of the between hyper-slices variability is defined as $\mathbf{X}_G = \begin{bmatrix} \mathbf{1}_{IK}\overline{\mathbf{x}}_{..1}^{\mathrm{T}} \\ \mathbf{1}_{IK}\overline{\mathbf{x}}_{..2}^{\mathrm{T}} \\ \vdots \\ \mathbf{1}_{IK}\overline{\mathbf{x}}_{..G}^{\mathrm{T}} \end{bmatrix} - \mathbf{1}_{IKG}\overline{\mathbf{x}}_{...}^{\mathrm{T}}$. The $IKG \times J$ data matrix of the between slices variability is defined as $\mathbf{X}_K = repmat\left(\begin{bmatrix} \mathbf{1}_I\overline{\mathbf{x}}_{.1.}^{\mathrm{T}} \\ \mathbf{1}_I\overline{\mathbf{x}}_{.2.}^{\mathrm{T}} \\ \vdots \\ \mathbf{1}_I\overline{\mathbf{x}}_{.K.}^{\mathrm{T}} \end{bmatrix}, G\right) - \mathbf{1}_{IKG}\overline{\mathbf{x}}_{...}^{\mathrm{T}}$, where the function $repmat(\mathbf{X_K}, G)$ repeats the matrix $\mathbf{X_K}$ vertically G times. The $IKG \times J$ variability data matrix of the interaction between hyper-slices and slices is defined as $\mathbf{X}_{KG} = \begin{bmatrix} \mathbf{1}_I\overline{\mathbf{x}}_{.11}^{\mathrm{T}} \\ \mathbf{1}_I\overline{\mathbf{x}}_{.21}^{\mathrm{T}} \\ \vdots \\ \mathbf{1}_I\overline{\mathbf{x}}_{.KG}^{\mathrm{T}} \end{bmatrix} - \begin{bmatrix} \mathbf{1}_{IK}\overline{\mathbf{x}}_{..1}^{\mathrm{T}} \\ \mathbf{1}_{IK}\overline{\mathbf{x}}_{..2}^{\mathrm{T}} \\ \vdots \\ \mathbf{1}_{IK}\overline{\mathbf{x}}_{..G}^{\mathrm{T}} \end{bmatrix} - repmat\left(\begin{bmatrix} \mathbf{1}_I\overline{\mathbf{x}}_{.1.}^{\mathrm{T}} \\ \mathbf{1}_I\overline{\mathbf{x}}_{.2.}^{\mathrm{T}} \\ \vdots \\ \mathbf{1}_I\overline{\mathbf{x}}_{.K.}^{\mathrm{T}} \end{bmatrix}, G\right) + \mathbf{1}_{IKG}\overline{\mathbf{x}}_{...}^{\mathrm{T}}$. The $IKG \times J$ data matrix of the within hyper-slices and slices variability is defined as $\mathbf{X}_W = \mathbf{X} - \begin{bmatrix} \mathbf{1}_I\overline{\mathbf{x}}_{.11}^{\mathrm{T}} \\ \mathbf{1}_I\overline{\mathbf{x}}_{.21}^{\mathrm{T}} \\ \vdots \\ \mathbf{1}_I\overline{\mathbf{x}}_{.KG}^{\mathrm{T}} \end{bmatrix}$.

It is worth noting that given an SSCP matrix (e.g., the total variability SSCP matrix $\mathbf{X_t}$), the covariance matrix is defined as $\mathbf{C} = \frac{1}{IK-1}\mathbf{X_t}$, while the correlation matrix (input matrix in PCA) is defined as $\mathbf{R} = \mathbf{S}^{-1/2}\mathbf{C}\,\mathbf{S}^{-1/2}$, where $\mathbf{S} = \mathrm{diag}(c_{11}, c_{22}, \ldots, c_{JJ})$ and c_{JJ} is the variance of the variable j.

1.7. Theoretical background of variability decomposition on unbalanced four-dimensional data cubes

According to Williams [41], in the case of unbalanced data cubes (for $p = 4$), the decomposition of variability is more complex and should be treated as a distinct case rather than as a special case of balanced data. Particularly, in the case of unbalanced data, Equation (1) no longer exists because the main effects (components of variability) of hyper-slices, slices, and interactions are not independent, as they overlap.

In Reference [41], the author suggests two methods for analyzing unbalanced data cubes. The first approach involves converting the unbalanced data cube into a balanced data cube by randomly removing observations from redundant slices and/or hyper-slices or by adding estimated observations to incomplete slices and/or hyper-slices using unweighted or weighted means.

The second approach involves the direct analysis of the unbalanced stacked data matrix through various approaches. In the methodological framework of ANOVA (or MANOVA), there are three basic approaches to calculating the sum of squares (SS), which appear in the international literature under various names.

The first approach is usually called "Type I" and is recommended when either the hyper-slices or the slices of the data cube are of conceptual or scientific interest to researchers. More specifically, this approach adjusts each main effect of the hyper-slices or slices for the one that precedes it in terms of significance, according to the researcher. In other words, either the variable of

hyper-slices or the variable of slices is added sequentially to the model, while the results obtained depend on the order in which the terms are entered into the model [42]. Consequently, multiple sequential analyses are required so that each term can be placed last in the model and adjusted for the others [42]. However, this approach is not suitable in the case of unbalanced four-dimensional data cubes due to its hierarchical nature.

The second approach is usually called "Type II" and is recommended when the interaction is not of statistical, conceptual, or scientific interest and is therefore disregarded, while the focus is independently on the hyper-slices and slices, which are adjusted for each other. This approach does not use parameter restrictions, meaning that alternative approaches to the model produce identical results [42]. This approach is also not appropriate in the case of unbalanced four-dimensional data cubes, as it ignores the interaction between hyper-slices and slices, which is considered crucial both in most agricultural experiments [43] and in the in-depth study of the structure and components of the cube variability.

The third approach is usually called "Type III" and is recommended when the focus is on both interaction and independently on hyper-slices and slices, which are adjusted for each other. For example, the SS for the variable of hyper-slices is calculated on the assumption that the variable of slices and their interaction have already been included in the model. This approach uses the following parameter restrictions [42]: $\sum_{g=1}^{G}\left(\bar{\mathbf{d}}'_{..g}-\bar{\mathbf{d}}'_{...}\right)=0$, $\sum_{k=1}^{K}\left(\bar{\mathbf{d}}'_{.k.}-\bar{\mathbf{d}}'_{...}\right)=0$, $\sum_{k=1}^{K}\left(\bar{\mathbf{d}}_{.kg}-\bar{\mathbf{d}}'_{.k.}-\bar{\mathbf{d}}'_{..g}+\bar{\mathbf{d}}'_{...}\right)=0$, $\forall g$ and $\sum_{g=1}^{G}\left(\bar{\mathbf{d}}_{.kg}-\bar{\mathbf{d}}'_{.k.}-\bar{\mathbf{d}}'_{..g}+\bar{\mathbf{d}}'_{...}\right)=0$, $\forall k$. where $\bar{\mathbf{d}}'_{..g}$ is the vector of unweighted marginal means of the hyper-slices, $\bar{\mathbf{d}}'_{.k.}$ is the vector of unweighted marginal means of the slices, $\bar{\mathbf{d}}'_{...}$ is the vector of unweighted marginal means of the stacked data matrix $\mathbf{D}$, and $\bar{\mathbf{d}}_{.kg}$ is the vector of weighted means for the k^{th} slice within the g^{th} hyper-slice.

It is worth noting that there is one more approach called "Type IV," which is a generalization of "Type III" and is used in cases where there are missing values and treatments in the datasets. If there are no missing values and treatments, the two methods ("Type III" and "Type IV") produce the same results [42].

However, there is no clear answer in international scientific literature as to which of these methodological approaches is the most appropriate, as opinions differ. More specifically, several studies argue that the ideal approach for unbalanced data is the Type III methodological approach [42]. For this reason, most software programs (e.g., IBM SPSS Statistics, SAS, etc.) perform analyses using the "Type III" method as the default [42, 44].

On the other hand, there are researchers who argue that the "Type III" approach does not correspond to interesting hypotheses [42, 44]. Although the "Type III" methodological approach seems reasonable, it is essentially unfounded, as it leads to paradoxes [42, 44]. An additional limitation of this approach is the method of calculating the SS. Particularly, after an extensive study of the international scientific literature, as well as the guide provided by the IBM SPSS Statistics software, it appears that there is no explicit mathematical formulation for calculating the SS. This fact is also confirmed by LaMotte [44].

In the context of variability decomposition on unbalanced four-dimensional data cubes, the "Type III" approach was not considered appropriate for the objectives of the present study. Specifically, the main objective of this study was the investigation of the variability structure and variability components of the data cubes through directly interpretable SSCP matrices based on the actual weighted means of the observed data. Therefore, the proposed methodology focused on descriptive variability decomposition and on the interpretation of the data cubes' variability components.

In contrast, the SSCP matrices of the "Type III" approach are projection-based quantities derived from differences between full and reduced general linear models [44]. Consequently, "Type III" SSCP matrices cannot generally be expressed as weighted cross-products of deviations of the raw observations from the weighted means of the slices and hyper-slices.

An additional alternative methodological approach is that of separate univariate analyses, known as the Unadjusted Main Effects method [41]. This method is similar to classical ANOVA and is analogous to the balanced data cube approach. More specifically, the only difference with the balanced data cube approach lies in the coefficients of weights (number of objects) of the combinations between hyper-slices and slices that differ. This method directly calculates the main effects of hyper-slices and slices without adjusting one for the other. This makes it ideal for in-depth study of the variability of unbalanced four-dimensional data cubes.

1.8. Mathematical background of variability decomposition on unbalanced four-dimensional data cubes

In this study, the most ideal methodological approach is the method of Unadjusted Main Effects, which is adapted to the multivariate nature of the stacked data matrices (combination of multiple univariate analyses).

Let $\mathbf{D}$ be an unbalanced stacked data matrix with different objects per slice and hyper-slice, with dimensions $N \times J$ and d_{ijkg} elements, where $i = 1, 2, \ldots, n_{kg}$, $j = 1, 2, \ldots, J$, $k = 1, 2, \ldots, K$, $g = 1, 2, \ldots, G$, $N = \sum_{g=1}^{G}\sum_{k=1}^{K} n_{kg}$, $n_k = \sum_{g=1}^{G} n_{kg}$, and $n_g = \sum_{k=1}^{K} n_{kg}$.

Also, $\mathbf{d}_{ikg} = \begin{bmatrix} d_{i1kg} \\ d_{i2kg} \\ \vdots \\ d_{iJkg} \end{bmatrix}_{J\times 1}$ is the J-dimensional vector of variables for the object i in the k^{th} slice within g^{th} hyper-slice, $\bar{\mathbf{d}}_{.k.} = \frac{1}{n_k}\sum_{g=1}^{G}\sum_{i=1}^{n_{kg}}\mathbf{d}_{ikg} = \begin{bmatrix} \bar{d}_{.1k.} \\ \bar{d}_{.2k.} \\ \vdots \\ \bar{d}_{.Jk.} \end{bmatrix}_{J\times 1}$ is the J-dimensional vector of weighted means for the k^{th} slice, $\bar{\mathbf{d}}_{..g} = \frac{1}{n_g}\sum_{k=1}^{K}\sum_{i=1}^{n_{kg}}\mathbf{d}_{ikg} = \begin{bmatrix} \bar{d}_{.1.g} \\ \bar{d}_{.2.g} \\ \vdots \\ \bar{d}_{.J.g} \end{bmatrix}_{J\times 1}$ is the J-dimensional vector of weighted means for the g^{th} hyper-slice, $\bar{\mathbf{d}}_{.kg} = \frac{1}{n_{kg}}\sum_{i=1}^{n_{kg}}\mathbf{d}_{ikg} = \begin{bmatrix} \bar{d}_{.1kg} \\ \bar{d}_{.2kg} \\ \vdots \\ \bar{d}_{.Jkg} \end{bmatrix}_{J\times 1}$ is the J-

dimensional vector of weighted means for the k^{th} slice within the g^{th} hyper-slice, $\bar{\mathbf{d}}_{\ldots} = \frac{1}{N}\sum_{g=1}^{G}\sum_{k=1}^{K}\sum_{i=1}^{n_{kg}}\mathbf{d}_{ikg} = \begin{bmatrix}\bar{d}_{.1..}\\ \bar{d}_{.2..}\\ \vdots\\ \bar{d}_{.J..}\end{bmatrix}_{J\times 1}$ is the J-dimensional vector of weighted means for the stacked data matrix $\mathbf{D}$, $\bar{\mathbf{d}}'_{.k.} = \frac{1}{G}\sum_{g=1}^{G}\bar{\mathbf{d}}_{.kg}$ is the J-dimensional vector of unweighted means of $\bar{\mathbf{d}}_{.kg}$ for the k^{th} slice within each hyper-slice (or else the marginal means of slices), $\bar{\mathbf{d}}'_{..g} = \frac{1}{K}\sum_{k=1}^{K}\bar{\mathbf{d}}_{.kg}$ is the J-dimensional vector of unweighted means of $\bar{\mathbf{d}}_{.kg}$ for the g^{th} hyper-slice within each slice (or else the marginal means of hyper-slices), and $\bar{\mathbf{d}}'_{\ldots} = \frac{1}{KG}\sum_{g=1}^{G}\sum_{k=1}^{K}\bar{\mathbf{d}}_{.kg}$ is the J-dimensional vector of total unweighted means of $\bar{\mathbf{d}}_{.kg}$ (or else the marginal means of the stacked data matrix $\mathbf{D}$).

Then, it is true that the $J \times J$ SSCP matrix of total variability is defined as $\mathbf{D}_t = \sum_{g=1}^{G}\sum_{k=1}^{K}\sum_{i=1}^{n_{kg}}(\mathbf{d}_{ikg} - \bar{\mathbf{d}}_{\ldots})(\mathbf{d}_{ikg} - \bar{\mathbf{d}}_{\ldots})^{\mathrm{T}}$. The $J \times J$ SSCP matrix of the variability between hyper-slices is defined as $\mathbf{D}_g = \sum_{g=1}^{G} n_g\left(\bar{\mathbf{d}}_{..g} - \bar{\mathbf{d}}_{\ldots}\right)\left(\bar{\mathbf{d}}_{..g} - \bar{\mathbf{d}}_{\ldots}\right)^{\mathrm{T}}$. The $J \times J$ SSCP matrix of the variability between slices is defined as $\mathbf{D}_k = \sum_{k=1}^{K} n_k\left(\bar{\mathbf{d}}_{.k.} - \bar{\mathbf{d}}_{\ldots}\right)\left(\bar{\mathbf{d}}_{.k.} - \bar{\mathbf{d}}_{\ldots}\right)^{\mathrm{T}}$. The $J \times J$ SSCP matrix of the variability of interaction between hyper-slices and slices is defined as $\mathbf{D}_{kg} = \sum_{g=1}^{G}\sum_{k=1}^{K} n_{kg}(\bar{\mathbf{d}}_{.kg} - \bar{\mathbf{d}}'_{.k.} - \bar{\mathbf{d}}'_{..g} + \bar{\mathbf{d}}'_{\ldots})(\bar{\mathbf{d}}_{.kg} - \bar{\mathbf{d}}'_{.k.} - \bar{\mathbf{d}}'_{..g} + \bar{\mathbf{d}}'_{\ldots})^{\mathrm{T}}$. The $J \times J$ SSCP matrix of the variability within hyper-slices and slices is defined as $\mathbf{D}_w = \sum_{g=1}^{G}\sum_{k=1}^{K}\sum_{i=1}^{n_{kg}}\left(\mathbf{d}_{ikg} - \bar{\mathbf{d}}_{.kg}\right)\left(\mathbf{d}_{ikg} - \bar{\mathbf{d}}_{.kg}\right)^{\mathrm{T}}$.

Furthermore, it is true that the $N \times J$ data matrix of the total variability is defined as $\mathbf{D}_T = \mathbf{D} - \mathbf{1}_N\bar{\mathbf{d}}_{\ldots}^{\mathrm{T}}$, where $\mathbf{1}_N$ is the N-vector of ones. The $N \times J$ data matrix of the between hyper-slices variability is defined as $\mathbf{D}_G = \begin{bmatrix}\mathbf{1}_{n_g}\bar{\mathbf{d}}_{..1}^{\mathrm{T}}\\ \mathbf{1}_{n_g}\bar{\mathbf{d}}_{..2}^{\mathrm{T}}\\ \vdots\\ \mathbf{1}_{n_g}\bar{\mathbf{d}}_{..G}^{\mathrm{T}}\end{bmatrix} - \mathbf{1}_N\bar{\mathbf{d}}_{\ldots}^{\mathrm{T}}$. The $N \times J$ data matrix of the between slices variability is defined as $\mathbf{D}_K = repmat\left(\begin{bmatrix}\mathbf{1}_{n_k}\bar{\mathbf{d}}_{.1.}^{\mathrm{T}}\\ \mathbf{1}_{n_k}\bar{\mathbf{d}}_{.2.}^{\mathrm{T}}\\ \vdots\\ \mathbf{1}_{n_k}\bar{\mathbf{d}}_{.K.}^{\mathrm{T}}\end{bmatrix}, G\right) - \mathbf{1}_N\bar{\mathbf{d}}_{\ldots}^{\mathrm{T}}$, where the function $repmat(\mathbf{D_K}, G)$ repeats the matrix $\mathbf{D_K}$ vertically G times. The $N \times J$ variability data matrix of the interaction between hyper-slices and slices is defined as $\mathbf{D}_{KG} = \begin{bmatrix}\mathbf{1}_{n_{kg}}\bar{\mathbf{d}}_{..11}^{\mathrm{T}}\\ \mathbf{1}_{n_{kg}}\bar{\mathbf{d}}_{..21}^{\mathrm{T}}\\ \vdots\\ \mathbf{1}_{n_{kg}}\bar{\mathbf{d}}_{..KG}^{\mathrm{T}}\end{bmatrix} - \begin{bmatrix}\mathbf{1}_{n_g}\bar{\mathbf{d}}_{..1}'^{\mathrm{T}}\\ \mathbf{1}_{n_g}\bar{\mathbf{d}}_{..2}'^{\mathrm{T}}\\ \vdots\\ \mathbf{1}_{n_g}\bar{\mathbf{d}}_{..G}'^{\mathrm{T}}\end{bmatrix} - repmat\left(\begin{bmatrix}\mathbf{1}_{n_k}\bar{\mathbf{d}}_{.1.}'^{\mathrm{T}}\\ \mathbf{1}_{n_k}\bar{\mathbf{d}}_{.2.}'^{\mathrm{T}}\\ \vdots\\ \mathbf{1}_{n_k}\bar{\mathbf{d}}_{.K.}'^{\mathrm{T}}\end{bmatrix}, G\right) + \mathbf{1}_N\bar{\mathbf{d}}_{\ldots}'^{\mathrm{T}}$. The $N \times J$ data matrix of the within hyper-slices and slices variability is defined as $\mathbf{D}_W = \mathbf{D} - \begin{bmatrix}\mathbf{1}_{n_{kg}}\bar{\mathbf{d}}_{.11}^{\mathrm{T}}\\ \mathbf{1}_{n_{kg}}\bar{\mathbf{d}}_{.21}^{\mathrm{T}}\\ \vdots\\ \mathbf{1}_{n_{kg}}\bar{\mathbf{d}}_{.KG}^{\mathrm{T}}\end{bmatrix}$.

For comparison purposes, the mathematical formulas [44] of SSCP matrices for the "Type III" approach are presented in Appendix A.

2. Materials and Methods

2.1. Datasets

The proposed strategy was implemented using the initial agricultural dataset named "Basil" (stacked data matrix **Y**), which covers basil plants cultivated and studied at the Aristotle University of Thessaloniki Farm between 2015 and 2017 [45]. Data from the 2016–2017 growing seasons at Stage III of plant development were selected for analysis. The sample consisted of 60 basil plants, seven quantitative characteristics of the plants, three irrigation levels, and five basil varieties.

In order to achieve imbalance between the slices and hyper-slices of the cube and to increase the volume of the dataset, the initial dataset was simulated using Python code (Appendix B). The simulation algorithm for the "Basil" dataset was as follows:

Step 1: Calculate the mean values of the variables per slice within each hyper-slice of the initial stacked data matrix **Y**.

Step 2: Calculate the standard deviations of the variables per slice within each hyper-slice of the initial stacked data matrix **Y**.

Step 3: Calculate the correlation matrices between the variables per slice within each hyper-slice of the initial stacked data matrix **Y**.

Step 4: Simulate two new datasets **X** ("Basil-Sim 1") and **D** ("Basil-Sim 2"), larger in volume, with equal and unequal number of basil plants per slice within each hyper-slice, with specific mean values, standard deviations, and correlations of characteristics per slice within each hyper-slice, approximating the Multinormal distribution.

The first dataset ("Basil-Sim 1") was considered as a balanced four-dimensional data cube $\underline{\mathbf{X}}$ with dimensions 500 × 7 × 5 × 3 and with independent slices and hyper-slices. The data cube $\underline{\mathbf{X}}$ was transformed into a stacked data matrix **X** (7,500 × 7) through vertical concatenation. The 1st dimension consisted of an equal number of different basil plants per slice (I = 500). The 2nd dimension consisted of the same seven continuous variables (traits) per slice (J = 7). Particularly, these were the measurements of Height (cm), Weight (gr), Leaf Area Index (LAI) (m^2/m^2), Leaf

Temperature (labeled as Leaf Temp) (°C), Photosynthesis (μmol CO_2 m^{-2} s^{-1}), Water Use Efficiency (WUE) (gr/L), and Water Potential (MPa). The 3rd dimension was a categorical variable denoting five slices (K = 5) corresponding to five basil varieties (Mrs Burns [labeled as Burns], Cinnamon, Sweet, Red Rubin [labeled as Rend], Thai). The 4th dimension was a categorical variable denoting three hyper-slices (G = 3) corresponding to three irrigation levels (40%, 70%, 100%). Also, the number of basil plants in the g^{th} hyper-slice across all slices was IG = 1,500, while the number of basil plants in the k^{th} slice across all hyper-slices was IK = 2,500.

The second dataset ("Basil-Sim 2") was considered as an unbalanced four-dimensional data cube $\underline{\mathbf{D}}$ with dimensions $n_{kg} \times 7 \times 5 \times 3$ and with independent slices and hyper-slices. The data cube $\underline{\mathbf{D}}$ was transformed into a stacked data matrix $\mathbf{D}$ (7,300 × 7) through vertical concatenation. The 1st dimension consisted of unequal number of different basil plants per slice ($n_{11} = n_{22}$ = 400 and the rest n_{kg} = 500, $n_{1.} = n_{2.}$ = 1,400 and the rest $n_{k.}$ = 1,500, $n_{.1} = n_{.2}$ = 2,400 and $n_{.3}$ = 2,500). The 2nd dimension consisted of the same seven continuous traits (Height, Weight, LAI, Leaf Temp, Photosynthesis, WUE, and Water Potential) per slice (J = 7). The 3rd dimension consisted of five slices (K = 5) corresponding to five basil varieties (Burns, Cinnamon, Sweet, Rend, Thai) and the 4th dimension consisted of three hyper-slices (G = 3) corresponding to three irrigation levels (40%, 70%, 100%).

2.2. Strategy

The steps of the proposed strategy were the following:

Step 1: Select the strategy for decomposing the total variability of unbalanced stacked data matrices.

Step 1.1: Convert the unbalanced stacked data matrix $\mathbf{D}$ into a balanced stacked data matrix $\mathbf{X'}$ by adding the corresponding unweighted mean values to the observation-deficient slices and/or hyper-slices.

Step 1.2: Apply the Unadjusted Main Effects method to the unbalanced stacked data matrix $\mathbf{D}$.

Step 2: Construct the data matrix $\mathbf{X_T}$ or $\mathbf{D_T}$ or $\mathbf{X'_T}$ with the total variability of the data cube.

Step 3: Construct the data matrix $\mathbf{X_G}$ or $\mathbf{D_G}$ or $\mathbf{X'_G}$ with the variability between hyper-slices.

Step 4: Construct the data matrix $\mathbf{X_K}$ or $\mathbf{D_K}$ or $\mathbf{X'_K}$ with the variability between slices.

Step 5: Construct the data matrix $\mathbf{X_{KG}}$ or $\mathbf{D_{KG}}$ or $\mathbf{X'_{KG}}$ with the variability of the interaction between hyper-slices and slices.

Step 6: Construct the data matrix $\mathbf{X_W}$ or $\mathbf{D_W}$ or $\mathbf{X'_W}$ with the variability within hyper-slices and slices of the cube.

Step 7: Apply PCA to the correlation matrices $\mathbf{R_T}$, $\mathbf{R_G}$, $\mathbf{R_K}$, $\mathbf{R_{KG}}$, and $\mathbf{R_W}$ of the variability matrices $\mathbf{X_T}$ or $\mathbf{D_T}$ or $\mathbf{X'_T}$, $\mathbf{X_G}$ or $\mathbf{D_G}$ or $\mathbf{X'_G}$, $\mathbf{X_K}$ or $\mathbf{D_K}$ or $\mathbf{X'_K}$, $\mathbf{X_{KG}}$ or $\mathbf{D_{KG}}$ or $\mathbf{X'_{KG}}$, and $\mathbf{X_W}$ or $\mathbf{D_W}$ or $\mathbf{X'_W}$.

Step 8: Visually compare the structure of the components of variability in terms of variables, using the PCA loading diagrams.

2.3. Method and software

The proposed strategy was based on PCA without rotation [7]. Alternatively, singular value decomposition or FA on the appropriate data matrix (i.e., the correlation matrix) could be used [20]. When data lies on non-Euclidean spaces, other dimensionality reduction methods can be used, such as KPCA [22]. For the purposes of interpretation and visualization, only the first two principal components were selected. R v4.3.2 was used for statistical analysis, while Python v3.10 via the Anaconda platform and Jupyter Notebook v6.4.5 were used for the dataset simulations.

3. Results

The results of the proposed strategy were presented below. The detailed PCA results for each four-dimensional data cube were presented in Appendix C (PCA loadings and communalities are presented in Tables C1–C3, the corresponding explained variance percentages are presented in Tables C4–C6, and the corresponding scree and score plots are presented in Figures C1–C10), while the R code for applying PCA to each variability data matrix was presented in Appendix D. Figure 3 (a)–(e) present the results of the balanced stacked data matrix $\mathbf{X}$(7,500 × 7) ("Basil-Sim 1"). Specifically, Figure 3(a) presents the results of the $\mathbf{X_T}$ data matrix of the total variability, Figure 3(b) presents the results of the $\mathbf{X_G}$ data matrix of the variability between hyper-slices, Figure 3(c) presents the results of $\mathbf{X_K}$ data matrix of the variability between slices, Figure 3(d) presents the results of $\mathbf{X_{KG}}$ variability data matrix of interaction, and Figure 3(e) presents the results of $\mathbf{X_W}$ data matrix of the variability within the hyper-slices and slices.

On the other hand, Figure 4 (a)–(e) present the results of the unbalanced stacked data matrix $\mathbf{D}$(7,300 × 7) ("Basil-Sim 2"). Particularly, Figure 4(a) presents the results of the $\mathbf{D_T}$ data matrix of the total variability, Figure 4(b) presents the results of the $\mathbf{D_G}$ data matrix of the variability between hyper-slices, Figure 4(c) presents the results of $\mathbf{D_K}$ data matrix of the variability between slices, Figure 4(d) presents the results of $\mathbf{D_{KG}}$ variability data matrix of interaction, and Figure 4(e) presents the results of $\mathbf{D_W}$ data matrix of the variability within the hyper-slices and slices.

Furthermore, Figure 5 (a)–(e) present the results of the balanced stacked data matrix $\mathbf{X'}$(7,500 × 7) ("Basil-Sim 2"). Specifically, Figure 5(a) presents the results of the $\mathbf{X'_T}$ data matrix of the total variability, Figure 5(b) presents the results of the $\mathbf{X'_G}$ data matrix of the variability between hyper-slices, Figure 5(c) presents the results of $\mathbf{X'_K}$ data matrix of the variability between slices, Figure 5(d) presents the results of $\mathbf{X'_{KG}}$ variability data matrix of interaction, and Figure 5(e) presents the results of $\mathbf{X'_W}$ data matrix of the variability within the hyper-slices and slices.

For comparison purposes, the application of the "Type III" approach on the unbalanced four-dimensional data cube $\underline{\mathbf{D}}$ ("Basil-Sim 2") is presented in Appendix E (the PCA loading diagram from each source of variability is presented in Figure E1, PCA loadings and communalities are presented in Table E1, the corresponding explained variance percentages are presented in Table E2, and the corresponding scree and score plots are presented in Figures E2–E6).

4. Discussion

The results of data cubes $\underline{\mathbf{X}}$, $\underline{\mathbf{D}}$, and $\underline{\mathbf{X'}}$ were common due to the analogous variability decomposition strategy that was followed between balanced and unbalanced data cubes. Furthermore, their common structure indicated the existence of a small imbalance between the data cubes.

In addition, the results of the unbalanced data cube $\underline{\mathbf{D}}$ were common between the methodological approach of Unadjusted

Figure 3

PCA loading diagram from: (a) the X_T data matrix of the total variability; (b) the X_G data matrix of the variability between hyper-slices; (c) the X_K data matrix of the variability between slices; (d) the X_{KG} variability data matrix of interaction; and (e) the X_W data matrix of the variability within hyper-slices and slices. The first two principal components (Dim 1 and Dim 2) define the axes along with their respective proportion of explained variance. Directional vectors indicate the loading values for each variable, including Height, Weight, Leaf Area Index (LAI), Leaf Temperature (Leaf Temp), Photosynthesis, Water Use Efficiency (WUE), and Water Potential (generated in R)

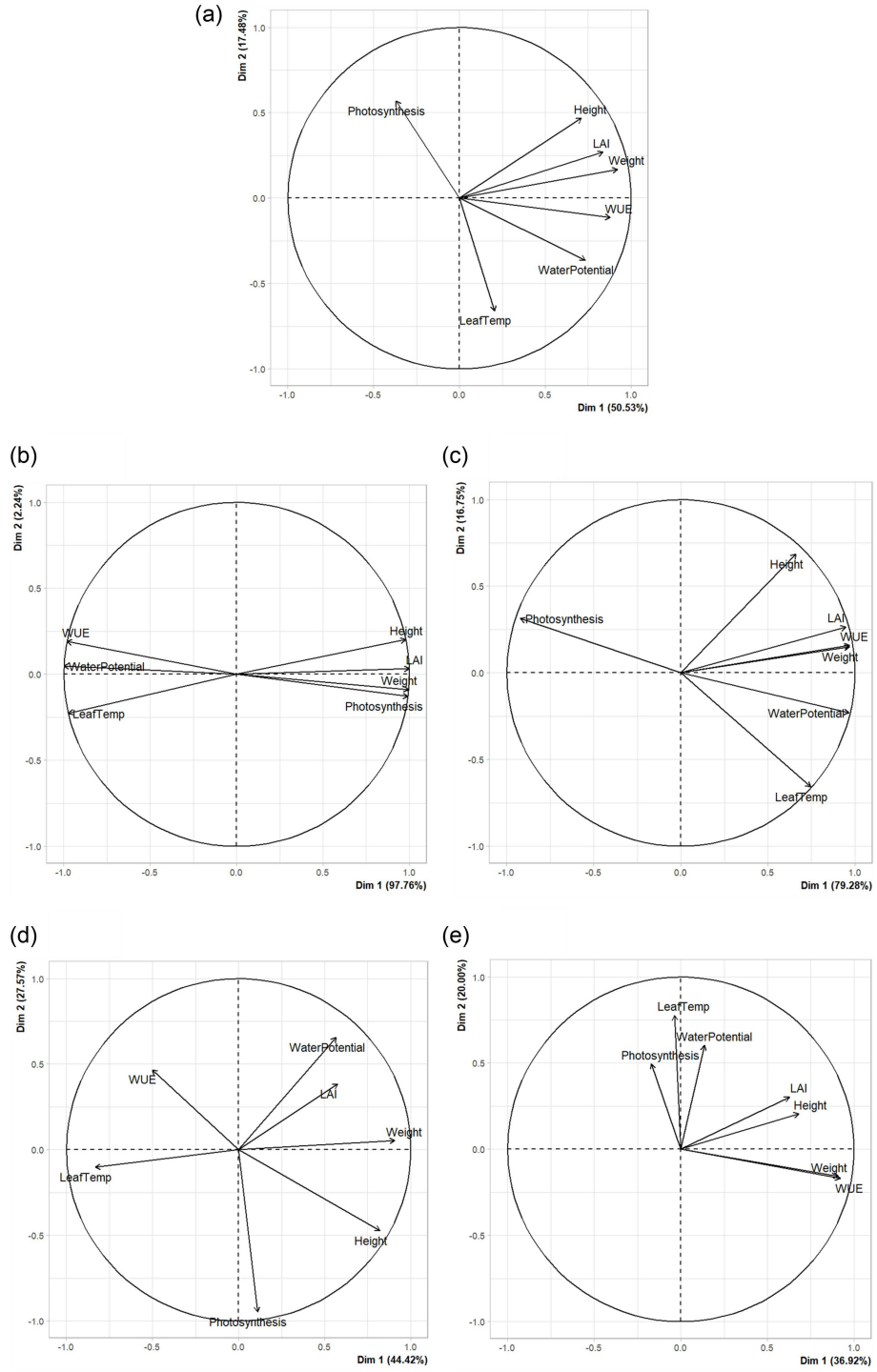


Figure 4

PCA loading diagram from: (a) the D_T data matrix of the total variability; (b) the D_G data matrix of the variability between hyper-slices; (c) the D_K data matrix of the variability between slices; (d) the D_{KG} variability data matrix of interaction; and (e) the D_W data matrix of the variability within hyper-slices and slices. The first two principal components (Dim 1 and Dim 2) define the axes along with their respective proportion of explained variance. Directional vectors indicate the loading values for each variable, including Height, Weight, Leaf Area Index (LAI), Leaf Temperature (Leaf Temp), Photosynthesis, Water Use Efficiency (WUE), and Water Potential (generated in R)

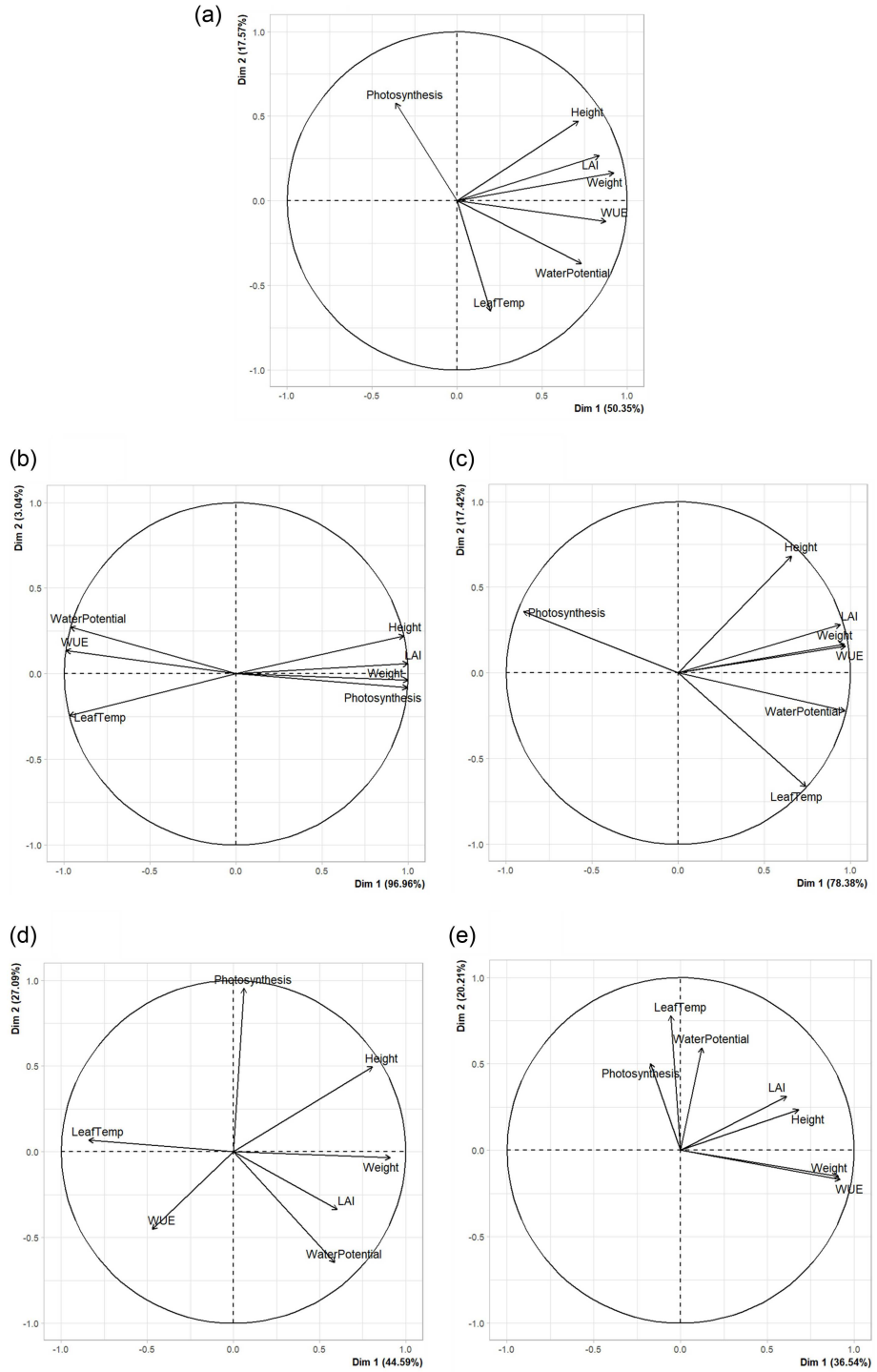
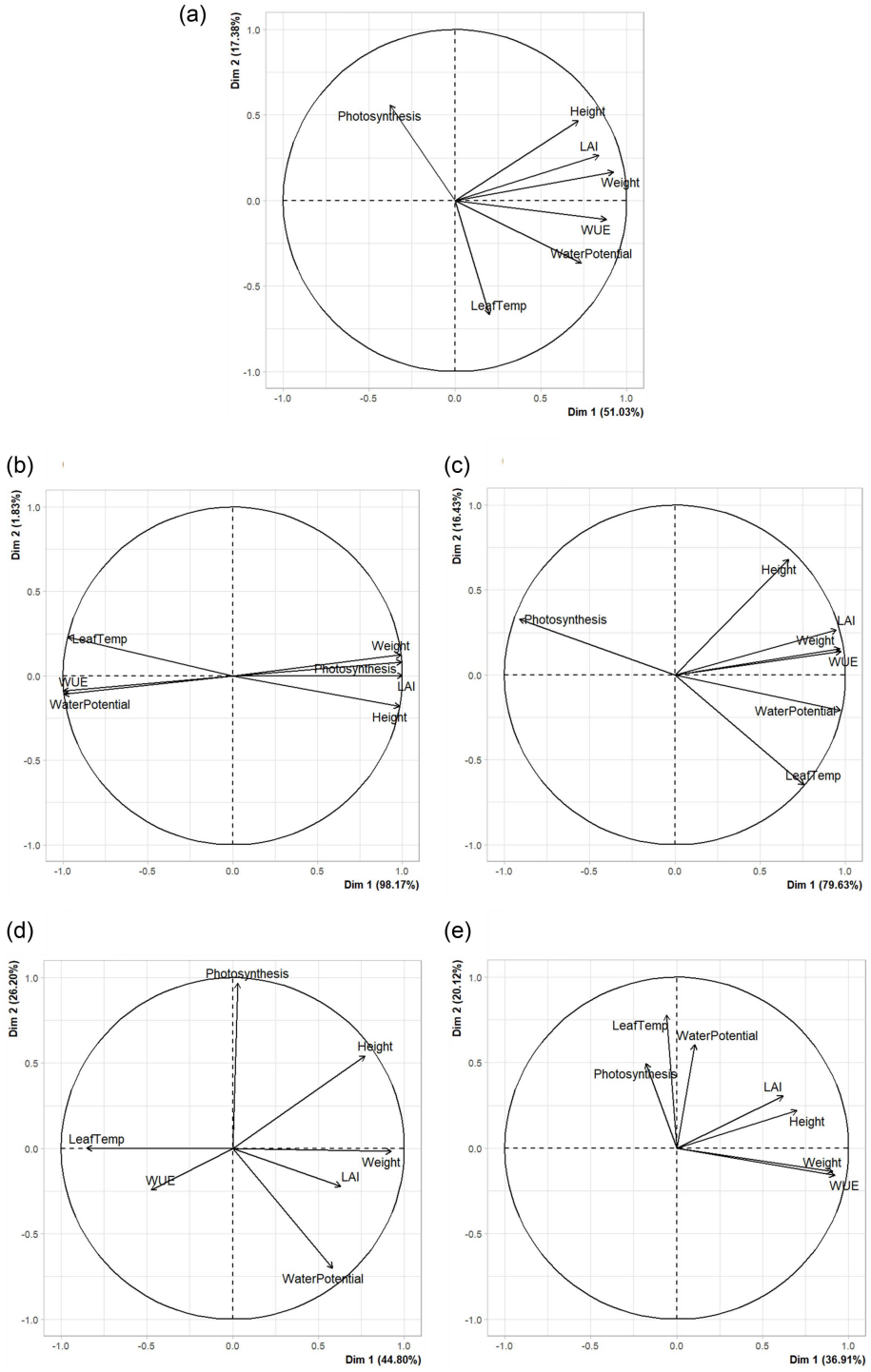


Figure 5

PCA loading diagram from: (a) the X'_T data matrix of the total variability; (b) the X'_G data matrix of the variability between hyper-slices; (c) the X'_K data matrix of the variability between slices; (d) the X'_{KG} variability data matrix of interaction; and (e) the X'_W data matrix of the variability within hyper-slices and slices. The first two principal components (Dim 1 and Dim 2) define the axes along with their respective proportion of explained variance. Directional vectors indicate the loading values for each variable, including Height, Weight, Leaf Area Index (LAI), Leaf Temperature (Leaf Temp), Photosynthesis, Water Use Efficiency (WUE), and Water Potential (generated in R)



Main Effects and the "Type III" approach (see Appendix E) due to the small imbalance of slices and hyper-slices. Therefore, the adjustments performed by the "Type III" approach were very small. However, the fact that the PCA results, and especially the variability structures of the PCA loading plots, were common for this particular data cube does not mean that the two methodological approaches are equivalent. As mentioned in Section 1.7, the "Type III" approach employs projection-based mathematical procedures and adjusted effects in order to decompose the total variability of an unbalanced four-dimensional data cube. Therefore, the resulting variability components represent adjusted sources of variation rather than variability structures derived directly from the observed data.

In contrast, the Unadjusted Main Effects approach resembles the classical ANOVA framework and allows the study of variability structures and components based directly on the observed data. For these reasons, the Unadjusted Main Effects method was considered the most appropriate methodological approach for the objectives of the present study, namely, the study of variability structures and components in unbalanced four-dimensional data cubes. Therefore, the discussion that follows corresponds to the unbalanced data cubes analyzed using the Unadjusted Main Effects method.

Regarding the data cubes $\underline{\mathbf{X}}$, $\underline{\mathbf{D}}$, and $\underline{\mathbf{X}}$', the total variance percentages explained by the first two principal components for the data matrices $\mathbf{X_T}$, $\mathbf{X_G}$, $\mathbf{X_K}$, $\mathbf{X_{KG}}$, and $\mathbf{X_W}$ were 68.01%, 100%, 96.03%, 71.99%, and 56.92%, respectively. The total variance percentages explained by the first two principal components for the data matrices $\mathbf{D_T}$, $\mathbf{D_G}$, $\mathbf{D_K}$, $\mathbf{D_{KG}}$, and $\mathbf{D_W}$ were 67.92%, 100%, 95.80%, 71.68%, and 56.75%, respectively. The total variance percentages explained by the first two principal components for the data matrices $\mathbf{X'_T}$, $\mathbf{X'_G}$, $\mathbf{X'_K}$, $\mathbf{X'_{KG}}$, and $\mathbf{X'_W}$ were 68.41%, 100%, 96.06%, 71%, and 57.03%, respectively.

As for the correlations between the variables, the data matrices $\mathbf{X_T}$ and $\mathbf{X_K}$ and $\mathbf{X'_T}$ and $\mathbf{X'_K}$, as well as the data matrices $\mathbf{D_T}$ and $\mathbf{D_K}$, had a common structure. In contrast, the remaining data matrices presented a different structure. As for the contribution of the variables to the first two principal components, all data matrices presented a different structure.

In other words, the total variability of the cube and the variability between slices (basil varieties) of the cube shared a common structure. In contrast, the variability between hyper-slices (irrigation levels), the variability of interaction, and the variability within hyper-slices and slices presented a different structure.

Regarding the biological interpretation of the results, the study of total variability showed that the first principal component represented the biological growth and hydrological status of basil plants. Specifically, high positive loadings were observed for height, weight, LAI, WUE, and water potential. On the other hand, the second principal component represented physiological parameters related to the crop microclimate, with a negative loading for leaf temperature and a positive loading for photosynthesis.

The study of variability between hyper-slices revealed a substantial effect of irrigation levels, as they affected all parameters positively or negatively. Specifically, high positive loadings were observed for height, weight, LAI, and photosynthesis, while high negative loadings were observed for leaf temperature, WUE, and water potential. These results indicated that higher irrigation levels enhanced basil plant growth and photosynthetic activity [46], while reducing WUE (possibly due to excessive water consumption) [47] and decreasing leaf temperature (possibly due to increased transpiration) [48].

The study of variability between slices showed that the five basil plant varieties differ mainly in their biological development and hydrological status, both of which showed high positive loadings in the first principal component. In contrast, photosynthesis showed a high negative loading in the first principal component. Height contributed slightly to differentiation among varieties through the second principal component, although it was not a dominant criterion of separation.

The study of variability of interaction between hyper-slices and slices revealed a substantial interaction effect between varieties and irrigation levels, as each variety responded differently to various irrigation levels. The first principal component represented the biological growth of basil plants, with height, weight, and LAI showing positive loadings, while leaf temperature and WUE showed negative loadings. On the other hand, the second principal component represented physiological characteristics of basil plants, such as photosynthesis (negative loading) and water potential (positive loading).

The study of variability within hyper-slices and slices revealed substantial internal variability within each irrigation level and basil variety. Particularly, WUE and basil plant weight showed substantial variability within the same slice. However, this variability might also be due to exogenous factors, such as differences in microclimate, experimental errors, and measurement errors [49].

Based on the above, the proposed strategy allowed for a detailed examination of the effect of hyper-slices and slices on the overall structure of the three data cubes by highlighting the dominant influence of hyper-slices (irrigation levels) on the variability of the data cube. Furthermore, the proposed strategy provided important and interesting information about the components and structure of variability through visualizations in lower dimensions. The proposed strategy could also be applied in addition to the MANOVA method, providing in-depth information about the variability of data cubes. Therefore, it was considered more important and safer to study the structure of the variability of a data cube in depth in order to extract important information about the effect and contribution of its slices and hyper-slices.

It is worth noting that the dimensionality reduction strategies for three-dimensional data cubes proposed by Kyrana et al. [1] can also be applied to four-dimensional data cubes. However, the variability decomposition strategy for unbalanced four-dimensional data cubes remains controversial, raising methodological and interpretive interest.

Moreover, to the best of the authors' knowledge, there are no comparative studies between balanced and unbalanced data cubes using the same dataset, with respect to the ways of decomposing the total variability. Consequently, the controversial case of unbalanced data cubes with $p \geq 4$ remains unaddressed. In other methodological approaches, the conceptual and structural role of the slices and/or hyper-slices of the cube, as well as their interaction, is not taken into account. It is worth noting that investigating the effect of the slices and/or hyper-slices on the overall data cube structure is of "vital" importance, as this could lead to simpler or more complex practical strategies for managing and analyzing the cube.

The present study provided a detailed presentation of the mathematical expressions of the variability data matrices. The row

and column elements of the variability data matrices are ready for the direct application of either PCA or another alternative dimensionality reduction method.

In terms of limitations, this study applied a specific dimensionality reduction strategy to particular and indicative simulated datasets. More specifically, the steps involved in this strategy and the dimensionality reduction method used depend on the objectives of each study and the observed data. Furthermore, the proposed dimensionality reduction strategy assumes linear relationships between the variables. Also, the R and Python programming languages, which were used for data analysis and simulation, have limitations regarding their versions and the written code. In addition, a specific method and strategy for the total variability decomposition were selected for the controversial case of the unbalanced four-dimensional data cubes. Therefore, caution should be exercised when generalizing the results and conclusions derived from the numerical data, analysis strategy, and software used in this study.

For future research, it is recommended that various clustering strategies be applied, in terms of the distance metric and the linkage method, either to the initial data cubes or to the PCA components' scores from each data variability matrix. Furthermore, the proposed dimensionality reduction strategy could be applied to data cubes with greater imbalance, in terms of sample size (number of objects), between slices and hyper-slices.

This study is considered to make a significant contribution to both the international scientific community and literature due to the alternative analysis strategy it proposes. This dimensionality reduction strategy provides a creative tool for managing and analyzing data, particularly when adapted to the objectives of each study. This strategy also serves as an important and alternative tool for data preprocessing in machine learning. Essentially, researchers can select the strategy and method that best align with their observed data and the objectives of their study. In other words, this study offers an alternative approach to managing and analyzing data in the form of four-dimensional data cubes.

Acknowledgments

We would like to thank Dr Iakovos Kalamartzis and the Laboratory of Agronomy at the School of Agriculture, Aristotle University of Thessaloniki, for granting us access to the experiment's primary data.

Ethical Statement

The "Basil" dataset used in this study was collected from a field experiment and is not publicly available. The experiment's primary data were provided by Professor Christos Dordas, Dr Iakovos Kalamartzis, and the Laboratory of Agronomy at the School of Agriculture, Aristotle University of Thessaloniki. Since the study involved no human participants or animals, ethical approval was not required.

Conflicts of Interest

The authors declare that they have no conflicts of interest to this work.

Data Availability Statement

Data are available from Professor Christos Dordas (co-author) upon reasonable request.

Author Contribution Statement

Zacharenia Kyrana: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Data curation, Writing – original draft, Visualization. **Nikolaos Papafilippou:** Writing – review & editing. **Emmanouil D. Pratsinakis:** Writing – review & editing. **Alexandra-Maria Michaelidou:** Writing – review & editing. **Christos Dordas:** Validation, Investigation, Resources, Data curation, Writing – review & editing. **Angelos Markos:** Methodology, Writing – review & editing. **George C. Menexes:** Conceptualization, Methodology, Validation, Investigation, Data curation, Writing – review & editing, Supervision, Project administration.

References

[1] Kyrana, Z., Pratsinakis, E. D., Papafilippou, N., Michaelidou, A. M., Kiranas, E., & Menexes, G. C. (2026). Dimensionality reduction in data cubes with independent slices. In S. D. Anastasiadou, T. G. Chadjipadelis, N. Koutsoupias, A. Markos, & G. Menexes (Eds.), *Modern international developments in data analysis and multivariate statistical analysis* (pp. 37–72). IGI Global Scientific Publishing. https://doi.org/10.4018/979-8-3693-9400-7.ch002

[2] Mantas, A., Yfantis, F., Bilidas, D., Stamoulis, G., Kondylatos, S., Prapas, I., . . . , & Koubarakis, M. (2024). Plato: A semantic data cube system using ontology-based data access technologies. *IEEE Access*, *12*, 130356–130374. https://doi.org/10.1109/access.2024.3453494

[3] Kroonenberg, P. M. (2008). *Applied multiway data analysis*. Wiley.

[4] Codesido, S., Hanafi, M., Gagnebin, Y., González-Ruiz, V., Rudaz, S., & Boccard, J. (2021). Network principal component analysis: A versatile tool for the investigation of multigroup and multiblock datasets. *Bioinformatics*, *37*(9), 1297–1303. https://doi.org/10.1093/bioinformatics/btaa954

[5] Kyrana, Z., Pratsinakis, E. D., Papafilippou, N., Markos, A., Dordas, C., & Menexes, G. C. (2025). Evaluating dimensionality reduction strategies on mixed-type datasets: A comparative analysis using Python, R and SPSS. *International Journal of Sustainable Agricultural Management and Informatics*, *11*(4), 445–469. https://doi.org/10.1504/IJSAMI.2024.10065202

[6] Jia, W., Sun, M., Lian, J., & Hou, S. (2022). Feature dimensionality reduction: A review. *Complex & Intelligent Systems*, *8*(3), 2663–2693. https://doi.org/10.1007/s40747-021-00637-x

[7] Hair, J. F., Black, W. C., Babin, B. J., & Anderson, R. E. (2010). *Multivariate data analysis: A global perspective* ((7th Ed.) ed.). Pearson Education.

[8] Holubova, H. (2023). A comparative analysis of the principal component method and parallel analysis in working with official statistical data. *Statistics in Transition new series*, *24*(1), 199–212. https://doi.org/10.59170/stattrans-2023-011

[9] Cruz, A., Carneiro, S., Pontes, S., Oliveira, J., Lima, J., Costa, V., . . . , & Fechine, P. (2023). Sensing materials: Optical sensing based on carbon quantum dots. In R. Narayan (Ed.), *Encyclopedia of sensors and biosensors* (pp. 542–559). Elsevier. https://doi.org/10.1016/b978-0-12-822548-6.00025-x

[10] Alhuzali, T., Beh, E. J., & Stojanovski, E. (2022). Multiple correspondence analysis as a tool for examining Nobel Prize data from 1901 to 2018. *PLOS One*, *17*(4), e0265929. https://doi.org/10.1371/journal.pone.0265929

[11] Tabacchi, G., Navarra, G. A., Scardina, A., Thomas, E., D'Amico, A., Gene-Morales, J., . . . , & Bellafiore, M. (2025). A multiple correspondence analysis of the fear of falling, sociodemographic, physical and mental health factors in older adults. *Scientific Reports*, *15*(1), 6341. https://doi.org/10.1038/s41598-025-89702-w

[12] Li, J., Wang, H., Chen, Y., Chen, Y., Jiao, J., Chen, T., Cao, X., Zhao, C., . . . Zhao, C. (2023). Can CATPCA be utilized for spatial modeling? A case of the generation susceptibility of gully head in a watershed. *CATENA*, *232*, 107456. https://doi.org/10.1016/j.catena.2023.107456

[13] Vasu, D., Tiwari, G., Sahoo, S., Dash, B., Jangir, A., Sharma, R. P., . . . , & Chandran, P. (2021). A minimum data set of soil morphological properties for quantifying soil quality in coastal agroecosystems. *CATENA*, *198*, 105042. https://doi.org/10.1016/j.catena.2020.105042

[14] Mothobi, N., van Ryneveld, L., & Graham, M. A. (2021). A cross-country comparison of mathematics teachers' beliefs about technology in education. *International Journal of Information and Communication Technology Education*, *17*(4), 1–13. https://doi.org/10.4018/IJICTE.20211001.oa13

[15] Zolanvari, M., Yang, Z., Khan, K., Jain, R., & Meskin, N. (2023). TRUST XAI: Model-agnostic explanations for AI with a case study on IIoT security. *IEEE Internet of Things Journal*, *10*(4), 2967–2978. https://doi.org/10.1109/jiot.2021.3122019

[16] Ufeli, C. P., Sattar, M. U., Hasan, R., & Mahmood, S. (2025). Enhancing customer segmentation through factor analysis of mixed data (FAMD)-based approach using K-Means and hierarchical clustering algorithms. *Information*, *16*(6), 441. https://doi.org/10.3390/info16060441

[17] Jihaoui, M. E., Abra, O. E. K., & Mansouri, K. (2025). Factors affecting student academic performance: A combined factor analysis of mixed data and multiple linear regression analysis. *IEEE Access*, *13*, 15946–15964. https://doi.org/10.1109/access.2025.3532099

[18] Eslami, A., Kohler, A., Qannari, E. M., & Bougeard, S. (2013). General overview of methods of analysis of multi-group datasets. *Revue des Nouvelles Technologies de l'Information*, *25*, 108–123.

[19] Roux, B., & Rouanet, H. (2005). *Geometric data analysis: From correspondence analysis to structured data analysis*. Kluwer Academic Publishers. https://doi.org/10.1007/1-4020-2236-0

[20] Smilde, B. J., Botman, E., de Ruiter, R. D., Smit, J. M., Teunissen, B. P., Lubbers, W. D., . . . , & Eekhoff, E. M. W. (2022). Monitoring and management of fibrodysplasia ossificans progressiva: Current perspectives. *Orthopedic Research and Reviews*, *14*, 113–120. https://doi.org/10.2147/ORR.S337491

[21] Rotari, M., Diaz, V. F., de Ketelaere, B., & Kulahci, M. (2024). An extension of PARAFAC to analyze multi-group three-way data. *Chemometrics and Intelligent Laboratory Systems*, *246*, 105089. https://doi.org/10.1016/j.chemolab.2024.105089

[22] Anowar, F., Sadaoui, S., & Selim, B. (2021). Conceptual and empirical comparison of dimensionality reduction algorithms (PCA. In *Computer Science Review* (*40*, 100378.KPCA, LDA, MDS, SVD, LLE, ISOMAP, LE, ICA, t-SNE). https://doi.org/10.1016/j.cosrev.2021.100378

[23] Roy, A. (2025). Quantifying the complexity of agricultural data for regression and classification problems. In P. K. Dutta, A. Hamad, A. K. Haghi, & P. K. Prabhakar (Eds.), *Food and Industry 5.0: Transforming the food system for a sustainable future* (pp. 171–183). Springer. https://doi.org/10.1007/978-3-031-76758-6_12

[24] Triana-Martinez, J., Álvarez-Meza, A., & Castellanos-Dominguez, G. (2025). Enhancing agricultural data interpretability and visualization with TabNet-driven feature extraction and local biplots. *Results in Engineering*, *27*, 106672. https://doi.org/10.1016/j.rineng.2025.106672

[25] Papafilippou, N., Kyrana, Z., Pratsinakis, E., Kiranas, E., Michailidou, A., Markos, A., & Menexes, G. (2026). Integrating data analysis methods with machine learning algorithms for mixed data types: Does this combination improve predictive models' accuracy? *Journal of Data Science and Intelligent Systems*, *4*(1), 39–50. https://doi.org/10.47852/bonviewjdsis52023906

[26] Eckhardt, C. M., Madjarova, S. J., Williams, R. J., Ollivier, M., Karlsson, J., Pareek, A., & Nwachukwu, B. U. (2023). Unsupervised machine learning methods and emerging applications in healthcare. *Knee Surgery, Sports Traumatology, Arthroscopy*, *31*(2), 376–381. https://doi.org/10.1007/s00167-022-07233-7

[27] Morales, E. F., & Escalante, H. J. (2022). A brief introduction to supervised, unsupervised, and reinforcement learning. In A. A. Torres-García, C. A. Reyes-García, L. Villaseñor-Pineda, & O. Mendoza-Montoya (Eds.), *Biosignal processing and classification using computational learning and intelligence: Principles, algorithms, and applications* (pp. 111–129). Academic Press. https://doi.org/10.1016/b978-0-12-820125-1.00017-8

[28] Amato, A., & di Lecce, V. (2023). Data preprocessing impact on machine learning algorithm performance. *Open Computer Science*, *13*(1), 20220278. https://doi.org/10.1515/comp-2022-0278

[29] Alanazi, A. (2022). Using machine learning for healthcare challenges and opportunities. *Informatics in Medicine Unlocked*, *30*, 100924. https://doi.org/10.1016/j.imu.2022.100924

[30] Banerjee, R., Bharti, Das., Srivastava, P., Ankita, V., Kataria, S., Ahmed, B., & Varshney, N. (2023). An overview of statistical techniques for analysis of data in agricultural research. In A. E. O. Elkhalifa (Ed.), *Emerging issues in agricultural sciences* (*8*, pp. 190–206). B P International. https://doi.org/10.9734/bpi/eias/v8/6853C

[31] Jones, G. P., Stambaugh, C., Stambaugh, N., & Huber, K. E. (2023). Analysis of variance. In A. E. M. Eltorai, J. A. Bakal, D. W. Kim, & D. E. Wazer (Eds.), *Translational radiation oncology* (pp. 171–177). Academic Press. https://doi.org/10.1016/b978-0-323-88423-5.00041-8

[32] Din, I. U., & Hayat, Y. (2021). ANOVA or MANOVA for correlated traits in agricultural experiments. *Sarhad Journal of Agriculture*, *37*(4), 1250–1259. https://doi.org/10.17582/journal.sja/2021/37.4.1250.1259

[33] de Melo, M. B., Daldegan-Bueno, D., Oliveira, M. G. M., & de Souza, A. L. (2022). Beyond ANOVA and MANOVA for repeated measures: Advantages of generalized estimated equations and generalized linear mixed models and its use in neuroscience research. *European Journal of Neuroscience*, *56*(12), 6089–6098. https://doi.org/10.1111/ejn.15858

[34] Pérez-Cova, M., Platikanov, S., Stoll, D. R., Tauler, R., & Jaumot, J. (2022). Comparison of multivariate ANOVA-based approaches for the determination of relevant variables in experimentally designed metabolomic studies. *Molecules*, *27*(10), 3304. https://doi.org/10.3390/molecules27103304

[35] Abegaz, F., Abedini, D., Dong, L., Westerhuis, J. A., van Eeuwijk, F., Bouwmeester, H., & Smilde, A. K. (2025). Analysis of microbiome high-dimensional experimental design data using generalized linear models and ANOVA simultaneous component analysis. *Frontiers in Microbiomes*, *4*, 1584516. https://doi.org/10.3389/frmbi.2025.1584516

[36] de Figueiredo, M., Giannoukos, S., Wüthrich, C., Zenobi, R., & Rutledge, D. N. (2023). A tutorial on the analysis of multifactorial designs from one or more data sources using AComDim. *Journal of Chemometrics*, *37*(7), e3384. https://doi.org/10.1002/cem.3384

[37] dos Santos, L. O., dos Santos, A. M. P., Ferreira, M. M., Ferreira, S. L., & Nepomuceno, A. F. S. (2022). The use of ANOVA-PCA and DD-SIMCA in the development of corn flour laboratory reference materials. *Food Chemistry*, *367*, 130748. https://doi.org/10.1016/j.foodchem.2021.130748

[38] de Haan, J. R., Wehrens, R., Bauerschmidt, S., Piek, E., van Schaik, R. C., & Buydens, L. M. C. (2007). Interpretation of ANOVA models for microarray data using PCA. *Bioinformatics*, *23*(2), 184–190. https://doi.org/10.1093/bioinformatics/btl572

[39] Smilde, A. K., Jansen, J. J., Hoefsloot, H. C. J., Lamers, R. A. N., van der Greef, J., & Timmerman, M. E. (2005). ANOVA-simultaneous component analysis (ASCA): A new tool for analyzing designed metabolomics data. *Bioinformatics*, *21*(13), 3043–3048. https://doi.org/10.1093/bioinformatics/bti476

[40] Ali, N., Jansen, J., van den Doel, A., Tinnevelt, G. H., & Bocklitz, T. (2021). WE-ASCA: The weighted-effect ASCA for analyzing unbalanced multifactorial designs—A Raman spectra-based example. *Molecules*, *26*(1), 66. https://doi.org/10.3390/molecules26010066

[41] Williams, J. D. (1972). Two-way fixed effects analysis of variance with disproportionate cell frequencies. *Multivariate Behavioral Research*, *7*(1), 67–83. http://dx.doi.org/10.1207/s15327906mbr0701_4

[42] Langsrud, Ø. (2003). ANOVA for unbalanced data: Use Type II instead of Type III sums of squares. *Statistics and Computing*, *13*(2), 163–167. https://doi.org/10.1023/A:1023260610025

[43] Genç, S., & Mendeş, M. (2022). A biological perspective on interpreting interaction effect. *Turkish Journal of Veterinary & Animal Sciences*, *46*(3), 535–542. https://doi.org/10.55730/1300-0128.4224

[44] LaMotte, L. R. (2020). A formula for Type III sums of squares. *Communications in Statistics – Theory and Methods*, *49*(13), 3126–3136. https://doi.org/10.1080/03610926.2019.1586933

[45] Kalamartzis, I., Papakaloudis, P., & Dordas, C. (2023). Basil (Ocimum basilicum) Landraces can be used in a water-limited environment. *Plants*, *12*(13), 2425. https://doi.org/10.3390/plants12132425

[46] Zhang, H., Cao, H., Zhao, Z., Dou, Z., Liao, Z., Bai, Z., Li, S., Fan, J., … Fan, J. (2025). Effects of irrigation interval and irrigation level on growth, photosynthesis, fruit yield, quality, and water-nitrogen use efficiency of drip-fertigated greenhouse tomatoes (Solanum lycopersicum L.). *Agronomy*, *15*(5), 1068. https://doi.org/10.3390/agronomy15051068

[47] Okasha, A. M., Deraz, N., Elmetwalli, A. H., Elsayed, S., Falah, M. W., Farooque, A. A., & Yaseen, Z. M. (2022). Effects of irrigation method and water flow rate on irrigation performance, soil salinity, yield, and water productivity of cauliflower. *Agriculture*, *12*(8), 1164. https://doi.org/10.3390/agriculture12081164

[48] Gräf, M., Immitzer, M., Hietz, P., & Stangl, R. (2021). Water-stressed plants do not cool: Leaf surface temperature of living wall plants under drought stress. *Sustainability*, *13*(7), 3910. https://doi.org/10.3390/su13073910

[49] Kemppinen, J., & Niittynen, P. (2022). Microclimate relationships of intraspecific trait variation in sub-arctic plants. *Oikos*, *2022*(12), e09507. https://doi.org/10.1111/oik.09507

How to Cite: Kyrana, Z., Papafilippou, N., Pratsinakis, E. D., Michaelidou, A.-M., Dordas, C., Markos, A., & Menexes, G. C. (2026). Dimensionality Reduction Strategy on Four-Dimensional Data Cubes with Independent Slices and Hyper-Slices Focusing on Variability Structure and Components. *Journal of Data Science and Intelligent Systems*. https://doi.org/10.47852/bonviewJDSIS62028503

Appendix A

The mathematical formulas of "Type III" SSCP matrices for unbalanced four-dimensional data cubes are presented below [44].

Let $\mathbf{D}$ be an unbalanced stacked data matrix with different objects per slice and hyper-slice, with dimensions $N \times J$ and d_{ijkg} elements, where $i = 1, 2, \ldots, n_{kg}$, $j = 1, 2, \ldots, J$, $k = 1, 2, \ldots, K$, $g = 1, 2, \ldots, G$, $N = \sum_{g=1}^{G} \sum_{k=1}^{K} n_{kg}$, $n_k = \sum_{g=1}^{G} n_{kg}$ and $n_g = \sum_{k=1}^{K} n_{kg}$.

Also, $\mathbf{d}_{ikg} = \begin{bmatrix} d_{i1kg} \\ d_{i2kg} \\ \vdots \\ d_{iJkg} \end{bmatrix}_{J\times 1}$ is the J-dimensional vector of variables for the object i in the k^{th} slice within g^{th} hyper-slice, $\bar{\boldsymbol{d}}_{...} = \frac{1}{N} \sum_{g=1}^{G} \sum_{k=1}^{K} \sum_{i=1}^{n_{kg}} \mathbf{d}_{ikg} = \begin{bmatrix} \bar{d}_{.1..} \\ \bar{d}_{.2..} \\ \vdots \\ \bar{d}_{.J..} \end{bmatrix}_{J\times 1}$ is the J-dimensional vector of weighted means for the stacked data matrix $\mathbf{D}$, $\bar{\mathbf{d}}_{.kg} = \frac{1}{n_{kg}} \sum_{i=1}^{n_{kg}} \mathbf{d}_{ikg} = \begin{bmatrix} \bar{d}_{.1kg} \\ \bar{d}_{.2kg} \\ \vdots \\ \bar{d}_{.Jkg} \end{bmatrix}_{J\times 1}$ is the J-dimensional vector of weighted means for the k^{th} slice within the g^{th} hyper-slice, $\widehat{\mathbf{M}} = \begin{bmatrix} \bar{\mathbf{d}}_{.11}^{\mathrm{T}} \\ \bar{\mathbf{d}}_{.21}^{\mathrm{T}} \\ \vdots \\ \bar{\mathbf{d}}_{.K1}^{\mathrm{T}} \\ \bar{\mathbf{d}}_{.12}^{\mathrm{T}} \\ \vdots \\ \bar{\mathbf{d}}_{.KG}^{\mathrm{T}} \end{bmatrix}$ is the matrix of observed cell means, with dimensions $KG \times J$ (each row of $\widehat{\mathbf{M}}$ is a J-dimensional mean vector), and $\mathbf{W} = \mathbf{D}_{\mathbf{n}}^{-\mathbf{1}} = diag(11, 21, K1, 12, \text{and } KG \text{ should be indices of } n)$ is the weight matrix, where $\mathbf{D}_n = diag(11, 21, K1, 12, \text{and } KG \text{ should be indices of } n)$ is the diagonal matrix of cell frequencies.

Regarding the SSCP matrix of hyper-slices, let $\mathbf{C_G}$ be a full-rank $(G-1) \times G$ contrast matrix and $\mathbf{L_G} = \mathbf{C_G} \otimes \frac{1}{K}\mathbf{1}_K^{\mathrm{T}}$ the "Type III" hypothesis matrix, with dimensions $(G-1) \times KG$, for the hyper-slice factor, where $\mathbf{1}_K^{\mathrm{T}}$ is the $1 \times K$ vector of ones and $\otimes$ denotes the Kronecker product.

Regarding the SSCP matrix of slices, let $\mathbf{C_K}$ be a full-rank $(K-1) \times K$ contrast matrix and $\mathbf{L_K} = \frac{1}{G}\mathbf{1}_G^{\mathrm{T}} \otimes \mathbf{C_K}$ the "Type III" hypothesis matrix, with dimensions $(K-1) \times KG$, for the slice factor, where $\mathbf{1}_G^{\mathrm{T}}$ is the $1 \times G$ vector of ones.

Regarding the SSCP matrix of interaction between hyper-slices and slices, let $\mathbf{L_{KG}} = \mathbf{C_G} \otimes \mathbf{C_K}$ the "Type III" hypothesis matrix, with dimensions $(G-1)(K-1) \times KG$, for the interaction factor.

Then, it is true that the $J \times J$ SSCP matrix of the total variability is defined as $\mathbf{D_t} = \sum_{g=1}^{G} \sum_{k=1}^{K} \sum_{i=1}^{n_{kg}} (\mathbf{d}_{ikg} - \bar{\mathbf{d}}_{...})(\mathbf{d}_{ikg} - \bar{\mathbf{d}}_{...})^{\mathrm{T}}$. The $J \times J$ SSCP matrix of the variability between hyper-slices is defined as $\mathbf{D_g} = (\mathbf{L_G}\widehat{\mathbf{M}})^{\mathrm{T}} (\mathbf{L_G}\mathbf{W}\mathbf{L_G}^{\mathrm{T}})^{-1} (\mathbf{L_G}\widehat{\mathbf{M}})$. The $J \times J$ SSCP matrix of the variability between slices is defined as $\mathbf{D_k} = (\mathbf{L_K}\widehat{\mathbf{M}})^{\mathrm{T}} (\mathbf{L_K}\mathbf{W}\mathbf{L_K}^{\mathrm{T}})^{-1} (\mathbf{L_K}\widehat{\mathbf{M}})$. The $J \times J$ SSCP matrix of the variability of interaction between hyper-slices and slices is defined as $\mathbf{D_{kg}} = (\mathbf{L_{KG}}\widehat{\mathbf{M}})^{\mathrm{T}} (\mathbf{L_{KG}}\mathbf{W}\mathbf{L_{KG}}^{\mathrm{T}})^{-1} (\mathbf{L_{KG}}\widehat{\mathbf{M}})$. The $J \times J$ SSCP matrix of the variability within hyper-slices and slices is defined as $\mathbf{D_w} = \sum_{g=1}^{G} \sum_{k=1}^{K} \sum_{i=1}^{n_{kg}} \left(\mathbf{d}_{ikg} - \bar{\mathbf{d}}_{.kg}\right)\left(\mathbf{d}_{ikg} - \bar{\mathbf{d}}_{.kg}\right)^{\mathrm{T}}$.

In other words, $\mathbf{D_g} = \mathbf{D}^{\mathrm{T}}(\mathbf{P_F} - \mathbf{P_{-G}})\mathbf{D}$, where $\mathbf{D}$ is the $N \times J$ response matrix (or else the stacked data matrix) and $\mathbf{P_F} = \mathbf{X}(\mathbf{X}^{\mathrm{T}}\mathbf{X})^{-}\mathbf{X}^{\mathrm{T}}$ is the projection matrix for the full model. Also, $\mathbf{X}$ is the $N \times q$ design matrix (where $q = 1 + G + K + KG$ is he total number of model parameters), $(\mathbf{X}^{\mathrm{T}}\mathbf{X})^{-}$ denotes a generalized inverse, which is required because factorial design matrices are typically rank-deficient, and $\mathbf{P_{-G}}$ is the projection matrix for a restricted model containing all terms except the effect of hyper-slices. Similarly, $\mathbf{D_k} = \mathbf{D}^{\mathrm{T}}(\mathbf{P_F} - \mathbf{P_{-K}})\mathbf{D}$, where $\mathbf{P_{-K}}$ is the projection matrix after removing the effect of slices, while $\mathbf{D_{kg}} = \mathbf{D}^{\mathrm{T}}(\mathbf{P_F} - \mathbf{P_{-KG}})\mathbf{D}$, where $\mathbf{P_{-KG}}$ is the projection matrix after removing the effect of interaction.

These matrices are obtained from linear hypotheses in the general linear model. Therefore, in the case of unbalanced data cubes, it is true that $\mathbf{D_t} \neq \mathbf{D_g} + \mathbf{D_k} + \mathbf{D_{kg}} + \mathbf{D_w}$, because hyper-slices and slices are non-orthogonal.

Appendix B

The code for simulating the data in the Python programming language is shown below.

```
# Library import for data processing
import pandas as pd
# Library import for numerical calculations
import numpy as np
# Library import for Multinormal Distribution
from scipy.stats import multivariate_normal
# Library import for calculating eigenvalues and eigenvectors
from scipy.linalg import eigh
# Set random seed for reproducibility
np.random.seed(42)
# Checking for the existence of positive semi-definite matrices
def is_positive_semi_definite(matrix):
eigenvalues = np.linalg.eigvals(matrix)
return np.all(eigenvalues >= -1e-8)
# Finding the nearest positive semi-definite matrix
def nearest_positive_semi_definite(matrix):
eigenvalues, eigenvectors = eigh(matrix)
eigenvalues[eigenvalues < 0] = 0
return eigenvectors @ np.diag(eigenvalues) @ eigenvectors.T
# Definition of vectors with the mean values per slice within each hyper-slice
mean_vectors = {‘
(Level ’1, ‘’Burns): np.array([65.10, 700.75, 3.30, 28.41, 6.24, 7.21, 3.61]),‘
(Level ’1, ‘’Cinnamon): np.array([53.15, 521.25, 2.14, 30.03, 7.46, 5.36, 3.66]),‘
(Level ’’1,’Sweet): np.array([64.10, 317.50, 1.47, 27.22, 9.49, 3.26, 2.83]),‘
(Level ’’’1,’’Rend): np.array([48.95, 192.25, 1.61, 27.24, 11.46, 1.98, 2.69]),‘
(Level ’’1,’Thai): np.array([30.90, 200.00, 0.77, 27.84, 8.68, 2.06, 2.67]),‘
(Level ’2, ‘’Burns): np.array([71.00, 858.50, 3.30, 29.24, 10.23, 5.44, 3.50]),‘
(Level ’2, ‘’Cinnamon): np.array([57.15, 601.25, 2.52, 28.60, 9.38, 3.81, 3.78]),‘
(Level ’’2,’Sweet): np.array([67.10, 350.75, 2.16, 26.75, 10.09, 2.22, 2.65]),‘
(Level ’’’2,’’Rend): np.array([50.10, 232.50, 1.49, 27.82, 10.88, 1.47, 2.82]),‘
(Level ’’2,’Thai): np.array([33.50, 270.50, 1.17, 27.16, 12.45, 1.71, 2.39]),‘
(Level ’3, ‘’Burns): np.array([84.20, 1071.50, 3.89, 26.35, 12.11, 4.90, 3.55]),‘
(Level ’3, ‘’Cinnamon): np.array([62.40, 766.75, 3.07, 28.53, 8.12, 3.51, 4.02]),‘
(Level ’’3,’Sweet): np.array([73.45, 324.00, 1.74, 27.27, 13.48, 1.48, 2.41]),‘
(Level ’’’3,’’Rend): np.array([54.10, 219.00, 1.89, 27.34, 12.71, 1.00, 2.50]),‘
(Level ’’3,’Thai): np.array([40.75, 219.33, 1.63, 27.61, 13.04, 1.00, 2.30])
}
# Definition of vectors with the standard deviations per slice within each hyper-slice
std_vectors = {‘
(Level ’1, ‘’Burns): np.array([3.04, 213.99, 1.24, 1.50, 2.16, 2.20, 0.32]),‘
(Level ’1, ‘’Cinnamon): np.array([5.13, 210.45, 0.52, 1.94, 1.74, 2.16, 0.62]),‘
(Level ’’1,’Sweet): np.array([5.26, 101.40, 0.79, 1.30, 1.51, 1.04, 0.58]),‘
(Level ’’’1,’’Rend): np.array([2.11, 53.11, 0.35, 1.37, 2.13, 0.55, 0.50]),‘
(Level ’’1,’Thai): np.array([4.11, 78.09, 0.39, 1.82, 1.55, 0.80, 0.36]),‘
(Level ’2, ‘’Burns): np.array([4.53, 108.71, 0.28, 0.49, 0.72, 0.69, 0.41]),‘
(Level ’2, ‘’Cinnamon): np.array([7.84, 103.38, 0.94, 1.12, 5.06, 0.65, 0.13]),‘
(Level ’’2,’Sweet): np.array([4.12, 98.21, 0.39, 0.65, 1.92, 0.62, 0.27]),‘
(Level ’’’2,’’Rend): np.array([1.91, 97.08, 0.53, 2.21, 1.49, 0.61, 0.52]),‘
(Level ’’2,’Thai): np.array([7.03, 35.30, 0.54, 2.04, 1.75, 0.22, 0.42]),‘
(Level ’3, ‘’Burns): np.array([6.58, 337.41, 0.38, 1.53, 3.76, 1.54, 0.28]),‘
(Level ’3, ‘’Cinnamon): np.array([10.06, 169.59, 0.91, 3.64, 2.82, 0.78, 0.52]),‘
(Level ’’3,’Sweet): np.array([3.19, 104.80, 0.54, 2.09, 3.02, 0.48, 0.31]),‘
(Level ’’’3,’’Rend): np.array([4.56, 55.60, 0.60, 2.12, 3.34, 0.25, 0.65]),‘
(Level ’’3,’Thai): np.array([10.46, 37.61, 1.00, 1.92, 2.87, 0.17, 0.65])
}
# Definition of correlation matrices per slice within each hyper-slice
correlation_matrices = {‘
(Level ’1, ‘’Burns): np.array([
[1.000, 0.871, 0.960, 0.052, -0.363, 0.871, 0.828],
[0.871, 1.000, 0.973, 0.034, -0.080, 1.000, 0.451],
[0.960, 0.973, 1.000, 0.063, -0.180, 0.973, 0.643],
[0.052, 0.034, 0.063, 1.000, 1.000, 0.034, 0.162],
[-0.363, -0.080, -0.180, 1.000, 1.000, -0.080, -0.775],
[0.871, 1.000, 0.973, 0.034, -0.080, 1.000, 0.451],
[0.828, 0.451, 0.643, 0.162, -0.775, 0.451, 1.000]
```

```
]),‘
(Level ’1, ‘’Cinnamon): np.array([
[1.000, 0.976, 0.900, -0.831, -0.956, 0.976, 0.248],
[0.976, 1.000, 0.871, -0.795, -0.962, 1.000, 0.348],
[0.900, 0.871, 1.000, -0.511, -0.984, 0.871, -0.156],
[-0.831, -0.795, -0.511, 1.000, 0.601, -0.795, -0.627],
[-0.956, -0.962, -0.984, 0.601, 1.000, -0.962, 0.315],
[0.976, 1.000, 0.871, -0.795, -0.962, 1.000, 0.348],
[0.248, 0.348, -0.156, -0.627, 0.315, 0.348, 1.000]
]),‘
(Level ’1, ‘’Sweet): np.array([
[1.000, 0.213, 0.491, 0.660, 0.930, 0.213, 0.427],
[0.213, 1.000, 0.081, -0.232, -0.359, 1.000, -0.484],
[0.491, 0.081, 1.000, 0.897, 0.717, 0.081, 0.822],
[0.660, -0.232, 0.897, 1.000, 0.859, -0.232, 0.951],
[0.930, -0.359, 0.717, 0.859, 1.000, -0.359, 0.685],
[0.213, 1.000, 0.081, -0.232, -0.359, 1.000, -0.484],
[0.427, -0.484, 0.822, 0.951, 0.685, -0.484, 1.000]
]),‘
(Level ’1, ‘’Rend): np.array([
[1.000, -0.493, -0.035, 0.890, 0.857, -0.493, 0.933],
[-0.493, 1.000, -0.656, -0.388, -0.915, 1.000, -0.550],
[-0.035, -0.656, 1.000, -0.362, 0.855, -0.656, -0.170],
[0.890, -0.388, -0.362, 1.000, 0.927, -0.388, 0.980],
[0.857, -0.915, 0.855, 0.927, 1.000, -0.915, 0.964],
[-0.493, 1.000, -0.656, -0.388, -0.915, 1.000, -0.550],
[0.933, -0.550, -0.170, 0.980, 0.964, -0.550, 1.000]
]),‘
(Level ’1, ‘’Thai): np.array([
[1.000, 0.533, 0.375, 0.489, -0.340, 0.533, -0.018],
[0.533, 1.000, 0.372, 0.434, -0.933, 1.000, 0.276],
[0.375, 0.372, 1.000, -0.542, -0.987, 0.372, -0.789],
[0.489, 0.434, -0.542, 1.000, -0.043, 0.434, 0.858],
[-0.340, -0.933, -0.987, -0.043, 1.000, -0.933, 0.123],
[0.533, 1.000, 0.372, 0.434, -0.933, 1.000, 0.276],
[-0.018, 0.276, -0.789, 0.858, 0.123, 0.276, 1.000]
]),‘
(Level ’2, ‘’Burns): np.array([
[1.000, 0.948, 0.672, -0.587, -0.909, 0.948, -0.423],
[0.948, 1.000, 0.795, -0.803, -0.914, 1.000, -0.258],
[0.672, 0.795, 1.000, -0.655, -0.366, 0.795, -0.494],
[-0.587, -0.803, -0.655, 1.000, 0.992, -0.803, -0.280],
[-0.909, -0.914, -0.366, 0.992, 1.000, -0.914, 0.125],
[0.948, 1.000, 0.795, -0.803, -0.914, 1.000, -0.258],
[-0.423, -0.258, -0.494, -0.280, 0.125, -0.258, 1.000]
]),‘
(Level ’2, ‘’Cinnamon): np.array([
[1.000, 0.994, 0.217, 0.851, -0.598, 0.994, -0.596],
[0.994, 1.000, 0.299, 0.879, -0.591, 1.000, -0.505],
[0.217, 0.299, 1.000, 0.155, 0.937, 0.299, 0.460],
[0.851, 0.879, 0.155, 1.000, -0.956, 0.879, -0.249],
[-0.598, -0.591, 0.937, -0.956, 1.000, -0.591, 0.601],
[0.994, 1.000, 0.299, 0.879, -0.591, 1.000, -0.505],
[-0.596, -0.505, 0.460, -0.249, 0.601, -0.505, 1.000]
]),‘
(Level ’2, ‘’Sweet): np.array([
[1.000, 0.749, 0.654, 0.216, -0.719, 0.749, -0.606],
[0.749, 1.000, -0.006, 0.436, -0.996, 1.000, -0.794],
[0.654, -0.006, 1.000, -0.263, 0.183, -0.006, -0.078],
[0.216, 0.436, -0.263, 1.000, -0.989, 0.436, 0.195],
[-0.719, -0.996, 0.183, -0.989, 1.000, -0.996, 0.828],
[0.749, 1.000, -0.006, 0.436, -0.996, 1.000, -0.794],
[-0.606, -0.794, -0.078, 0.195, 0.828, -0.794, 1.000]
]),‘
(Level ’2, ‘’Rend): np.array([
[1.000, -0.639, -0.537, -0.218, 0.757, -0.639, -0.016],
[-0.639, 1.000, 0.277, 0.315, -0.847, 1.000, 0.435],
[-0.537, 0.277, 1.000, 0.894, 0.944, 0.277, 0.662],
[-0.218, 0.315, 0.894, 1.000, 0.644, 0.315, 0.921],
```

```
[0.757, -0.847, 0.944, 0.644, 1.000, -0.847, 0.321],
[-0.639, 1.000, 0.277, 0.315, -0.847, 1.000, 0.435],
[-0.016, 0.435, 0.662, 0.921, 0.321, 0.435, 1.000]
]),‘
(Level ’2, ‘’Thai): np.array([
[1.000, 0.094, 0.876, 0.346, 0.281, 0.094, 0.283],
[0.094, 1.000, 0.176, 0.949, 0.841, 1.000, 0.923],
[0.876, 0.176, 1.000, 0.304, -0.137, 0.176, 0.177],
[0.346, 0.949, 0.304, 1.000, 0.976, 0.949, 0.987],
[0.281, 0.841, -0.137, 0.976, 1.000, 0.841, 0.986],
[0.094, 1.000, 0.176, 0.949, 0.841, 1.000, 0.923],
[0.283, 0.923, 0.177, 0.987, 0.986, 0.923, 1.000]
]),‘
(Level ’3, ‘’Burns): np.array([
[1.000, 0.733, 0.337, -0.940, -0.168, 0.733, -0.117],
[0.733, 1.000, -0.025, -0.723, 0.974, 1.000, -0.282],
[0.337, -0.025, 1.000, -0.584, -0.703, -0.025, 0.886],
[-0.940, -0.723, -0.584, 1.000, 0.977, -0.723, -0.196],
[-0.168, 0.974, -0.703, 0.977, 1.000, 0.974, -0.478],
[0.733, 1.000, -0.025, -0.723, 0.974, 1.000, -0.282],
[-0.117, -0.282, 0.886, -0.196, -0.478, -0.282, 1.000]
]),‘
(Level ’3, ‘’Cinnamon): np.array([
[1.000, 0.431, 0.678, -0.003, -0.136, 0.431, 0.304],
[0.431, 1.000, 0.652, -0.257, -0.947, 1.000, 0.844],
[0.678, 0.652, 1.000, 0.477, -0.732, 0.652, 0.192],
[-0.003, -0.257, 0.477, 1.000, 0.814, -0.257, -0.735],
[-0.136, -0.947, -0.732, 0.814, 1.000, -0.947, -0.942],
[0.431, 1.000, 0.652, -0.257, -0.947, 1.000, 0.844],
[0.304, 0.844, 0.192, -0.735, -0.942, 0.844, 1.000]
]),‘
(Level ’3, ‘’Sweet): np.array([
[1.000, 0.789, 0.981, -0.451, 0.373, 0.789, -0.679],
[0.789, 1.000, 0.873, -0.064, -0.273, 1.000, -0.429],
[0.981, 0.873, 1.000, -0.447, 0.228, 0.873, -0.715],
[-0.451, -0.064, -0.447, 1.000, -0.920, -0.064, 0.928],
[0.373, -0.273, 0.228, -0.920, 1.000, -0.273, -0.626],
[0.789, 1.000, 0.873, -0.064, -0.273, 1.000, -0.429],
[-0.679, -0.429, -0.715, 0.928, -0.626, -0.429, 1.000]
]),‘
(Level ’3, ‘’Rend): np.array([
[1.000, 0.691, -0.023, -0.312, 0.335, 0.691, -0.805],
[0.691, 1.000, -0.077, 0.429, -0.582, 1.000, -0.300],
[-0.023, -0.077, 1.000, -0.396, -0.648, -0.077, -0.485],
[-0.312, 0.429, -0.396, 1.000, -0.782, 0.429, 0.724],
[0.335, -0.582, -0.648, -0.782, 1.000, -0.582, -0.272],
[0.691, 1.000, -0.077, 0.429, -0.582, 1.000, -0.300],
[-0.805, -0.300, -0.485, 0.724, -0.272, -0.300, 1.000]
]),‘
(Level ’3, ‘’Thai): np.array([
[1.000, 0.054, 0.168, 0.980, 0.998, 0.054, 0.132],
[0.054, 1.000, -0.212, 0.229, 0.087, 1.000, 0.976],
[0.168, -0.212, 1.000, 0.040, 0.956, -0.212, 0.001],
[0.980, 0.229, 0.040, 1.000, 0.956, 0.229, 0.282],
[0.998, 0.087, 0.956, 0.956, 1.000, 0.087, 0.296],
[0.054, 1.000, -0.212, 0.229, 0.087, 1.000, 0.976],
[0.132, 0.976, 0.001, 0.282, 0.296, 0.976, 1.000]
])}
# Definition of correlation matrices as positive semi-definite
adjusted_correlation_matrices = {}
for key, corr_matrix in correlation_matrices.items():
if not is_positive_semi_definite(corr_matrix):
print(’’fAdjusting correlation matrix for {key} to be positive semi-definite’’.)
adjusted_corr_matrix = nearest_positive_semi_definite(corr_matrix)
adjusted_correlation_matrices[key] = adjusted_corr_matrix
else:
adjusted_correlation_matrices[key] = corr_matrix
# Calculation of covariance matrices
covariance_matrices = {}
```

```
for key, corr_matrix in adjusted_correlation_matrices.items():
std_vector = std_vectors[key]
cov_matrix = np.diag(std_vector) @ corr_matrix @ np.diag(std_vector)
covariance_matrices[key] = cov_matrix
# Creation of data approximating the Multinormal Distribution and definition
    of the number of objects, slices and hyper-slices
num_plants = 1500
num_irrigation_levels = 3
num_varieties = 5
plants_per_group = 500
data_list = []
for irrigation_level in range(1, num_irrigation_levels + 1):
for variety in `['Burns, ``Cinnamon, ``Sweet, ``Rend, ``Thai]:
key = ('fLevel {irrigation_level'}, variety)
mean = mean_vectors[key]
cov_matrix = covariance_matrices[key]
mv_data = multivariate_normal.rvs(mean=mean, cov=cov_matrix, size=plants_per_group)
df = pd.DataFrame(mv_data, columns``=[``Height, ```Weight, ```LAI, ```LeafTemp,
    ```Photosynthesis, ```WUE, ```WaterPotential])
df`['Irrigation_Level] = 'fLevel {irrigation_level'}
df`['Variety] = variety
data_list.append(df)
# Combination of all matrices
simulated_data = pd.concat(data_list, ignore_index=True)
# Saving simulated data as a.csv file
simulated_data.to_csv`(simulated_4D.'csv, index=False)
print``(Simulated dataset saved as `simulated_4D.'``csv)
# Verification of the correlation matrices of the initial data with those of the simulated data
new_correlation_matrices = {}
for irrigation_level in range(1, num_irrigation_levels + 1):
for variety in `['Burns, ``Cinnamon, ``Sweet, ``Rend, ``Thai]:
key = ('fLevel {irrigation_level'}, variety)
group_data = simulated_data[
(simulated_data`['Irrigation_Level] == 'fLevel {irrigation_level'}) &
(simulated_data`['Variety] == variety)
]
corr_matrix = group_data.iloc[:,:7].corr()
new_correlation_matrices[key] = corr_matrix
for key, corr_matrix in correlation_matrices.items():
print(''fOriginal Correlation Matrix for {key''}:)
print(corr_matrix)
print(''fNew Correlation Matrix for {key''}:)
print(new_correlation_matrices[key])
print``(\''n)
```
```

Appendix C

The main PCA results (loadings, communalities, percentages of explained variance, scree plots, and score plots) for each four-dimensional data cube are presented below.

Table C1

PCA loadings (only loadings with an absolute value greater than or equal to 0.450 are presented) and communalities of the first two principal components (PC1 and PC2), for each variable (Height, Weight, Leaf Area Index [LAI], Leaf Temperature [Leaf Temp], Photosynthesis, Water Use Efficiency [WUE], and Water Potential), of the balanced data cube $\underline{X}$ ("Basil-Sim 1"). Irrigation levels correspond to the variable of hyper-slices, with three levels (40%, 70%, and 100%). Varieties correspond to the variable of slices, with five levels (Burns, Cinnamon, Sweet, Rend, and Thai)

		PCA loadings		
Variability	Variable	PCI	PC2	Communality
Total	Height	.711		.724
	Weight	.920		.875
	LAI	.836		.771
	Leaf Temp		–.660	.478
	Photosynthesis		.568	.460
	WUE	.877		.783
	Water Potential	.733		.670
Between hyper-slices	Height	.979		1.000
	Weight	.996		1.000
	LAI	.999		1.000
	Leaf Temp	–.973		1.000
	Photosynthesis	.992		1.000
	WUE	–.981		1.000
	Water Potential	–.998		1.000
Between slices	Height		.685	.908
	Weight	.969		.961
	LAI	.947		.967
	Leaf Temp	.745		.990
	Photosynthesis	–.921		.946
	WUE	.969		.964
	Water Potential	.965		.986
Interaction	Height	.821		.900
	Weight	.908		.827
	LAI	.574		.480
	Leaf Temp	–.834		.707
	Photosynthesis		–.947	.907
	WUE	–.496		.475
	Water Potential		.649	.752
Within hyper-slices and slices	Height	.682		.507
	Weight	.911		.855
	LAI	.628		.486
	Leaf Temp		.777	.605
	Photosynthesis		.495	.274
	WUE	.920		.876
	Water Potential		.603	.382

Figure C1
Scree plot of PCA for the total variability of data cubes X, D, and X' (generated in SPSS)

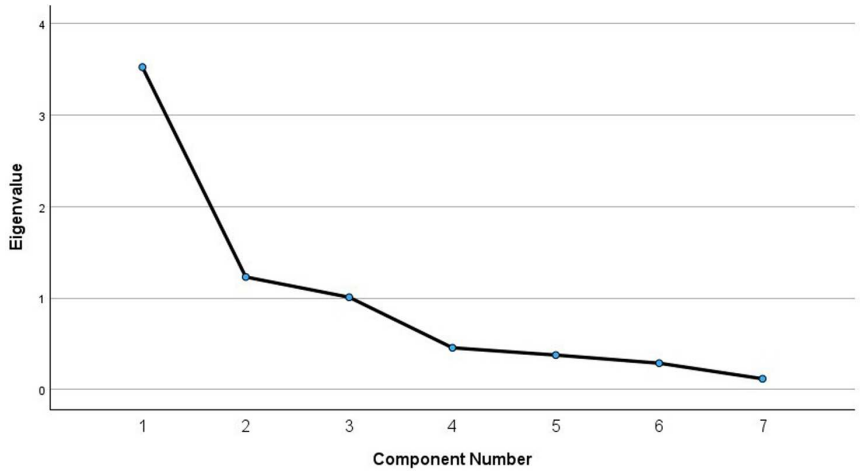


Figure C2
Score plot of the first two principal components (PC1 and PC2) for the total variability of data cubes X, D, and X' (generated in SPSS)

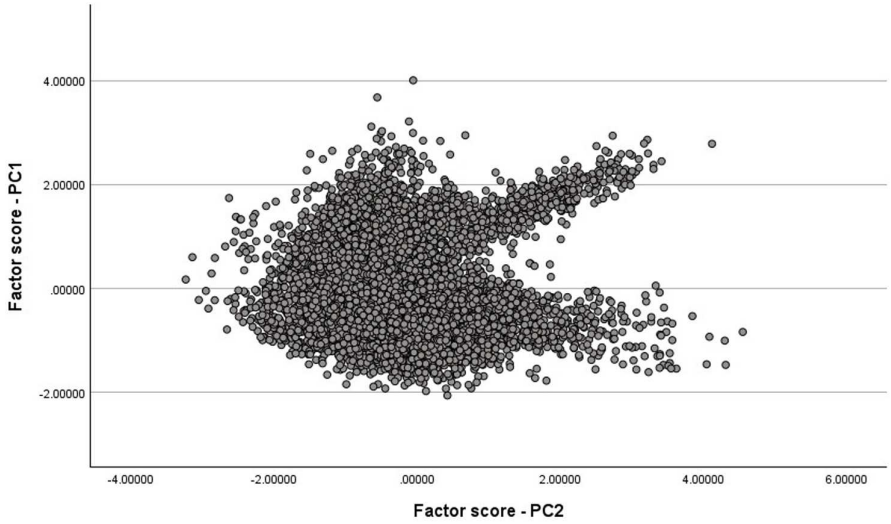


Figure C3
Scree plot of PCA for the between hyper-slices variability of data cubes X, D, and X' (generated in SPSS)

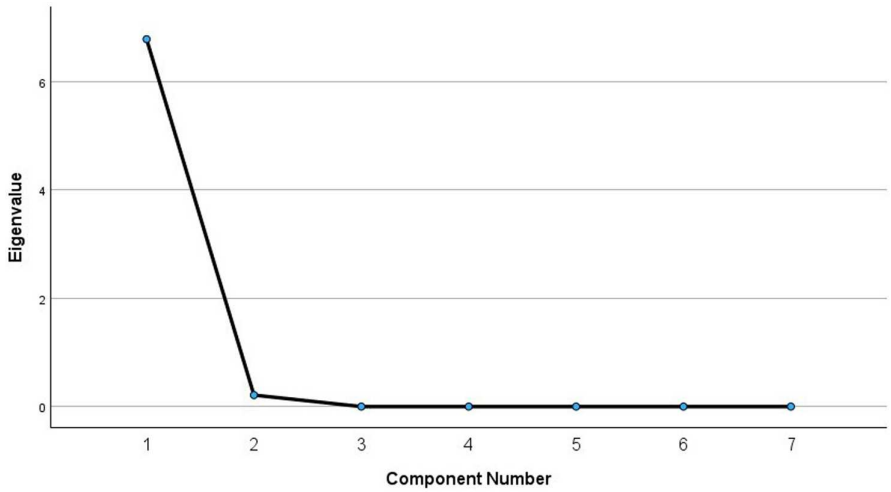


Table C2

PCA loadings (only loadings with an absolute value greater than or equal to 0.450 are presented) and communalities of the first two principal components (PC1 and PC2), for each variable (Height, Weight, Leaf Area Index [LAI], Leaf Temperature [Leaf Temp], Photosynthesis, Water Use Efficiency [WUE], and Water Potential), of the unbalanced data cube D ("Basil-Sim 2"). Irrigation levels correspond to the variable of hyper-slices, with three levels (40%, 70%, and 100%). Varieties correspond to the variable of slices, with five levels (Burns, Cinnamon, Sweet, Rend, and Thai)

		PCA loadings		
Variability	Variable	PCI	PC2	Communality
Total	Height	.713		.732
	Weight	.921		.875
	LAI	.836		.770
	Leaf Temp		–.651	.462
	Photosynthesis		.576	.462
	WUE	.876		.781
	Water Potential	.731		.672
Between hyper-slices	Height	.975		1.000
	Weight	.999		1.000
	LAI	.999		1.000
	Leaf Temp	–.970		1.000
	Photosynthesis	.997		1.000
	WUE	–.991		1.000
	Water Potential	–.963		1.000
Between slices	Height		.684	.901
	Weight	.964		.958
	LAI	.942		.967
	Leaf Temp	.741		.991
	Photosynthesis	–.899		.935
	WUE	.971		.967
	Water Potential	.967		.987
Interaction	Height	.800		.897
	Weight	.909		.828
	LAI	.607		.478
	Leaf Temp	–.839		.708
	Photosynthesis		.956	.917
	WUE	–.457		.404
	Water Potential		–.628	.761
Within hyper-slices and slices	Height	.680		.518
	Weight	.911		.852
	LAI	.610		.470
	Leaf Temp		.779	.610
	Photosynthesis		.502	.282
	WUE	.919		.874
	Water Potential		.593	.367

Table C3

PCA loadings (only loadings with an absolute value greater than or equal to 0.450 are presented) and communalities of the first two principal components (PC1 and PC2), for each variable (Height, Weight, Leaf Area Index [LAI], Leaf Temperature [Leaf Temp], Photosynthesis, Water Use Efficiency [WUE], and Water Potential), of the balanced data cube $\underline{X}$' ("Basil-Sim 2"). Irrigation levels correspond to the variable of hyper-slices, with three levels (40%, 70%, and 100%). Varieties correspond to the variable of slices, with five levels (Burns, Cinnamon, Sweet, Rend, and Thai)

		PCA loadings		
Variability	Variable	PCI	PC2	Communality
Total	Height	.718		.734
	Weight	.922		.878
	LAI	.839		.773
	Leaf Temp		–.667	.485
	Photosynthesis		.557	.453
	WUE	.880		.787
	Water Potential	.738		.678
Between hyper-slices	Height	.982		1.000
	Weight	.993		1.000
	LAI	1.000		1.000
	Leaf Temp	–.971		1.000
	Photosynthesis	.997		1.000
	WUE	–.996		1.000
	Water Potential	–.989		1.000
Between slices	Height		.683	.912
	Weight	.967		.959
	LAI	.947		.967
	Leaf Temp	.755		.990
	Photosynthesis	–.912		.939
	WUE	.975		.969
	Water Potential	.972		.988
Interaction	Height	.772		.892
	Weight	.923		.852
	LAI	.629		.446
	Leaf Temp	–.853		.727
	Photosynthesis		.969	.939
	WUE	–.468		.287
	Water Potential		–.700	.829
Within hyper-slices and slices	Height	.700		.538
	Weight	.906		.840
	LAI	.619		.476
	Leaf Temp		.781	.613
	Photosynthesis		.496	.278
	WUE	.919		.869
	Water Potential		.606	.378

Table C4
Percentages of variance explained by the first two principal components (PC1 and PC2) for the balanced data cube X ("Basil-Sim 1")

Variability	Principal component	% of variance	% of total variance
Total	PC1	50.53	
	PC2	17.48	68.01
Between hyper-slices	PC1	97.76	
	PC2	2.24	100.00
Between slices	PC1	79.28	
	PC2	16.75	96.03
interaction	PC1	44.42	
	PC2	27.57	71.99
Within hyper-slices and slices	PC1	36.92	
	PC2	20.00	56.92

Table C5
Percentages of variance explained by the first two principal components (PC1 and PC2) for the unbalanced data cube D ("Basil-Sim 2")

Variability	Principal component	% of variance	% of total variance
Total	PC1	50.35	
	PC2	17.57	67.92
Between hyper-slices	PC1	96.96	
	PC2	3.04	100.00
Between slices	PC1	78.38	
	PC2	17.42	95.80
Interaction	PC1	44.59	
	PC2	27.09	71.68
Within hyper-slices and slices	PC1	36.54	
	PC2	20.21	56.75

Table C6
Percentages of variance explained by the first two principal components (PC1 and PC2) for the balanced data cube X' ("Basil-Sim 2")

Variability	Principal component	% of Variance	% of total variance
Total	PC1	51.03	
	PC2	17.38	68.41
Between hyper-slices	PC1	98.17	
	PC2	1.83	100.00
Between slices	PC1	79.63	
	PC2	16.43	96.06
Interaction	PC1	44.80	
	PC2	26.20	71.00
Within hyper-slices and slices	PC1	36.91	
	PC2	20.12	57.03

Figure C4
Score plot of the first two principal components (PC1 and PC2) for the between hyper-slices variability of data cubes X, D, and X'. Irrigation levels correspond to the variable of hyper-slices, with three levels (40%, 70%, and 100%) (generated in SPSS)

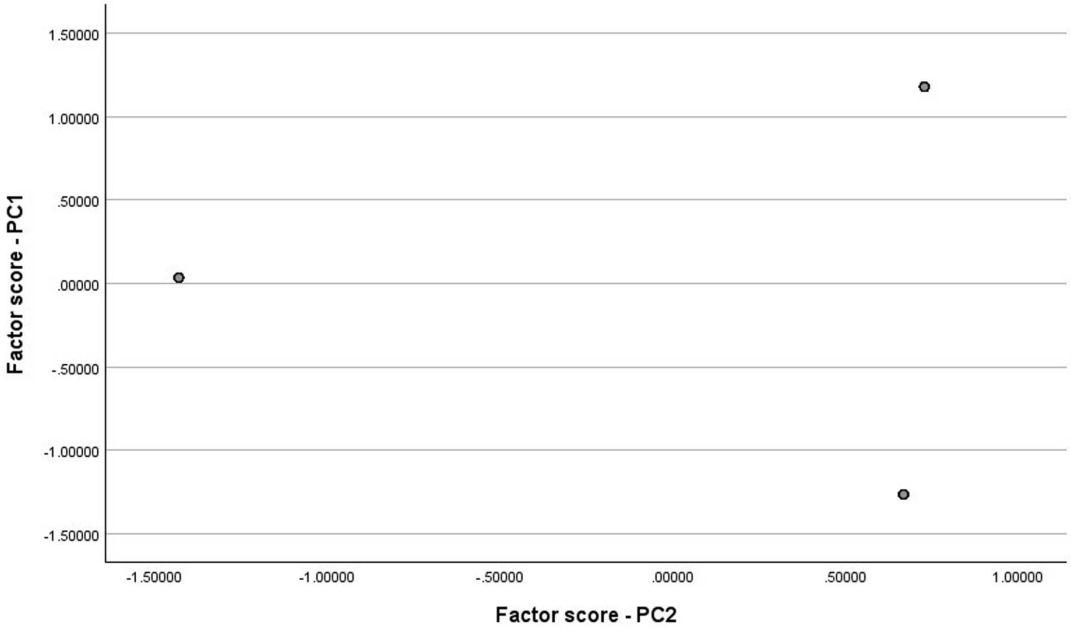


Figure C5
Scree plot of PCA for the between slices variability of data cubes X, D, and X' (generated in SPSS)

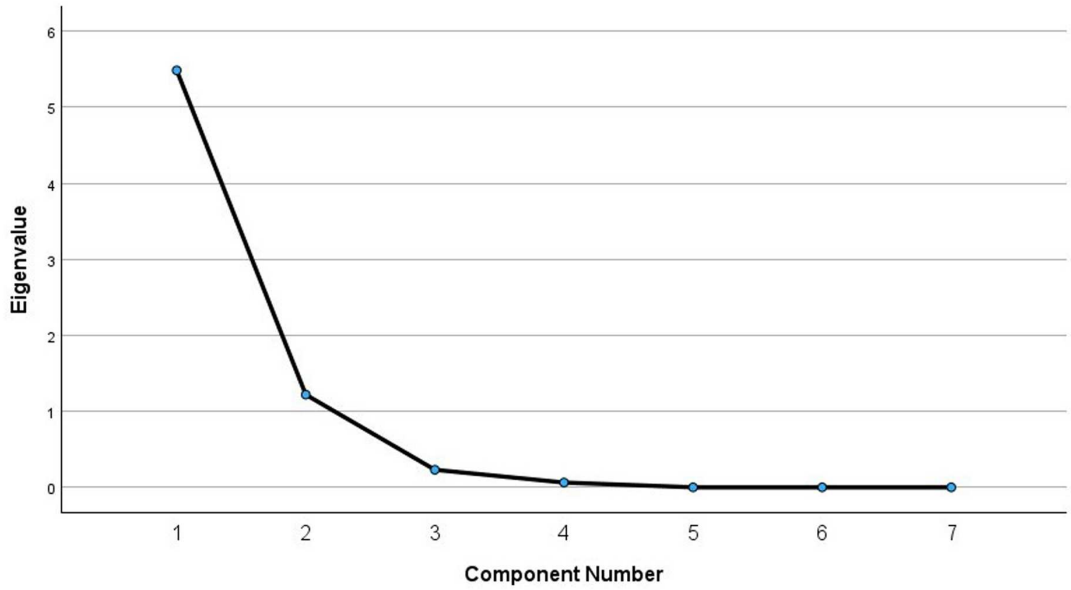


Figure C6

Score plot of the first two principal components (PC1 and PC2) for the between slices variability of data cubes X, D, and X'. Varieties correspond to the variable of slices, with five levels (Burns, Cinnamon, Sweet, Rend, and Thai) (generated in SPSS)

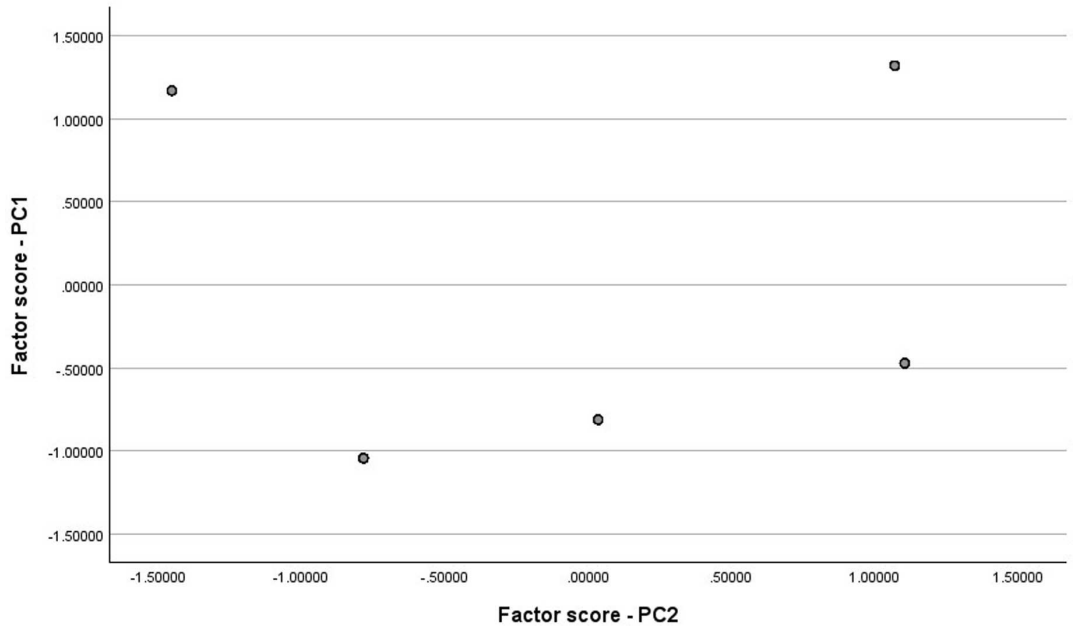


Figure C7

Scree plot of PCA for the interaction between hyper-slices and slices variability of data cubes X, D, and X' (generated in SPSS)

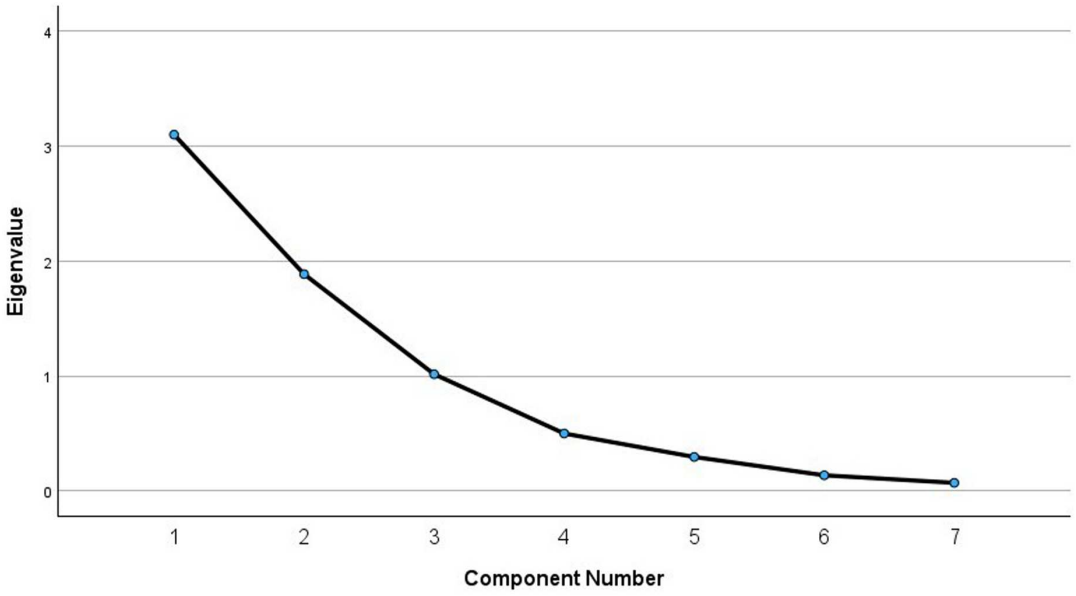


Figure C8

Score plot of the first two principal components (PC1 and PC2) for the interaction between hyper-slices and slices variability of data cubes X, D, and X'. Irrigation levels correspond to the variable of hyper-slices, with three levels (40%, 70%, and 100%). Varieties correspond to the variable of slices, with five levels (Burns, Cinnamon, Sweet, Rend, and Thai) (generated in SPSS)

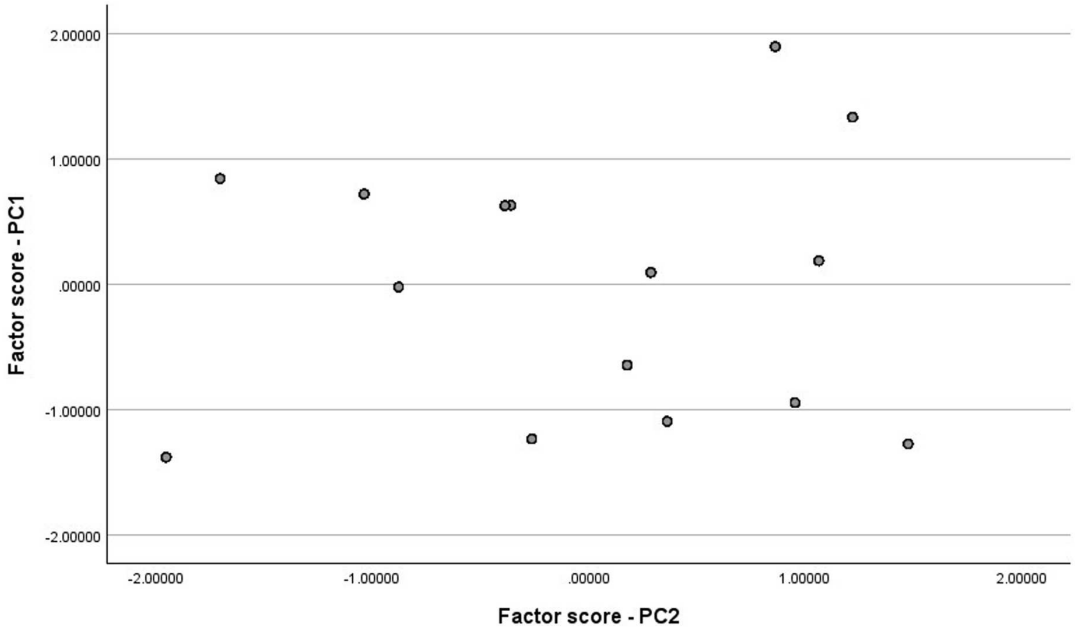


Figure C9

Scree plot of PCA for the within hyper-slices and slices variability of data cubes X, D, and X' (generated in SPSS)

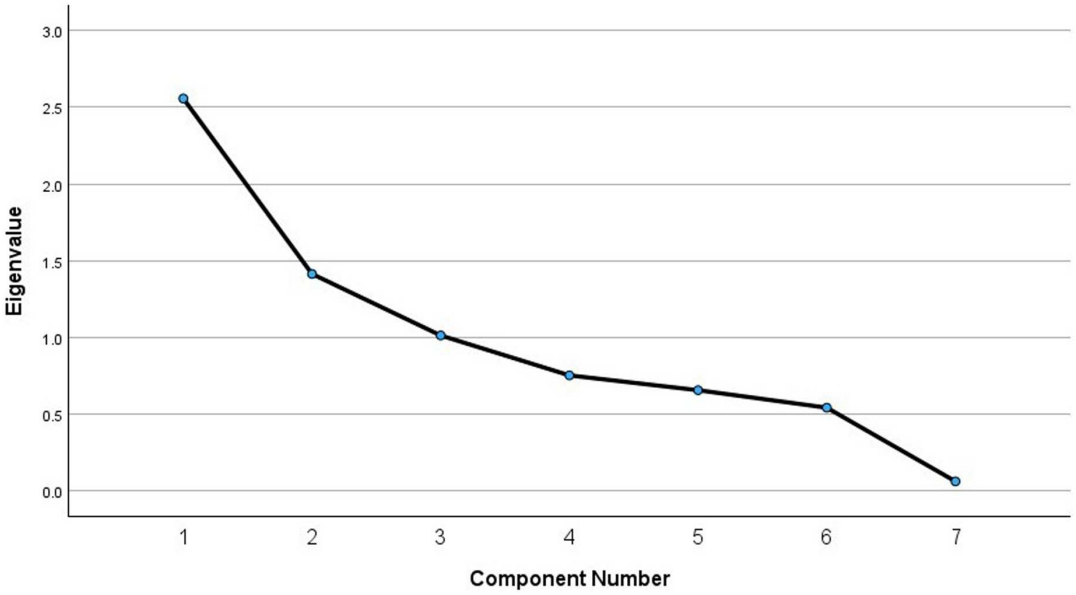
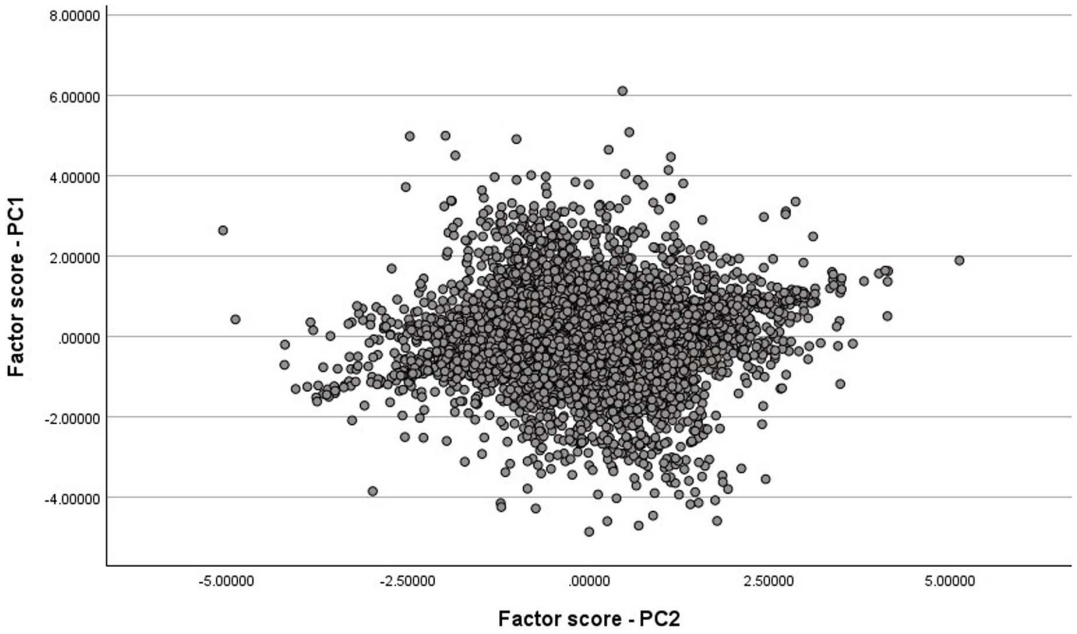


Figure C10
Score plot of the first two principal components (PC1 and PC2) for the within hyper-slices and slices variability of data cubes X, D, and X' (generated in SPSS)



Appendix D

The R code for applying PCA to each variability data matrix of the balanced data cubes is presented below.

```
# Import library for reading.xlsx files
library(readxl)
# Import libraries for data processing and management
library(dplyr)
library(tidyr)
library(purrr)
# Importing libraries for applying PCA using the ‘’’’PCA command
library(factoextra)
library(FactoMineR)
# Import data from Excel
data = read_excel‘’(C:/Users. . . /./ Basil-Sim 1.’’xlsx)
# Defining symbols and preparing data
data = data %>%
rename(G = ‘Irrigation_Level‘,
K = Variety)
vars = c‘’(’’Height, ‘’’’Weight, ‘’’’LAI, ‘’’’LeafTemp, ‘’’’Photosynthesis, ‘’’’WUE, ‘’’’WaterPotential)
X = data %>% select(all_of(vars)) %>% as.matrix()
I = data %>% count(G, K) %>% pull(n) %>% unique()
stopifnot(length(I) == 1)
I = I[1]
K_levels = sort(unique(data$K))
G_levels = sort(unique(data$G))
K = length(K_levels)
G = length(G_levels)
J = ncol(X)
# Finding the total means (. . . x)
global_mean = colMeans(X) # 1 x J
# Finding the data matrix of total variability (□_□)
X_T = X
X_T = X_T - matrix(rep(global_mean, each = nrow(X_T)), ncol = J, byrow = FALSE)
# Finding the means per hyper-slice (x..g)
X_bar_g = data %>%
group_by(G) %>%
summarise(across(all_of(vars), mean),.groups = ‘’’’drop) %>%
arrange(G)
# Finding the data matrix of the between hyper-slices variability (□_G)
X_G = as.matrix(X_bar_g[match(data$G, X_bar_g$G), -1]) # IKG x J
X_G = X_G - matrix(rep(global_mean, each = nrow(X_G)), ncol = J, byrow = FALSE)
# Finding the means per slice (x..k)
X_bar_k = data %>%
group_by(K) %>%
summarise(across(all_of(vars), mean),.groups = ‘’’’drop) %>%
arrange(K)
# Finding the data matrix of the between slices variability (□_K)
X_K = as.matrix(X_bar_k[match(data$K, X_bar_k$K), -1]) # IKG x J
X_K = X_K - matrix(rep(global_mean, each = nrow(X_K)), ncol = J, byrow = FALSE)
# Finding the means per slice within each hyper-slice (x.kg)
X_bar_kg = data %>%
group_by(G, K) %>%
summarise(across(all_of(vars), mean),.groups = ‘’’’drop) %>%
arrange(G, K)
X_KG_raw = as.matrix(X_bar_kg[match(interaction(data$G, data$K),
 interaction(X_bar_kg$G, X_bar_kg$K)), -c(1,2)])
# Finding the variability data matrix of interaction (□_KG)
X_KG = X_KG_raw - as.matrix(X_bar_g[match(data$G, X_bar_g$G), -1]) - as.matrix(X_bar_k[match
    (data$K, X_bar_k$K), -1]) + matrix(rep(global_mean, each = nrow(X_KG_raw)), ncol = J, byrow = FALSE)
# Finding the data matrix of the within hyper-slices and slices variability (□_W)
X_W = X - X_KG_raw
# Finding of the SSCP matrices
SSCP_G = t(X_G) %*% X_G
SSCP_K = t(X_K) %*% X_K
SSCP_KG = t(X_KG) %*% X_KG
SSCP_W = t(X_W) %*% X_W
SSCP_T = t(X_T) %*% X_T
# Application of PCA to each variability data matrix (extraction of the first two principal
```

```
    components and the corresponding plot)
PCA(X_T, scale.unit = TRUE, ncp = 2, graph = TRUE)
PCA(X_G, scale.unit = TRUE, ncp = 2, graph = TRUE)
PCA(X_K, scale.unit = TRUE, ncp = 2, graph = TRUE)
PCA(X_KG, scale.unit = TRUE, ncp = 2, graph = TRUE)
PCA(X_W, scale.unit = TRUE, ncp = 2, graph = TRUE)
```

The R code for applying PCA to each variability data matrix of the unbalanced data cube is presented below.

```
# Import data from Excel
data = read_excel‘‘(C:/Users. . . /./ Basil-Sim 2.’’xlsx)
# Defining symbols and preparing data
data = data %>%
rename(G = ‘Irrigation_Level‘,
K = Variety)
vars = c‘‘(’’Height, ‘‘’’Weight, ‘‘’’LAI, ‘‘’’LeafTemp, ‘‘’’Photosynthesis,
    ‘‘’’WUE, ‘‘’’WaterPotential)
D = data %>% select(all_of(vars)) %>% as.matrix()
N = nrow(D)
G_levels = unique(data$G)
K_levels = unique(data$K)
G = length(G_levels)
K = length(K_levels)
J = ncol(D)
# Finding the total weighted means (. . . d)
global_mean_weighted = colSums(D) / N
# Finding the weighted means per hyper-slice (d..g)
D_bar_g = data %>%
group_by(G) %>%
summarise(across(all_of(vars), ~ sum(.) / n()),.groups = ‘‘’’drop) %>%
arrange(G)
# Finding the weighted means per slice (d..k)
D_bar_k = data %>%
group_by(K) %>%
summarise(across(all_of(vars), ~ sum(.) / n()),.groups = ‘‘’’drop) %>%
arrange(K)
# Finding the weighted means per slice within each hyper-slice (d.kg)
D_bar_kg = data %>%
group_by(G, K) %>%
summarise(across(all_of(vars), mean),.groups = ‘‘’’drop) %>%
arrange(G, K)
# Finding the unweighted means per hyper-slice (d..g□)
D_bar_g_unweighted = D_bar_kg %>%
group_by(G) %>%
summarise(across(all_of(vars), mean),.groups = ‘‘’’drop)
# Finding the unweighted means per slice (d..k□)
D_bar_k_unweighted = D_bar_kg %>%
group_by(K) %>%
summarise(across(all_of(vars), mean),.groups = ‘‘’’drop)
# Finding the total unweighted means (. . . d□)
global_mean_unweighted = D_bar_kg %>%
summarise(across(all_of(vars), mean)) %>%
as.numeric()
# Finding the data matrix of total variability (D_□)
D_T = D - matrix(global_mean_weighted, nrow = N, ncol = length(vars), byrow = TRUE)
# Finding the data matrix of the between hyper-slices variability (D_G)
D_G = matrix(0, nrow = N, ncol = length(vars))
for (g in 1:G) {
idx = which(data$G == G_levels[g])
D_G[idx,] = matrix(as.numeric(D_bar_g[g, -1]), nrow = length(idx), ncol = length(vars), byrow = TRUE) -
matrix(global_mean_weighted, nrow = length(idx), ncol = length(vars), byrow = TRUE)
}
# Finding the data matrix of the between slices variability (D_K)
D_K = matrix(0, nrow = N, ncol = length(vars))
for (k in 1:K) {
idx = which(data$K == K_levels[k])
group_means = as.numeric(D_bar_k[k, vars])
D_K[idx,] = matrix(group_means, nrow = length(idx), ncol = length(vars), byrow = TRUE) -
matrix(global_mean_weighted, nrow = length(idx), ncol = length(vars), byrow = TRUE)
}
```

```
# Finding the variability data matrix of interaction (D_KG)
D_KG = matrix(0, nrow = N, ncol = length(vars))
for (i in 1:nrow(data)) {
g = which(G_levels == data$G[i])
k = which(K_levels == data$K[i])
kg_mean = as.numeric(D_bar_kg %>% filter(G == G_levels[g], K == K_levels[k]) %>% select(all_of(vars)))
g_mean = as.numeric(D_bar_g_unweighted %>% filter(G == G_levels[g]) %>% select(all_of(vars)))
k_mean = as.numeric(D_bar_k_unweighted %>% filter(K == K_levels[k]) %>% select(all_of(vars)))
D_KG[i,] = kg_mean - g_mean - k_mean + global_mean_unweighted
}
# Finding the data matrix of the within hyper-slices and slices variability (D_W)
D_bar_kg_matrix = matrix(0, nrow = N, ncol = length(vars))
for (i in 1:nrow(data)) {
g = which(G_levels == data$G[i])
k = which(K_levels == data$K[i])
D_bar_kg_matrix[i,] = as.numeric(D_bar_kg %>% filter(G == G_levels[g], K == K_levels[k])
    %>% select(all_of(vars)))
}
D_W = D - D_bar_kg_matrix
# Defining the names of each column (variable) in each variability data matrix
colnames(D_T) = c``(''Height, ```Weight, ```LAI, ```LeafTemp, ```Photosynthesis,
    ```WUE, ```WaterPotential)
colnames(D_G) = c``(''Height, ```Weight, ```LAI, ```LeafTemp, ```Photosynthesis,
    ```WUE, ```WaterPotential)
colnames(D_K) = c``(''Height, ```Weight, ```LAI, ```LeafTemp, ```Photosynthesis,
    ```WUE, ```WaterPotential)
colnames(D_KG) = c``(''Height, ```Weight, ```LAI, ```LeafTemp, ```Photosynthesis,
    ```WUE, ```WaterPotential)
colnames(D_W) = c``(''Height, ```Weight, ```LAI, ```LeafTemp, ```Photosynthesis,
    ```WUE, ```WaterPotential)
# Finding of the SSCP matrices
SSCP_G = t(D_G) %*% D_G
SSCP_K = t(D_K) %*% D_K
SSCP_KG = t(D_KG) %*% D_KG
SSCP_W = t(D_W) %*% D_W
SSCP_T = t(D_T) %*% D_T
# Application of PCA to each variability data matrix (extraction of the first two
    principal components and the corresponding plot)
PCA(D_T, scale.unit = TRUE, ncp = 2, graph = TRUE)
PCA(D_G, scale.unit = TRUE, ncp = 2, graph = TRUE)
PCA(D_K, scale.unit = TRUE, ncp = 2, graph = TRUE)
PCA(D_KG, scale.unit = TRUE, ncp = 2, graph = TRUE)
PCA(D_W, scale.unit = TRUE, ncp = 2, graph = TRUE)
```
```

Appendix E

The PCA results of the unbalanced four-dimensional data cube ("Basil-Sim 2"), with the "Type III" approach, are presented below.

Figure E1

PCA loading diagram from: (a) the D_t SSCP matrix of the total variability; (b) the D_g SSCP matrix of the variability between hyper-slices; (c) the D_k SSCP matrix of the variability between slices; (d) the D_{kg} SSCP matrix of the interaction between hyper-slices and slices variability; and (e) the D_w SSCP matrix of the variability within hyper-slices and slices. The first two principal components (Dim 1 and Dim 2) define the axes along with their respective proportion of explained variance. Directional vectors indicate the loading values for each variable, including Height, Weight, Leaf Area Index (LAI), Leaf Temperature (Leaf Temp), Photosynthesis, Water Use Efficiency (WUE), and Water Potential (generated in R)

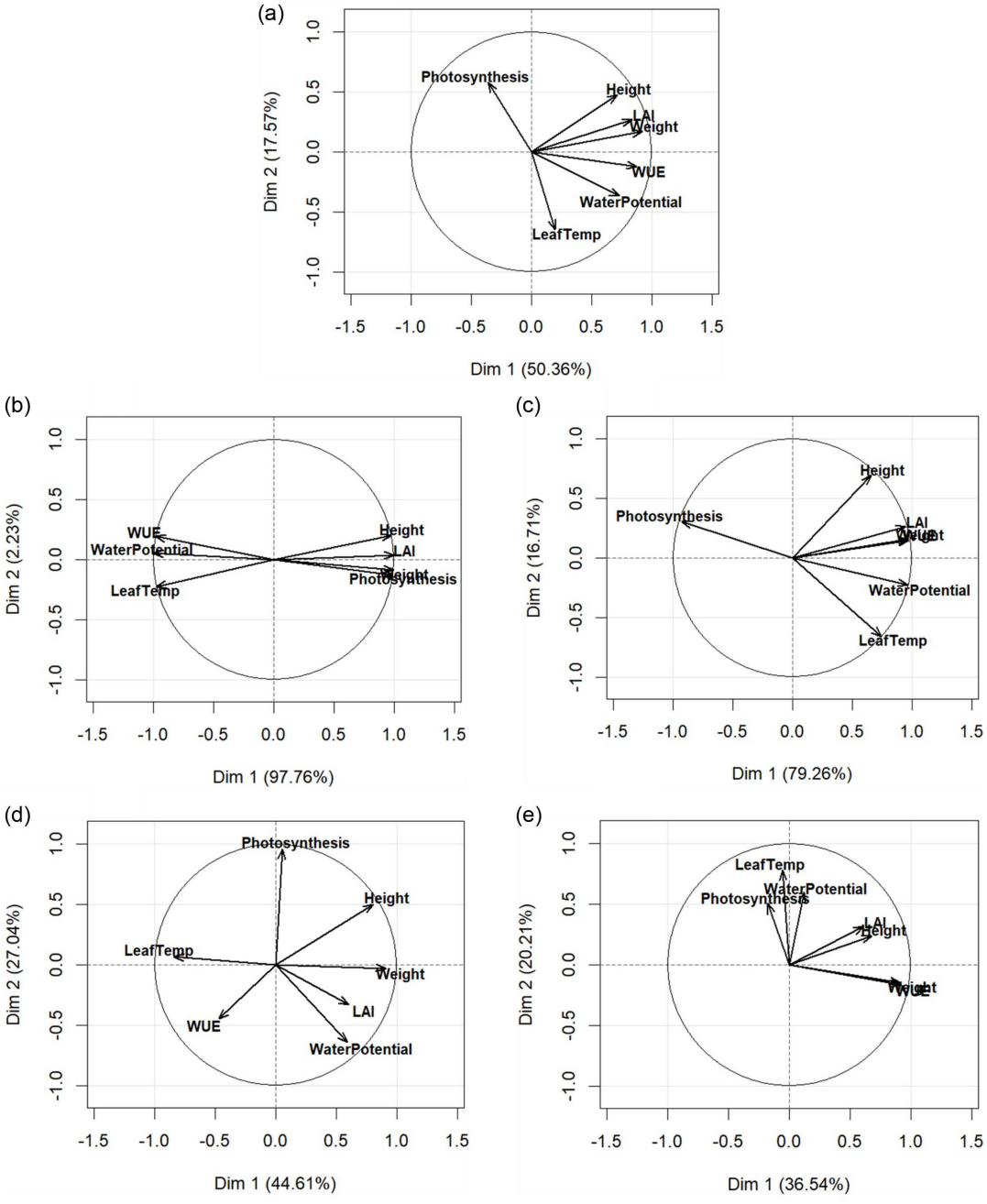


Table E1

PCA loadings (only loadings with an absolute value greater than or equal to 0.450 are presented) and communalities of the first two principal components (PC1 and PC2), for each variable (Height, Weight, Leaf Area Index [LAI], Leaf Temperature [Leaf Temp], Photosynthesis, Water Use Efficiency [WUE], and Water Potential), of the balanced data cube D ("Basil-Sim 2"). Irrigation levels correspond to the variable of hyper-slices, with three levels (40%, 70%, and 100%). Varieties correspond to the variable of slices, with five levels (Burns, Cinnamon, Sweet, Rend, and Thai)

		PCA loadings		
Variability	Variable	PCI	PC2	Communality
Total	Height	.713		.732
	Weight	.921		.875
	LAI	.836		.770
	Leaf Temp		–.651	.462
	Photosynthesis		.576	.462
	WUE	.876		.781
	Water Potential	.731		.672
Between hyper-slices	Height	.980		1.000
	Weight	.996		1.000
	LAI	.999		1.000
	Leaf Temp	–.974		1.000
	Photosynthesis	.992		1.000
	WUE	–.982		1.000
	Water Potential	-.999		1.000
Between slices	Height		.691	.901
	Weight	.969		.958
	LAI	.947		.967
	Leaf Temp	.744		.991
	Photosynthesis	–.924		.935
	WUE	.970		.967
	Water Potential	.966		.987
Interaction	Height	.805		.896
	Weight	.911		.831
	LAI	.604		.475
	Leaf Temp	–.841		.712
	Photosynthesis		.955	.915
	WUE	–.468		.422
	Water Potential		–.644	.765
Within hyper-slices and slices	Height	.680		.518
	Weight	.911		.852
	LAI	.610		.470
	Leaf Temp		.779	.610
	Photosynthesis		.502	.282
	WUE	.919		.874
	Water Potential		.593	.367

Table E2
Percentages of variance explained by the first two principal components (PC1 and PC2) for the unbalanced data cube D ("Basil-Sim 2")

Variability	Principal component	% of variance	% of total variance
Total	PC1	50.36	
	PC2	17.57	67.93
Between hyper-slices	PC1	97.76	
	PC2	2.23	99.99
Between slices	PC1	79.26	
	PC2	16.71	95.97
Interaction	PC1	44.61	
	PC2	27.04	71.65
Within hyper-slices and slices	PC1	36.54	
	PC2	20.21	56.75

Figure E2
Scree plot of PCA for the total variability of the unbalanced data cube D ("Basil-Sim 2") (generated in SPSS)

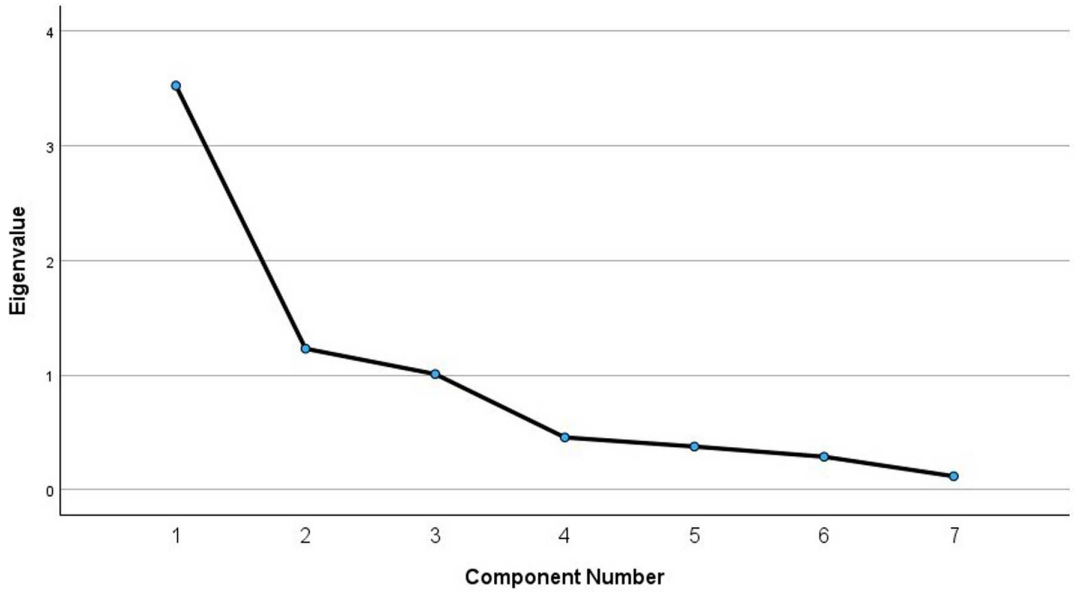


Figure E3
Scree plot of PCA for the between hyper-slices variability of the unbalanced data cube D ("Basil-Sim 2") (generated in SPSS)

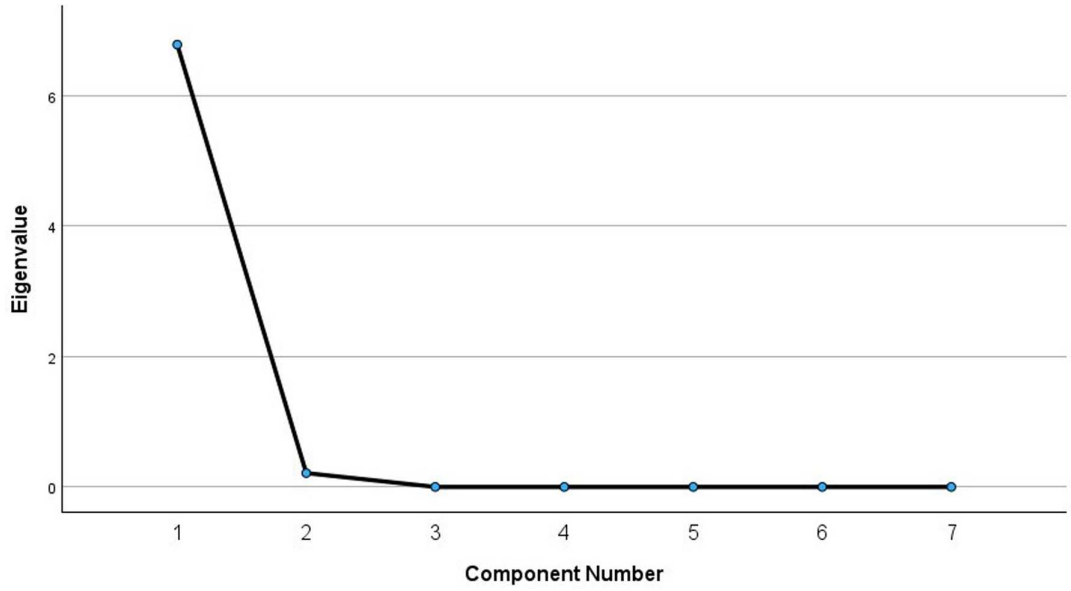


Figure E4
Scree plot of PCA for the between slices variability of the unbalanced data cube D ("Basil-Sim 2") (generated in SPSS)

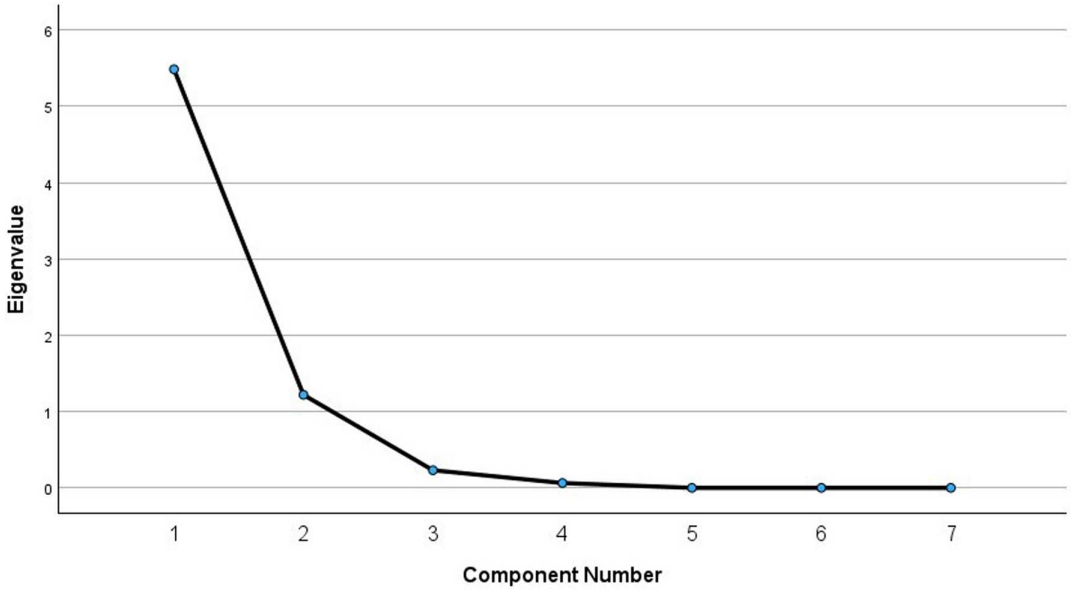


Figure E5
Scree plot of PCA for the interaction between hyper-slices and slices variability of the unbalanced data cube D ("Basil-Sim 2") (generated in SPSS)

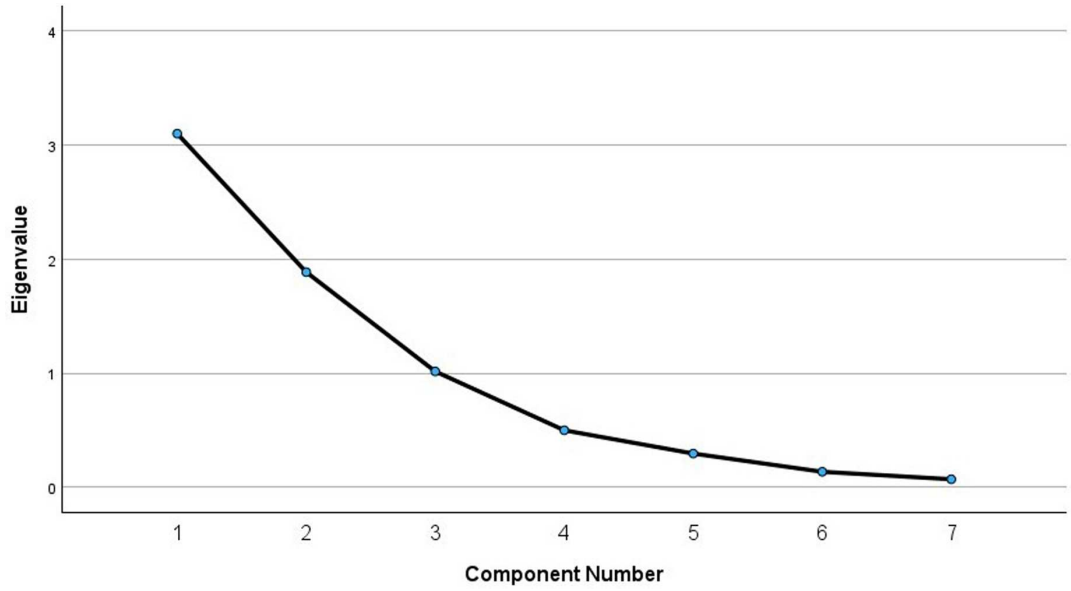


Figure E6
Scree plot of PCA for the within hyper-slices and slices variability of the unbalanced data cube D ("Basil-Sim 2") (generated in SPSS)

