Bertinetto, Carlo G.; Vuorinen, Tapani

Influence of Pre-Processing and Distance on Spectral Classification: A Simulation Study

Published in:
Vibrational Spectroscopy

DOI:
10.1016/j.vibspec.2014.08.005

Published: 01/01/2014

Please cite the original version:
Influence of Pre-Processing and Distance on Spectral Classification: a Simulation Study

Carlo G. Bertinetto*, Tapani Vuorinen

Department of Forest Products Technology, School of Chemical Technology, Aalto University, P.O. Box 16300, 00076 Aalto, Finland.

Abstract

An investigation on the influence of pre-processing on the recognition of chemically similar areas in a spectral image, using simulated data. Fictitious spectra of mixtures of five components at varying concentrations were corrupted by different types of noise to mimic typical signals from Raman imaging. They were then processed by various combinations of pre-processing functions, including baseline correction, smoothing, normalization and Principal Components (PC) compression, and by two clustering algorithms (k-means and agglomerative hierarchical clustering) to recognize the original mixtures. The clusters obtained by the different pre-processing combinations and distance metrics were evaluated by statistical parameters (Rand index and silhouette coefficient) and visual inspection. Perhaps the best performing on the basis of all considered criteria is the combination using an adaptive polynomial detrending, a slight smoothing, normalization by the total signal intensity and compression by 4 PCs (spanning 80% of the total variance). More detailed analysis was also carried out on subsets of the whole.

* Corresponding author. Tel.: +358 947001; fax: +358 947024259. E-mail address: carlo.bertinetto@aalto.fi (Carlo G. Bertinetto).
data with a particular type of noise and on the influence of each single pre-
processing/clustering variable.

**Keywords:** Pre-processing; Raman microimaging; Cluster Analysis; Simulated spectra;
Rand index

1. Introduction

Studies involving spectroscopic measurements are increasingly coupled with
computational methods to extract more information from the often huge amount of data
that can be produced by nowadays instruments. The most advanced analysis and
prediction techniques, such as multivariate modeling, curve resolution and machine
learning, usually require a very high data quality to reach their full potential, e.g. detecting
very small components or grasping complex and subtle relationships. Whenever this
requirement is not achievable, the analysis is commonly preceded by pre-processing, a
term that comprises several operations to remove non-relevant sources of variation,
instrumental artifacts and/or non-linear behavior [1]. These operations can be very diverse
and depend on the specific type of spectroscopy and scope of the investigation. For
example, IR spectra are often subject to smoothing, derivatization, multiplicative scatter
correction and wavenumber selection, whereas mass spectra can undergo alignment, dead
time correction or de-isotoping. Comprehensive reviews of pre-processing methods can
be found elsewhere [1−4]. Several software packages, such as Unscrambler, Grams,
Matlab, R and SIMCA, as well as software built-in with spectroscopic instruments, offer
a wide range of pre-processing tools.

It is commonly assumed that pre-processing improves the spectral analysis and leads to
more accurate, simple and robust models [5]. However, if not performed in the right way,
it can also introduce unwanted variation that will influence all the following steps of the analysis and may thus hamper the successful outcome of the entire experiment [2]. The choice of an optimal pre-processing strategy is thus critical, but little systematic study has so far been carried out to provide clear guidelines on when to use or to avoid certain methods or combination of methods. Users often make these decisions without following objective criteria, a risky approach considering that the effects of each numerical pre-processing step are usually not transparent when dealing with large data sets. Moreover, recent research [2] produced results that contrast the conventional wisdom on the matter, stressing the need for a deeper understanding.

Probably the biggest hurdle in conducting an exhaustive investigation on pre-processing is the huge dimension of the variable space when taking into account all methods, parameters, data characteristics and goals of the analysis. Much of the acquired knowledge is scattered in the literature among many works, since researchers often try a few different pre-processing options any time they conduct a chemometric study. Such knowledge is fragmented, occasionally conflicting and too case-related to be of valuable use for other scientists. Works that address the issue more systematically can be found on IR [1,2,5–8] and some on Raman [9,10] spectroscopy. They all optimize pre-processing on a supervised modeling problem, either a multivariate regression or a PLS-based classification, usually with 2 classes. Engel et al. [2] identify three main approaches for the assessment and selection of a pre-processing strategy: (i) trial and error; (ii) visual inspection; (iii) data quality metrics. The first approach, in which different strategies are applied and the best performing according to the analysis goal is selected, is the most robust and reliable, but can sometimes be computationally too expensive for a practical use. In the second one, the pre-processed data are inspected by simply looking at them
and checking if any artifacts are still visible. It is a simple approach and thus widely employed, but it is subjective and often inaccurate. The third approach consists of using data quality metrics to quantify the presence of artifacts in the data; it is a useful compromise between the former two, although suitable metrics are not always available and they are nevertheless dependent on the specific analysis. A few authors [5,7,10] combined the trial-and-error approach with genetic algorithms to find the best combination of pre-processing functions among a wide range of possibilities. They found optimal or near-optimal solutions in a relatively short time, but provided little or no interpretation of the effects of the considered operations on the final outcome. In some cases, they even suggest discarding well-performing solutions that are deemed “odd” in favor of nearly equally good ones that can be more easily explainable by prior knowledge [5].

The aim of the present work is to investigate the influence of some important pre-processing functions on a problem of unsupervised spectral recognition using more than two classes. This problem has received less attention than the supervised sort, but it is frequently encountered in many applications ranging from the classification of drugs [11] to the identification of regions of interest in hyperspectral images [12,13]. The present analysis is done on simulated spectra, an approach which does not guarantee absolute fidelity to the physical world, but which allows for a more precise evaluation of each variable. In particular, the data are made to resemble Raman microimaging spectra, including their typical noise and distortions, such as broad sloping baselines caused by fluorescence of the sample, Gaussian noise and peak height differences caused by variations in laser intensity or in sample depth [10,14,15]. Some of these effects occur in other types of spectroscopy as well; the results of this study can therefore be partly
generalized. Cosmic radiation, a common type of noise affecting all charge-coupled
device detectors, is not taken into account here because its correction is relatively easy to
do and evaluate, even visually [16,17]. The pre-processing functions and parameters
considered here are restricted to a limited number of widely used options, with their order
defined beforehand according to sensible principles, to keep the variable space within a
manageable size. The scope of this paper is indeed to provide broad guidelines rather than
seeking the best possible strategy, which is usually dependent on the specific context.
Nevertheless, this investigation includes the most important pre-processing operations for
this kind of data.

The simulation experiment consists of: (i) defining fictitious spectra of mixtures of five
components at varying concentrations and corrupted by different types of noise; (ii)
processing them by combinations of selected functions for noise removal and data
compression; (iii) recognizing the original mixtures by clustering algorithms [18], namely
\( k \)-means clustering and agglomerative hierarchical clustering. Clustering algorithms,
which perform an unsupervised classification of data, have already been employed in
several works on spectral imaging, both as a main analysis tool [19–21] and as a treatment
prior to other techniques [22,23], but to our knowledge this is the first systematic study
about the influence of pre-processing on the clustering of spectra. The results are
discussed in detail by statistically evaluating the clustering accuracy, on the whole data
set as well as subsets with a particular type of noise, and by examining trends among each
pre-processing/clustering variable.

2. Data set and method

2.1. Simulated spectra
Five ‘pure component’ spectra, indicated by the letters A to E, were defined as vectors of 1000 elements containing sums of various Lorentzian curves convoluted by a Gaussian (amplitude = 1; standard deviation = 5 points;) and normalized by their Euclidean norm. These five spectra, plotted in Fig. 1a, were linearly combined into ‘mixture’ spectra with 10 different proportions and normalized by their amplitude ([max – min]). Figure 1b shows a histogram of these proportions, from which it can be observed that component A imitates a ‘background’ substance present in large amounts everywhere in the sample (as could be an embedding medium), while E represents a ‘trace’ compound. These mixtures can be visually grouped into 4 classes: three with a higher abundance of A, B and C, respectively, and a fourth with roughly equal concentrations of components A-D. This partition is confirmed by k-means clustering both on the mixture spectra and on the concentration matrix, using either Euclidean, city-block or cosine distance.

Each of these ten spectra was added a baseline distortion and shot noise. The baseline distortion could be a straight line decreasing from 1 to 0 over the abscissa range, or a 3rd order polynomial intercepting values of 0, 0.5, 1 and 0.7 at vector elements n. 1, 200, 700 and 1000, respectively. Each baseline was multiplied by 0, 0.5, 1, 2, 4 or 8 to modify its baseline-to-signal ratio (BSR). Shot noise was taken from five measurements in air by an Alpha300 R Confocal Raman microscope (Witec GmbH, Germany), out of which a section of 1151 spectral points was selected, encompassing the range between 637 and 3240 cm\(^{-1}\). They were mean-centered and sigma-scaled, with sigma derived by a Gaussian fit of their probability density function, approximated by their histogram plot. 31 vectors of 1000 elements were then carved out of every 1151-long vector by choosing different starting points. The number of these vectors was then increased four-fold by reflection on the X or Y axis or both, giving a total pool of 620 noise vectors. These were randomly
assigned to the mixture spectra and divided by 100, 70, 30, 20, 10 or 5 to modify their signal-to-noise ratio (SNR). Among the most noisy of the resulting spectra, two sharp peaks deriving from oxygen and nitrogen Raman signals were visible. These peaks were considered useful for the simulation because they look like leftovers from an imperfect cosmic ray removal, a rather frequent case in Raman image analysis. The previous translations and reflections of the noise vector ensure that these peaks do not occur always at the same positions and thus have a more random-like behavior.

Finally, each baseline- and noise-added spectrum was appropriately multiplied to set their Euclidean norm to 1, 2, 4 or 8. The total number of simulated spectra is 2640 and all the parameters that define them are listed in Table 1. A few examples of the final noise-added spectra are depicted in Fig. 2.

2.2. Pre-processing

The pre-processing consisted of four steps: baseline treatment, smoothing, normalization and Principal Component (PC) compression [24], performed in this order except where specified below. The baseline was treated either with an adaptive detrending or with a derivatization. The former is an in-house written method [25] consisting of 5th order polynomial iterative fitting of the peak-free spectral points recognized by an algorithm based on the continuous wavelet transform. The latter is a Savitzky-Golay 2nd derivative using a 21-point window and a 2nd order polynomial. The smoothing method was either a Whittaker smoother [26], using a 2nd derivative and a smoothness parameter $\lambda = 100$, or a wavelet-based smoothing obtained by setting to zero all the detail coefficients from a decomposition at level 3 using the discrete Meyer wavelet. This wavelet was chosen according to the energy/entropy criterion among a set of 45 wavelets including the
Daubechies, Symlet and Coiflet families [27]. Such smoothing method is expected to be more specifically adapted to these instrumental noise patterns. The normalization could be either according to the 1-norm or to the 2-norm, also known as city-block and Euclidean norm, respectively. The number of retained PCs was chosen to span either 90% or 80% of the total variance in the data matrix; this number can be very different depending on the other pre-processing choices.

When the baseline was treated with a 2nd order derivative, smoothing was carried out first, to avoid noise amplification caused by derivatization. The numerical parameters used for baseline correction and smoothing were determined by visual inspection on a few sample spectra and by comparison with other works in the literature that are more focused on this aspect [5,9,10]. As mentioned in the introduction, the purpose of this study is to evaluate main pre-processing approaches rather than do a very detailed optimization.

2.3. Cluster analysis

Clustering is performed by two commonly used methods: $k$-Means Clustering (KMC) and Agglomerative Hierarchical Clustering (AHC) [18]. In KMC, $k$ clusters are each represented by a centroid in the spectral space and every data point (in the present case: every spectrum) is assigned to the nearest centroid. The centroids are then recalculated by averaging the elements of each cluster and all data points are reassigned to these new centroids; this step is repeated until the centroid positions do not change. Because the result is influenced by the random initialization of centroids, the algorithm was repeated 1000 times with different initial conditions and the solution with the lowest sum of squared errors (SSE) was retained. Because such SSE value was reached in more than half of the repetitions in almost all simulations and because trials with a particle swarm
optimizer could not improve this outcome, it was considered as a reasonable approximation of the optimal SSE. The number of clusters $k$ was set as either 4 or 10; distances were calculated according to the city-block metric, the Euclidean metric or based on the cosine angle between data points.

By contrast, AHC starts by considering every data point as a separate cluster, then merges the two clusters with minimum mutual distance and repeats this operation until a single cluster remains. In addition to a distance metric, HCA needs a criterion to determine the distance between clusters: Ward’s method was used here, for which the distance is defined as the increase in SSE when two clusters are merged. It is somewhat similar to SSE-optimized KMC, with the difference that the SSE is minimized for single merging steps instead of the final clusters. This method is widely employed because of its accuracy and robustness to noise; indeed, preliminary trials on our data showed that it performed clearly better than other widely used criteria, such as single, complete and average linkage. Because Ward's method is typically formulated in terms of Euclidean distance, only this metric was used for AHC simulations. In AHC the number of clusters does not need to be determined in advance, but in order to have a more uniform comparison with KMC, the merging was stopped when 4 and 10 clusters were formed, respectively (the final number of clusters will be denoted as $k$ for AHC simulations as well).

All combinations of the mentioned pre-processing and clustering variables were tried, for a total of 648 simulations. All the employed simulation options (including the one of doing nothing) and the abbreviations used to indicate them in the following sections are summarized in Table 2. The calculations were performed using MATLAB® version 8.2 R2013b (The Mathworks, USA).
3. Results and discussion

3.1. Overall evaluation

The pre-processing combinations are primarily evaluated by the accuracy of the resultant clusters. As secondary evaluation criteria, the cluster separation, some computational issues and the performance on certain subsets of the original data are also considered here.

The results with 4 and 10 clusters are treated separately because they correspond to two different and not fully comparable problems: the former mimics the typical operation of identifying chemically similar (though not identical) points in an image, whereas the latter is a more demanding task meant to test the capability of the pre-processing methods in discerning the exact mixture.

The clustering accuracy is quantified by the Rand index ($R$) [28], which measures how a clustering matches a known set of classes, in the present case either the original mixtures or the four classes indicated in Fig. 1b. $R$ (formula shown in Eq. A.1, Appendix A) can have values between 0 and 1, with 1 corresponding to a perfect cluster-class match. The fifteen pre-processing/clustering combinations that yielded the highest $R$ for both KMC and AHC are ranked and listed in Table 3; in the following paragraphs they will often be referred to with their $R$-wise rank number. For KMC, the best ones are \{k = 4, WPD, NoS, 2N, 90%[76], CB\} and \{k = 10, WPD, NoS, 1N, noPCA, cos\}, respectively (see Table 2 for abbreviations). AHC generally yielded a lower $R$ than KMC, although for a few combinations it was similar or higher. The best ones are \{k = 4, WPD, WavS, 1N, 80%\ [4]\} and \{k = 10, WPD, WavS, 2N, 80% [4]\}, which correspond to the 7th and 3rd best outcome for KMC with 4 and 10 clusters, respectively. For both KMC and AHC, the next few combinations in the ranking after the first have an $R$ value almost as good, suggesting that they also are reasonable choices.
The pre-processing function that appears almost everywhere in Table 3 is WDP; other
functions are frequent for certain calculations: CB (appearing in the first ten combinations
for KMC, \( k = 4 \)), 80\% (in the 2\textsuperscript{nd} to 7\textsuperscript{th} combination for KMC, \( k = 10 \)), 1N (in the first
four combinations for AHC, \( k = 4 \)) and 2N (in the first five combinations for AHC, \( k =
10 \)). For KMC, some of the best clusterings (including the very best) are obtained without
any smoothing, whereas all the best clustering by AHC have either WavS or WS. This is
probably due to the stepwise and non-revocable definition of clusters by the AHC
algorithm, which implies that any erroneous initial assignment of a noisy point cannot be
corrected later. No particular prevalence or trend is observed for the remaining pre-
processing functions.

Another meaningful evaluation criterion is cluster separation, especially useful to refine
the analysis among equally accurate clusterings. A higher separation increases the
probability that the result is robust and not out of chance from these particular data. The
separation is quantified by the average silhouette coefficient \((S)\) [29] over all data
(formula shown in Eq. A.2, Appendix A), whose value ranges between -1 and 1 for
perfectly undistinguishable and perfectly separated clusters, respectively. \( S \) for the most
accurate clusterings is indicated in Table 3 next to the \( R \) value. For KMC, some of the
clusterings with the best \( R \), including the very best, do not score well according to the \( S \)
criterion, as can be visualized by the \( S \) vs \( R \) plot in Fig. 3. Following the \( R \)-based ranking,
the first results with \( S > 0.5 \) are the 7\textsuperscript{th} for \( k = 4 \) (highlighted by a circle in Fig. 3) and
those from 2\textsuperscript{nd} to 7\textsuperscript{th} for \( k = 10 \). On the other hand, the clusters from AHC, especially the
most accurate ones, have a much higher \( S \), probably as a result of the characteristics of
the AHC algorithm mentioned above.
It is worth noticing that some of the best combinations in Table 3 don’t use any PC compression. Although this choice may not be detrimental towards the quality of the results, in some contexts it can be unpractical because of the required computation time, approximately 100 times longer than when using PCA. It may therefore be convenient to use equally good (R-wise) pre-processing combinations that include PCA when dealing with large data sets. Another interesting observation is that some of the best results obtained in these simulations come from a number of PCs lower than 5, which is the actual number of independent components known to be in the data. This fact suggests that such pre-processing strategies were not able to recover all the information out of the added noise or that they might have oversimplified the data, as is qualitatively illustrated in the example in Fig. 4, which plots the first 6 PCs after WPD, WavS and 1N. All the features of pure component spectra can be observed in these PCs, including those of the ‘trace’ component E, indicated by ovals. However, several noise-related features, highlighted by shadings, are hardly distinguishable from those of component E: any cutoff in the number of PCs will therefore either exclude E or include unwanted noise.

Further information can be obtained by visual inspection. Figure 5 depicts some selected clusterings as color maps: the data points from the same mixture are aligned in rows, with the SNR decreasing in blocks going from left to right; a perfect clustering appears as homogeneous horizontal stripes. For $k = 4$, the most accurate results from both KMC and AHC match all the correct classes except for a few noisy data points, see Fig. 5a (only KMC shown). On the other hand, the clustering with highest $R$ for $k = 10$ does not correctly distinguish all mixtures and, as can be seen in Figs. 5b (KMC) and 5c (AHC), at least two pairs of mixtures are always clustered together even on data with little noise. Exceptions to this rule are the 6th best clusterings by both KMC and AHC and the 7th best
by KMC (shown in Fig. 5d), which cluster together only mixtures 2 and 3, at least on data with high SNR. The below-optimal \( R \) value of these last-mentioned clusterings is caused by a poor performance on noisy data, especially for mixtures 1-3.

These tendencies, i.e. better clustering of noisy data by the pre-processing combinations ranked 1\(^{st}\) to 5\(^{th}\) in Table 3 (10 clusters) and of clean data by \{WPD, WhS, 1N, 80\%[3], Eucl/CB\}, are observed even when \( k \) is increased up to 15 (not shown here). The choice of optimal pre-processing is thus dependent on the type of data, with the 6\(^{th}\) combination (or 7\(^{th}\) for KMC) seeming more appropriate when the SNR is high. In practice, it is even more so considering that it is usually preferred to have very reliable results on clean data and that the noisiest signals are often discarded as outliers. Moreover, the combination \{WPD, WavS/WhS, 1N, 80\%[4], Eucl/CB\} is probably the most interesting because it obtains good clusterings in all four simulations: 7\(^{th}\) best for \{KMC, \( k = 4 \)\} (and, as already observed, best among those with \( S > 0.5 \)), 1\(^{st}\) for \{AHC, \( k = 4 \)\} and 6\(^{th}\) for \{KMC/AHC, \( k = 10 \)\} (with the best clustering of low-noise data). Hence, this pre-processing seems a more general data treatment suitable for more than one analysis task.

A peculiar result is observed for \{KMC, \( k = 4 \), WPD, WhS, N2, 80\%[4], CB\}, which has the 21\(^{st}\) highest \( R \) with 4 clusters, shown in Fig. 5e. For the high-SNR data points, mixture n. 9 is (wrongly) clustered together with mixtures n. 6-7, but it is instead clustered correctly for the noisy points. No satisfactory explanation was found and this outcome might be product of chance.

3.2. Results on subsets of data

To provide a more detailed picture of the clusterings obtained in our simulations, the Rand index was calculated for subsets of the data, which included: (i) spectra with low SNR (=
308 30, 20, 10, 5); (ii) spectra with high BSR (= 2, 4, 8); (iii) spectra with linear baselines only; (iv) spectra with polynomial baselines only. Figure 6 plots the Rand statistics of the same pre-processing/clustering combinations shown in Table 3, for all data as well as the mentioned subsets. The series with low SNR is always considerably below the overall $R$, especially for AHC, see Figs. 6c and 6d. Particularly notable are the low values for the 6th and 7th combination with $k = 10$ for both KMC and AHC, which indicate a poor clustering of noisy data as was observed visually in Fig. 5d. The high-BSR series stays slightly below the overall $R$ most of the times for KMC (Figs. 6a and 6b), whereas for AHC it is more erratic. Subsets with linear or polynomial baseline often have different $R$, but there is no regularity over which outperforms the other. Considering that the baseline correction method (WPD) is the same for all but one of the plotted combinations, the spectral recognition with respect to the baseline type is clearly influenced by the other pre-processing functions as well.

3.3. Dependence on single pre-processing functions
To gain more insight on the effect of each pre-processing/clustering parameter, trends in the $R$ and $S$ statistics for the whole data set were examined on sections of the solution space in which only a single variable was changed at a time. For KMC, 81 sections were produced for each operation (baseline correction, smoothing, normalization, PC compression and clustering with different distance metrics) and number of clusters (4 and 10), for a total of 810. For AHC, 27 sections were produced for each operation (which did not include the choice of distance metric), for a total of 216. Due to the high number and redundancy of these trends, the ones with low $R$ values are ignored from the analysis and only the most representative are shown in Fig. 7. Moreover, since practically the same
trends were observed for KMC and AHC, all examples are taken from the former except where specified. It must be pointed out that it is not possible to perfectly separate the effect of one pre-processing function from the other, as was already noticed when commenting about baseline types at the end of Section 3.2. Another evident example is the calculation of the Savitzky-Golay 2nd derivative, which has an influence on shot noise as well.

3.3.1. Dependence on the baseline correction

The R trend is WPD > S2D > noBC in nearly all cases, see Fig. 7a (crosses), as hinted also by the quasi-absence of S2D in Table 3. An exception are the trends using $k = 4$, WhS, noN/1N/2N, noPCA and cos, which are S2D > WPD > noBC, see Fig. 7a (triangles). As is already widely known, these numbers show that baseline correction is very important for a good spectral recognition. More debated is whether detrending is better than derivatives and, according to these results, it is. This outcome, also reinforced by the fact that in a few spectra the detrending was not optimal, suggests that derivatives remove more information than was previously thought. Moreover, spectra treated with S2D need many more PCs (about 4-5 times more) to span the same variance, implying that derivatization leads to an unnecessary increase in the problem’s complexity.

3.3.2. Dependence on the smoothing

Smoothing has a very small influence on the clustering accuracy and all tried options produce roughly the same R, see Fig. 7b (triangles). The only exception is with combinations that yield a high R by AHC, for which WavS ≈ WhS > NoS (see Fig. 7b, crosses), as was already commented in Section 3.1. The other effect of smoothing is a
considerable increase in $S$, see Fig. 7b (filled blue circles), and great reduction (up to eight-fold) of the number of PCs needed to describe the same variance in not-smoothed data. Contrary to the expectations about using a wavelet adapted to the noise patterns, no significant difference is observed between applying WhS and WavS. From these results, smoothing does not appear to be absolutely necessary for the analysis task, but it can make the data less blurred and the clustering more simple and stable.

It is important to point out that in the present simulation this operation was performed with the help of background knowledge about peak shapes. In practice, when such knowledge is not available it is better to smooth only to a minimum degree. The small gains derived from an optimal smoothing are usually not worth the risk of oversmoothing and consequent loss of information.

3.3.3. Dependence on the normalization

The $R$ trends show that normalization is extremely important for an accurate clustering, see Fig. 7c (crosses). The only exception are the simulations that use noN, noPCA and cos, see Fig 6c (circles), because the cosine distance levels off the differences in total intensities, However, the use of PCA and cosine distance without normalization severely distorts the spectral information, see Fig. 7c (squares). It must be stressed that in this analysis spectra are recognized on the basis of the relative signal proportions; if the absolute signal intensities were more meaningful, other types of normalization (such as internal standards) or none at all would be preferable. For $k = 4$, most of the $R$ trends have $1N > 2N$, see Fig. 7c (triangles); for $k = 10$ no significant difference is observed between the two normalization methods.
In the literature, the Euclidean normalization (2N) is employed more often, but there are no clear opinions about which one is better. Out of common sense, if the purpose is to normalize by the total spectral intensity the 1-norm is more appropriate because it is more directly related to the number of interacting photons, but the 2-norm should be preferable for noisy spectra because it gives less weight to small noise peaks. To seek some evidence of this reasoning among our results, we computed the average $R$ using 1N and 2N, respectively, for all data as well as for the low-SNR subset introduced in Section 3.2. The 129 combinations with lowest $R$, for both the 4-cluster and 10-cluster rankings, were excluded from this average. With $k = 4$, these values were 0.8638 (1N) and 0.8623 (2N) for all data and 0.8366 (1N) and 0.8434 (2N) for low-SNR data. With $k = 10$, they were 0.8966, 0.9005 and 0.8857, 0.8929, respectively. In both cases the difference $R(2N) - R(1N)$ is greater for noisy data than for the whole data set; this is in line with our hypothesis, although these differences are too small to be an unequivocal proof.

3.3.4. Dependence on the compression with principal component analysis

PC compression does not have any significant influence on the $R$ values, see Fig. 7d (crosses), except for the already mentioned sharp decrease when using PCA, cosine distance and no normalization. A similar effect is observed also for simulations using $k = 4$, S2D, 1N/2N, cos, see Fig. 7d (triangles). There is no great difference in $R$ between spanning 80\% and 90\% of the total variance. On the other hand, the S plots always follow the trend 80\% > 90\% > noPCA, see Fig. 7d (filled blue circles), indicating that a limited number of PCs leads to a better cluster separation. The determination of the exact number of PCs for best separation is out of the scope of this paper.
3.3.5. **Dependence on the distance metric**

The distance metric presents the most various $R$ trends: $\text{CB} \approx \text{Eucl} > \cos$ for $k = 4$, WPD, 1N (Fig. 7e, crosses); $\cos >> \text{Eucl} > \text{CB}$ for $k = 4$, WhS, S2D, 1N/2N, noPCA (Fig. 7e, triangles); $\text{CB} > \text{Eucl} \approx \cos$ for $k = 4$, WDP, noS, 1N/2N (Fig. 7e, circles); no significant dependence is observed for the other series. For $k = 10$, the trends are similar but much less pronounced, often nearly constant. The $S$ trend is always $\cos > \text{Eucl} > \text{CB}$ (Fig 6e, filled blue circles) for both 4 and 10 clusters, which comes in favor of using cosine distance if it is not in conflict with other pre-processing choices.

4. **Conclusion**

The presented investigation explored the influence of commonly used pre-processing operations on the problem of chemical recognition from spectral data based on the proportion of its constituting signals. The analysis was carried out through $k$-means and agglomerative hierarchical clustering on a set of spectra that simulate a Raman image. The results were evaluated by visual inspection and statistical parameters, with the aim of providing guidelines for future studies and applications.

As expected, no simple relationship was found and the optimal pre-processing combination depends on the type of data and evaluation criteria. A few combinations have been identified as the best choices that produce the most accurate clustering according to the a-priori known chemical mixtures. In particular, the one using an adaptive polynomial detrending [25], some smoothing, normalization by the total intensity and compression by 4 PCs (representing 80% of the total variance) simultaneously achieved excellent cluster accuracy and separation, computational speed and range of applicability.
Nevertheless, it is not the best performing for very noisy spectra, as shown by color plots and statistics on subsets of the whole data.

The analysis of individual pre-processing functions revealed some interesting trends. Baseline correction is very important for an accurate clustering and the employed detrending nearly always outperforms 2\textsuperscript{nd} derivatization in terms of both cluster accuracy and problem simplicity (i.e. it required less PCs). On the other hand, smoothing has little influence on the accuracy, but it improves separation and simplicity. Normalization is necessary for this type of spectral recognition, except for the case using cosine distance and no PC compression. Using the 2-norm appears slightly better than 1-norm for noisy data.

PC compression does not have any significant effect on the cluster accuracy, but the separation is best when selecting a not too high number of PCs, here represented by the one spanning 80\% of the data variance. A more complicated dependency is observed for the clustering distance metric, investigated only for the $k$-means method. The Euclidean and city-block distances are equally accurate in many cases, though the former seems slightly favored for coarser clustering (i.e. low number of clusters). The cosine distance yields the highest cluster separation but is accurate only with certain pre-processing choices.

The results shown in this paper, although they do not exhaustively cover the whole issue, offer a clearer picture of which pre-processing functions and distances improve the analysis of spectral images and which should instead be avoided.

Acknowledgements

The financial contribution by the Academy of Finland is gratefully acknowledged.
Appendix A: formulae of statistical parameters used for clustering evaluation

Rand index ($R$):

$$R = \frac{f_{00} + f_{11}}{f_{00} + f_{01} + f_{10} + f_{11}}$$  \hspace{1cm} (A.1)  

where:

$f_{00} =$ number of pairs of data points having a different class and a different cluster  
$f_{01} =$ number of pairs of data points having a different class and the same cluster  
$f_{10} =$ number of pairs of data points having the same class and a different cluster  
$f_{11} =$ number of pairs of data points having the same class and the same cluster

Silhouette coefficient, average value for all data points ($S$):

$$S = \frac{1}{N} \sum_{i=1}^{N} \left( \frac{b_i - a_i}{\max(a_i, b_i)} \right)$$  \hspace{1cm} (A.2)  

where:

$N =$ total number of data points  
$a_i =$ average distance from data point $i$ to all other points in cluster $I : i \in I$  
$b_i =$ average distance from data point $i$ to the points of the neighboring cluster $J : i \notin J$  
and $b_i$ is minimum among all clusters

References

DOI:10.1016/j.trac.2013.04.015.


Figures

Figure 1. Simulated spectra: (a) Spectra of pure components (arbitrary units). (b) Mixture proportions; the parentheses on the right side indicate the classes used for the clustering with \( k = 4 \). For more details on these plots see Table 1.
Figure 2. Examples of simulated spectra after adding baseline distortion, shot noise and multiplying the total intensity: (a) mixture n. 1, linear baseline (BSR = 0.5), SNR = 30, norm = 1; (b) mixture n. 4, polynomial baseline (BSR = 4), SNR = 100, norm = 4; (c) mixture n. 9, polynomial baseline (BSR = 0.5), SNR = 5, norm = 8. For abbreviations see Table 1.
**Figure 3.** Plot of $R$ vs. $S$ for the 50 most accurate clusterings by the $k$-means method with $k = 4$ (top) and $k = 10$ (bottom), respectively. The dashed line marks $S = 0.5$; the circle in the top plot highlights the 7th most accurate clustering.
Figure 4. Plot of the first 6 PCs after WPD, WavS and 1N. These PCs explain 87% of data variance; 80% and 90% are spanned by 4 and 11 PCs, respectively. Spectral features from the ‘trace’ component E (ovals) are mixed with and hardly distinguishable from several noise-related ones (gray shading).
Figure 5. Color-based visualization of clusterings obtained from certain preprocessing/clustering combinations. Each square represents a data point, identified by the mixture number, the SNR (increasing in blocks from left to right) and the BSR preceded by “L” or “P” for linear and polynomial baseline, respectively (the BSR sequence is repeated for each SNR block). The norm is not indicated for simplicity, since all these combinations include a normalization step that cancels the intensity variations. Figure 5a: \{KMC, k = 4, WPD, NoS, 2N, 90\%[76], CB\} (highest R). Figure 5b: \{KMC, k = 10, WPD, NoS, 1N, noPCA, cos\} (highest R with 10 clusters). Figure 5c: \{AHC, k = 10, WPD, WavS, 2N, 80\%[4]\} (highest R by AHC with 10 clusters). Figure 5d: \{KMC, k = 10, WPD, WhS, 1N, 80\%[3], CB\} (7th highest R by KMC with 10 clusters). Figure 5e: \{KMC, k = 4, WPD, WhS, N2, 80\%[4], CB\} (R = 0.938, 21st highest R by KMC with 4 clusters). For abbreviations see Table 2.
Figure 6. Rand statistics of the pre-processing/clustering combinations from Table 3 for all data (filled dots) and for the following subsets: spectra with SNR = 30, 20, 10, 5 (stars); spectra with BSR = 2, 4, 8 (crosses); spectra with linear baselines (triangles); spectra with polynomial baselines (circles). Figure 6a: KMC, $k = 4$; Figure 6b: KMC, $k = 10$; Figure 6c: AHC, $k = 4$; Figure 6d: AHC, $k = 10$;
Figure 7. R and S trends in selected sections of the solution space focusing on a single pre-processing/clustering variable. The examples are all taken from KMC results, except where specified. Figure 7a, dependence on the baseline correction method: R for \( k = 10, \) WhS, 2N, 80%, Eucl \{triangles\} and \( k = 4, \) WhS, 1N, noPCA, cos \{crosses\}. Figure 7b, dependence on the smoothing method: R for \( k = 10, \) WPD, 1N, 80%, Eucl \{triangles\} and \{AHC, \( k = 4, \) WPD, 1N, 90\%\} \{crosses\}; S for \( k = 4, \) WPD, 1N, 80%, CB \{filled blue circles\}. Figure 7c, dependence on the normalization method: R for \( k = 10, \) WPD, WavS, 80%, CB \{crosses\}, \{k = 4, \) WhS, S2D, noPCA, cos \{circles\}, \{k = 4, WPD, WavS, 90%, cos \{squares\} and \{k = 4, WPD, WhS, 90%, Eucl \{triangles\}. Figure 7d, dependence on the PC compression: R for \( k = 10, \) WPD, WhS, 2N, Eucl \{crosses\} and \{k = 4, WhS, S2D, 1N, cos \{triangles\}; S for \{k = 4, WPD, WhS, 1N, cos \{filled blue circles\}. Figure 7e, dependence on the distance metric: R for \{k = 4, WPD, WavS, 1N, 80\%[4] \{crosses\}, \{k = 4, WhS, S2D, 1N, noPCA \{triangles\} and \{k = 4, WDP, noS, 2N, 90\%[76] \{circles\}; S for \{k = 10, WPD, WavS, 1N, 90\% \{filled blue circles\}. For abbreviations see Table 2.