# A kernel-based approach for fault diagnosis in batch processes

R. Vitale[a], O.E. de Noord[b], A. Ferrer[a]

[a]Departamento de Estadística e Investigación Operativa Aplicadas y Calidad, Universitat Politècnica de València, 46022, Valencia, Spain

[b]Shell Global Solutions International B.V., Shell Technology Centre Amsterdam, PO Box 38000, 1030 BN Amsterdam, The Netherlands

## Summary

This article explores the potential of kernel-based techniques for discriminating on- and off-specification batch runs, combining Kernel-PLSDA and three common approaches to analyze batch data by means of bilinear models: Landmark Features Extraction, BatchWise Unfolding and VariableWise Unfolding. Gower's idea of pseudo-sample projection is exploited to recover the contribution of the initial variables to the final model and visualize those having the highest discriminant power. The results show the proposed approach provides an effective fault discrimination and enables a correct identification of the discriminant variables in the considered case studies.

**Keywords**: kernel-based methods, pseudo-sample projection, batch processes, fault discrimination, fault diagnosis.

## 1. Introduction

The presence of complex non-linear relationships in data may represent a difficult issue to solve when one tries to model them by means of the most common tools in chemometrics, such as Principal Component Analysis (PCA), Principal Component

Regression (PCR) or Partial Least Squares Regression (PLSR). In fact, these methods are not able to describe the underlying structure of datasets that are affected by severe non-linearities, since they assume this structure is linear [1]. In recent years, many techniques have been proposed to handle such kind of situations: those based on non-linear PLS [2-3] and neural networks [4] have been the most exploited ones. Unfortunately, these approaches often encompass many adjustable parameters, are time and memory-consuming and may suffer from overfitting and local minima. In order to avoid these issues, the so-called kernel methods have been developed [5]. These techniques, which also comprehend Support Vector Machines (SVM) [6], have been broadly used for solving non-linear problems in chemistry [7-8], biology [9], informatics [10-11] and continuous process chemometrics [12-13]. Their basic principle is common: before modeling the data, a transformation of the original input space into a higher dimensional one, the feature space, is performed by using specific kernel functions. This permits to describe non-linear relationships in a linear form and to solve the problem under study by means of classical linear methods. Hence, performing, for instance, PCA, PLS or PLS Discriminant Analysis (PLSDA) after the data matrix transformation results in Kernel-PCA (K-PCA), Kernel-PLS (K-PLS) and Kernel-PLSDA (K-PLSDA), respectively. Unfortunately, the information about the weights or the contributions of the original variables is lost. Different possibilities [8, 14-16] to overcome this limitation exist, but authors often abstain from resorting to them, since they do not permit to graphically visualize the relation between variables and final models. Krooshof *et al*. [17] extended the idea of the non-linear biplots, described by Gower and Hardings [18], to recover and visualize this

specific information. In this case, the importance and influence of the variables is evaluated by constructing artificial samples, also known as pseudo-samples, whose projection onto the space of the model gives information about their contribution to it. This has been tested only on simulated datasets and in some metabolomic studies [19-20].

The first aim of this article is to explore the potential of K-PLSDA for fault detection in batch process analysis. Industrial batch processes generate massive amounts of data, which are recorded for online treatment or posterior analysis. In particular, during each batch run, $m = 1; 2; \ldots ; M$ variables are measured at $t = 1; 2; \ldots ; T$ time points. Data collected for $i = 1; 2; \ldots ; I$ batches are arranged in a three-way array ($I \times M \times T$). Even though techniques for directly modeling this structure exist, the most widely used approach to extract exploitable information from this kind of data is to rearrange this three-way array into a matrix and then fit a bilinear model by means of one of the aforementioned chemometric tools [21]. The three most common unfolding strategies to perform this rearrangement are VariableWise Unfolding (VWU), BatchWise Unfolding (BWU) and Landmark Feature Extraction (LFE). VWU unfolds the original three-way array to a new matrix ($IT \times M$) by preserving the variable direction. BWU unfolds the initial structure to a new array ($I \times TM$) by preserving the batch direction. LFE defines $F$ landmark features of the evolution of the $M$ variables in each batch and organizes them in a new matrix ($I \times F$). A good survey of these techniques can be found in [22].

This article will be focused on the analysis of historical batch operations for the troubleshooting of specific problems occurred during particular process runs. The identification of the variables, which evolve differently with respect to an *in-*

*control* situation, (the so-called fault diagnosis) is a key point in such cases. Unfortunately, classical tools such as the contributions plots are useless if one wants to appeal to kernel-based methods for batch process analysis, due to the transformation of the original data matrix. For this reason, a new method based on pseudo-sample projection is proposed here for recognizing those variables, which deviates from the Normal Operation Conditions.

## 2. Materials and methods

### *2.1 Datasets*

In this paper, three datasets are considered. The first is a simulated data array containing the evolution of 10 variables at 25 sampling times in 30 different batches: 15 are evolving under Normal Operation Conditions (NOC), while the remaining 15 are faulty due to an increment in the variance of some variables. The second one relates to a polymerization process described in [23] and consists of 23 batches (18 NOC and 5 off-specification) during which 10 variables are measured at 100 time points. In this case, both VWU and BWU were applied to the original three-way array. The third dataset was described in [24] and contains the values of 8 landmark features extracted from the variable trajectories of 71 batches (33 NOC, 10 on-specification but presenting an abnormally high quantity of residual solvent, and 28 off-specification) of a pharmaceutical spray drying process. In contrast with the original article, the second group of 10 batches was excluded from the analysis in order to enable a simpler discrimination between on-specification and off-specification runs, as for the previous datasets.

*2.2 Kernel transformation*

The framework of the different kernel-based techniques is based on the so-called kernel transformation, which is sketched in a general form in SI.1.

Its mathematical formulation is given by:

$$K(\mathbf{x}_i,\mathbf{x}_j) = \langle\phi(\mathbf{x}_i), \phi(\mathbf{x}_j)\rangle \quad (1)$$

where $\mathbf{x}_i$ and $\mathbf{x}_j$ are two row vectors belonging to the original dataset, to which a non-linear mapping function $\phi$ is applied, while < and > denote the inner product.

Therefore, the initial $N \times J$ data array, **X**, where $N$ is the number of observations and $J$ the number of measured variables, is transformed into a new square symmetric $N \times N$ kernel matrix, **K,** in which each position contains a value representing the dissimilarity or distance between two different observations. When dealing with kernel-based techniques, it is not necessary to know the mapping function *a priori*: there are many generic kernel functions one can use in order to obtain **K** and all of them exhibit two fundamental properties: i) they project the original data onto a high dimensional space, the feature space; ii) they provide a way to calculate the inner product between observations in this feature space.

The former permits to describe in a linear way possible non-linear relationships in the data. The latter makes all the algorithms of the classical multivariate linear techniques, which only use the inner product matrix (PCA, PLS and Fisher Discriminant Analysis, FDA, as demonstrated by Cao *et al*. [1]), suitable for being applied in the higher dimensional feature space. For the purposes of this article, only three kernel transformations, the linear, the 2$^{nd}$-order polynomial and the Gaussian (executed by Radial Basis Functions) will be taken into account. Their

mathematical formulations are listed in Table 1, together with the symbols of their possible adjustable parameters.

[INSERT HERE TABLE 1]

*2.3 Pseudo-sample projection*

A pseudo-sample corresponds to a particular observation that carries all the weight in one single variable. For example, the vector [0, 0, …, 1, 0, …, 0], represents one of the possible pseudo-samples associated to the variable $x_j$ of a specific dataset. By projecting an observation like this onto the latent structure of a classical 1-LV PLSDA model, the score for this new sample is calculated as follows:

$$t_{\text{new}} = [0, 0, \ldots, 1, 0, \ldots, 0]\mathbf{w}^* = w_j \quad (2)$$

This score is equal to the $j$-th value of the weighting vector, $\mathbf{w}^*$, and, thus, gives information about the contribution of variable $x_j$ to the model. Creating for each variable a pseudo-sample matrix, $\mathbf{P}_j$, which contains in the $j$-th column values ranging from the minimum to the maximum of that variable and 0 in all the other entries, and projecting it onto the latent space, trajectories of points are constructed according to the equation:

$$\mathbf{P}_j\mathbf{w}^* = \begin{bmatrix} 0, \ldots, 0, \min(\mathrm{x}_j), 0, \ldots, 0 \\ \cdots \\ \cdots \\ 0, \ldots, 0, \max(\mathrm{x}_j), 0, \ldots, 0 \end{bmatrix}\mathbf{w}^* = \begin{bmatrix} \min(\mathrm{x}_j)w_j \\ \cdots \\ \cdots \\ \max(\mathrm{x}_j)w_j \end{bmatrix} \quad (3)$$

It is straightforward to generalize this result to the case in which more than 1 LV is considered. Here, the matrix resulting from the previous operation defines the geometrical locus of all the points lying along the direction determined by the origin of the latent space and the point whose coordinates are defined by the

weights of the *j*-th variable on the *A* calculated LVs. In a classical PLSDA model, representing these points does not provide any additional information, but, as will be shown later, it is possible to get an idea from this kind of plot about how the original variable evolves in the latent space when kernel-based methods are applied. In fact, Postma *et al*. [19] demonstrated pseudo-sample projection permits to recover the information related to the contribution of the original variables when dealing with a Euclidean distance matrix, say **D**. In addition, **D** (double-centered) is directly generated applying a linear kernel transformation to a generic mean-centered dataset (see Appendix I for the details). Thus, it is possible to resort to this strategy even when one uses K-PLSDA. In this case, it is only needed to transform each pseudo-sample array into a pseudo-sample kernel one by the same transformation as for the matrix used for constructing the model. The result is a $P \times N$ array, which contains information about the dissimilarity between the $P$ pseudo-samples and the $N$ observations of the training set. The mathematical formulation of this extension is described in Appendix II. Moreover, this is valid not only in case one is exploiting a linear function to transform the analyzed data. The pseudo-sample projection can be used when dealing with all the kernel transformations, provided that they generate sets of distances which may be embedded in a Euclidean space [18].

The whole procedure used in this article for building kernel-based models and recovering the information about the influence of the original variables comprises the following steps: i) Autoscale the original data matrix, **X**; ii) Transform the autoscaled dataset into a kernel matrix, **K,** by a specific kernel function; iii) Double-center **K** so that:

$$\mathbf{K}_c = \mathbf{K} - \overline{\mathbf{K}}_j - \overline{\mathbf{K}}_n + \overline{\mathbf{K}}_{nj} \quad (4)$$

where $\overline{\mathbf{K}}_j$, $\overline{\mathbf{K}}_n$ and $\overline{\mathbf{K}}_{nj}$ contain the column means, the row means and the overall mean of the **K** matrix, respectively; iv) Fit a PLSDA model on $\mathbf{K}_c$; v) Create a pseudo-sample matrix, $\mathbf{P}_j$, for each one of the original *J* variables as described before; vi) Apply to each pseudo-sample matrix the same kernel transformation as for the training data in order to obtain a pseudo-sample kernel matrix, $\mathbf{P}_j^{\mathbf{K}}$; vii) Double-center every $\mathbf{P}_j^{\mathbf{K}}$ so that:

$$\mathbf{P}_{jc}^{\mathbf{K}} = \mathbf{P}_j{}^{\mathbf{K}} - \overline{\mathbf{K}}_j - \overline{\mathbf{P}}_{jp}{}^{\mathbf{K}} + \overline{\mathbf{K}}_{nj} \quad (5)$$

where the *p*-th row of $\overline{\mathbf{P}}_{jp}{}^{\mathbf{K}}$ contains the mean of the *p*-th row of $\mathbf{P}_j^{\mathbf{K}}$. Notice that $\overline{\mathbf{K}}_n$ is substituted by the term $\overline{\mathbf{P}}_{jp}{}^{\mathbf{K}}$ since the total number of rows of $\mathbf{P}_j^{\mathbf{K}}$ is usually different from the number of rows of **K**; viii) Project each *j*-th pseudo-sample kernel matrix onto the latent structure, as follows

$$\mathbf{T}_{j,ps} = \mathbf{P}_{jc}^{\mathbf{K}}\mathbf{W}^{*\mathbf{K}} \quad (6)$$

where $\mathbf{W}^{*\mathbf{K}}$ corresponds to the weighting matrix of the K-PLSDA model; ix) Plot the predicted scores, $\mathbf{T}_{j,ps}$, for recovering the information about the contribution of each original variable to the K-PLSDA model.

## 3. Results and discussion

### *3.1 Simulated example*

A 750x2 set of scores following trimmed circular trajectories was generated, creating 2 classes of 15 different trajectories of 25 observations each, as shown in Figure 1.

[INSERT HERE FIGURE 1]

Every trimmed circular profile represents a proper batch score trajectory, which defines its evolution in the latent variable space and might have been obtained applying PCA on a VWU data array. So, multiplying this set of scores by a 2x10 transposed matrix of loadings, calculated building a PCA model on real process data, a 750x10 dataset was constructed, which contains the evolution of 10 variables at 25 sampling times in 30 different runs. As shown in Figure 2, this results in two classes of batches characterized by differences in the variance of the measured variables and not in their mean values (e.g. due to sensor or controller faults).

[INSERT HERE FIGURE 2]

To verify whether pseudo-sample projection enables the correct identification of the discriminant variables, three columns of the resulting dataset were substituted by three white noise vectors.

Finally, the whole array was divided into a training and a test set, containing 500 (20 complete batches) and 250 (10 complete batches) observations, respectively. Batch selection was randomly performed class-wise.

Four different cross-validated classification models, with a growing degree of non-linearity, were built on the simulated data. The performance of the final models is summarized in Table 2.

[INSERT HERE TABLE 2]

Clearly, the two classes cannot be satisfactorily separated by classical PLSDA and K-PLSDA with a linear kernel transformation. However, resorting to non-linear kernel functions permits to correctly discriminate most of the observations belonging to the two different categories for both training and test set. The best

correct classification rate is obtained by the 2nd-order polynomial kernel model, whose scores and predicted class values plots are shown in Figure 3.

[INSERT HERE FIGURE 3]

Here, it is important to take into account that each one of the represented symbols corresponds to a specific time point of a particular batch. K-PLSDA is then able to correctly discriminate most of the time samples in which the process is progressing under Normal Operation Conditions or not.

The highest discrimination ability of this model is reasonable, considering that the differences between the two classes under study are associated to the variance of the measured variables, which results, indeed, in a quadratic transformation of the original data.

In order to check whether pseudo-sample projection permits to recover the information about the discriminant power of the original variables, for each column of the simulated data matrix, a 20×10 pseudo-sample array was built, transformed and projected onto the model space as described in Section 2.3. Figure 4 shows the obtained outcomes.

[INSERT HERE FIGURE 4]

The different trajectories represent the predicted scores calculated from the pseudo-sample kernel matrices constructed for all the original variables (numbered from 1 to 10). The blue dotted line corresponds to the discriminant direction between the centers of gravity of the two classes of observations, obtained from Figure 3a. The font-size of the numerical characters constituting each trajectory increases in correspondence of regions of the latent space where the respective variables assume higher values and *viceversa*. So, comparing this

237 graph with the scores plot in Figure 3a, it is rather clear that the second class (red
238 squares) contains batch runs associated to either higher or lower values of the
239 labeled variables than those belonging to the first group (blue dots).
240 In order to define an objective criterion for evaluating the discriminant power of
241 the original variables, the cosine of the angle formed by the blue dotted line and
242 each trajectory was calculated. These values are listed in Table 3 and clearly
243 indicate that all the variables except $x_8$, $x_9$ and $x_{10}$ have good discriminant power
244 (i.e. angle cosines close to 1). Notice that for these latter there are no clear
245 trajectories (see Figure 4) and then the cosine of the angles cannot be precisely
246 calculated. This is coherent with the simulated data shown in Figure 2 where the
247 variables with differences in variance between on– and off-specification batches
248 are $x_1$ to $x_7$.
249 [INSERT HERE TABLE 3]
250 *3.2 VWU/K-PLSDA (polymerization process)*
251 The polymerization dataset under consideration contains observations related to
252 on- and off-specification batches, but the time period in which their evolution
253 differs is unknown. In order to identify it, a preliminary exploratory K-PCA model
254 was built, using a linear kernel transformation, on all the NOC process runs.
255 Hotelling's $T^2$ and SPE (Squared Prediction Error) statistics were calculated for the
256 remaining faulty ones after their projection onto the latent variable space. The
257 resulting $T^2$ and SPE control charts are shown in SI.2.
258 It is straightforward to identify that the initial time interval of the process (the first
259 15 time points) is where the off-specification batches have different evolution than
260 the on-specification ones. In this case, using classical PCA instead of K-PCA would

261 have returned very similar results (not shown). Therefore, in order to discriminate
262 the two classes, only this period was considered in the following step of data
263 analysis. So, the initial VWU matrix was reduced to a 345×10 one, which was then
264 divided into a training and a test set, containing 225 and 120 observations,
265 respectively. Their selection was performed leaving outside the training set all the
266 time samples associated to 6 on-specification and 2 off-specification batches,
267 randomly chosen. A linear kernel transformation was applied to the calibration
268 data and a cross-validated 2-LV PLSDA model was built on the resulting 225×225
269 kernel matrix. Its performance was evaluated in terms of $R^2$ and $Q^2$, showing
270 values of 94.7% and 94.3%, respectively. In order to assess its prediction ability
271 the observations of the test set were transformed in the same way as those of the
272 training set (generating a kernel test matrix with dimension 120×225), projected
273 onto its latent structure and, according to their predicted *y* values, assigned to one
274 of the two considered classes. Results are displayed in Figure 5.

275 [INSERT HERE FIGURE 5]

276 The K-PLSDA scores plot on the two latent variables (Figure 5a) shows a perfect
277 separation between the observations belonging to the different categories. Since
278 the model was built after the transformation of a VWU data matrix, as for the
279 previous case, each represented symbol corresponds to a specific time point of a
280 particular batch. As will be shown, this is a fundamental difference with respect to
281 the other described approaches based on BWU and LFE.

282 The good discrimination is corroborated by the plot of the predicted class values
283 (Figure 5b). 100% correct classification rate is obtained both in training and test
284 sets for the two categories. As aforementioned, plotting directly the loadings or the

weights of models like this is totally uninformative since the kernel matrix only contains values of dissimilarity between observations. This is the reason why pseudo-sample projection is needed to recover the information about the contribution of the original variables to the discrimination between the two classes. For each column of the VWU data matrix, a 20×10 pseudo-sample array was built, transformed and projected onto the model space. Figure 6 shows the obtained outcomes.

[INSERT HERE FIGURE 6]

The values of the cosine of the angle formed by the blue dotted line and each trajectory are summarized in Table 4.

[INSERT HERE TABLE 4]

Variables $x_4$, $x_7$, $x_9$ and $x_{10}$ are proved to be the most significant ones with values of this cosine clearly higher than the other ones.

Also in this case, the font-size of the numerical characters of each trajectory increases in correspondence to regions of the latent space where the respective variables assume higher values and *viceversa*. Therefore, comparing this graph with the scores plot in Figure 5a, it is possible to infer that off-specification batches are characterized by higher values of variables $x_7$, $x_9$ and $x_{10}$ and lower values of variable $x_4$ in comparison to the on-specification ones, as confirmed by representing their original temporal evolution, shown in SI.3.

Similar results are obtained from a PLSDA model without a kernel transformation, as highlighted in SI.4. The plot is associated to a specific time point of the interval during which the off-specification batches evolve differently from the others, but the displayed profile is consistent with all the other analyzed time samples.

*3.3 BWU/K-PLSDA (polymerization process)*

The same procedure described in Section 2.3 was then applied to the BWU data matrix (23×1000) from the polymerization process. A linear kernel was chosen for the transformation of the original array. Since only few observations (i.e. batches) were available for the calibration of the model, it was not possible to evaluate its predictive ability via an external test set. As will be discussed later, permutation tests were used for overcoming this limitation. PLSDA was applied on the resulting 23×23 kernel matrix. Results are shown in Figure 7.

[INSERT HERE FIGURE 7]

The final Leave-One-Out CV 2-LV model shows $R^2$ and $Q^2$ values of 97.5% and 95.9%, respectively. The separation of the two classes is perfect, leading to 100% correct classification rate both in calibration and cross-validation. Unlike the VWU case, here each represented symbol corresponds to a whole batch: therefore, the discrimination highlights the difference between on- and off-specification batches. Due to the structure of the original dataset, 1000 pseudo-sample trajectories showing the importance of a particular variable measured at a specific time spot were constructed, each one constituted by 20 points. Representing all these trajectories would have made the plot uninterpretable. For this reason, only those related to the time period, during which the difference in the evolution of the batches was detected, according to the initial K-PCA analysis discussed in Section 3.2, were included in SI.5. The graph is divided in 10 sections as the number of original variables. Inside every section, the pseudo-sample trajectories for the respective variable at the different considered time points are represented. The blue dotted line corresponds to the class discriminant direction. As in the VWU

case, the cosine of the angle formed by each trajectory and this direction was selected as criterion of variable importance. In SI.5, only the pseudo-samples trajectories characterized by a value of the amplitude of this angle lower than 30° are black-coloured. Variables $x_4$, $x_9$ and $x_{10}$ are found to have high contributions to the model approximately for the whole interval under study, while variable $x_7$ is significant only in part of this period. This is also shown by plotting the values of the cosine of the angles formed by the series of respective trajectories and the discriminant direction with respect to the batch time, as illustrated in Figure 8.

[INSERT HERE FIGURE 8]

Variables $x_2$, $x_5$, $x_6$ also proved to have high significance in small periods. In such cases, a further investigation of the original variable trajectories is always needed to properly identify the root causes generating problems during the process.

As for the previous examples, the pseudo-sample plot (SI.5) is built using larger bullet-size in correspondence of the zones of the latent space in which the respective variable assumes higher values and *viceversa*. Hence, it is straightforward to conclude that on-specification batches are characterized by lower values of variables $x_7$, $x_9$ and $x_{10}$ and higher values of variable $x_4$ in comparison to the off-specification runs, which exactly corresponds to the outcomes discussed before.

In order to validate the final model, a permutation test [26] was performed. SI.6 shows the validation plots obtained for the BWU kernel matrix. The difference between the two categories under study is proved to be statistically significant ($p$-value $< \frac{1}{2000} = 0.005$ for both $R^2$ and $Q^2$).

*3.4 LFE/K-PLSDA (pharmaceutical spray drying process)*

A K-PLSDA model was built on the LFE data matrix (61×8) from the pharmaceutical spray drying process. Among the initial observations, 12 (7 on-specification and 5 off-specification) were found having abnormally high residuals and therefore were excluded from the final classification in order not to jeopardize its quality. A further K-PLSDA model was then constructed on the reduced LFE dataset (49×8). Since a linear kernel transformation did not provide good results, a Radial Basis Function was applied to the original array. The $\sigma$ parameter was optimized by leave-one-out cross-validation and fixed at a value of 0.8. Smaller values would have generated over-fitting and hardly interpretable pseudo-sample trajectories. A cross-validated 2-LV PLSDA model was built on the resulting 49×49 kernel matrix. Its performance was assessed according to the values of $R^2$ (73.8%) and $Q^2$ (43.6%). Figure 9 displays the K-PLSDA scores plot and the predicted *y* values for all the observations in calibration and cross-validation.

[INSERT HERE FIGURE 9]

Also in this case, each represented symbol corresponds to a whole process run. Here, the model does not guarantee high performance as those described in the previous examples. This is due to the fact that the selected landmark features have quite low correlation to the quality of the batches [24]. Nevertheless, resorting to a K-PLSDA model enabled a satisfying discrimination even dealing with a dataset like this (73.08% and 91.30% correct classification rate in cross-validation for the two categories, respectively). In order to recover the information about the original variables, a 20×8 pseudo-sample matrix was constructed per each column of the initial LFE array. The resulting pseudo-sample trajectories are represented in Figure 10.

[INSERT HERE FIGURE 10]

The graph is built in the same way as SI.5. As stated by Gower [18], non-linear kernel transformations lead to non-linear pseudo-sample trajectories. Since it is impossible to univocally define an angle between the separation direction and a curved line, the interpretation of the variable importance is not straightforward. On the other hand, by inspecting the plot, it is rather clear that the only variables, whose pseudo-sample evolution is correlated to the blue dotted line, are $x_1$, $x_2$ and $x_8$. All the other trajectories cover circular paths (variables $x_4$, $x_5$ and $x_6$) or show a nearly linear trend with a direction almost orthogonal to the discriminant one (variable $x_3$ and $x_7$). As in the previous cases, larger font-sizes indicate regions of the latent space in which the labeled variables assume higher values. So, it is easy to infer off-specification batches are characterized by lower values of variable $x_8$ and by higher values of variables $x_1$ and $x_2$. The obtained outcomes are coherent with the conclusions reached in the original article by García-Muñoz *et al.* [24], where it is detailed "a high-quality product is also associated with low solvent level in the collector tank (variable $x_1$)", "batches that progress faster (with higher values of $x_8$) tend to be those with high product quality" and "a low temperature in the dryer at the end of stage 1 (variable $x_2$) might also seem desirable".

For assessing the model performance, a permutation test was executed, due to the small number of observations constituting the dataset. The results are shown in SI.7.

The model is found to be statistically significant with respect to the other permuted classifications ($R^2$ $p$-value = 0.003, $Q^2$ $p$-value < 0.005). However, even if the $Q^2$ of the final model is always larger than those calculated modifying the class

label of the single observations, its $R^2$ is lower than some obtained after the class randomization. This aspect might be a caution indicator of the presence of variables whose contribution is unrelated to the class of the objects [27]. This issue is quite common when dealing with LFE [22]. In general, selecting a set of landmark features, which summarize the evolution and the differences between on-specification and off-specification batches in a proper way, may not be obvious: this may often lead to less reliable results when dealing with such kind of datasets than directly operating on the evolution of the measured variables during time.

## 4. Comparison between K-PLSDA and classical PLSDA models

The analysis of the simulated dataset highlighted the main advantage of using non-linear kernel-based classification methods over classical PLSDA. In fact, when complex data structures have to be modeled, such bilinear technique leads to low and unsatisfactory correct classification rates, which jeopardizes the fault detection. In such cases, exploiting non-linear classifiers radically improves the quality of the discrimination and the identification of the process runs, which did not progress under Normal Operation Conditions. This is also confirmed by the results obtained in the second case study. In fact, for the first real dataset, for both the VWU and BWU matrices, resorting to K-PLSDA for discriminating NOC batches from faulty ones did not result in significantly better performance than building a classical PLSDA model (results not shown). This similarity is a consequence of the fact that a linear kernel transformation permitted to obtain satisfying correct classification rates for the two considered classes, which means the original data were not affected by strongly non-linear relationships [1] and, therefore, they

might have been analyzed by means of conventional bilinear approaches, obtaining very similar outcomes.

On the other hand, when the LFE matrix was dealt with for the second real dataset, the best discrimination between the two categories under study was obtained by a kernel transformation performed using a radial-basis function (RBF). Here, if one compares the RBF K-PLSDA scores and *y*-predicted plots, displayed in Figure 9, with the ones constructed when a classical PLSDA model is built on the original matrix, shown in Figure 11, it is possible to verify the clear improvement in the separation between the observations belonging to the different classes, achieved when the kernel-based method is applied.

[INSERT HERE FIGURE 11]

## 5. Conclusions

In this article, a novel approach for fault discrimination and diagnosis in batch processes was proposed. It combines the ability of kernel-based classification techniques (in particular K-PLSDA) of dealing with complex non-linear data structures with the power of pseudo-sample projection (originally conceived by John Gower) for recovering the information related to the contribution of the initial variables to the final model, which permits to overcome one of the main drawbacks of these methods.

K-PLSDA shows similar performance to classical PLSDA, when linear transformations are appropriate for the datasets under study, but leads to better discrimination between the classes in case non-linear functions are needed for modelling more complex data structures, as clearly highlighted by the analysis of

453 the simulated and LFE datasets. In both scenarios, the pseudo-sample projection
454 enables a correct identification of the discriminant variables. For all these reasons,
455 the authors' suggestion for practical users is to resort to non-linear K-PLSDA when
456 standard bilinear techniques provide unsatisfactory outcomes.
457 Moreover, it was seen that the described strategy may constitute a powerful
458 method for detecting differences in the variance of the variable trajectories
459 measured during batch runs and then could represent an important crossroad in
460 this specific field of statistical process monitoring and control.
461 These satisfying results can be certainly considered a good starting point for
462 implementing this strategy as a complementary tool for Batch Multivariate
463 Statistical Process Control (BMSPC) methods.
464

465 **Appendix I**

466 *Relationship between the Euclidean Distance Matrix,* **D***, and the inner product*
467 *matrix,* $\mathbf{X}\mathbf{X}^{\mathrm{T}}$

468 The Euclidean distance between two observations contained in a generic data
469 matrix $\mathbf{X}_{(N\times M)}$, $\mathbf{x}_i$ and $\mathbf{x}_j$, is:

470 $$d_{i,j} = \left\|\mathbf{x}_i - \mathbf{x}_j\right\|^2 = (\mathbf{x}_i - \mathbf{x}_j)^{\mathrm{T}}(\mathbf{x}_i - \mathbf{x}_j) = \mathbf{x}_i{}^{\mathrm{T}}\mathbf{x}_i + \mathbf{x}_j{}^{\mathrm{T}}\mathbf{x}_j - 2\mathbf{x}_i{}^{\mathrm{T}}\mathbf{x}_j \quad (7)$$

471 Let **F** be the inner product matrix so that:

472 $$\mathbf{F} = \mathbf{X}\mathbf{X}^{\mathrm{T}} \quad (8)$$

473 The Euclidean distance matrix is then defined as:

474 $$\mathbf{D} = \mathbf{f}\mathbf{1}^{\mathrm{T}} + \mathbf{1}\mathbf{f}^{\mathrm{T}} - 2\mathbf{F} \quad (9)$$

475 where $\mathbf{f} =$ diag($\mathbf{F}$) and $\mathbf{1} = (1, 1, \dots, 1)^{\mathrm{T}}$. Centering **X** so that:

476 $$\widetilde{\mathbf{X}} = \mathbf{X} - \frac{1}{n}\mathbf{1}\mathbf{1}^{\mathrm{T}}\mathbf{X} \quad (10)$$

477 it is obtained:

478 $$\widetilde{\mathbf{F}} = \widetilde{\mathbf{X}}\widetilde{\mathbf{X}}^{\mathrm{T}} =$$

479 $$(\mathbf{X} - \frac{1}{n}\mathbf{1}\mathbf{1}^{\mathrm{T}}\mathbf{X})(\mathbf{X} - \frac{1}{n}\mathbf{1}\mathbf{1}^{\mathrm{T}}\mathbf{X})^{\mathrm{T}} = \mathbf{F} - \frac{1}{n}\mathbf{F}\mathbf{1}\mathbf{1}^{\mathrm{T}} - \frac{1}{n}\mathbf{1}\mathbf{1}^{\mathrm{T}}\mathbf{F} + \frac{1}{n^2}\mathbf{1}\mathbf{1}^{\mathrm{T}}\mathbf{F}\mathbf{1}\mathbf{1}^{\mathrm{T}} \quad (11)$$

480 Consider the double-centered Euclidean distance matrix:

481 $$\mathbf{B} = -\frac{1}{2}\mathbf{H}\mathbf{D}\mathbf{H}^{\mathrm{T}} \quad (12)$$

482 where $\mathbf{H} = \mathbf{I} - \frac{1}{n}\mathbf{1}\mathbf{1}^{\mathrm{T}}$. So:

483 $$\mathbf{B} = -\frac{1}{2}\mathbf{H}(\mathbf{f}\mathbf{1}^{\mathrm{T}} + \mathbf{1}\mathbf{f}^{\mathrm{T}} - 2\mathbf{F})\mathbf{H}^{\mathrm{T}} \quad (13)$$

484 Since:

485 $$\mathbf{f}\mathbf{1}^{\mathrm{T}}\mathbf{H}^{\mathrm{T}} = \mathbf{f}\mathbf{1}^{\mathrm{T}}(\mathbf{I} - \frac{1}{n}\mathbf{1}\mathbf{1}^{\mathrm{T}})^{\mathrm{T}} = \mathbf{f}\mathbf{1}^{\mathrm{T}} - \mathbf{f}(\frac{\mathbf{1}^{\mathrm{T}}\mathbf{1}}{n})\mathbf{1}^{\mathrm{T}} = 0 \quad (14)$$

486 it is verified:

487 $$\mathbf{H}\,\mathbf{f}\mathbf{1}^{\mathrm{T}}\mathbf{H}^{\mathrm{T}} = 0 = \mathbf{H}\mathbf{1}\mathbf{f}^{\mathrm{T}}\mathbf{H}^{\mathrm{T}} \quad (15)$$

488 Therefore:

489 $$\mathbf{B} = \mathbf{H}\mathbf{F}\mathbf{H}^{\mathrm{T}} = (\mathbf{I} - \frac{1}{n}\mathbf{1}\mathbf{1}^{\mathrm{T}})\mathbf{F}(\mathbf{I} - \frac{1}{n}\mathbf{1}\mathbf{1}^{\mathrm{T}})^{\mathrm{T}} = \mathbf{F} - \frac{1}{n}\mathbf{F}\mathbf{1}\mathbf{1}^{\mathrm{T}} - \frac{1}{n}\mathbf{1}\mathbf{1}^{\mathrm{T}}\mathbf{F} +$$

490 $$\frac{1}{n^2}\mathbf{1}(\mathbf{1}^{\mathrm{T}}\mathbf{F}\mathbf{1})\mathbf{1}^{\mathrm{T}} = \widetilde{\mathbf{F}} \quad (16)$$

491 that is:

492 $$\mathbf{B} = -\frac{1}{2}\mathbf{H}\mathbf{D}\mathbf{H}^{\mathrm{T}} = \widetilde{\mathbf{X}}\widetilde{\mathbf{X}}^{\mathrm{T}} \quad (17)$$

493 The Euclidean distance matrix **D** after double-centering is equal to the inner
494 product matrix $\widetilde{\mathbf{X}}\widetilde{\mathbf{X}}^{\mathrm{T}}$.

495

496 **Appendix II**

497 *Extension of the pseudo-samples projection to the feature space*

Suppose one has built a 1-LV PLSDA model on a double-centered Euclidean distance matrix, $\mathbf{B}_{(N\times N)}$, obtained based on the distances between the observations in $\mathbf{X}_{(N\times M)}$. The scores of the objects belonging to the training set are calculated as:

$$\mathbf{t}_{(N\times 1)} = \mathbf{B}_{(N\times N)}\mathbf{w}^{*\mathbf{B}}{}_{(N\times 1)} \quad (18)$$

where $\mathbf{w}^{*\mathbf{B}}$ represents the weighting vector obtained from $\mathbf{B}$, which does not show the contribution of the $M$ original variables. Substituting (17) in (18):

$$\mathbf{t}_{(n\times 1)} = \widetilde{\mathbf{X}}_{(N\times M)}\,\widetilde{\mathbf{X}}^{\mathrm{T}}{}_{(M\times N)}\mathbf{w}^{*\mathbf{B}}{}_{(N\times 1)} \quad (19)$$

Rewriting formula (18):

$$\mathbf{t}_{(N\times 1)} = \widetilde{\mathbf{X}}_{(N\times M)}\left(\widetilde{\mathbf{X}}^{\mathrm{T}}{}_{(M\times N)}\mathbf{w}^{*\mathbf{B}}{}_{(N\times 1)}\right) = \widetilde{\mathbf{X}}_{(N\times M)}\mathbf{w}^{*\prime}{}_{(M\times 1)} \quad (20)$$

where $\mathbf{w}^{*\prime}$ actually contains information about the influence of the $M$ original variables on the model.

Projecting the pseudo-sample $[0, \ldots, 0, 1, 0, \ldots, 0]_{(1\times M)}$ onto the PLSDA latent structure and calculating the respective predicted score results in:

$$t_{new} = [0, \ldots, 0, 1, 0, \ldots, 0]_{(1\times M)}\,\widetilde{\mathbf{X}}^{\mathrm{T}}{}_{(M\times N)}\mathbf{w}^{*\mathbf{B}}{}_{(N\times 1)} = [0, \ldots, 0, 1, 0, \ldots, 0]_{(1\times M)}\,\mathbf{w}^{*\prime}{}_{(M\times 1)}$$

$$= w_j^{*\prime} \quad (21)$$

$t_{new}$ is exactly equal to the $j$-th value of the vector $\mathbf{w}^{*\prime}$. When using a series of pseudo-samples instead of only one, trajectories of points are constructed, whose evolution gives an idea about how the original variables contribute to the final model. Since the linear kernel matrix exactly corresponds to the inner product one, it is straightforward to infer this outcome is valid when dealing with the respective data transformation. However, as stated by Gower [18], the same property is verified when dealing with all those generating sets of distances, which may be embedded in a Euclidean space. The Euclidean nature of the Gaussian kernel is particularly clear since it is calculated as a function of the Euclidean distance [28].

## References


[1] Cao D, Liang Y, Xu Q, Hu Q, Zhang L, Fu G. Exploring nonlinear relationships in chemical data using kernel-based methods. *Chemometr. Intell. Lab.* 2011; **107**:106-115

[2] Walczak B, Massart D. The Radial Basis Functions – Partial Least Squares approach as a flexible non-linear regression technique. *Anal. Chim. Acta* 1996; **331**:177-185

[3] Walczak B, Massart D. Application of Radial Basis Functions – Partial Least Squares to non-linear pattern recognition problems: diagnosis of process faults. *Anal. Chim. Acta* 1996; **331**:187-193

[4] Gasteiger J, Zupan J. Neural Networks in Chemistry. *Angew. Chem. Int. Ed. Engl.* 1993; **32**:503-527

[5] Schölkopf B, Smola A. *Learning with Kernels* (1st edn). MIT Press: Cambridge, UK, 2002

[6] Li H, Liang Y, Xu Q. Support vector machines and its applications in chemistry. *Chemometr. Intell. Lab.* 2009; **95**:188-198

[7] Williams P. Influence of water on prediction of composition and quality factors: the Aquaphotomics of low moisture agricultural materials. *J. Near Infrared Spectrosc.* 2009; **17**: 315-328

[8] Tan C, Li M. Mutual information-induced interval selection combined with kernel partial least squares for near-infrared spectral calibration. *Spectrochim. Acta Pt. A-Mol. Biomol. Spectrosc.* 2008; **71**:1266-1273

[9] Embrechts M, Ekins S. Classification of Metabolites with Kernel-Partial Least Squares (K-PLS). *Drug Metab. Dispos.* 2007; **35**:325-327

546 [10] Arenas-Garcia J, Camps-Valls G. Efficient Kernel Orthonormalized PLS for
547 Remote Sensing Applications. *IEEE Trans. Geosci. Remote Sens.* 2008;
548 **46**:2872-2881
549 [11] Struc V, Pavesic N. Gabor-based kernel partial-least-squares
550 discrimination features for face recognition. *Informatica* 2009; **20**:115-
551 138
552 [12] Sun R, Tsung F. A kernel-distance-based multivariate control chart using
553 support vector methods. *Int. J. Prod. Res.* 2003; **41**:2975-2989
554 [13] Lee J, Yoo C, Choi S, Vanrolleghem P, Lee I. Nonlinear process monitoring
555 using kernel principal component analysis. *Chem. Eng. Sci.* 2004; **59**:223-
556 234
557 [14] Bennett K, Embrechts M. *Advances in Learning Theory: Methods, Models*
558 *and Applications* (1st edn). IOS Press: Amsterdam, The Netherlands,
559 2003, 227-249
560 [15] Kewley R, Embrechts M, Breneman C. Data strip mining for the virtual
561 design of pharmaceuticals with neural networks. *IEEE Trans. Neural*
562 *Netw.* 2000; **11**:668-679
563 [16] Ūstūn B, Melssen W, Buydens L. Visualisation and interpretation of
564 Support Vector Regression models. *Anal. Chim. Acta* 2007; **595**:299-309
565 [17] Krooshof P, Ūstūn B, Postma G, Buydens L. Visualization and Recovery of
566 the (Bio)chemical Interesting Variables in Data Analysis with Support
567 Vector Machine Classification. *Anal. Chem.* 2010; **82**:7000-7007
568 [18] Gower J, Harding S. Nonlinear biplots. *Biometrika* 1988; **75**:445-455

569 [19] Postma G, Krooshof P, Buydens L. Opening the kernel of kernel partial
570 least squares and support vector machines. *Anal. Chim. Acta* 2011;
571 **705**:123-134
572 [20] Smolinska A, Blanchet L, Coulier L, Ampt K, Luider T, Hintzen
573 R, Wijmenga S, Buydens L. Interpretation and Visualization of Non-Linear
574 Data Fusion in Kernel Space: Study on Metabolomic Characterization of
575 Progression of Multiple Sclerosis. *PLoS ONE* 2012; **7**:e38163
576 [21] Camacho J, Picó J, Ferrer A. Bilinear modelling of batch processes. Part I:
577 theoretical discussion. *J. Chemometr.* 2008; **22**:299-308
578 [22] Wold S, Kettaneh-Wold N, MacGregor J, Dunn K. *Comprehensive*
579 *Chemometrics* (1st edn), vol.2. Elsevier B.V.: Oxford, UK, 2009, 163-197
580 [23] Nomikos, P, MacGregor J. Multivariate SPC Charts for Monitoring Batch
581 Processes. *Technometrics* 1995; **37**:41-59
582 [24] García-Muñoz S, Kourti T, MacGregor J, Mateos A, Murphy G.
583 Troubleshooting of an Industrial Batch Process Using Multivariate
584 Methods. *Ind. Eng. Chem. Res.* 2003; **42**:3592-3601
585 [25] Pérez N, Ferré J, Boqué R. Calculation of the reliability of classification in
586 discriminant partial least-squares binary classification. *Chemometr. Intell.*
587 *Lab.* 2009; **95**:122-128
588 [26] Lindgren F, Hansen B, Karcher W, Sjöström M, Eriksson L. Model
589 validation by permutation tests: Applications to variable selection. *J.*
590 *Chemometr.* 1996; **10**:521-532
591 [27] Quintás G, Portillo N, García-Cañaveras J, Vicente Castell J, Ferrer A,
592 Lahoz A. Chemometric approaches to improve PLSDA model outcome for

predicting human non-alcoholic fatty liver disease using UPLC-MS as a metabolic profiling tool. *Metabolomics* 2012; **8**:86-98

[28] Courrieu P. Straight monotonic embedding of data sets in Euclidean space. *Neural Networks* 2002; **15**:1185-1196

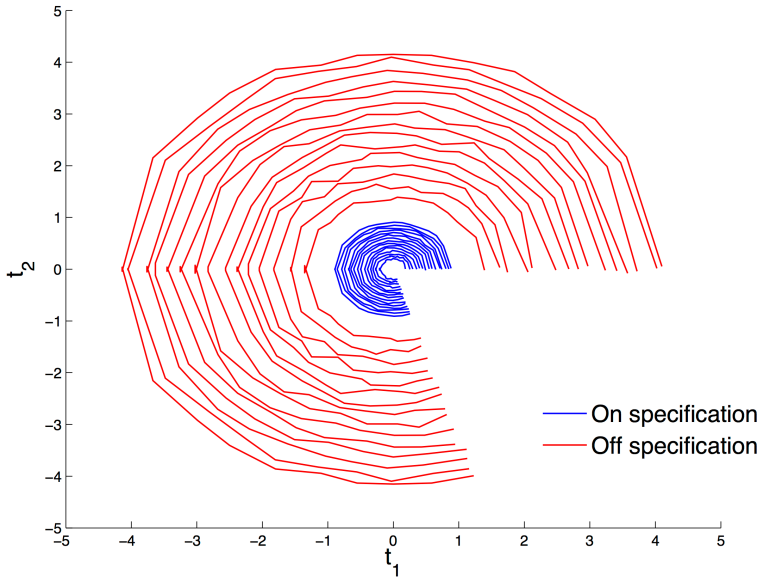


**Figure 1:** On-specification and off-specification simulated batch score trajectories

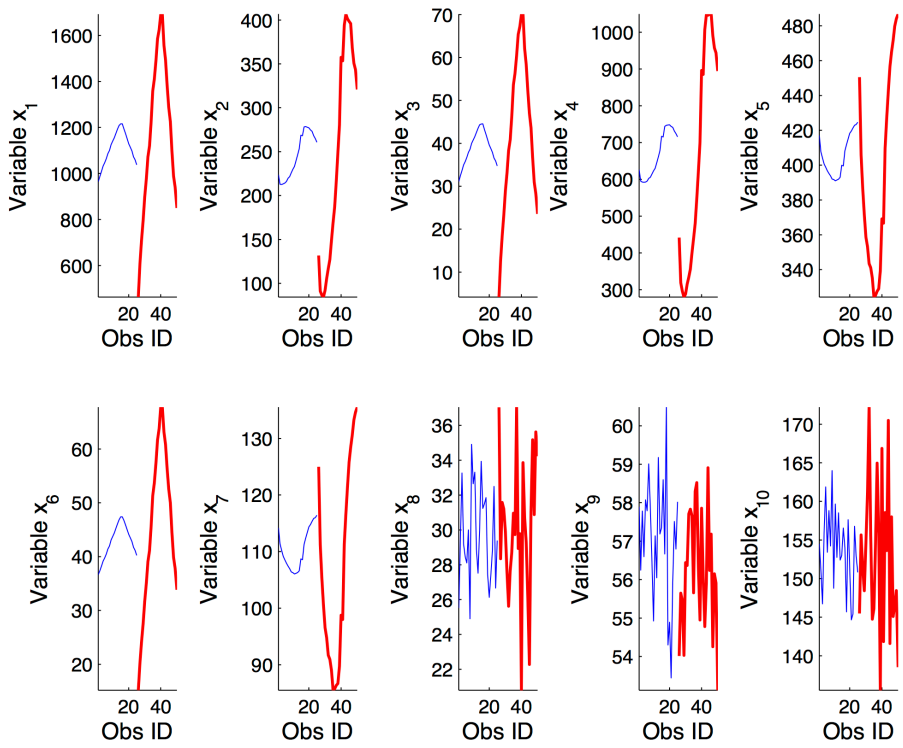


**Figure 2:** Temporal evolution of the variables of the simulated dataset for an on-specification (blue thin line) and an off-specification (red thick line) batch of the training set.

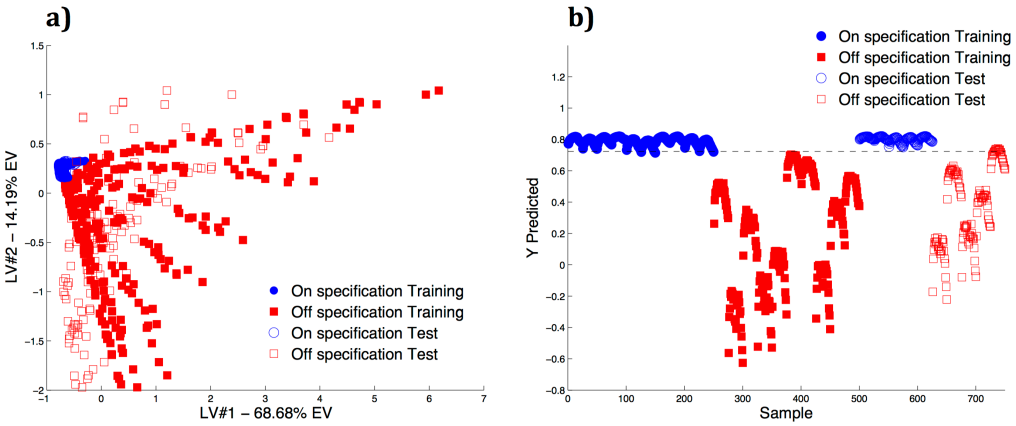


**Figure 3:** a) PLSDA scores plot of the 2nd-order polynomial kernel model built on the simulated data matrix and b) predicted *y* values for both training and test sets. The black dotted line represents the probability threshold, calculated according to the Bayes' theorem [25]. (EV: Explained Variance).

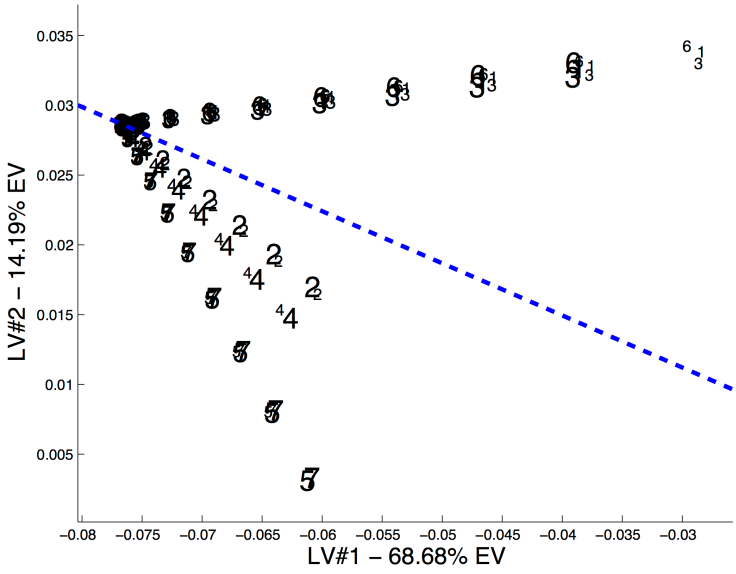


**Figure 4:** Pseudo-sample predicted scores plot for the 2nd-order polynomial kernel model built on the simulated dataset. The blue dotted line represents the discriminant direction between the centers of gravity of the two considered classes. (EV: Explained Variance).

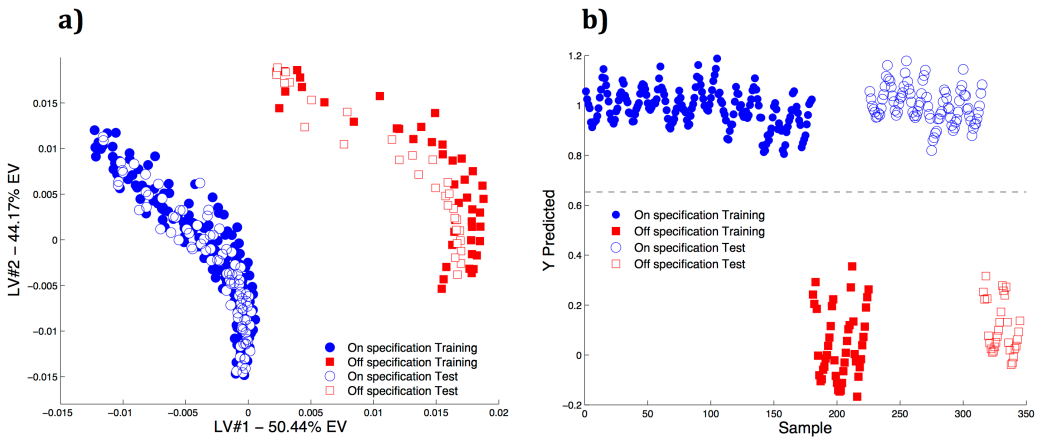


**Figure 5:** a) PLSDA scores plot of the model built on the reduced VWU kernel data matrix and b) predicted *y* values for both training and test sets. The black dotted line represents the probability threshold, calculated according to the Bayes' theorem [25]. (EV: Explained Variance). Its use is justified by verifying that the response values calculated by the model for the observations of the training set are normally distributed within each single class.

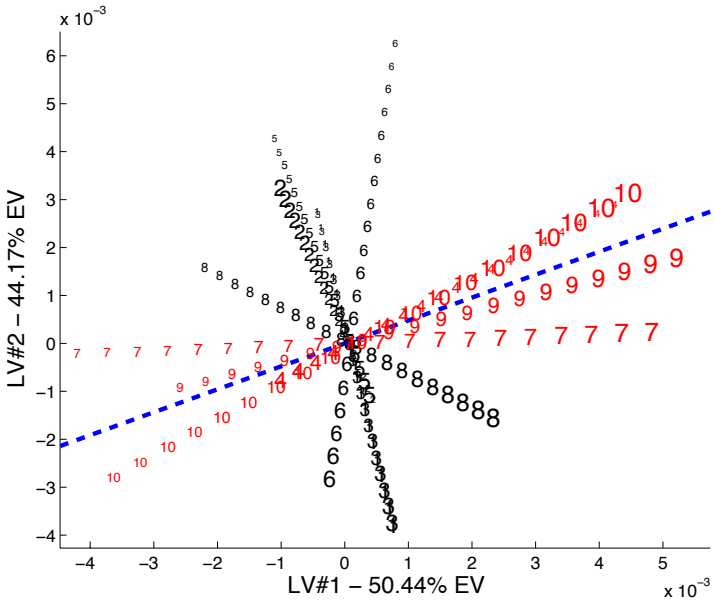


**Figure 6:** Pseudo-sample predicted scores plot for the reduced VWU kernel matrix. The blue dotted line represents the discriminant direction between the centers of gravity of the two considered classes. (EV: Explained Variance).

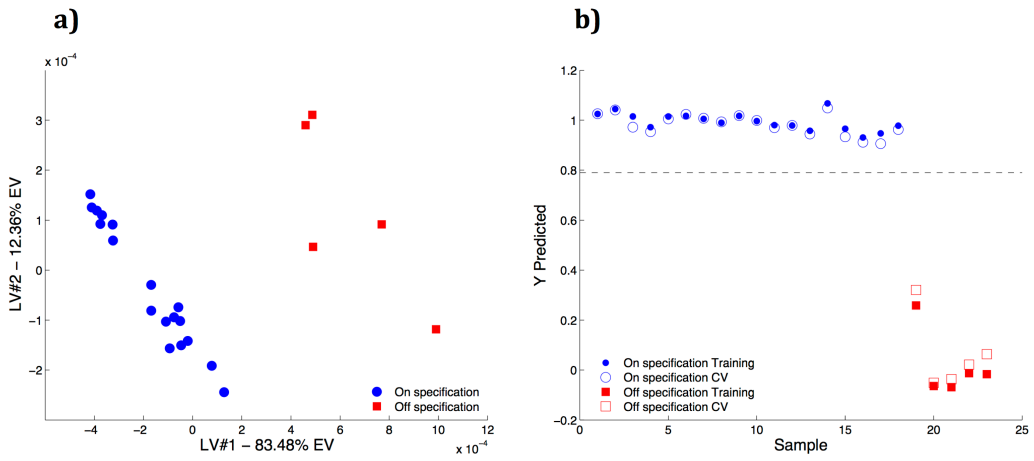


**Figure 7:** a) PLSDA scores plot of the model built on the BWU data matrix and b) predicted *y* values for both training set and cross-validation. The black dotted line represents the probability threshold, calculated according to the Bayes' theorem [25]. (EV: Explained Variance).

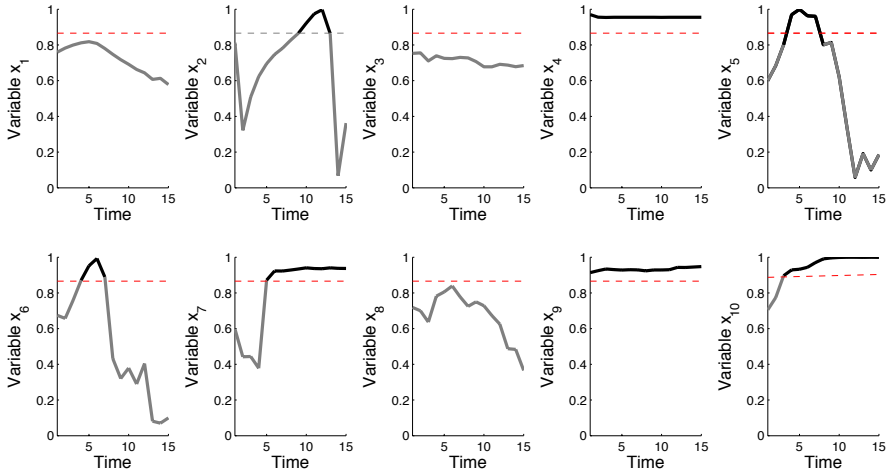


**Figure 8:** Values of the cosine of the angles formed by the pseudo-sample trajectories, related to each one of the original variables, and the discriminant direction at the different time points under study. The red dotted line represents the reference value of the cosine of a 30° angle.

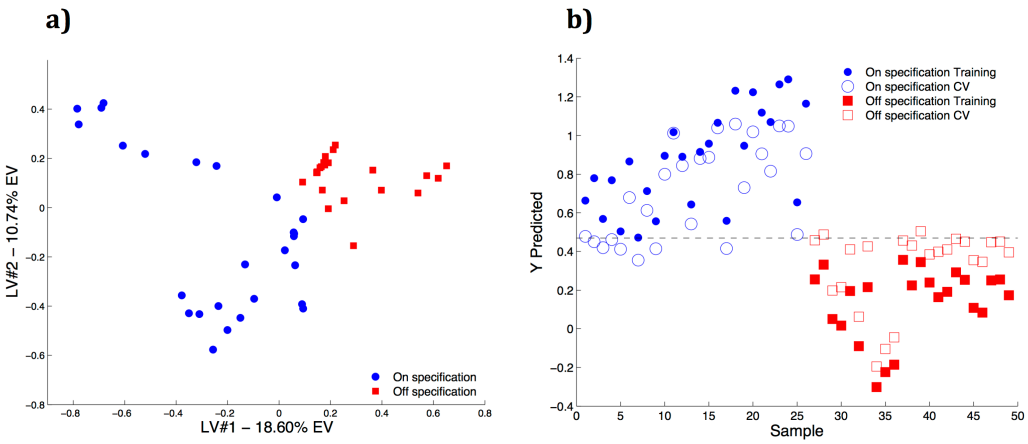


**Figure 9:** a) PLSDA scores plot of the model built on the LFE kernel data matrix and b) predicted *y* values for both training set and cross-validation. The black dotted line represents the probability threshold, calculated according to the Bayes' theorem [25]. (EV: Explained Variance).

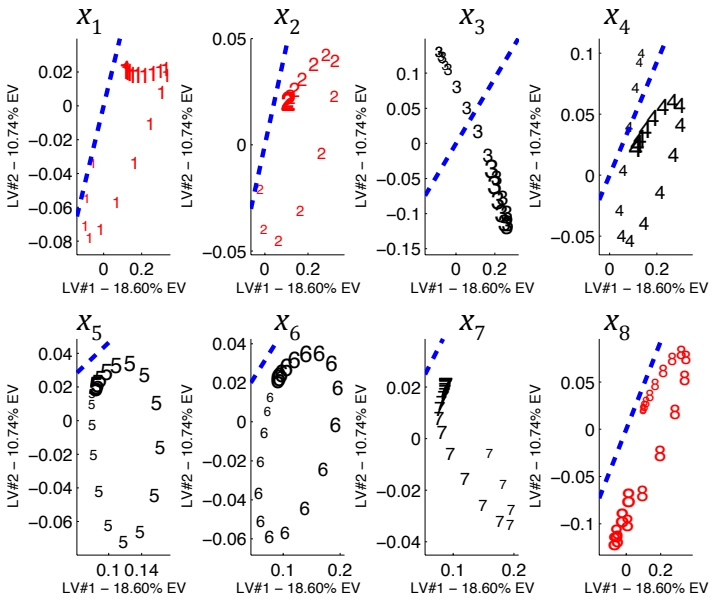


**Figure 10:** Pseudo-samples predicted scores plot for the LFE kernel matrix. Each subplot contains a pseudo-sample trajectory for a specific variable. The blue dotted line represents the discriminant direction between the centers of gravity of the two considered classes. (EV: Explained Variance).

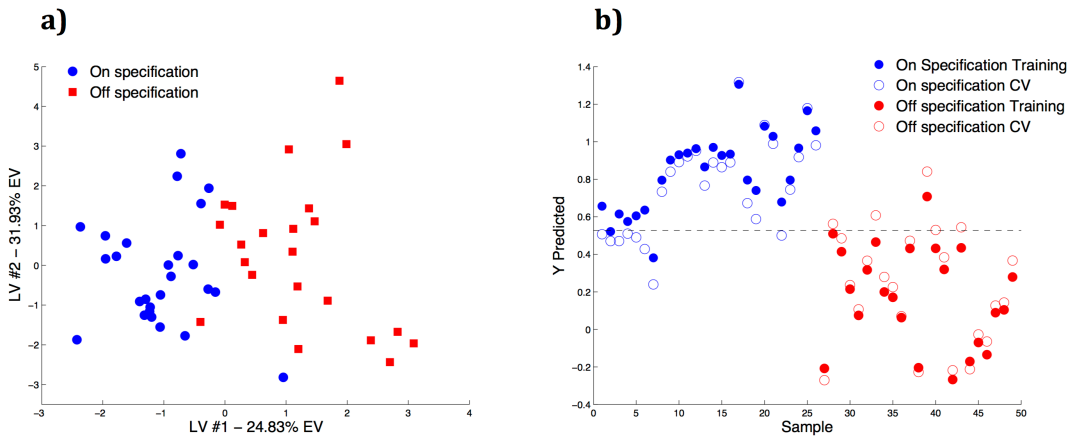


**Figure 11:** a) Scores and b) *y*-predicted plots obtained building a classical PLSDA model on the original LFE data matrix. The black dotted line represents the probability threshold, calculated according to the Bayes' theorem [25]. Correct classification rate in CV: 69.2% (on-specification), 78.3% (off-specification). (EV: Explained Variance)

**Table 1:** Kernel functions used in this article and list of their adjustable parameters.

| Kernel Type | Kernel Function | Adjustable parameters |
|---|---|---|
| Linear | $\mathbf{x}_i{}^{\mathrm{T}}\mathbf{x}_j$ | None |
| 2nd-order polynomial | $(\mathbf{x}_i{}^{\mathrm{T}}\mathbf{x}_j)^{\mathbf{2}}$ | None |
| Gaussian | $\exp(-\frac{\left\|\mathbf{x}_i-\mathbf{x}_j\right\|^2}{2\sigma})$ | σ |

**Table 2:** Latent variable number and correct classification rate of the 4 models built on the simulated dataset

| | | Correct classification rate (%) | | | |
|---|---|---|---|---|---|
| | LV | Training I class | Training II class | Test I class | Test II class |
| PLSDA | 2 | 96.0 | 46.0 | 100 | 42.4 |
| K-PLSDA (linear kernel) | 2 | 95.6 | 45.2 | 100 | 42.4 |
| K-PLSDA ($2^{nd}$-order polynomial kernel) | 2 | 98.4 | 100 | 100 | 92.8 |
| K-PLSDA (rbf kernel, $\sigma$=0.5) | 2 | 100 | 99.2 | 100 | 87.2 |

**Table 3:** Values of the cosine of the angles formed by each pseudo-sample trajectory and the class discriminant direction (simulated data matrix).

| Var. 1 | Var. 2 | Var. 3 | Var. 4 | Var.5 | Var.6 | Var. 7 | Var. 8 | Var. 9 | Var. 10 |
|---|---|---|---|---|---|---|---|---|---|
| 0.88 | 0.97 | 0.89 | 0.92 | 0.80 | 0.87 | 0.81 | - | - | - |

**Table 4:** Values of the cosine of the angles formed by each pseudo-sample trajectory and the class discriminant direction (VWU data matrix).

| Var. 1 | Var. 2 | Var. 3 | Var. 4 | Var.5 | Var.6 | Var. 7 | Var. 8 | Var. 9 | Var. 10 |
|---|---|---|---|---|---|---|---|---|---|
| 0.27 | 0.11 | 0.27 | 0.99 | 0.18 | 0.53 | 0.92 | 0.49 | 0.99 | 0.98 |