LU-TP 23-09
April 2023

Quantification of the Performance of Algorithms for spectra Baseline Correction

Jerry Huang

Computational Biology and Biological Physics, Lund University

Master thesis supervised by Carl Troein

LUND
UNIVERSITY

Abstract

Spectroscopy serves as a vital tool in both scientific research and industrial applications. In spectral analysis, baseline correction is important in order to be able to efficiently extract essential features. Several algorithms for baseline correction have been developed, including Asls (Asymmetric Least Squares algorithm), arPLS (Asymmetrically Reweighted Penalized Least Squares algorithm), airPLS (Adaptive Iteratively Reweighted Penalized Least Squares algorithm), and MSBC (Multiple Spectra Baseline Correction algorithm). In this paper, a computational framework is devised to assess the efficacy of these four algorithms, based on principal component analysis, K-means clustering, confusion matrix and Silhouette analyses. Comprehensive computational experiments are conducted on synthetic and real Fourier transform infrared spectroscopy data. Drawing from our findings, we deduce that baseline correction significantly helps our spectral analysis by extracting information. Asls can be used to sort spectra into different clusters, and MSBC is able to attain consistent baselines and corrected spectra across all data. Moreover, we suggest potential avenues for refining baseline correction algorithms.

Populärvetenskaplig beskrivning

Imagine that your left eye is a human eye while the other one is from a fly, what will you see? The vision you have is quite different from the one for a fly. If your right eye becomes a fly one, the pictures from the human eye and the fly eye will combine, forming a new one. You can sense wavelengths visible to the human and those visible to the fly, and that is the working principle for spectral technology, multiple spectra extend the wavelength range you can sense and give information from different wavelength bands.

However, there are still some difficulties hard to be solved with the spectra technology. For example, the human eye can only receive light energy signals from objects in three spectral bands: red, green, and blue. They are what we often call the three primary colors of luminescence, but in fact, we can see more subtle colors such as orange, purple, turquoise, and so on produced by the combination of these three colors. However, we are not unable to distinguish the difference between pure yellow and red-green mixed colors, which is also called "metamerism". But hyperspectral imaging makes it easy to tell the difference.

Hyperspectral imagers (HSIs) can be compared to hundreds or thousands of single-point detectors closely arranged together and focusing on an area at the same time, each working independently and acquiring spectral information about its pixel. Each pixel in these images out of the imager has its spectrum, and each spectrum contains hundreds of spectral bands. Such a capability of hyperspectral imaging allows people to see the spectral signal at every distinguishable spatial position in a scene, that is, more dimensions of information are obtained. Therefore, hyperspectral imaging has a wide range of application scenarios, including art identification, crop health, coastline mapping, forests, mineral exploration, urban and industrial infrastructure, production line product quality, environmental monitoring, and so on.

The spectral result can be a metaphor for a series of mountains, they consist of mountainsides and peaks. You can regard the result as peaks of signal added to a continuous mountainside, or say baseline. The most important information is hidden in those peaks. To distinguish this information, we are supposed to do baseline correction, which means we should cut off the mountainside. After cutting away the baseline, the bare signal peak will come out, demonstrating the vital feature of the sample, making it easier for us to analyze the sample.

To accomplish this goal, many methods were invented and they give variant results. It remains a hard question to quantify how good these results are. We can make up artificial spectra by adding peaks to a virtual baseline, then do baseline correction to the spectra with these methods and compare the result with the given baseline. We can also do such corrections with these methods to the multiple spectra we made up or the real spectra, then check if it will be easier to sort the corrected spectra into different kinds, such as sample and background.

Contents

List of acronyms

HSI Hyperspectral imaging
FTIR Fourier-transform infrared spectra
Asls Asymmetric Least Squares
arPLS Asymmetrically reweighted penalized least squares algorithm
airPLS Adaptive iteratively reweighted penalized least squares algorithm
MSBC Multiple spectra baseline correction algorithm
PCA principal component analysis
λ the smoothness parameter of baseline correction
p the asymmetric reweight parameter of baseline correction

List of Figures

List of Tables

1 Introduction

The baseline of a spectrum is the underlying curve that is not directly related to the spectral features we are interested in. It is a common scenario that the collected spectra data, which consists of sharp characteristic peaks, are superimposed upon a continuous, slowly varying baseline [4]. Such a scenario is shown in Fig.1. The difficulties in spectral analysis, such as the failure to distinguish samples from the background, due to the baseline can be thorny problems. Various sources such as instrument noise, background signals, the effect of scattering, and other types of interference, can lead to the formation of a baseline [5].

To guarantee the effectiveness of the spectral analysis, baseline corrections are supposed to be implemented on the raw data. Baseline correction is a common spectral preprocessing procedure. This procedure removes the baseline from a spectrum to improve the accuracy of the spectroscopic analysis.

Baseline correction is important for several reasons. First, it can improve the performance of spectral analysis by reducing the effects of scattering, impurities, and other sources of errors [5]. Second, it can enhance the visual feature of spectral images by removing systematic variations, usually from spectral apparatus and measuring instruments, that may be present in different samples or measurements. Finally, the visual appearance of the spectral image can be improved by the baseline correction. It will remove unwanted background signals and enhance the contrast[5], making it easier to identify and interpret spectral characteristics. Overall, baseline correction is an important step in the analysis of spectral data that can improve the accuracy, explainability, and reliability of the results.

There are several algorithms to perform baseline correction, including polynomial fitting [6] and wavelet smoothing [7, 8]. The choice of method depends on the specific property of different kinds of spectrum, the reason for the formation of baselines, and the time cost. In the Theory background section, the theoretical background of the spectra, the four baseline correction algorithms that are tested, and several numerical analysis methods used to analyze their performance are discussed. The information on the data and the experimental environment are given in the Datasets and the experimental environment section. In the Result section, we will develop a computational scheme to quantify the performance of different algorithms based on their abilities to sort spectra into the right clusters and demonstrate the test results in both synthetic data and real data, and finally draw conclusions.

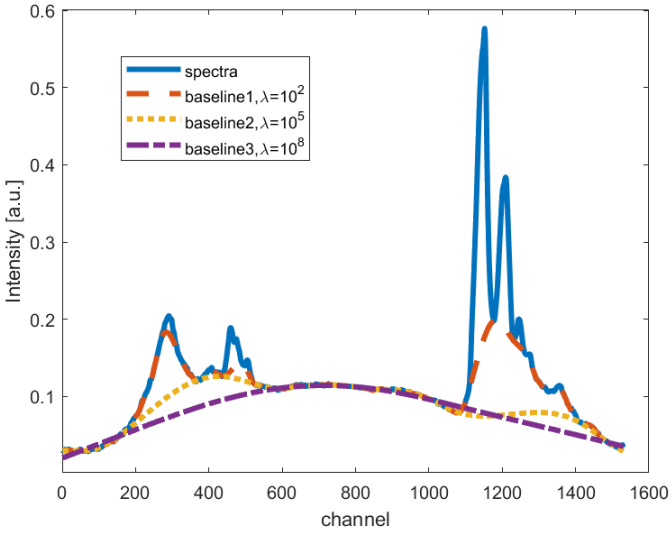


Figure 1: An example of spectra and calculated baselines. The baselines are calculated by the arPLS algorithm. Different parameters can cause different baselines.

2 Theory background

2.1 Spectra

Spectra refers to the distribution or representation of the intensity of electromagnetic radiation, such as refracted and reflected light from a sample, as a function of their wavelengths or frequencies. Spectra play an important role in many fields and have been used for a long period in science and industrial applications for understanding the properties and behavior of matter at the atomic and molecular levels.

Spectra can be measured and analyzed using various techniques, such as spectroscopy, spectrometry, or spectrography. These methods involve the use of instruments and detectors to measure the intensity of radiation at different wavelengths or frequencies, enabling scientists to extract valuable information about the source or the interaction with matter. Spectral analysis plays a crucial role in understanding the fundamental properties of matter, studying astronomical objects, characterizing materials, and advancing scientific knowledge across a wide range of disciplines.

2.1.1 Fourier-transform infrared spectra

Fourier-transform infrared (FTIR) spectroscopy is a powerful analytical technique used to obtain information about the molecular composition and structure of a sample based on its interaction with infrared light. In FTIR spectroscopy, the sample is exposed to in-

frared radiation, and different chemical compounds absorb infrared light at characteristic frequencies due to the specific vibrational and rotational motions of the molecules within the sample. When infrared light passes through the sample, certain frequencies are absorbed by the sample, while others are transmitted or scattered [9, 10]. Plot the resulting absorption or transmission of light as a function of the wavelength, we can see the spectral fingerprint of the sample.

The FTIR spectrum consists of a series of peaks and troughs, each corresponding to a specific vibrational or rotational transition within the molecules present in the sample. These peaks are characteristic of the functional groups and chemical bonds present in the sample, providing valuable information about its molecular composition and structure [10]. By analyzing the positions, shapes, and intensities of the peaks, scientists can identify and quantify the various components in a sample, as well as investigate molecular interactions, chemical reactions, and material properties. The FTIR spectra would have a wandering baseline changing constantly due to the background, such as light source, atmospheric environment, and impurity in samples [11]. It is an underlying curve that we want to remove for a better quantification or presentation of spectra.

2.1.2 Hyperspectra

Hyperspectral imaging (HSI), is an advanced imaging technique that allows the collection of high-resolution spectral data from a scene or an object in a non-destructive and non-intrusive way. Hyperspectral images consist of many narrow, contiguous spectral bands, typically distributing in the whole electromagnetic spectrum, including but not limited to the visible, near-infrared, and shortwave infrared regions.

Compared to the traditional spectra, to name a few, Raman or infrared spectra, hyperspectral provides a higher resolution and much more details about the observed sample. Because hyperspectral demonstrates the response signal from the sample to a wider wavelength band. In a hyperspectral image, each pixel has a spectrum, recording how a specific area responds to lights at different wavelengths across the electromagnetic spectrum, allowing the researchers to identify the samples based on their characteristics of spectral behaviors. Therefore, the hyperspectra image can be used to measure both spatial and spectral information from samples [12].

Hyperspectra can be regarded as a more advanced form of multi-spectra that provides a higher level of detail and resolution in the spectral domain. The largest difference between them is that typically only a few discrete wavelength bands, usually 3 to 10, are required to acquire multiple spectra. However, hundreds or even thousands of spectral bands will be involved in the acquisition of a hyperspectral image. Additionally, in multiple spectra, the spectral bands are wider and typically chosen to capture the averaged response to a wide band from the object being imaged. This strategy will certainly bring a larger error. While in hyperspectral imaging, wavelength bands are finer and much narrow, allowing for a more detailed analysis of the object's spectral features. This technique has been

applied in domains such as remote sensing [13], agriculture [14], mineral exploration [15], and biomedical imaging [16].

2.2 Baseline correction algorithms

2.2.1 Asymmetric least squares algorithm

As introduced in the initial section, the baseline of a spectrum refers to the underlying curve that isn't directly linked to the specific spectral characteristics under investigation. This baseline's emergence can be attributed to factors such as illumination conditions, atmospheric influences, sample impurities, and more. Typically, this baseline is overlaid by characteristic peaks, and its removal enhances the clarity and accuracy of spectral analysis and presentation.

The Asymmetric Least Squares (Asls) baseline correction algorithm is widely employed for eliminating baseline variations from spectra. Operating in a non-linear fashion, this method aims to minimize the sum of squared differences between the original spectrum and the estimated baseline, along with the difference of the fitted baseline itself.

The mathematical foundation of the asymmetric least squares algorithm can be traced back to the work of Whittaker a century ago[1]. In 2003, Paul H. C. Eilers encountered this algorithm during a review of literature and subsequently transformed it into a statistical smoothing technique founded on sparse matrices[17].

Now, let's delve into the theoretical underpinnings of the Asls algorithm from a probabilistic perspective. After performing baseline correction, the resulting baseline should ideally exhibit smoothness—devoid of sharp spikes or abrupt jumps. Additionally, it should align with the trend of the raw data without excessive deviations. For simplification, we posit that in the absence of uncertainties or irregularities arising from observational errors, a smooth curve should emerge. Here, we precise the word "smooth" by defining it as making the differences, in any order, as small as possible [1].

Suppose that we have done n observations and attained n data y_1, y_2, y_3, ... y_n, which are under the influence of uncertainties or irregularities. While z_1, z_2, z_3, ... z_n are the true value of the corresponding observed quantity. We can consider the following hypothesis: the true value which should have been observed, for example, for y_1, located between z_1 and $z_1 + \sigma$, where σ is a small constant number. Postulating the Gaussian law of error, the probability that this hypothesis is true is

$$\frac{h_1}{\sqrt{\pi}} e^{-{h_1}^2 (y_1 - z_1)^2} \sigma.$$

where h_1 is a constant showing how precise this measurement can be [1]. Similarly, the

probability that the true value of the second observation lies between z_2 and $z_2 + \sigma$ is

$$\frac{h_2}{\sqrt{\pi}} e^{-{h_2}^2 (y_2 - z_2)^2} \sigma.$$

Apply this hypothesis to all observed data points, the corresponding probability is

$$\frac{h_1 h_2 h_3 ... h_n}{\pi^{\frac{n}{2}} e^F \sigma^{-n}}$$

Here F is the sum:

$$F = {h_1}^2 (y_1 - z_1)^2 + {h_2}^2 (y_2 - z_2)^2 + ... + {h_n}^2 (y_n - z_n)^2.$$

If the data have been sampled uniformly, the degree of smoothness of the data sequence can be defined as:

$$S = (z_2 - z_1)^2 + (z_3 - z_2)^2 + ... + (z_n - z_{n-1})^2.$$

The quantity S expresses how smooth the curve is numerically, and the quantity F expresses how the fitted curve deviates from the original data. We can assume that the prior possibility that the hypothesis is true is $ce^{-\lambda S}\sigma^n$, where c and λ are constants [1].

If p_s donates the probability of the s^{th} hypothesis before a phenomenon is observed, and P_s donates the probability of an observed phenomenon on the assumption of the truth of the s^{th} hypothesis, the most probable hypothesis after the phenomenon has been observed is that for which the product $P_s p_s$ is greatest [1].

Applying this theory to our case. $P_s = \frac{h_1 h_2 h_3 ... h_n}{\pi^{\frac{n}{2}} e^F \sigma^{-n}}$, $p_s = ce^{-\lambda S}\sigma^n$. That is to say, the probable hypothesis, or, the most probable value set z_1, z_2, z_3, ... z_n will minimize the quantity $Q = \lambda S + F$ [1]. We can write this optimization problem in a more mathematical way [4]:

$$\mathbf{z} = \arg\min_{\mathbf{z}} [\sum_i w_i (y_i - z_i)^2 + \lambda \sum_i (\Delta z_i)^2].$$

We can set the weight parameter w_i^t in the following way: if $y_i > z_i^{t-1}$, $w_i^t = p$; if $y_i < z_i^{t-1}$, $w_i^t = 1 - p$, with $0 < p \ll 1$. Here, i is the number index of channels, and t is the number index of iterations. This formulation assigns considerably greater weights to negative deviations from the baseline compared to positive ones. The parameter p embedded within the Asls algorithm defines the extent of asymmetry within the weight function. A heightened p value emphasizes fitting positive peaks within the spectrum. Conversely, a reduced p value directs the algorithm's focus toward fitting negative peaks in the spectrum. When p equals 0.5, the algorithm effectively detrends along each data point. The iterative process converges nicely after approximately 10 iterations when initiated with appropriate conditions. This is due to the convex nature of the function Q in relation to $\mathbf{z}$ [18].

The λ parameter controls the smoothness of the fitted baseline. We can imagine it as a force applied on both sides of a rope. When the λ parameter is higher, the force is stronger, leading to the fitted curve's smoothness. In contrast, a small parameter will result in a rough baseline. From Fig.3, we can see how the results vary with different p and λ.

To adapt to the Matlab environment, we introduce matrices and vectors and use a difference matrix to substitute the icon Δ. For example, for a vector with 5 components, its difference matrix $\mathbf{D}$ could be:

$$\mathbf{D} = \begin{bmatrix} -1 & 1 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 \\ 0 & 0 & -1 & 1 & 0 \\ 0 & 0 & 0 & -1 & 1 \end{bmatrix}$$

With matrices and vectors introduced, we calculate the partial derivatives of Q to $\mathbf{z}$:

$$\frac{\partial Q}{\partial \mathbf{z}^T} = -2\mathbf{W}(\mathbf{y} - \mathbf{z}) + 2\lambda \mathbf{D}^\mathbf{T}\mathbf{D}\mathbf{z}.$$

At the optimized result, this derivative is 0. We can gain the following matrix linear equation:

$$(\mathbf{W} + \lambda \mathbf{D}^\mathbf{T}\mathbf{D})\mathbf{z} = \mathbf{W}\mathbf{y}.$$

Here $\mathbf{W}$ is the weight parameter matrix [17].

When sparse matrices are being used, the equation above can be easily solved fast. It smoothes even a large amount of data up to 100 000 values in a few seconds on a common personal computer. This algorithm can also be used to interpolate or extrapolate missing data, fit both positive and negative peaks by altering the p parameter, and have a good continuous control of the smoothness of line through the λ parameter. However, when applying this algorithm in spectra research, we should remember that it bases on the assumption that the potential physics mechanism leads to a smooth baseline.

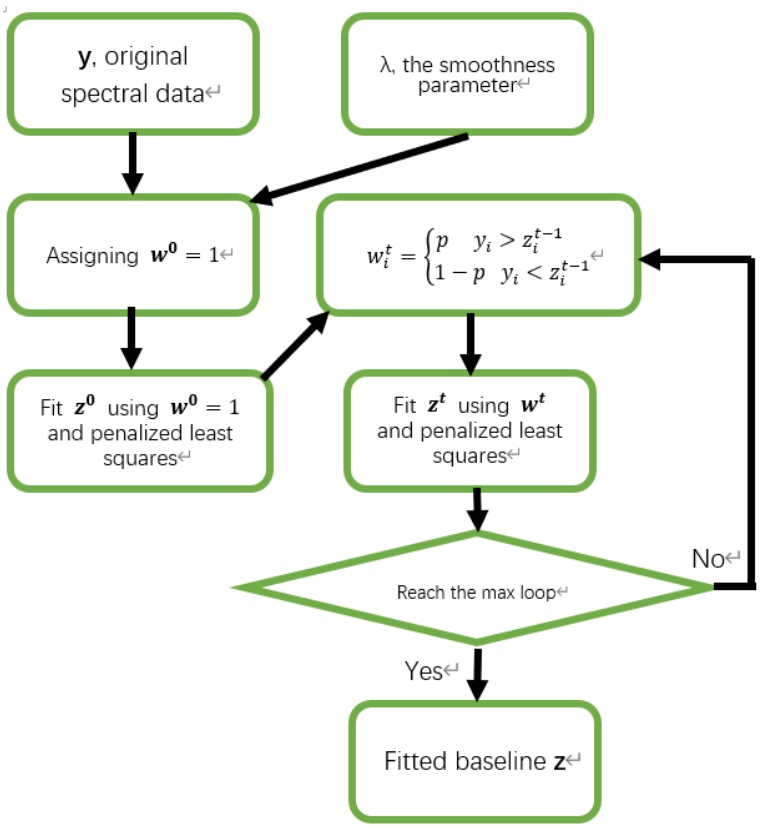
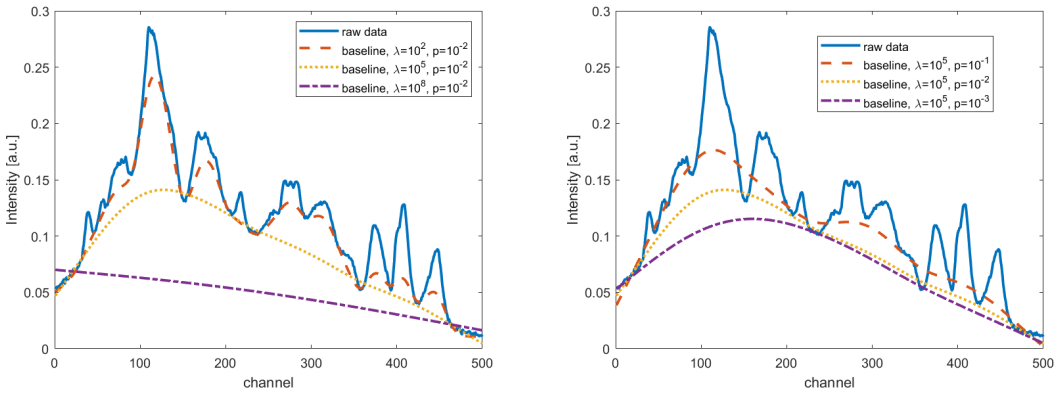


Figure 2: Flow chart describing the framework of the Asls algorithm [1].



(a) (b)

Figure 3: One correction result for the Asls algorithm, based on the synthetic data. (a)The raw data and baselines calculated with λ changing, $\lambda = 10^2$ (red), 10^5 (yellow), 10^8 (purple), $p = 10^{-2}$; (b)calculated with p changing, $\lambda = 10^5$, $p = 10^{-1}$ (red), 10^{-2} (yellow), 10^{-3} (purple).

2.2.2 Adaptive iteratively reweighted penalized least squares algorithm

While the Asymmetric Least Squares (Asls) algorithm offers significant utility, it also presents certain limitations. One notable drawback is the need to optimize two distinct parameters, which might lead to redundancy in efforts and outcomes. Additionally, the asymmetry parameter (referred to as the "p" parameter) remains uniform across all baseline region points, even though its value may vary. In response to this limitation, an innovative approach was introduced by Zhang et al. in 2010 – the Adaptive Iteratively Reweighted Penalized Least Squares (airPLS) algorithm. This algorithm seeks to harness the variances between the previously fitted baseline and the original signals. It achieves this by dynamically and iteratively adjusting the weight parameter, thereby enhancing adaptability and precision in baseline correction [2].

This algorithm extended the Asls algorithm mainly on the weight parameter updating rule and the termination criterion. After inputting the spectral data and the smoothness parameter, the next step is to set the initial weight value $W \equiv 1$, then w of t^{th} iteration for i^{th} sample point is updated adaptively according to the following rules:

$$w_i^t = \begin{cases} 0 & y_i > z_i^{t-1} \\ e^{\frac{t(y_i - z_i^{t-1})}{|\mathbf{d}^t|}} & y_i < z_i^{t-1} \end{cases}$$

Here vector $\mathbf{d}^t$ consists of negative elements of the differences between the original data $\mathbf{y}$ and the last iteration result $\mathbf{z}^{t-1}$ [2].

To jump out of the loop, we can either set the maximum iteration time or the termination criterion. In [2], the criterion is defined as:

$$|\mathbf{d}^t| < 0.001 \times |\mathbf{y}|.$$

This design makes sure that the fitted baseline is under the spectra in most cases.

Compared with the Asls algorithm, the airPLS has a better performance when it comes to spectrums with smaller peaks. And obviously, the airPLS algorithm spends less time and iterations to reach a satisfactory convergence result than the Asls algorithm does [2]. Fig.4 shows the flow chart of the airPLS algorithm. Fig.4 and Fig.5 show the flow chart of the airPLS algorithm and how the result varies with λ.

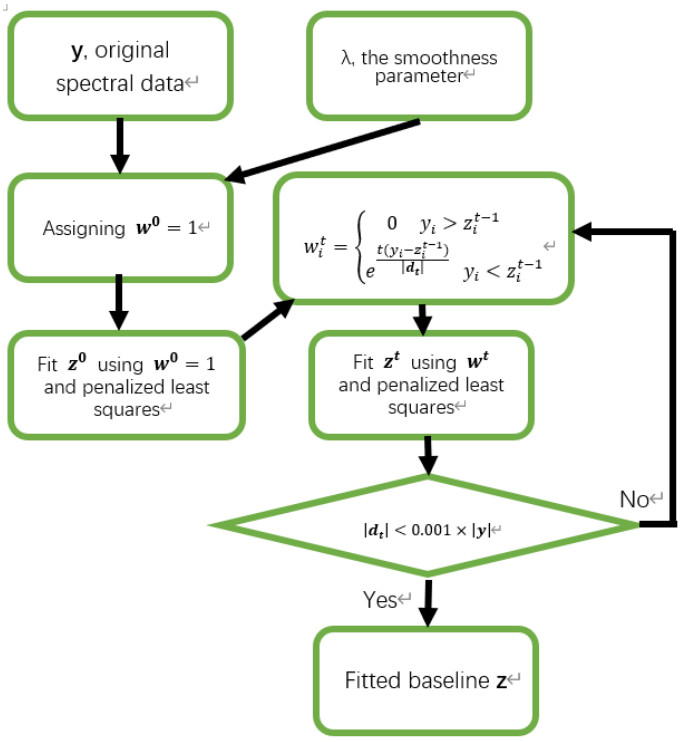


Figure 4: Flow chart describing the framework of the airPLS algorithm [2].

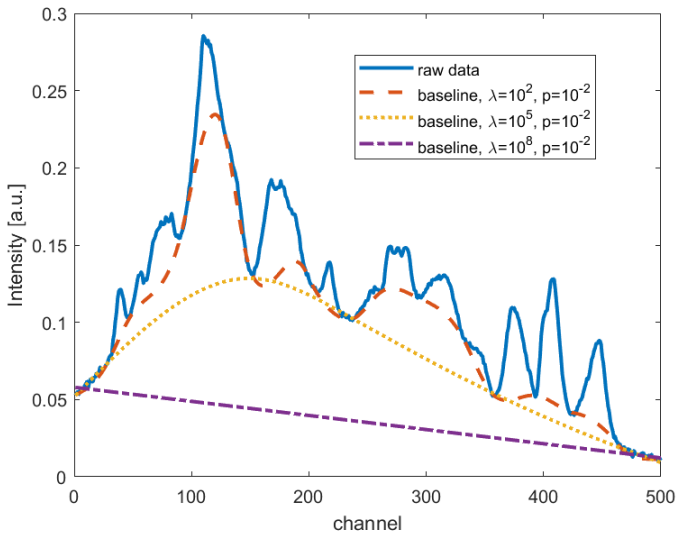


Figure 5: One correction result for the airPLS algorithm, based on the synthetic data. The raw data and baselines calculated with with λ changing, $\lambda = 10^2$ (red), 10^5 (yellow), 10^8 (purple).

2.2.3 Asymmetrically reweighted penalized least squares algorithm

While the Asymmetric Least Squares (Asls) algorithm and the airPLS algorithm exhibit effectiveness in scenarios with minimal noise, their performance diminishes significantly in the presence of high noise levels. This is primarily attributed to the inherent asymmetry of the weight parameter both above and below the computed baseline. Consequently, the results generated tend to fall below the noise band, thus leading to an underestimation of the actual baseline, especially in situations characterized by substantial noise.

Addressing this limitation, in 2015, Baek et al. introduced a novel approach that capitalizes on a distinct weighting scheme. This scheme involves the utilization of the generalized logistic function to counteract the adverse effects of high noise levels. This innovative method presents a promising solution to rectify the aforementioned challenge, enabling more accurate and reliable baseline corrections even in noisy conditions [3].

When using the Asls or the airPLS algorithm to do the baseline correction, if the amplitude of the noise is relatively high, those data points under the baseline from the former iteration will probably be eliminated because of their high weight. While the ones above the baseline will be reserved because their weights are 0. Therefore, the result in the noise band is overestimated. Therefore, giving equal or similar weight to either case is desirable as additive noise is equally distributed along a baseline [3]. To this end, a new weighting scheme based on the generalized logistic function is proposed in the following way:

$$w_i^t = f(y_i - z_i^{t-1}, m_d, \sigma_d).$$

Here m_d and σ_d are the average and the standard deviation of d_t respectively. The vector d_t is defined in section 2.2.2. The logistic function is defined as follows [3]:

$$f(d, m, \sigma) = [1 + e^{2(d+m-2\sigma)/\sigma}]^{-1}.$$

If the difference between the data points and the baseline is σ, the weight is approaching 1, and the chance of the point being eliminated is higher. The amplitude of the random noise signal seldom exceeds 3σ. Therefore, for the point deviates over 3σ, their weights approach 0 and can be remained as signal peaks. When the relative difference between the new weight vector and the last one $|\mathbf{w}^t - \mathbf{w}^{t-1}|/|\mathbf{w}^t|$ is smaller than a specific ratio, the iteration is terminated. Fig.6 and Fig.7 show the flow chart of the arPLS algorithm and how the result varies with λ.

Compared with the Asls and the airPLS algorithm, the arPLS algorithm will perform better when the noise level is high. Another great advantage is that this algorithm is not sensitive to the choice of the smoothness parameter [3]. However, with the foregoing termination criterion, the fall into endless iteration is often observed.

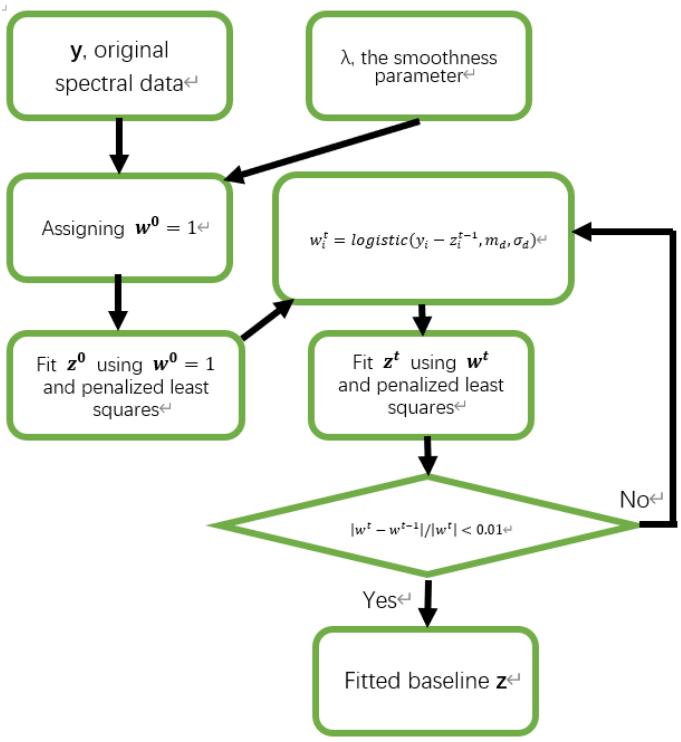


Figure 6: Flow chart describing the framework of the arPLS algorithm [3].

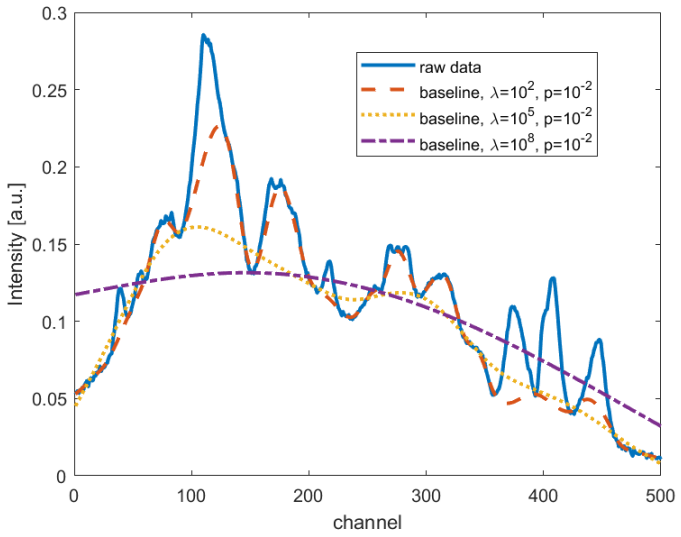


Figure 7: One correction result for the arPLS algorithm, based on the synthetic data. The raw data and baselines calculated with with λ changing, $\lambda = 10^2$ (red), 10^5 (yellow), 10^8 (purple).

2.2.4 Multiple spectra baseline correction algorithm

The acceleration of baseline correction computation was a notable development in 2010 when Peng et al. introduced the Multiple Spectra Baseline Correction algorithm (MSBC). This innovative approach builds upon the Asls method [4] and is rooted in the idea that when a sample is measured repeatedly, the resulting baselines exhibit minor variations. This assumption is especially effective in scenarios involving multiple spectra and hyperspectra datasets.

The MSBC algorithm leverages this assumption to enhance its computational efficiency. To achieve this, Peng introduced a novel term into the optimization function. This added term serves the purpose of penalizing differences observed between corrected spectra. This strategic addition ensures that the MSBC algorithm generates notably consistent baselines across different measurements. Consequently, the algorithm enhances its ability to swiftly and accurately correct baselines in a variety of scenarios, thus contributing to more streamlined data processing.

Assume there are m spectra and n wavelengths (channels), the optimization problem can be written as:

$$\begin{aligned}\mathbf{z}_k = \arg\min_{\mathbf{z}_k}[&\sum_{u=1}^{m}\sum_{v=1,v>u}^{m}||(\mathbf{y}_u-\mathbf{z}_u)-(\mathbf{y}_v-\mathbf{z}_v)||^2\\ &+\mu\sum_{u=1}^{m}(\mathbf{y}_u-\mathbf{z}_u)^T\mathbf{W}_u(\mathbf{y}_u-\mathbf{z}_u)+\sum_{u=1}^{m}\lambda_u(\Delta\mathbf{z}_u)^2].\end{aligned} \tag{2.1}$$

Here $\mathbf{z}_k$ is the k^{th} baseline to be estimated, $\mathbf{y}_u$ is the u^{th} column vectors including n channels, and $\mathbf{W}_u$ is the u^{th} $n\times n$ diagonal weight matrix with diagonal elements $w_i(j)=p$ if $\mathbf{y}_i(j)>\mathbf{z}_i(j)$ and $w_i(j)=1-p$ otherwise. The equation above can be turned into the equivalent form:

$$\begin{aligned}\mathbf{z}_k = \arg\min_{\mathbf{z}_k}[&\textstyle\sum_{u=1}^{m}||(\mathbf{y}_u-\mathbf{z}_u)-\frac{1}{m}\sum_{v=1}^{m}(\mathbf{y}_v-\mathbf{z}_v)||^2\\ &+m\mu\sum_{u=1}^{m}(\mathbf{y}_u-\mathbf{z}_u)^T\mathbf{W}_u(\mathbf{y}_u-\mathbf{z}_u)+\sum_{u=1}^{m}m\lambda_u(\Delta\mathbf{z}_u)^2].\end{aligned} \tag{2.2}$$

The proof of this equivalent form is beyond the scope of this paper. Anyone interested can consult the reference [4]. From the equation above we can see that the spectra corrected by the MSBC method must be close to the average corrected one [4]. The equivalence can greatly accelerate the calculation of baselines.

We should remember that up to now the method is based on the assumption that all

spectra share the same baseline. However, this assumption can be relaxed by introducing a relaxation factor a. The optimization problem thus can be extended as:

$$\begin{aligned}(\mathbf{z}_k, a_k) = \arg\min_{\mathbf{z}_k, a_k} [\textstyle\sum_{u=1}^{m} ||(\mathbf{y}_u - \mathbf{z}_u) - a_k \frac{1}{m} \sum_{v=1}^{m} (\mathbf{y}_v - \mathbf{z}_v)||^2 \\ + m\mu \sum_{u=1}^{m} (\mathbf{y}_u - \mathbf{z}_u)^T \mathbf{W}_u (\mathbf{y}_u - \mathbf{z}_u) + \sum_{u=1}^{m} m\lambda_u (\Delta \mathbf{z}_u)^2].\end{aligned} \tag{2.3}$$

Where the relaxation factor a_k demonstrates how k^{th} corrected spectra are similar to the average one. If we denote $\theta = \frac{1}{m} \sum_{v=1}^{m} (\mathbf{y}_v - \mathbf{z}_v)$, it can be proved that the relaxation factor a_k can be calculated as:

$$a_k = (\theta^T \theta)^{-1} \theta^T (\mathbf{y}_v - \mathbf{z}_v)$$

and if we denote $\gamma_k = a_k(2 - a_k)$, the solution of the baseline $\mathbf{z}_k$ is [4]:

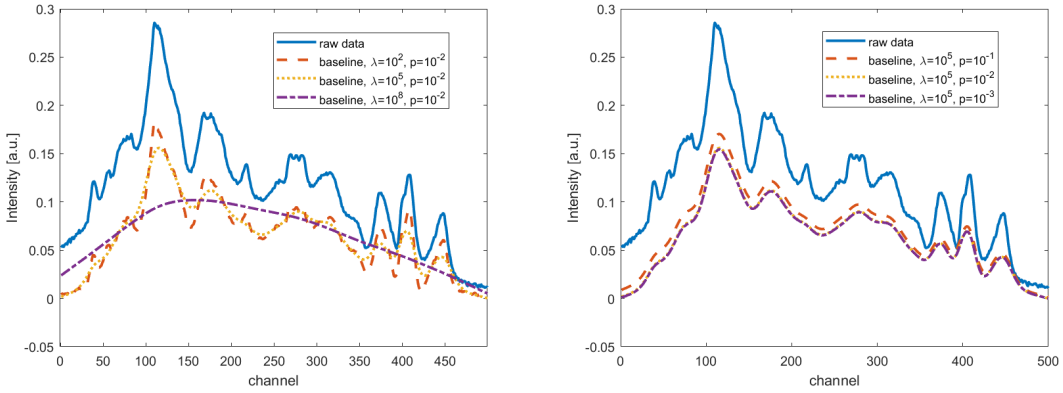
$$\begin{aligned}\mathbf{z}_k = [(m - \gamma_k)\mathbf{E} + m\mu \mathbf{W}_k + m\lambda_k \mathbf{D}^\mathbf{T} \mathbf{D}]^{-1} \\ [(m - \gamma_k)\mathbf{y}_k - \gamma_k \sum_{i=1, i \neq k}^{m} (\mathbf{y}_i - \mathbf{z}_i) + m\mu \mathbf{W}_k \mathbf{y}_k].\end{aligned} \tag{2.4}$$

Here $\mathbf{D}$ is the differential matrix, and $\mathbf{E}$ is the n^{th} order identity matrix.

In most cases, after some iterations, all the relaxation factors will approach 1, which means that all fitted baselines will be close to each other. The advantage of this algorithm is that every newly fitted baseline can provide information for other ones to be estimated to make them close to the same potential baseline [4]. Nevertheless, if the samples in the measured area share two different potential baselines, probably this algorithm will have a bad performance due to its penalty on the difference between baselines. Therefore, in contrast, we need a new algorithm to exaggerate such differences when dealing with this case. Such a method is still waiting to be invented.

2.3 Computational methods

After baseline corrections, some computational methods need to be implemented to measure the performance of corresponding correction algorithms. For example, the principal component analysis can be done on the corrected spectra to describe them with only a few principal vectors, we can visualize their coefficients by scattering plots to find possible regularity. The K-means clustering method can be used to separate the measured area into several pixie clusters according to their individual spectra, to compare with the original sample or the labels. And the confusion matrix or Silhouette plot can quantify the



(a) (b)

Figure 8: One correction result for the MSBC algorithm, based on the synthetic data. (a)The raw data and baselines calculated with with λ changing, $\lambda = 10^2$ (red), 10^5 (yellow), 10^8 (purple), $p = 10^{-2}$; (b)calculated with p changing, $\lambda = 10^5$, $p = 10^{-1}$ (red), 10^{-2} (yellow), 10^{-3} (purple).

accuracy or quality of the clusters, assessing the effectiveness of the corrected spectra. In this part, we are going to discuss the principles of these methods.

2.3.1 Principal component analysis

The principal component analysis (PCA) method is used to reduce the data dimension able to extract the features of the raw data. It is commonly used in data analysis to extract relevant information from large datasets. It can be applied to various types of data, including images, spectra, and time-series data. A measured value can be multiple-dimensional. If we use the PCA method, we can take out some vectors representing the measured data most efficiently, and calculate their corresponding coefficients. Such vectors are called the principal components of the data. It is a powerful tool for exploratory data analysis and can help to reveal underlying patterns and relationships in complex datasets.

The principle of PCA is to find a new coordinate in the n-dimensional space (the order index n is chosen by users, and usually is smaller than the dimension of the original data) where the new axes are aligned with the directions of maximum variance in the data. Because each data point projected into the axis with the larger variance brings more information. For a more detailed elaboration, please consult the reference [19].

2.3.2 K-means clustering method

K-means clustering is a popular unsupervised machine learning algorithm used for clustering data points into groups based on the Euclidean distance. The algorithm partitions a

dataset into k clusters where each data point belongs to the cluster with the nearest mean. The value of k is a hyperparameter that must be specified by the user.

The algorithm works by randomly selecting k initial cluster centers, typically called centroids, from the data points. Each data point is then assigned to the cluster whose centroid is closest to it, usually based on Euclidean distance. The mean of each cluster is then calculated, and the centroid is moved to this new mean. This process of reassigning data points and recalculating the mean of each cluster is repeated until convergence, which occurs when the assignment of data points to clusters no longer changes.

2.3.3 Confusion matrix

A confusion matrix is a table used in classification problems to evaluate the performance of a clustering method. It summarizes the algorithm's performance by showing the number of correct and incorrect predictions made by the algorithm, as well as the types of errors it is making.

If the variables are binary, there are only four elements in the matrix: true positives (TP), false positives (FP), false negatives (FN), and true negatives (TN). TP stands for the cases where the model predicted the positive class and the actual class was also positive. The accuracy of clusters, or say, the fraction of correct predictions, is (TP+TN)/(TP+FN+TN+FP) [20].

If there are n categories, the confusion matrix will be n^{th} order and contain n^2 elements. Elements on the diagonal are the numbers of correct predictions. While the others mean how many data points belonging to a label are sorted to other wrong clusters.

If we are not sure about the mapping relation between the labels and the cluster results, we can measure the accuracy by pair error. For a multiple-variable confusion matrix, we first sum up all the elements in the same label and calculate the number of pairs they can form within the same label by $n \times (n-1)/2$, then sum up the numbers. After that, we calculate every number of pairs in the matrix elements individually and sum up all the results. The cluster accuracy is the ratio between the two numbers.

2.3.4 Silhouette coefficient and Silhouette plot

The Silhouette coefficient is a measure of how similar an observation is to its own cluster compared to other clusters, and ranges from -1 to 1, with higher values indicating better clustering. The formula to calculate the Silhouette coefficient for a single data point i is as follows:

$$s_i = \frac{b_i - a_i}{max(a_i, b_i)}$$

where a_i is the average Euclidean distance between data point i and all other data points within the same cluster, and b_i is the smallest average Euclidean distance of i to any other cluster, of which i is not a member. The denominator takes the maximum of a_i and b_i to ensure that the coefficient falls between -1 and 1. A coefficient of 1 indicates that the object is well-matched to its own cluster and poorly matched to neighboring clusters. A coefficient of 0 indicates that the object is poorly matched to its own cluster, while a coefficient of -1 indicates that the object is better matched to neighboring clusters than to its own cluster.

The Silhouette plot is a visualization tool that displays the Silhouette coefficient for each data point in a cluster analysis. It is used to assess the quality of clustering and to identify the optimal number of clusters in the data. The silhouette plot displays a horizontal bar chart for each observation in the data, where the length of the bar represents the Silhouette coefficient for that observation. The silhouette plot is often used in combination with other clustering evaluation techniques to help identify the optimal number of clusters.

3 Datasets and the experimental environment

The samples contain synthetic data and real data. The synthetic data is made up by a Python 3 program. This Python program produces a file including 1024 synthetic Fourier-transform infrared (FTIR) spectra with 20 random peak positions, amplitudes, noises, and their corresponding baselines with 8 random peaks to imitate real vibrational spectroscopy data. Each spectrum has 500 wavelength channels. These spectra are three classes with labels, where each class is generated by mixing virtual chemicals with different proportions so that different amplitudes can be generated in the characteristic frequency band of the chemicals. And labels are used for checking the accuracy of unsupervised learning. The individual spectra deviate a little bit from the class average, and on top of this, there is relative and absolute noise.

The real data is some files containing 4096 individual FTIR spectra of fungal hyphae consisting of 1530 wavelength channels. The image of the sample fungal picked from a wavelength channel in random is shown in Fig.6.

The programs are written in-house in Matlab Version R2020b (MathWorks, Inc.) and run on a personal computer with an Intel Core i7 processor, an Intel iRISXe graphics processor, and a Windows XP operating system.

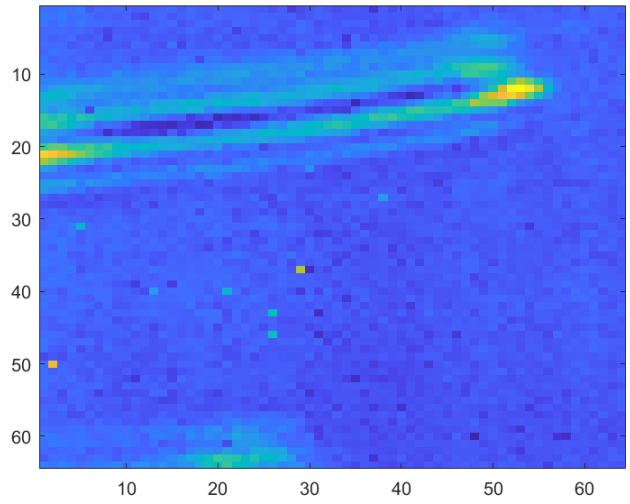


Figure 9: Image of a fungal sample (64×64 pixels) picked from a wavelength channel in random. The light intensity is higher in the yellowish region and lower in the bluish region. The fungus sample is located in the upper part of the image, and in the lower part there is a spot of light due to background lighting.

4 Results

We assume that, if the best baseline is found, the corrected spectra should be the most characteristic. It means each pixel can be sorted into different clusters most easily and clearly. Based on this assumption, a computational scheme to quantify the performance of these algorithms is developed.

For synthetic data, the computational scheme first is to do the principal component analysis on the corrected spectrum given by different methods and parameters. This step is supposed to calculate their principal coefficients. PCA was also done to the original spectrum to make a comparison. After that, with the K-means clustering method implemented, these corrected spectra were divided into different clusters for further analysis according to their coefficients of the components. Then, we will check the mapping relation between the labels and the clusters by the confusion matrix, and calculate their cluster accuracies to find the best λ parameter for clustering (accuracy-optimal point). Finally, the average Silhouette values to quantify the clustering performance of these baseline correction algorithms will be calculated, to see whether they are higher than that from spectrum without correction. A plot showing how the error (the difference between the given baselines and the fitted baselines) changes with λ was also drawn to find the error-optimized point. The average Silhouette value at this point was also calculated to make a comparison. Some other 2D or 3D scattering plots at optimized parameters are shown to give an intuition on how baseline corrections help cluster.

For real data, we will also try to figure out how the clustering ability changes with the smoothness parameter λ, to find the best working parameters for each method. Because there are no labels in the real data for us to check the correctness of the result, after PCA and the K-means method, we can only try to measure the performance of these algorithms by the average Silhouette value. Due to the reason that the value is affected by both number of clusters n and λ parameter, an attempt was made to find the point (n, λ) that maximizes the average Silhouette value (Silhouette optimal point). The optimized Silhouette value was compared with that from corrected spectra without correction. Also, we plotted some images extracted from the clustered results at the Silhouette optimal point, to make a comparison with the original one and check the effectiveness of these algorithms intuitively.

4.1 Synthetic data

The error between given baselines and results is calculated and integrated over 20 synthetic files, and the parameter space of λ ranges from 10 to 10^{10}. As shown in Fig.10(a), the errors from these four methods share almost the same tendency. They decrease with the logarithm of the smoothness parameter λ increasing. They come to the minimum point and climb up again. AirPLS, which seems to have the best performance, reached the lowest error of 5.39$\pm$0.015 when the λ is 10^6. The second one is Asls, the minimum error is 5.56$\pm$0.010 at $\lambda = 10^{7.5}$. MSBC and arPLS come to a minimum of 5.52$\pm$0.006 and 8.42$\pm$0.010, at $\lambda = 10^{9.5}$ and $\lambda = 10^6$ respectively.

Fig.10(b) shows how the cluster accuracy calculated from the corrected spectra changes with the λ parameter. The spectrum corrected by the Asls algorithm is much easier to be clustered almost perfectly, especially when $\log \lambda$ is between 3 and 5, then the accuracy drops down drastically if continue increasing the smoothness parameter. The accuracy from airPLS has a similar trend, but the average level is lower. The accuracy calculated from MSBC is relatively stable until $\log \lambda$ reaches 6, then it experiences a sudden jump. The accuracy from arPLS stays unstable across the whole parameter space. For Asls, the accuracy-optimal point is $\lambda = 10^5$, giving an almost max accuracy with a variance that can be neglected. The second one is arPLS, giving an accuracy of 0.88$\pm$0.01 at $\lambda = 10^5$. The followings are MSBC and airPLS, their highest accuracy are 0.76$\pm$0.05 and 0.66$\pm$0.03 at $\lambda = 10^9$ and 10^3 respectively. Fig.10(c)(d) are plots of a randomly selected raw spectrum, its given, and fitted baselines, the fitted baselines are calculated by these methods at their error- and accuracy-optimal points. From these plots, we can see that the fitted baselines tend to imitate the given baseline at error-optimal points, but at accuracy-optimal points, the results tend to fit the spectral peaks. The influence of these behaviors to the clustering ability of the correction algorithms will be shown in the following text. In Fig.10(b), the tendency of the accuracy-λ curve given by MSBC is quite different from that given by three other methods, when λ is high it performs better, we will also talk about the reason latter.

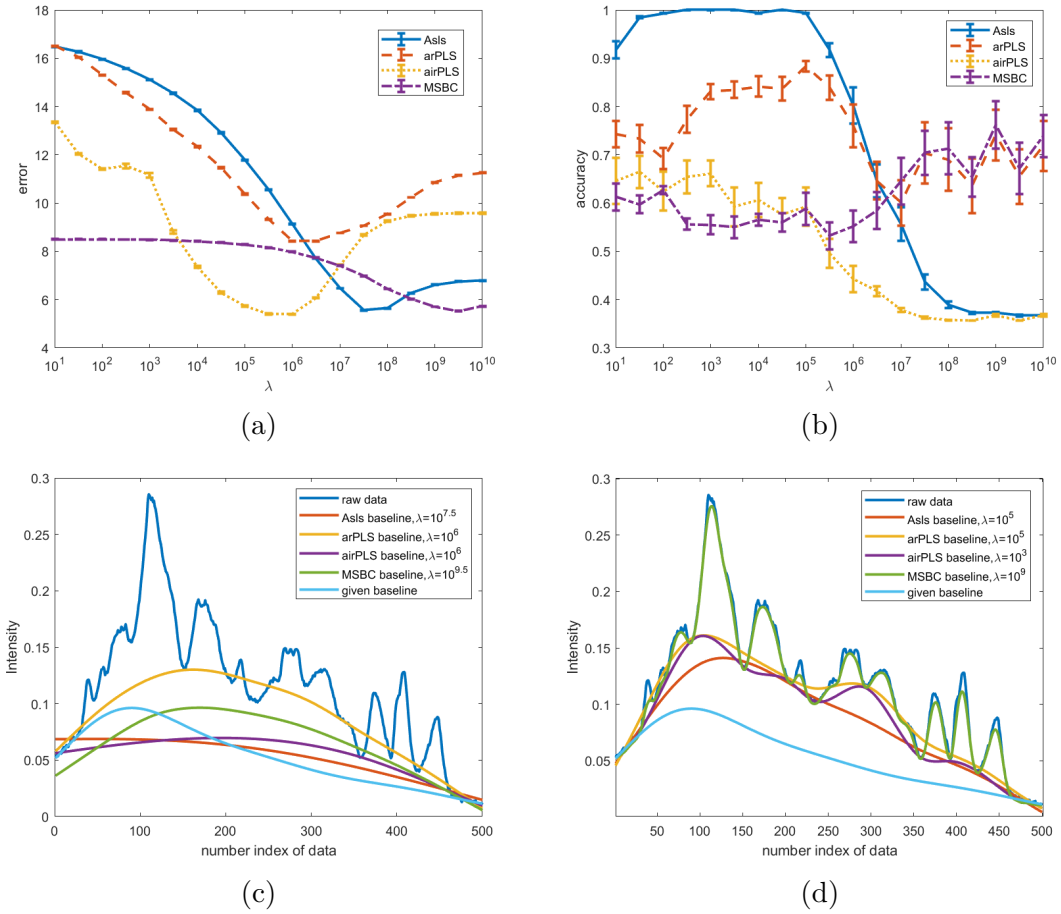


Figure 10: (a) The difference between given baselines and calculated baselines from the four algorithms, and (b) the clustering accuracy vs λ, tested on 20 synthetic 3-cluster datafiles, the p parameter is set as 0.1 for Asls and MSBC. The blue, red, yellow and purple curves are given by Asls, arPLS, airPLS, and MSBC respectively. The error bar is calculated over results from these 20 datafiles. The error optimal points for the four algorithms are $10^{7.5}, 10^6, 10^6$ and $10^{9.5}$ for Asls, arPLS, airPLS, and MSBC respectively; the accuracy optimal points for the four algorithms are $10^5, 10^5, 10^3$ and 10^9 for Asls, arPLS, airPLS, and MSBC respectively. The error bars indicate the variance over the 20 datafiles (c) The plot of a randomly selected raw datafile, the given baseline, and calculated baselines from the four methods at their error optimal points, and (d) accuracy optimal points. The blue and light blue curves are the raw data and the given baseline. The red, yellow, purple, and green lines are given by Asls, arPLS, airPLS, and MSBC respectively. Calculated baselines at error optimal points tend to be smoother and similar to the given baseline; at accuracy optimal points, they tend to fit the peaks in the raw data.

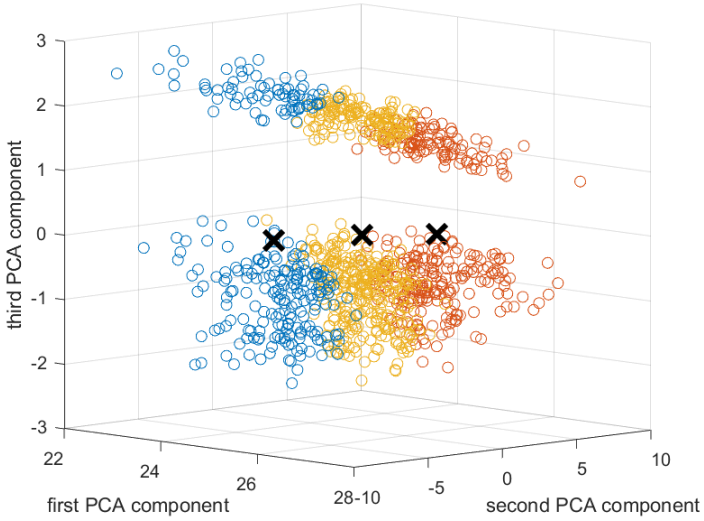
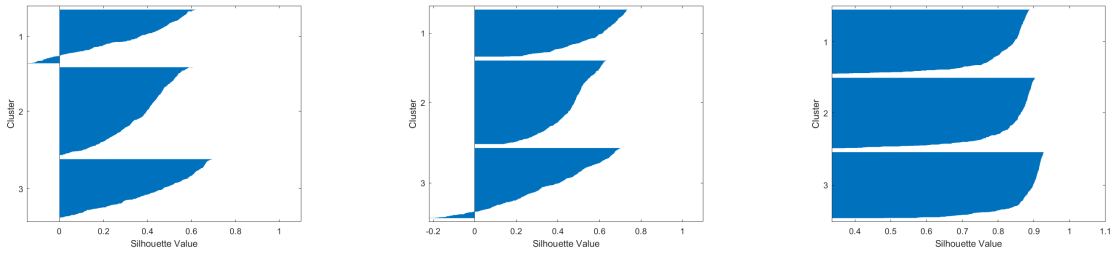


Figure 11: 3D scatter plot of the PCA results. Each point stands for a spectrum. The three main axes are the first three principal components. The different colors express different clusters given by the K-means method. The spectrum used here is the same as the one used in Fig.10 (c)(d). Obviously, this result is not what we want because the clusters are supposed to be divided horizontally, while the result given by the K-means is vertical. Without correction, two individual clusters are mixed together in the lower part, making them indistinguishable.



(a) (b) (c)

Figure 12: Silhouette plot for cluster results from (a) spectrum without baseline correction, (b) spectrum corrected by Asls at error-optimal point ($\lambda = 10^{7.5}$), and (c) corrected by Asls at accuracy-optimal point ($\lambda = 10^{5}$). The spectrum used here is the same as the one used in Fig.10 (c)(d). The negative values in the (a) and (b) subplot imply the cluster result's ineffectiveness. The fullness of the (c) subplot proves that Asls do better at the accuracy-optimal point.

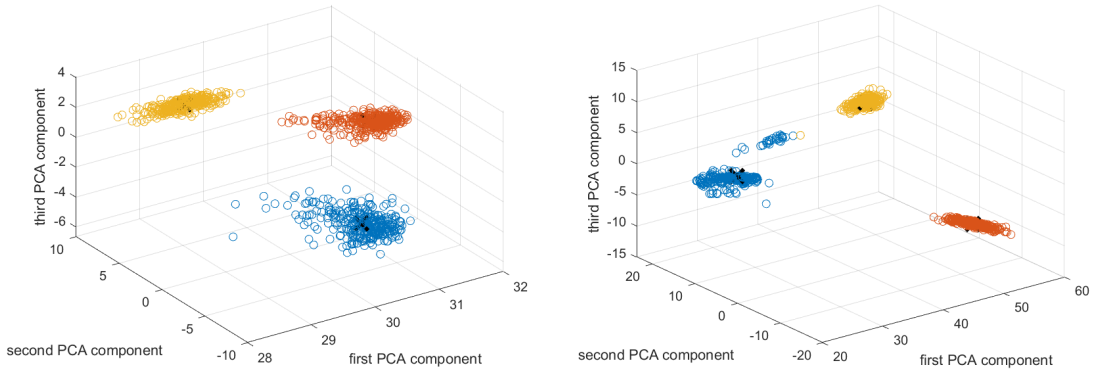
To show how the baseline correction helps to cluster, we can do the principal component analysis of the original data and the corrected one, and make a 3D scattering plot, where the axes are the coefficients of principal components, to make a comparison and demonstrate how baseline correction extracts spectral information. Fig.11 shows a scattering plot of a 3-cluster synthetic spectral datafile without any baseline corrections. Different colors stand for cluster results divided by the K-means method. The black cross is the mass center of every cluster. The K-means method fails to sort the PCA results from the raw data with no corrections. The clusters should be divided horizontally, while the resulting border is vertical. Because without baseline correction, two clusters mix together in the lower part, making them indistinguishable. Its corresponding Silhouette values are certainly low, some are even negative, which are not appreciated in classification and can be seen in Fig.12(a).

The scatter plots and the Silhouette plots of the PCA result after Asls baseline corrections at error- and accuracy-optimal points are demonstrated in Fig.12(b)(c), Fig.13, and Fig.14. As shown in Fig.13, the K-means method succeeded to divide the PCA results from the spectra corrected by these four methods at their accuracy-optimal points. Three clusters got separated together with their mass centers. However, for results from the spectra corrected at their error-optimal points, the division does not go well, after correction, two clusters of the three still tangle together, making the K-means method fail. These results are consistent with the phenomenon we saw in Fig.10(a)(b), that the accuracy value from Asls stays at a high level when the error is still high. These visual properties can be quantified by the Silhouette plots in Fig.12, the fullness of the plots of Asls clustered results proves that compared with the error-optimal point, they do better in extracting information to help classify spectra at the accuracy-optimal point. This conclusion can also be verified by the Silhouette values listed in Table 1, at accuracy optimal points the values are much higher than that at error-optimal points.

Table 1: The average Silhouette value from spectra corrected by different methods at their accuracy- or error-optimal points.

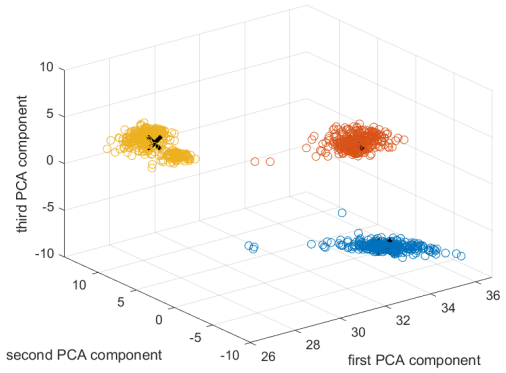
average Silhouette value	no correction	Asls	arPLS	airPLS	MSBC
accuracy optimal	0.4343	0.8369	0.9165	0.9266	0.6389
error optimal	0.4343	0.4486	0.5479	0.4804	0.5598

To figure out why the tendency of the accuracy-λ curve for MSBC is different from the other, we should go into the principle of the MSBC algorithm. For other algorithms, at low λ parameters, the calculated baseline will fit the spectral peaks, and exaggerate the spectral features, leading to a high accuracy; at high parameters, the fitted baseline becomes smoother or even a straight line, the original spectrum is almost remained, PCA cluster results mix together and become indistinguishable, leading to a low accuracy. As for MSBC, it will penalize the difference between spectra, this fact means that MSBC tends to compress the cluster. At a low λ point, clusters can be compressed into a narrow space, making the K-means method fail, leading to low accuracy. At a high λ point, MSBC almost

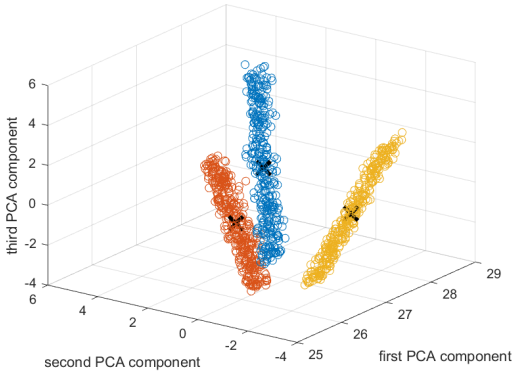


(a)

(b)



(c)



(d)

Figure 13: The scatter plots of the principal coefficient clusters after baseline corrections at accuracy-optimal points. (a) Asls, $\lambda = 10^5$; (b) arPLS, $\lambda = 10^5$; (c) airPLS, $\lambda = 10^3$; (d) MSBC, $\lambda = 10^9$. The dataset used here is the same as the one used in Fig.10 (c)(d). Obviously, at accuracy-optimal points, all four methods sort the data into three different clusters clearly.

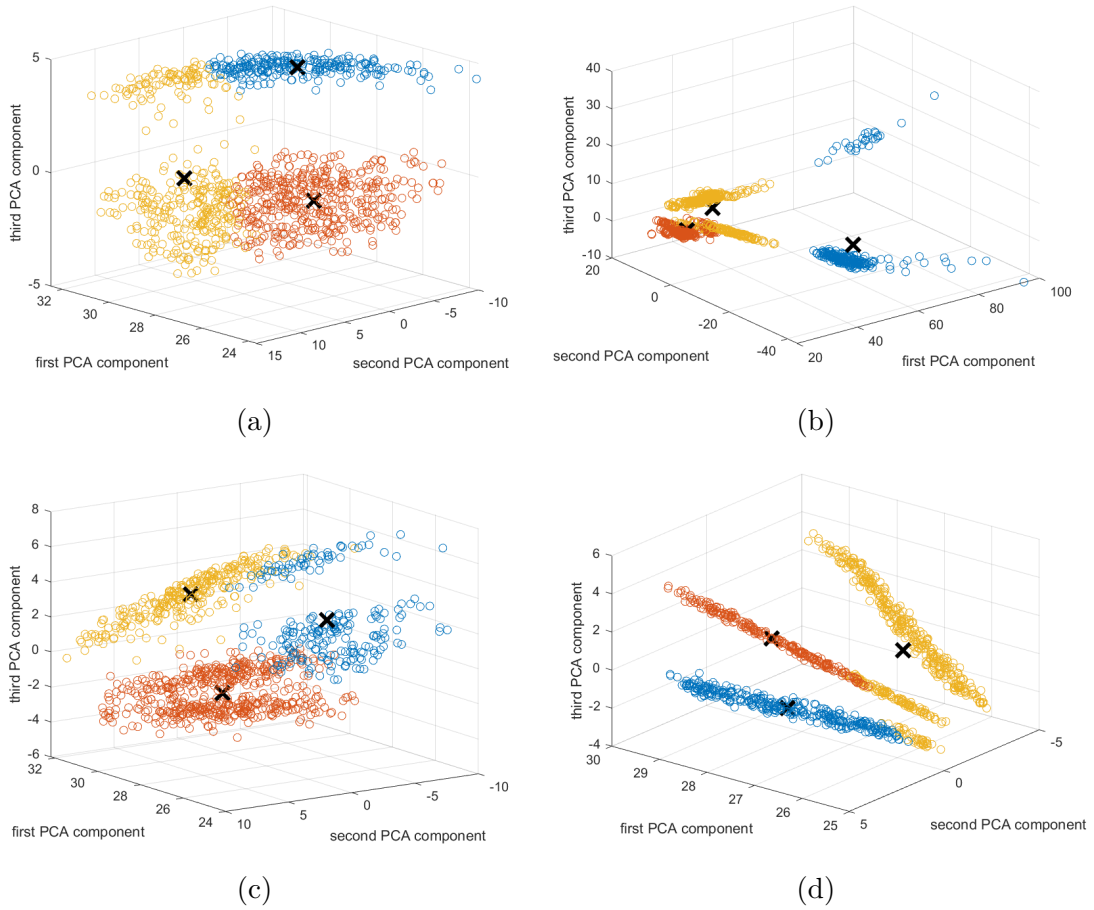


Figure 14: The scatter plots of the principal coefficient clusters after baseline corrections at error-optimal points. (a) Asls, $\lambda = 10^{7.5}$; (b) arPLS, $\lambda = 10^{6}$; (c) airPLS, $\lambda = 10^{6}$; (d) MSBC, $\lambda = 10^{9.5}$. The spectrum used here is the same as the one used in Fig.10 (c)(d). At error-optimal points, the clustered results tend to mix together, implying that at the corresponding parameters, baseline correction algorithms can't sort data as clearly as they did at accuracy-optimal points.

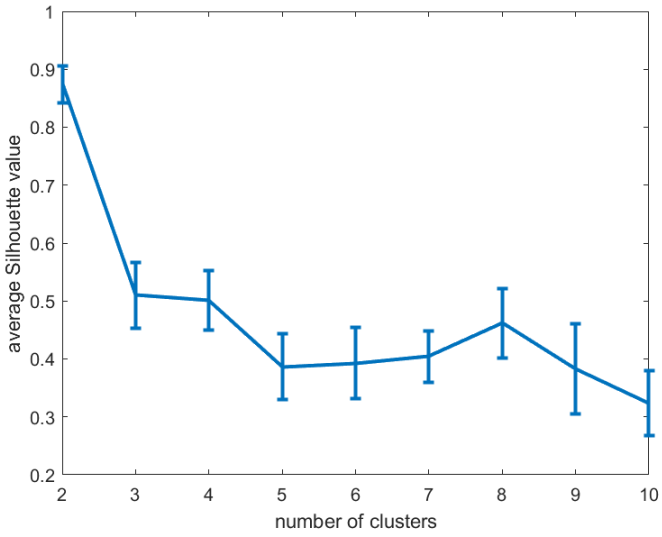


Figure 15: The average Silhouette value for spectral clusters of the real data without baseline correction. It implies that the sample spectrum only has two different clusters. The error bar shows the variance calculated over all 4096 spectra in the real data file.

does nothing to the original data, the cluster structure is remained, making the accuracy a bit higher.

4.2 Real data

For the real data, because there are no labels or number of clusters for reference, we can only use the K-means method with different numbers of clusters, on the PCA result after no correction, to check its possible number of clusters from the average Silhouette value. From Fig.15 we can see, the reasonable number range of clusters is 2-3.

To quantify the performance of these algorithms, we need to find the Silhouette-optimal point (n, λ) to calculate the max Silhouette value and make a comparison with the result without correction. Fig.16 shows four 3D bar subplots demonstrating the relationship between the Silhouette value of clusters and the number of clusters & log of lambda. Both Asls and MSBC perform better when the number of clusters (K parameter) is low and the log of lambda is high. While the plot for arPLS and airPLS look like valleys. For Asls and MSBC, the Silhouette-optimal point is $\lambda = 10^{10}, n = 2$. For arPLS and airPLS, the optimal points are $\lambda = 10^5, n = 2$ and $\lambda = 10^7, n = 2$ respectively. These results imply that there are only two kinds of samples in the original data (fungi and the background).

To make a comparison with the result without correction visually, in Fig.17, we plotted the scatter plots of the principal coefficient clusters after baseline corrections at Silhouette-optimal points and without correction. In Fig.18, the original image of a wavelength and

five corresponding images after being clustered at the Silhouette-optimal point by the K-means method are shown. It's hard to judge from Fig.17 which algorithm performs better, and hard to see if the baseline correction promotes separation, the corrected spectra were divided into a large cluster and a small one. Visually, the airPLS result has the best separation, while the arPLS result was divided into three clusters, which is different from results given by other methods, implying its failure.

For the image result in Fig.18, only a single image is used but it's necessary to check if the spectra images after correction and clustered by the K-means method look reasonable. In the later five subplots, different pixel colors stand for different clusters. The image of cluster results without baseline correction can clearly distinguish the observed fungi, but it's also sensitive to the background illumination. For example, the light spot in the (a) subplot background is reserved in subplot (b) after being clustered. The fungi remain distinguishable in the image of Asls, airPLS, and MSBC, but the light spot is diminished, which means that these baseline correction algorithms can help us focus on the main spectra features. But for the MSBC image, due to the reason that differences between spectra are over-penalized, some fungi pixels are regarded as background. The fungi in the arPLS subplot (d) are blurred in the noise, which verifies our guess in the last paragraph.

To quantify their performance better, the max Silhouette values from different methods at optimal points are listed in Table 2. We can see that except MSBC, neglecting the fact that the arPLS result in Fig.18(d) is unreasonable, their max Silhouette values are higher than the value from the spectra without correction. Therefore, we can conclude that the baseline correction can help us to find spectral features and classify the samples by clustering spectra.

Table 2: The max Silhouette value from spectra corrected by different methods

no correction	Asls	arPLS	airPLS	MSBC
0.8741	0.8930	0.9699	0.9551	0.8661

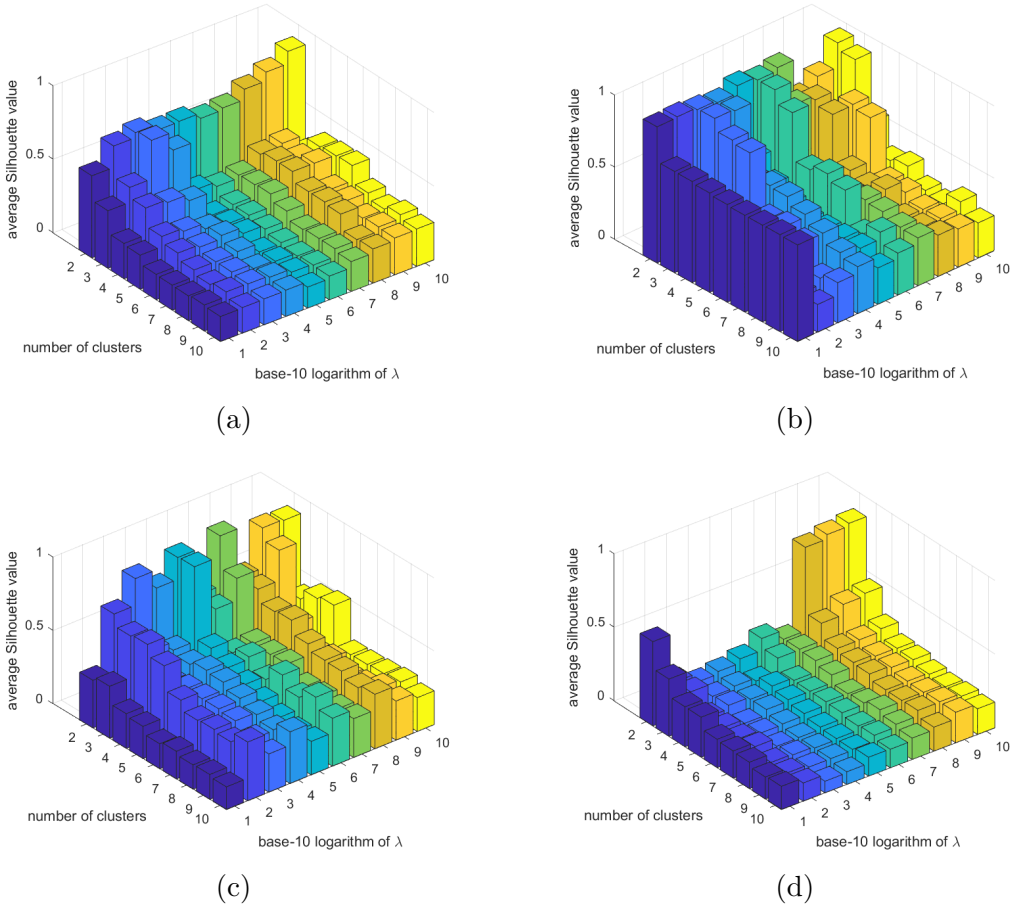


Figure 16: The Silhouette value of clusters vs the number of clusters & log of lambda. (a) Asls; (b) arPLS; (c) airPLS; (d) MSBC. The data file used here is the fungi spectral image.

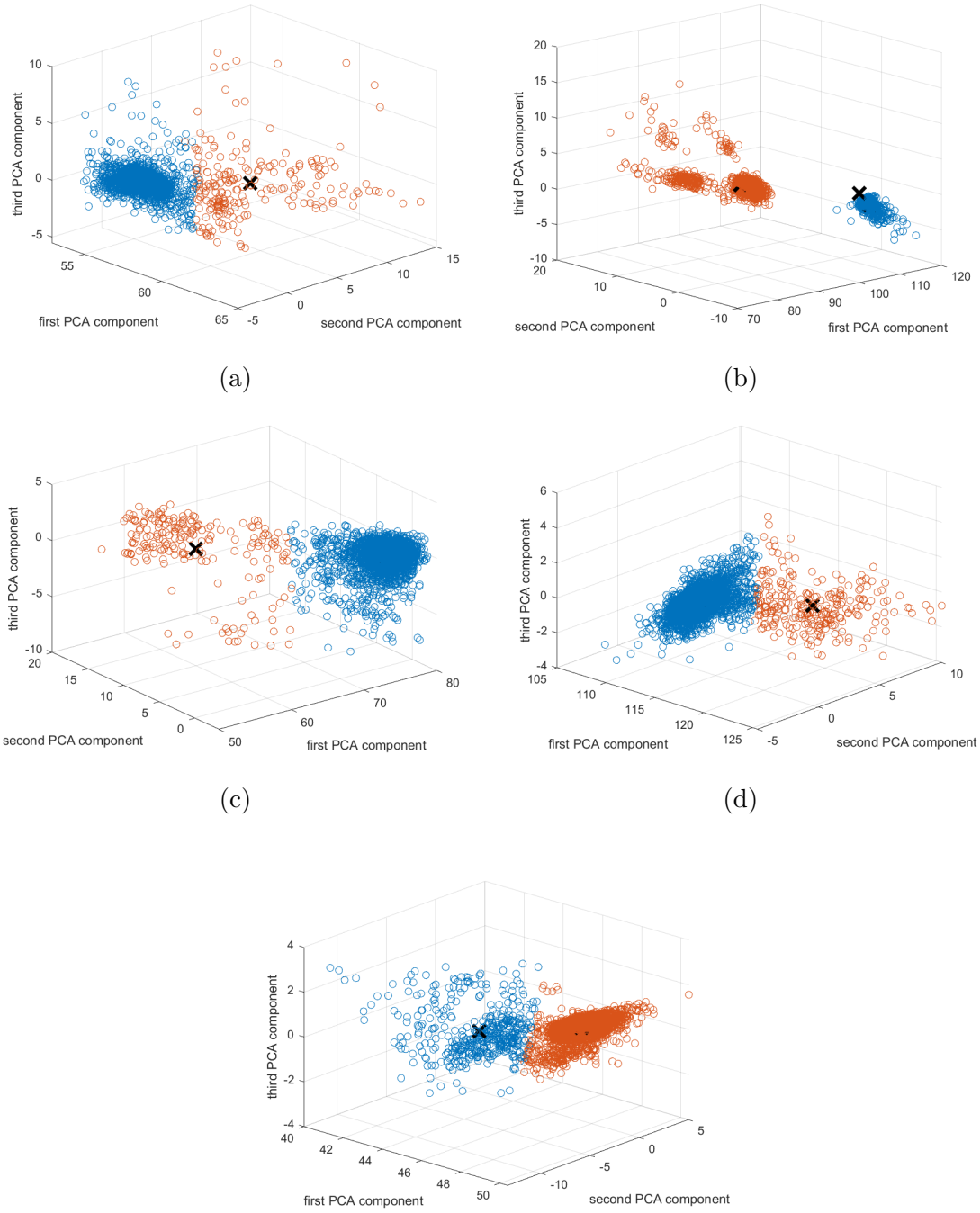


Figure 17: The scatter plots of the principal coefficient clusters after baseline corrections at Silhouette-optimal points. (a) Asls, $\lambda = 10^{10}, n = 2$; (b) arPLS, $\lambda = 10^{5}, n = 2$; (c) airPLS, $\lambda = 10^{7}, n = 2$; (d) MSBC, $\lambda = 10^{10}, n = 2$; (e) no correction, $n = 2$. The Silhouette optimal points are attained from the Fig.16. Visually, the airPLS result has the best separation. The data file used here is the fungi spectral image.

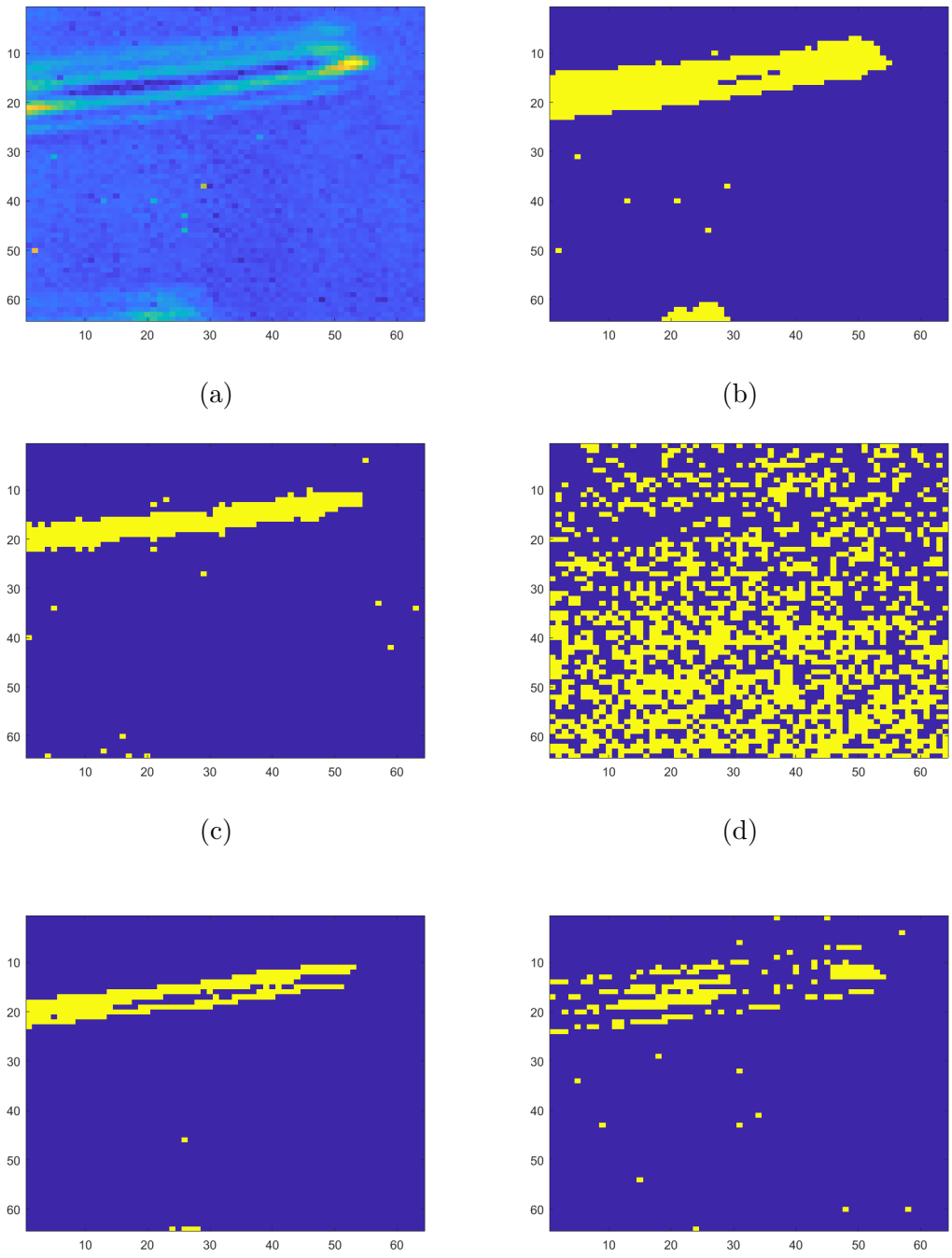


Figure 18: The original image of a wavelength and five corresponding images after being corrected and clustered by the K-means method. The baseline corrections are conducted at the Silhouette optimal points of each method. (a) original image of a wave channel; (b) without correction, $n = 2$; (c) Asls, $\lambda = 10^{10}, n = 2$; (d) arPLS, $\lambda = 10^{5}, n = 2$; (e) airPLS, $\lambda = 10^{7}, n = 2$; (f) MSBC, $\lambda = 10^{10}, n = 2$.

5 Discussions and conclusions

With many examples of baseline correction results and comparisons, it is obvious that baseline correction assists our spectral analysis by extracting information and sorting spectra into the right clusters.

The selection of the correction algorithm is contingent upon specific requirements. If the aim is to attain consistent baselines and corrected spectra across all data, among the four evaluated algorithms, MSBC exhibits superior performance owing to its incorporation of penalties for discrepancies among individual spectra. However, this might come at the cost of clustering accuracy. Conversely, if the objective is to extract feature information, Asls stands as the optimal choice. It has the capability to segment the captured pixels into distinct clusters utilizing PCA and the K-means method, while retaining the primary visual characteristics of the original images.

It is possible to improve the baseline correction algorithms by combining the difference penalization of MSBC and the weights updating function of arPLS to design a new algorithm. But the calculation could be highly time-consuming.

Acknowledgments

Special thanks to my supervisor Carl Troein for providing the raw data, Python codes generating synthetic data, and his kind guide. Also thanks to my girlfriend Zhang Hui for her support during my writing.

References

[1] Edmund T Whittaker. On a new method of graduation. *Proceedings of the Edinburgh Mathematical Society*, 41:63–75, 1922.

[2] Zhi-Min Zhang, Shan Chen, and Yi-Zeng Liang. Baseline correction using adaptive iteratively reweighted penalized least squares. *Analyst*, 135(5):1138–1146, 2010.

[3] Jianfeng Ye, Ziyang Tian, Haoyun Wei, and Yan Li. Baseline correction method based on improved asymmetrically reweighted penalized least squares for the raman spectrum. *Applied Optics*, 59(34):10933–10943, 2020.

[4] Jiangtao Peng, Silong Peng, An Jiang, Jiping Wei, Changwen Li, and Jie Tan. Asymmetric least squares for multiple spectra baseline correction. *Analytica chimica acta*, 683(1):63–68, 2010.

[5] Quanjie Han, Qiong Xie, Silong Peng, and Baokui Guo. Simultaneous spectrum fitting and baseline correction using sparse representation. *Analyst*, 142(13):2460–2468, 2017.

[6] Feng Gan, Guihua Ruan, and Jinyuan Mo. Baseline correction by improved iterative polynomial fitting with automatic threshold. *Chemometrics and Intelligent Laboratory Systems*, 82(1-2):59–65, 2006.

[7] Fang Qian, Yihui Wu, and Peng Hao. A fully automated algorithm of baseline correction based on wavelet feature points and segment interpolation. *Optics & Laser Technology*, 96:202–207, 2017.

[8] Xianchun Shen, Shubin Ye, Liang Xu, Rong Hu, Ling Jin, Hanyang Xu, Jianguo Liu, and Wenqing Liu. Study on baseline correction methods for the fourier transform infrared spectra with different signal-to-noise ratios. *Applied Optics*, 57(20):5794–5799, 2018.

[9] Wolfgang Demtröder. *Atoms, molecules and photons*, volume 3. Springer, 2010.

[10] Sune Svanberg. Laser spectroscopy. In *Atomic and Molecular Spectroscopy: Basic Aspects and Practical Applications*, pages 339–454. Springer, 2023.

[11] Xianchun Shen, Shubin Ye, Liang Xu, Rong Hu, Ling Jin, Hanyang Xu, Jianguo Liu, and Wenqing Liu. Study on baseline correction methods for the fourier transform infrared spectra with different signal-to-noise ratios. *Appl. Opt.*, 57(20):5794–5799, Jul 2018.

[12] Jonghee Yoon, Alexandru Grigoroiu, and Sarah E Bohndiek. A background correction method to compensate illumination variation in hyperspectral imaging. *PLoS One*, 15(3):e0229502, 2020.

[13] Freek D Van der Meer, Harald MA Van der Werff, Frank JA Van Ruitenbeek, Chris A Hecker, Wim H Bakker, Marleen F Noomen, Mark Van Der Meijde, E John M Carranza, J Boudewijn De Smeth, and Tsehaie Woldai. Multi-and hyperspectral geologic remote sensing: A review. *International Journal of Applied Earth Observation and Geoinformation*, 14(1):112–128, 2012.

[14] FM Lacar, MM Lewis, and IT Grierson. Use of hyperspectral imagery for mapping grape varieties in the barossa valley, south australia. In *IGARSS 2001. Scanning the Present and Resolving the Future. Proceedings. IEEE 2001 International Geoscience and Remote Sensing Symposium (Cat. No. 01CH37217)*, volume 6, pages 2875–2877. IEEE, 2001.

[15] Marleen F Noomen. *Hyperspectral reflectance of vegetation affected by underground hydrocarbon gas seepage*. Wageningen University and Research, 2007.

[16] Guolan Lu and Baowei Fei. Medical hyperspectral imaging: a review. *Journal of biomedical optics*, 19(1):010901–010901, 2014.

[17] Paul HC Eilers. A perfect smoother. *Analytical chemistry*, 75(14):3631–3636, 2003.

[18] Paul HC Eilers and Hans FM Boelens. Baseline correction with asymmetric least squares smoothing. *Leiden University Medical Centre Report*, 1(1):5, 2005.

[19] Gilbert Strang. *Linear algebra and its applications.* Belmont, CA: Thomson, Brooks/Cole, 2006.

[20] Jonas Kalderstam, Patrik Edén, Pär-Ola Bendahl, Carina Strand, Mårten Fernö, and Mattias Ohlsson. Training artificial neural networks directly on the concordance index for censored data using genetic algorithms. *Artificial intelligence in medicine*, 58(2):125–132, 2013.