Check for updates

DOI: 10.1177/0003702819885002

[1]Multiple Constrained Reweighted Penalized Least Squares for Spectral Baseline Correction

Guofeng Yanga, Jiacai Dai*[a], Xiangjun Liua,[b], Meng Chena, Xiaolong Wu [a]

[a] School of Geoscience and Technology, Southwest Petroleum University, Chengdu 610500, Sichuan Province, China

[b] State Key Laboratory of Oil and Gas Reservoir Geology and Exploitation, Southwest Petroleum University, Chengdu 610500, Sichuan Province, China

Corresponding author email: daijc@swpu.edu.cn

Abstract

Baseline drift occurs in various measured spectra, and the existence of a baseline signal will influence qualitative and quantitative analyses. Therefore, it is necessary to perform baseline correction or background elimination before spectral analysis. In this paper, a multiple constrained asymmetric least squares method (mcaLS) based on the penalized least squares principle is proposed for baseline correction. The method takes both baseline and peak characteristics into account. Based on the prior knowledge that the left and right boundaries of characteristic peaks should be symmetrical, additional constraints of penalized least squares are added, which ensure the symmetry of spectra. The experimental results of the proposed method on simulated spectra are compared with existing baseline correction methods to verify the accuracy and adaptability of the proposed method. The method is also successfully applied to the baseline correction of real spectra. The results show that it can be effective for estimating the baseline. In addition, this method can also be applied to the baseline correction of other similar spectral signals.

Keywords: Spectroscopy process, baseline correction, penalized least squares, constraint condition

INTRODUCTION

Spectroscopic technology, such as Raman, infrared (IR), and radiation spectroscopy, is an important method for performing qualitative or quantitative analysis of chemical and physical

[1] Fi

DOI: 10.1177/0003702819885002

properties of materials. Spectroscopy has been widely used in geochemistry,[1] organic chemistry,[2] biological science[3] and other domains. However, because of spectral interference, for instance fluorescence in Raman spectroscopy or cosmic-ray radiation in radiation spectroscopy, the measured spectra always contain random noise and a baseline, which blur the useful signal and interfere with the calculation of material composition. Therefore, baseline correction of spectra is a crucial step in spectroscopy analysis.

In recent years, many mathematical methods have been proposed to eliminate baselines, including polynomial fitting,[4,5] wavelet transform,[6–8] optimization,[9,10] spline curve fitting,[11] sparse representation,[12,13] morphology,[14–17] statistics-sensitive nonlinear iterative peak-clipping (SNIP),[18–20] asymmetric weighted least squares,[21–24] and so on. These methods have positive significance in baseline correction. Among these methods, the asymmetric least squares is a method that has received a great deal of attention. This method estimates the spectral baseline by setting appropriate asymmetric weights and smoothing parameters, and the weights are iteratively updated. Eilers[21] adopted an asymmetric least squares smoothing (asLS) method to correct baselines. AsLS adopts a fixed-weight update scheme, which gives less weight to positive deviations with respect to the baseline. On the basis of asLS, Zhang et al.[22] proposed an adaptive iteratively reweighted penalized least squares method (airPLS) using an exponential expression to iteratively update weights thereby improving the convergence speed of the algorithm. He et al.[23] proposed an improved asymmetric least squares (IasLS) method by adding a first derivative smoothing constraint to the asLS algorithm, that improves the accuracy of the baseline estimation. To eliminate the adverse effect of noise in weight updating, Baek et al.[24] used a generalized logistic function combined with the noise mean and standard deviation to update weights and proposed the asymmetrically reweighted penalized least squares method (arPLS). ArPLS iteratively estimates the noise level and correspondingly adjusts the weights. Therefore, this algorithm has good noise resistance. However, these methods attempt to separate the baseline from the spectrum by describing baseline characteristics, but take less consideration of the characteristics of spectral peaks, which may not provide sufficient constraints to the estimated baseline. Thus, it is easy to produce insufficient or excessive baseline subtractions in complex spectra.

Similarly, the SNIP algorithm also describes the baseline features in an iterative way. The SNIP algorithm believes that the baseline value of each channel should be below the spectral

peak. Thus, this algorithm is a multipass clipping loop and clips out the peak region by replacing its value with the minima within the region. SNIP is also an efficient algorithm for baseline correction. However, the main problem with this algorithm is that it is very sensitive to the parameters, especially the clipping window, and the selection of parameters is often based on experience.

In this study, based on the principle of penalized least squares, a multiple constrained asymmetric least squares baseline correction method (mcaLS) is proposed. In this method, the characteristics of both baseline and spectral peaks are taken into account. When there is no baseline, the spectral peaks should be symmetric. Therefore, based on the left and right boundaries of spectral peaks, new constraints are set for penalized least squares and a new cost function is constructed. The additional constraints can effectively guarantee the symmetry of the peak shape so that the accuracy of the baseline estimation will be improved.

THEORY

Asymmetric Penalized Least Squares Method

The asymmetric penalized least squares method uses a Whittaker smoother to obtain a slowly varying estimation of the baseline. This method adopts asymmetric weights and smoothing parameters to balance the fidelity and roughness of spectral data by least squares, and the fitted result can be regarded as the spectral baseline. Assuming that the given spectrum vector $\boldsymbol{x} = [x_1, x_2, x_3, \ldots, x_m]^T$ and the baseline can be expressed as vector $\boldsymbol{z} = [z_1, z_2, z_3, \ldots, z_m]^T$, the cost function of penalized least squares is as follows:

$$F = \arg\min_{z} \sum_{i=1}^{m} \omega_i (x_i - z_i)^2 + \lambda \sum_{i=2}^{m-1} (\Delta^2 z_i)^2 \qquad (1)$$

where λ is the smoothing parameter and ω_i is the weight of x_i and can be written as a diagonal matrix $\boldsymbol{W}$:

$$\boldsymbol{W} = \begin{bmatrix} \omega_1 & 0 & \cdots & & 0 \\ 0 & \omega_2 & & & \\ \vdots & & \ddots & & \vdots \\ & & & \omega_{m-1} & 0 \\ 0 & & \cdots & 0 & \omega_m \end{bmatrix}_{m \times m} \qquad (2)$$

Δ^2 is second-order difference operator and $\Delta^2 z_i = (z_{i+1} - z_i) - (z_i - z_{i-1}) = z_{i+1} - 2z_i + z_{i-1}$. Thus, Δ^2

DOI: 10.1177/0003702819885002

can be expressed as the m–2×m matrix $\boldsymbol{D}$:

$$\boldsymbol{D}=\begin{bmatrix} 1 & -2 & 1 & 0 & \cdots & 0 & 0 & 0 \\ 0 & 1 & -2 & 1 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 1 & -2 & 1 & 0 \\ 0 & 0 & 0 & \cdots & 0 & 1 & -2 & 1 \end{bmatrix}_{m-2\times m} \quad (3)$$

Using matrices $\boldsymbol{D}$ and $\boldsymbol{W}$, the cost function can be written as:

$$F(\boldsymbol{z})=(\boldsymbol{x}-\boldsymbol{z})^T\boldsymbol{W}(\boldsymbol{x}-\boldsymbol{z})+\lambda\boldsymbol{z}^T\boldsymbol{D}^T\boldsymbol{D}\boldsymbol{z} \quad (4)$$

When the cost function is minimized, the value of the partial derivative with respect to the vector $\boldsymbol{z}$ is 0. Therefore, the solution of matrix Eq. 5 is given by Eq. 6.

$$\frac{\partial F}{\partial \boldsymbol{z}}=-2\boldsymbol{W}(\boldsymbol{x}-\boldsymbol{z})+2\lambda\boldsymbol{D}^T\boldsymbol{D}\boldsymbol{z}=0 \quad (5)$$

$$\hat{\boldsymbol{z}}=(\lambda\boldsymbol{D}^T\boldsymbol{D}+\boldsymbol{W})^{-1}\boldsymbol{W}\boldsymbol{x} \quad (6)$$

To effectively estimate the baseline, different weights should be assigned for the data in the spectrum based on the residual x_i–z_i. When the residual of a channel is less than 0, the channel will be given a large weight and a small weight will be set when the residual is larger than 0. In Eilers,[14] a simple scheme was proposed to achieve asymmetric weight adjustment by introducing the asymmetric parameter p. The weight scheme is as shown in Eq. 7.

$$\omega_i=\begin{cases} p & x_i>z_i \\ 1-p & x_i\le z_i \end{cases} \quad (7)$$

Since the value of parameter p ranges from 0.001 to 0.1, when the raw spectral data x_i from the ith channel are less than the fitting baseline value z_i, the weight of x_i is large, while the weight is small if x_i is larger than z_i. The weight of each channel is updated based on Eq. 7 in each iteration until the difference between the kth iteration and the (k+1)th iteration meets the accuracy requirement, at this time vector $\boldsymbol{z}$ is the estimated baseline.

Due to the inevitable noise interference in the actual measurement, the pure baseline signal will fluctuate to some extent in the region without a spectral peak, which may cause the estimated baseline to deviate from the actual baseline. To eliminate the influence of noise on the baseline estimation, Baek, et al. proposed a new weighting scheme in Chen and Dai[17] based on the generalized logistic function combining the mean and standard deviation of the noise. As shown in Eqs. 8 and 9, for the pure baseline region affected by noise, each channel should have a

similar weight, while the weight of the channels in the spectral peak regions should be close to 0.

$$\omega_i = \begin{cases} logistic(x_i - z_i, m_{d^-}, \sigma_{d^-}) & x_i > z_i \\ 1 & x_i \le z_i \end{cases} \quad (8)$$

$$logistic(d, m, \sigma) = \frac{1}{1 + \exp(2(d - (-m + 2\sigma))/\sigma)} \quad (9)$$

where m_{d^-} and σ_{d^-} are the mean and the standard deviation of $\boldsymbol{d}^-$, respectively. Given $\boldsymbol{d}=\boldsymbol{x}-\boldsymbol{z}$, $\boldsymbol{d}^-$ is a part of $\boldsymbol{d}$ that is defined in the region where $x_i<z_i$.

Multiple Constrained Asymmetric Least Squares Method (mcaLS)

Both the asLS algorithm and arPLS algorithm estimate the baseline by describing the characteristics of the baseline signal, that is, the spectrum is regarded as the superposition of the peak signal on the baseline signal, and the baseline should be below the spectral peaks. Therefore, when the spectra data are smaller than the fitting curve data, they will be given a greater weight. However, neither of the two methods takes into account the characteristics of measured spectral peaks, which makes it easy to cause an excessive or insufficient deduction of the baseline after processing and can result in a certain degree of deformation of characteristic peaks. According to the symmetry of spectral peaks, the intensities near the left and right boundaries of characteristic peaks should be similar. Based on this principle, we added additional constraints to the cost function and proposed the multiple constrained asymmetric least squares method to estimate the baseline. The new cost function is shown in Eq. 10.

$$F = \sum_{i=1}^{m} \omega_i (x_i - z_i)^2 + \lambda_1 \sum_{i=2}^{m-1} (\Delta^2 z_i)^2 + \lambda_2 \sum_{j}^{n} \sum_{i=1}^{p} [(x_{Lji} - z_{Lji}) - (x_{Rji} - z_{Rji})]^2 \quad (10)$$

where the first and second terms are the same as those in the above algorithms, representing the fidelity and smoothness of the baseline respectively. The third term represents the symmetry of the spectral peaks, and λ_1, λ_2 indicate the penalty coefficients of smoothness and symmetry, respectively, reflecting the importance of the constraints.

Parameter n is the number of the determined spectral peak regions and parameter p is the number of the selected spectral data near the left (right) spectral boundary of the jth peak region. Parameters x_{Lji} and z_{Lji} are the spectral intensity and the fitted baseline intensity of the ith channel near the left boundary of the jth peak region respectively. Parameters x_{Rji} and z_{Rji} are the spectral

intensity and the fitted baseline intensity of the *i*th channel near the right boundary of the *j*th peak region, respectively. The new cost function can also be written as a matrix equation, as shown in Eq. 11, so that the baseline can be iteratively estimated.

$$F(\boldsymbol{z}) = (\boldsymbol{x}-\boldsymbol{z})^T \boldsymbol{W}(\boldsymbol{x}-\boldsymbol{z}) + \lambda_1 \boldsymbol{z}^T \boldsymbol{D}^T \boldsymbol{D}\boldsymbol{z} + \lambda_2 (\boldsymbol{x}-\boldsymbol{z})^T \boldsymbol{E}^T \boldsymbol{E}(\boldsymbol{x}-\boldsymbol{z}) \quad (11)$$

where $\boldsymbol{E}$ is the boundary coefficient matrix, the form of which is shown in Eq. 12. The number of columns of $\boldsymbol{E}$ is equal to the spectral length m, and the number of rows is determined by the number of the selected spectral peak regions n. In matrix $\boldsymbol{E}$, the data near the left boundary of the peak regions are set as 1, while the data near the right boundary are set as –1, and the remaining data are set as 0. Data from multiple channels near the boundaries can be selected, resulting in a reduction of the impact of noise in the spectra. By means of constraint conditions, the intensities at the left and right boundaries of the peak regions can be guaranteed to be approximately equal, which conforms with the characteristics of the pure spectral peaks.

$$\boldsymbol{E} = \begin{bmatrix} & & & & & & \overbrace{\quad\quad}^{Peak\ \ region} & & & & & & & & & & \\ 0 & \cdots & 0 & 1 & \cdots & 1 & 0\cdots Peak\cdots 0 & -1 & \cdots & -1 & 0 & \cdots & 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & 0 & 1 & \cdots & 1 & \underbrace{0\cdots Peak\cdots 0}_{Peak\ \ region} & -1 & \cdots & -1 & 0 & \cdots & 0 \\ & & & & & & & & & & & & & & & \end{bmatrix} \quad (12)$$

Similarly, the partial derivative of Eq. 11 with respect to the vector $\boldsymbol{z}$ is obtained. When the partial derivative is 0, the baseline estimation can be obtained, as shown in Eq. 14.

$$\frac{\partial F}{\partial \boldsymbol{z}} = -2\boldsymbol{W}(\boldsymbol{x}-\boldsymbol{z}) + 2\lambda_1 \boldsymbol{D}^T \boldsymbol{D}\boldsymbol{z} - 2\lambda_2 \boldsymbol{E}^T \boldsymbol{E}(\boldsymbol{x}-\boldsymbol{z}) = 0 \quad (13)$$

$$\hat{\boldsymbol{z}} = (\boldsymbol{W} + \lambda_1 \boldsymbol{D}^T \boldsymbol{D} + \lambda_2 \boldsymbol{E}^T \boldsymbol{E})^{-1} (\lambda_2 \boldsymbol{E}^T \boldsymbol{E}\boldsymbol{x} + \boldsymbol{W}\boldsymbol{x}) \quad (14)$$

In terms of the weighting scheme, mcaLS still adopts the weighting scheme of arPLS. We believe this is an excellent design, because it takes into account the impact of random noise on the baseline estimation, allowing it to prevent an overestimation of the baseline owing to the influence of noise in the pure baseline area.

A key step in mcaLS is to determine the spectral peak regions. The boundary coefficient matrix can be directly determined if the characteristic peak positions and the calibration of the peak width in the spectrum are known. Otherwise, peak detection is required. At present, many research achievements in peak detection and peak region determination have been made, such as the derivative method,[25] wavelet transform,[26] and Markov chain smoothing,[27] among others,

which will not be elaborated in detail in this study. The greatest challenge in peak area determination is noise, the intensity of each point within the peak region should be greater than the noise level, and the peak region starts at a point higher than the noise level and moves along channels until a point lower than the noise level is found.[28] We prefer to use the smooth derivative method to determine the peak region, which is a fast and sensitive peak detection method. This method detects peaks by searching downward zero-crossing in the smoothed first derivative, which exceeds the slope threshold and amplitude threshold. The required peak regions can be obtained, while those that are too low, too wide or too narrow can be ignored. In addition, two methods can be applied to reduce the error caused by noise: the first is to increase the number of the selected channels in the boundary coefficient matrix $\boldsymbol{E}$; the second is to filter the spectral vector $\boldsymbol{x}$ to suppress noise.

The proposed algorithm is summarized as follows:

Algorithm 1: Multiple constrained asymmetric least squares (mcaLS)

Step 1. Input the raw spectral vector $\boldsymbol{x}$, smoothing parameter λ_1, symmetry parameter λ_2, and iterative termination precision δ.

Step 2. Determine the evaluation parameters：

2.1 Initialize weight matrix $\boldsymbol{W}$ and second-order difference matrix $\boldsymbol{D}$

2.2 Determine the boundary coefficient matrix $\boldsymbol{E}$ based on peak regions.

Step 3. Iterative operation：

3.1 Calculate parameter matrix $\boldsymbol{L} = (\boldsymbol{W} + \lambda_1 \boldsymbol{D}^T \boldsymbol{D} + \lambda_2 \boldsymbol{E}^T \boldsymbol{E})$

3.2 Calculate the baseline vector $\boldsymbol{z} = \boldsymbol{L}^{-1}(\lambda_2 \boldsymbol{E}^T \boldsymbol{E}\boldsymbol{x} + \boldsymbol{W}\boldsymbol{x})$

3.3 Update the weight matrix based on Eq. (8) and Eq. (9).

3.4 Check the terminal condition: If $||\mathbf{z}(k) - \mathbf{z}(k-1)|| < \delta$， then go to Step4; else return to Step3.

Step 4. Output baseline vector $\boldsymbol{z}$.

In terms of the performance of the algorithm, mcaLS has three advantages. First, the algorithm does not need to determine all the peak regions in the spectrum. Generally speaking, selecting parts of typical peak regions can achieve good results. Second, the boundary constraints are still valid for overlapping peaks, because the intensities at the left and right boundaries of the overlapping peaks are still approximately equal. Finally, the algorithm does not need to accurately determine the boundaries of the peak regions. In general, the pure baseline area near

DOI: 10.1177/0003702819885002

the boundaries can be selected to set constraints. These three advantages can improve the robustness of mcaLS in practical applications.

Experiment and Discussion

Experiments on Simulated Spectra

To illustrate the applicability of mcaLS in spectral baseline correction, two kinds of synthetic spectra were used to verify the performance of the algorithm. The synthetic spectrum has 256 channels and consists of three components: the peak signal, baseline signal and noise signal. The spectral peak signal is composed of multiple Gaussian functions in the form of:

$$p(i) = H \exp(-\frac{(i-c)^2}{2\sigma^2}) \quad (15)$$

where H is the peak height, c is the peak position, and σ is the standard deviation and it is directly proportional to the peak width. The synthetic spectra include multiple peaks with different widths, heights and positions. The spectral peak parameters are shown in Table I. We designed four kinds of spectral peaks in the synthetic spectra, including an isolated narrow peak (Peak 1), an isolated wide peak (Peak 2), partial overlapping peaks (Peak 3, Peak 4) and overlapping peaks (Peak 5, Peak 6).

Table I. Peak parameters of the synthetic spectrum.

Peak	H	c	σ^2
1	50	40	40
2	60	100	20
3	60	110	20
4	30	150	110
5	40	200	40
6	20	210	80

The baseline signal in the synthetic spectrum is expressed in two forms: a quadratic polynomial function and an exponential function. The functional expressions are shown in Eqs. 16 and 17. The synthetic spectra containing two types of baselines are shown in Figure 1. In the figure, the vertical dotted lines indicate the spectral boundaries and the bold lines are the

DOI: 10.1177/0003702819885002

constraint boundaries, which are used for constructing the matrix ***E***.

$$b_1 = -0.0006i^2 + 0.1i + 130 \quad (16)$$

$$b_2 = 190\exp(-\frac{i}{500}) \quad (17)$$

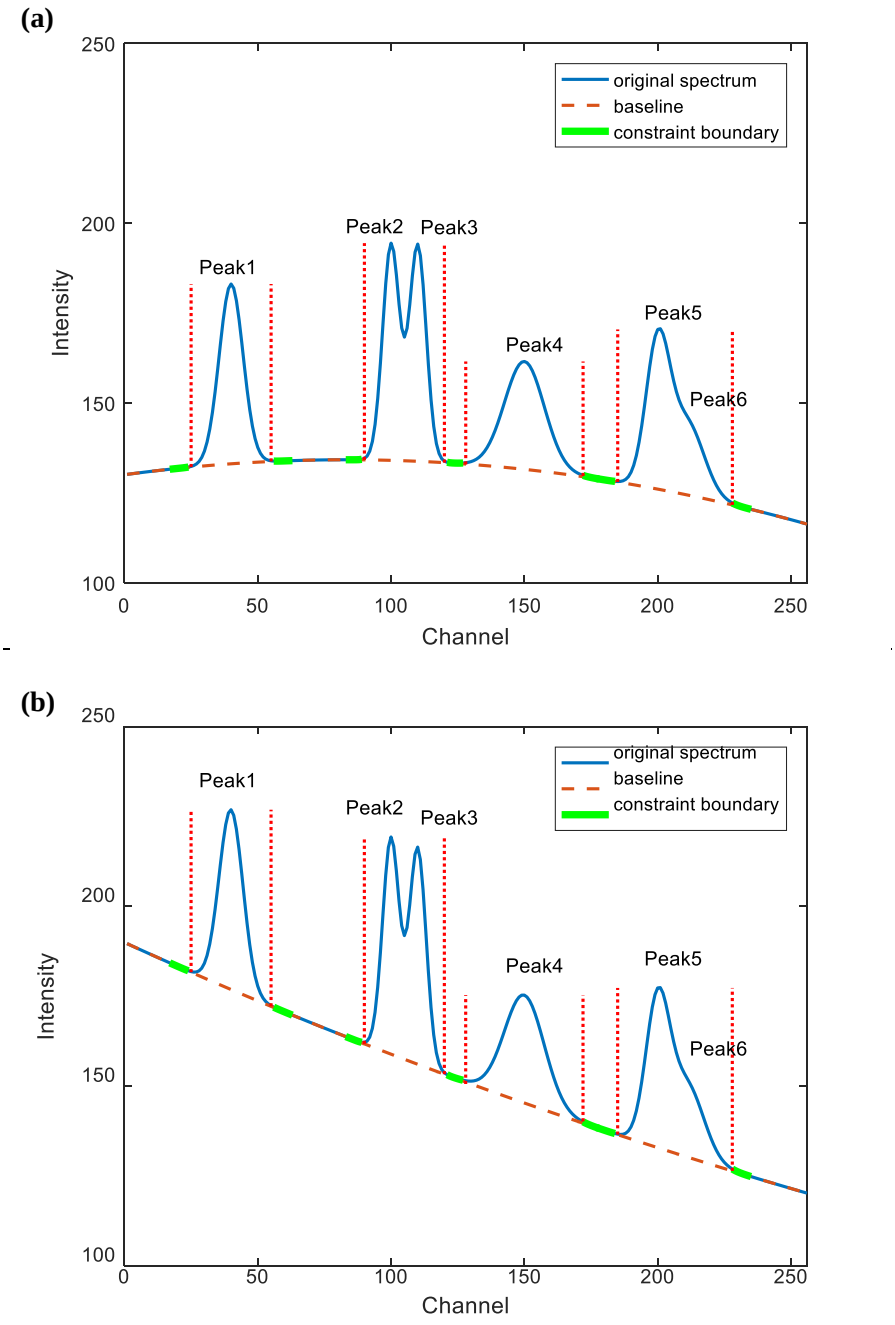


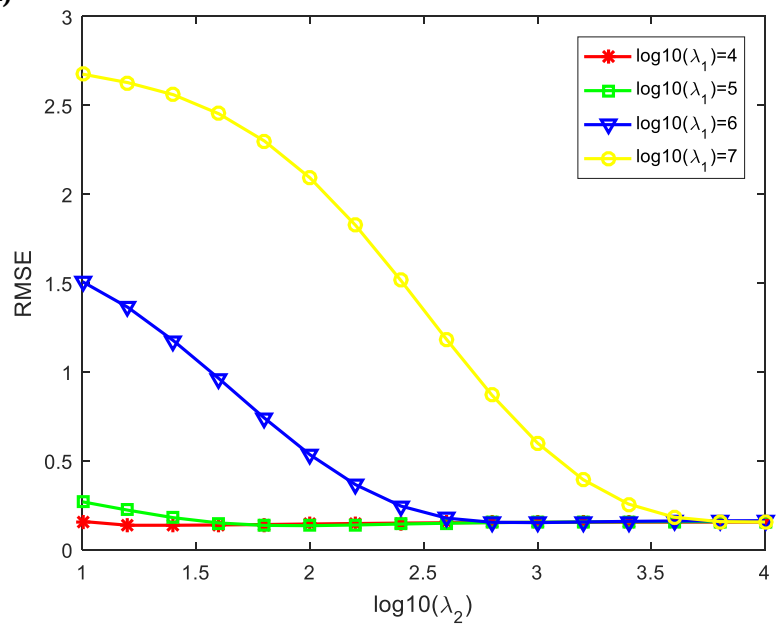
Figure 1. (a) Synthetic spectrum with a quadratic polynomial baseline. (b) Synthetic spectrum with an exponential baseline (the vertical dotted lines are the spectral boundaries)

DOI: 10.1177/0003702819885002

For synthetic spectra, the effect of baseline correction can be evaluated by the root mean square error (RMSE) between the actual baseline and the estimated baseline. Before using the mcaLS algorithm for baseline correction, it is necessary to determine the penalty parameters first. Baek[24] et al. discussed the smoothing parameter selection of the arPLS algorithm, which uses a logistic function weight updating scheme, and noted that the lowest RMSE was obtained when the smoothing parameter was approximately 10^5.

For the mcaLS algorithm, we should focus on the influences of the smoothing parameter λ_1 and symmetry parameter λ_2. Figure 2 shows the RMSEs of the mcaLS algorithm under different λ_1 and λ_2. It is obvious that whether for a quadratic polynomial baseline or exponential baseline, RMSE is more sensitive to λ_2 as λ_1 increases. When λ_1 is less than 10^5, the RMSEs of the estimated baselines vary slowly with λ_2. Moreover, for different values of λ_1, there is a convergence when λ_2 is larger than $10^{3.5}$. It can be seen from Figure 2 that the lowest RMSE is obtained when λ_1 is close to 10^5 and λ_2 is close to 10^2. Furthermore, the result for the exponential baseline is superior to that for the quadratic polynomial baseline.

(a)



DOI: 10.1177/0003702819885002

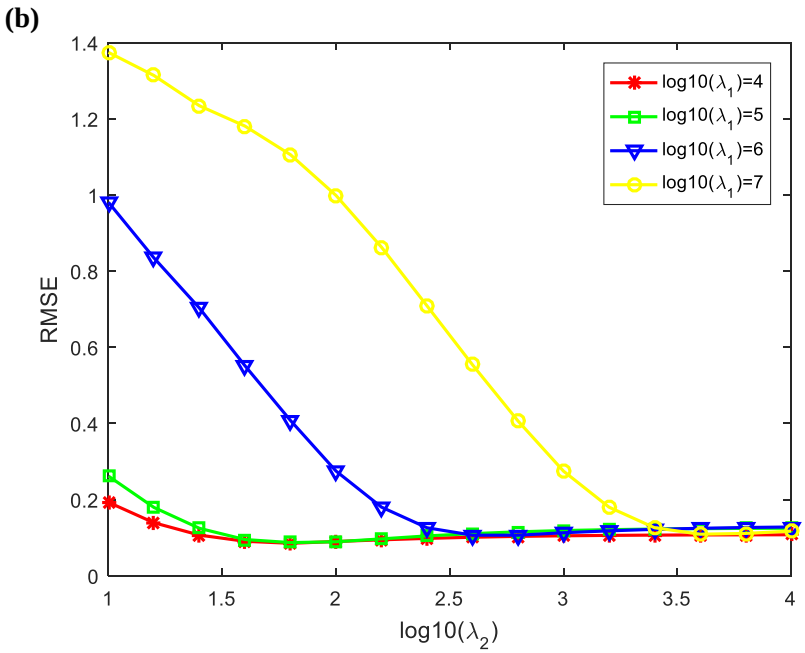


Figure 2 RMSE of mcaLS for various smoothing parameters (λ_1) and symmetry parameters (λ_2): (a) quadratic polynomial baseline and (b) exponential baseline.

To process the synthetic spectra, AsLS, arPLS, and SNIP were used and the results were compared with mcaLS. The parameters setting of arPLS and asLS refers to the research results of Baek[24] and Eilers[21]. SNIP determined the width of the clipping window through mathematical experiments. The results of the four algorithms are shown in Table II. It can be seen that mcaLS has the minimum RMSE for both the polynomial baseline and exponential baseline. Compared with arPLS, the RMSE of the polynomial baseline estimation is reduced by 69.04%, while the RMSE of the exponential baseline estimation is nearly 1/10 of that of the arPLS algorithm. This result indicates that the baseline estimated by mcaLS is the closest to the actual baseline and that the spectral peaks after correction have the best fidelity.

Table II. RMSE of the baseline estimation.

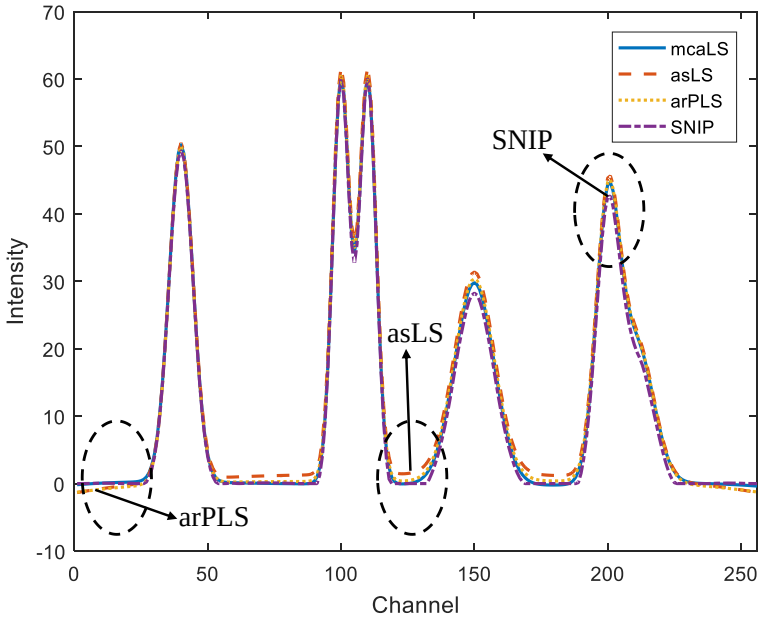
Method	Parameters	RMSE	
		Quadratic polynomial baseline	Exponential baseline
asLS	$\lambda=10^5$ p=0.01	1.04	0.86
arPLS	$\lambda=10^5$ δ=0.001	0.43	0.94

DOI: 10.1177/0003702819885002

SNIP	clipping window width=13	1.03	1.18
mcaLS	$\lambda_1=10^5$ $\lambda_2=10^2$ $\delta=0.001$	0.13	0.09

The spectral peak signals obtained after baseline correction by means of the four algorithms are shown in Figure 3. The oval marks in Figure 3 show that both the quadratic polynomial baseline and exponential baseline corrected by asLS are not accurate, that is, the estimated baseline is insufficient or excessive, which is more obvious in the processing results of the quadratic polynomial baseline. ArPLS produced an obvious distortion at the endpoints after processing of the polynomial baseline and exponential baseline. Although SNIP did not have negative areas after correction in two kinds of baselines, a peak height reduction appeared, which indicated that the spectral peaks was excessively subtracted. McaLS guaranteed the authenticity of spectral peaks to the greatest extent, whether in the polynomial baseline or the exponential baseline, and there was no distortion after the process.

(a)



DOI: 10.1177/0003702819885002

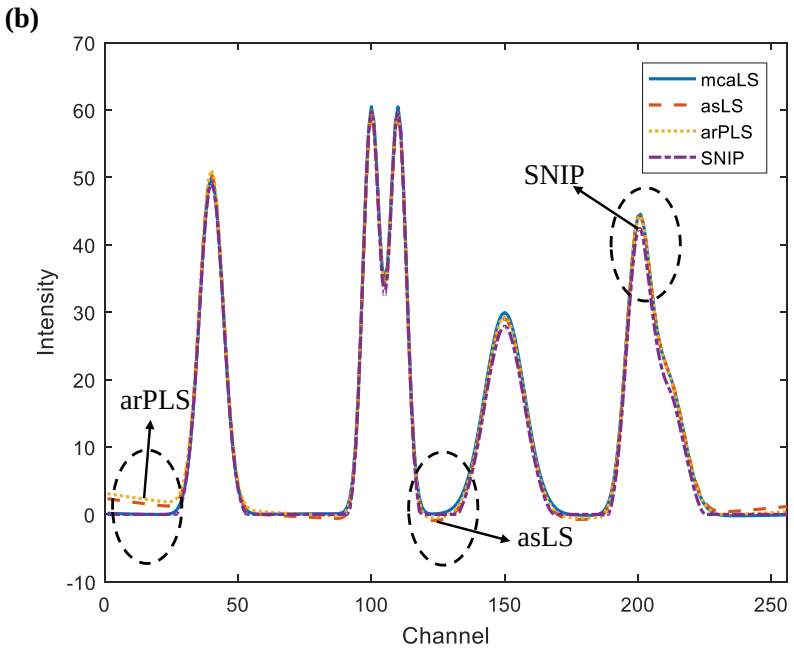


Figure 3 (a) Quadratic polynomial baseline correction effect. (b) Exponential baseline correction effect.

The determination of the matrix ***E*** depending on the widths of the intervals on the two flanks of the peak region. The constraints established by the matrix ***E*** reflect the symmetry of the spectral peaks. Each row in the matrix ***E*** describes the constraint boundary of one of the peak regions (isolated peak or overlapping peaks) in the spectrum. We can study the impact of the size of the constraint boundaries on baseline correction by varying the number of elements with values of 1 and –1 in each row of the matrix ***E***. The size of the constraint boundaries can be represented by the ratio of the length of the whole constraint boundaries to the length of the whole non-peak regions (CNR). The RMSEs of the baseline correction under different CNRs are shown in Table III. It can be seen that the size of the constrained boundaries has little influence on the effect of baseline correction. The difference of RMSE under different CNRs is small.

Table III. RMSE of baseline correction using mcaLS with different CNRs.

CNR	Quadratic polynomial baseline	Exponential baseline
12%	0.13	0.09

DOI: 10.1177/0003702819885002

24%	0.18	0.076
33%	0.19	0.12
57%	0.15	0.074

To study the influence of the number of the selected peak regions on the algorithm, we selected a different number of spectral peak regions to set the constraint conditions. According to the determined peak region boundaries, the synthetic spectrum can be divided into four peak regions, including two isolated peaks and two overlapping peaks. The baseline correction results are shown in Table IV. Compared with other methods, the baseline estimated by mcaLS still has a smaller RMSE. However, it is important to note that the selected peak regions should be located at different positions of the whole spectrum so as to ensure that the constraints are valid for the whole spectrum.

Table IV. RMSE of mcaLS with a different number of selected peak regions.

peak region	quadratic polynomial baseline	exponential baseline
1, (2,3), 4	0.31	0.41
1, 4, (5,6)	0.19	0.09
1, (2,3), (5,6)	0.18	0.075
1, (5,6)	0.18	0.086
(2,3), (5,6)	0.26	0.37

Experiments Using Real NIR Spectra

The real spectral data, which are used to verify the validity of the mcaLS algorithm, is composed of three kinds of spectra, including near infrared (NIR) spectrum, IR spectrum, and Raman spectrum. In the processing, we directly used the peak region boundaries of the spectrum, which are represented by the vertical dotted lines in the following figures, to construct the matrix $\boldsymbol{E}$. It means that the peak region boundaries are the constraint boundaries. For each figure, the dash line in the upper panel is the estimated baseline and the dash line in the lower panel is zero line. The solid line in the upper panel is the raw spectrum and the solid line in the lower panel is the corrected spectrum.

The NIR spectra data set is measured from corn samples and consists of 80 spectra

measured on three different NIR spectrometers. The wavelength range is 1100–2498 nm at 2 nm intervals (700 channels). The dataset is provided by Eigenvector Research, Inc. and is available from the website: http://www.eigenvector.com/data/Corn/index.html.

We selected one of the raw NIR spectra measured on spectrometer mp5 to test the mcaLS algorithm and the result is shown in Figure 4. As seen from the figure, the raw spectrum showed significant baseline drift. Most of the absorption peaks of spectra have extremely asymmetric morphological characteristics due to the presence of the baseline. The mcaLS algorithm was used to estimate the spectral baseline with the parameters: $\lambda_1=10^5$, $\lambda_2=10^2$. It is apparent that the spectrum corrected by mcaLS is close to the zero line, and that the characteristic peaks are not lost. In addition, after correction, the peak shape of the spectra was basically symmetrical and the intensities at the left and right boundaries of peak regions were almost the same.

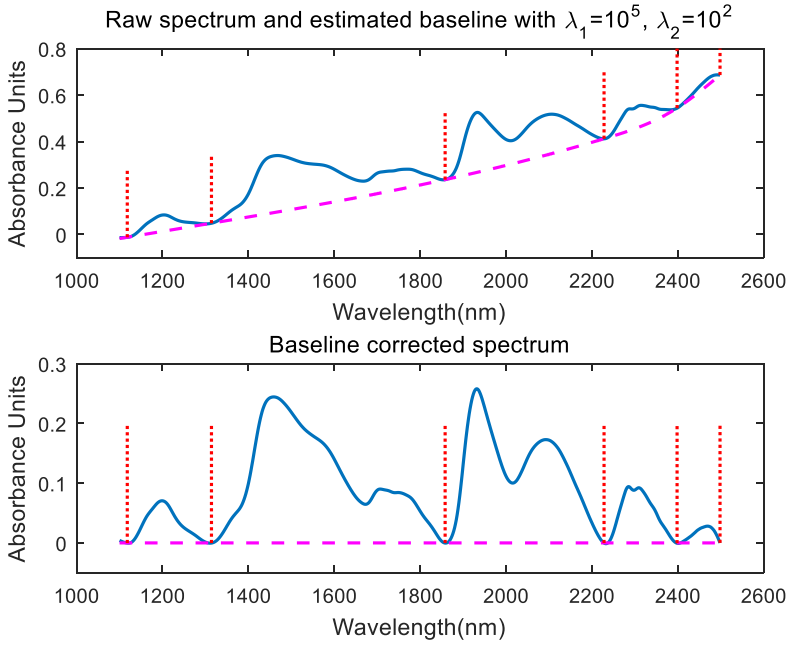


Figure 4. Baseline estimation with mcaLS applied to a NIR spectrum.

The second application example is the IR spectral data set of marzipan samples[29], the raw IR spectrum and the result are shown in Figure 5. The IR spectrum contains more overlapping peaks and narrow peaks. In this example, we put multiple overlapping peaks into one peak region and set constraints according to the selected peak region boundaries. The parameters of the algorithm are also: $\lambda_1= 10^5$, $\lambda_2 = 10^2$. The corrected spectrum is shown in the lower panel of

DOI: 10.1177/0003702819885002

Figure 5. We can find that the constraints are also valid for overlapping peaks. The baseline estimation performance can be improved by setting constraints on the whole spectrum segment.

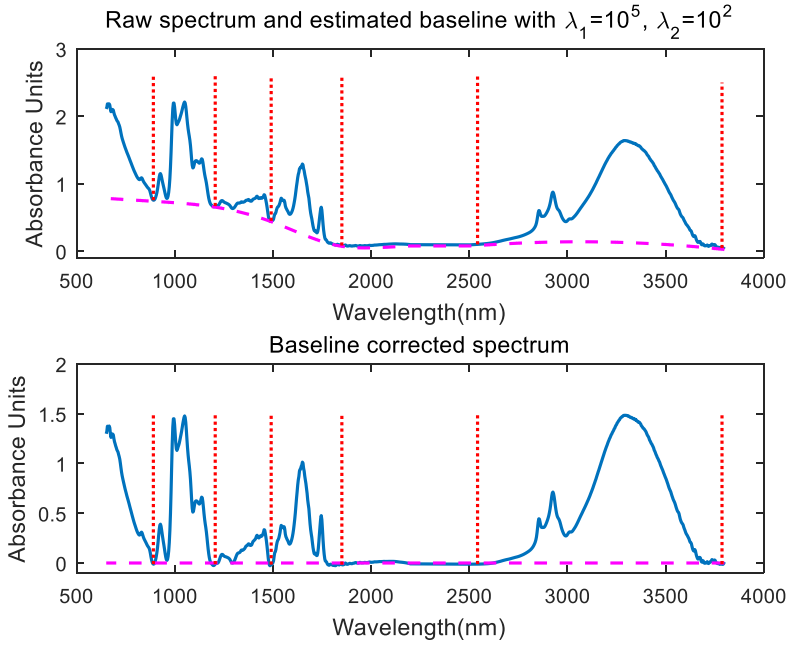


Figure 5. Baseline estimation with mcaLS applied to an IR spectrum.

The Raman spectrum data set is the third application example, which measured from 16 pork carcasses taken from the daily production stock of a slaughterhouse,[3] and one of the spectra is plotted in Figure 6. Compared with the last two examples, the Raman spectrum has more channels. The peak height difference in the Raman spectrum is relatively large and the baseline of the Raman spectrum is approximate to a polynomial baseline. We corrected the spectrum with $\lambda_1 = 10^5$, $\lambda_2 = 10^3$. In this example, we adjusted the parameters slightly and the estimated baseline is also a satisfactory estimation. This result is consistent with the relationship between RMSE and parameters λ_1, λ_2. Moreover, the weak peaks in the spectrum are not lost.

DOI: 10.1177/0003702819885002

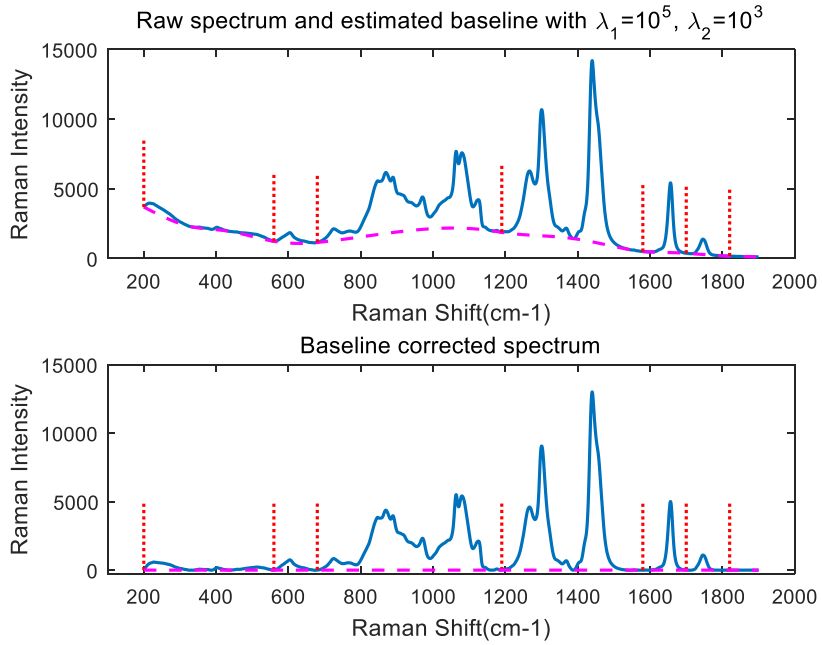


Figure 6. Baseline estimation with mcaLS applied to a Raman spectrum.

According to the above examples, we can see that the mcaLS algorithm can deal with overlapping peaks and weak peaks. If the spectrum contains little noise, the mcaLS algorithm is not sensitive to the widths of the constraint boundaries. However, if the signal-to-noise ratio of the spectrum is low, filtering or expanding the width of the constraint boundaries is necessary to reduce the adverse effects of noise. For parameters λ_1 and λ_2, although we provide a set of reference values, we still hope that the user can adjust the parameters based on the actual performance.

CONCLUSION

In this paper, a new method for baseline correction based on multiple constrained asymmetric least squares is proposed for measured spectra. Based on the principle of penalized least squares, the cost function, which can accurately describe the spectral baseline and peak characteristics, is established by using the symmetry of characteristic spectral peaks, and the baseline estimation is realized by iteration. The accuracy and robustness of the algorithm are verified by the processing results of synthetic spectra and real spectra.

However, qualitative and quantitative spectral analysis is a complex and systematic task, that requires a variety of preprocessing, such as spectral denoising, spectral peak detection,

baseline correction, and so on. The algorithm proposed in this paper depends on peak region determination to some extent, and the existence of noise can also have a negative impact on baseline estimation. Therefore, other preprocessing methods need to be improved in the next step, which will further improve the processing performance of the proposed algorithm.

Conflicts of Interest

The authors declared that they have no conflicts of interest to this work. We declare that we do not have any commercial or associative interest that represents a conflict of interest in connection with the work submitted.

Acknowledgments

This project was supported by the National Natural Science Foundation of China (No.41804141) and the author thanks for the support by the State Key Laboratory of Oil and Gas Reservoir Geology and Exploitation, Southwest Petroleum University, Sichuan province, China.

Supplemental Material

All supplemental material, including additional comparative information and algorithm codes, is available in the online version of the journal.

References

1. 1. W. Wu, N. Wei, L. Li. “Quantitative Analysis of Neutron-Capture Gamma-Ray Energy Spectra Using Direct Demodulation”. Geophysics. 2014. 79(2): D91–D98.
2. 2. J. Ethève, P. Huguet, C. Innocent, J. Bribes. “Electrochemical and Raman Spectroscopy Study of a Nafion Perfluorosulfonic Membrane in Organic Solvent–Water Mixtures”. J. Phys. Chem. B. 2001. 105(19): 4151–4154.
3. L.B. Lyndgaard, K.M. Sørensen, F. Berg, S. Engelsen. “Depth Profiling of Porcine Adipose Tissue by Raman Spectroscopy”. J. Raman Spectrosc. 2012 43(4): 482–489.
4. F. Gan, G.H. Ruan, J.Y. Mo. “Baseline Correction by Improved Iterative Polynomial Fitting with Automatic Threshold”. Chemom. Intell. Lab. Syst. 2006. 82(1): 59–65.
5. A. Cao, A.K. Pandya, G.K. Serhatkulu, R.E. Weber, et al. “A Robust Method for Automated Background Subtraction of Tissue Fluorescence”. J. Raman Spectrosc. 2010. 38(9):

1199–1205.

6. C.M. Galloway, E.C.L. Ru, P.G. Etchegoin. “An Iterative Algorithm for Background Removal in Spectroscopy by Wavelet Transforms”. Appl. Spectrosc. 2009. 63(12): 1370–1376.
7. H. Shin, M.P. Sampat, J.M. Koomen, M.K. Markey. “Wavelet-Based Adaptive Denoising and Baseline Correction for MALDI TOF MS”. Integr. Biol. 2010. 14(3): 95–285.
8. Y. Hu, J. Zhou, J. Tang, S. Xiao. “The Application of Complex Wavelet Transform to Spectral Signals Background Deduction”. Chromatographia. 2013. 76(11–12): 687–696.
9. S. He, S. Fang, X. Liu, W. Zhang, et a. “Investigation of A Genetic Algorithm Based Cubic Spline Smoothing for Baseline Correction of Raman Spectra”. Chemom. Intell. Lab. Syst. 2016. 152: 1–9.
10. S. Guo, T. Bocklitz, P. Jürgen. “Optimization of Raman-Spectrum Baseline Correction in Biological Application”. Analyst. 2016. 141(8): 2396–2404.
11. J.J.D. Rooi, P.H.C. Eilers. “Mixture Models for Baseline Estimation”. Chemom. Intell. Lab. Syst. 2012. 117: 56–60.
12. Q. Han, Q. Xie, S. Peng, B. Guo. “Simultaneous Spectrum Fitting and Baseline Correction Using Sparse Representation”. Analyst. 2017. 142(13): 2460–2468.
13. X. Ning, I.W. Selesnick, L. Duval “Chromatogram Baseline Estimation and Denoising Using Sparsity (BEADS)”. Chemom. Intell. Lab. Syst. 2014.139:156–167.
14. R. Perezpueyo, M.J. Soneira, S. Ruizmoreno. “Morphology-Based Automated Baseline Removal for Raman Spectra of Artistic Pigments”. Appl. Spectrosc. 2010. 64(6): 595–600.
15. H. Liu, Z. Zhang, S. Liu, L. Yan, et al. “Joint Baseline-Correction and Denoising for Raman Spectra”. Appl. Spectrosc. 2015. 69(9): 1013–1022.
16. Z. Li, D.J. Zhan, J.J. Wang, J. Huang, et al. “Morphological Weighted Penalized Least Squares for Background Correction”. Analyst. 2013. 138(16): 4483.
17. Y. Chen, Y.L. Dai. “An Automated Baseline Correction Method Based on Iterative Morphological Operations”. Appl. Spectrosc. 2018. 72(5): 731–739.
18. C.G. Ryan, E. Clayton, W.L. Griffin, S.H. Sie, et al. “SNIP, a Statistics-Sensitive Background Treatment for the Quantitative Analysis of PIXE Spectra in Geoscience Applications”. Nucl. Instrum. Meth. B. 1988. 34(3): 396–402.
19. M. Morhac. “An Algorithm for Determination of Peak Regions and Baseline Elimination in

Spectroscopic Data". Nucl. Instrum. Meth. A. 2009. 600(2): 478–487.

20. R. Shi, X. Tuo, H. Zheng, X. Yan, et al. "Step-Approximation SNIP Background-Elimination Algorithm for HPGe". Nucl. Instrum. Meth. A. 2018. 885: 60–66.
21. P.H.C. Eilers. "Parametric Time Warping". Anal. Chem. 2004. 76(2): 404–411.
22. Z.M. Zhang, S. Chen, Y.Z. Liang. "Baseline Correction Using Adaptive Iteratively Reweighted Penalized Least Squares". Analyst. 2010. 135(5): 1138–1146.
23. S. He, W. Zhang, L. Liu, Y. Huang, et al. "Baseline Correction for Raman Spectra Using an Improved Asymmetric Least Squares Method". Anal. Methods. 2014. 6(12): 4402–4407.
24. SJ. Baek, A. Park, Y.J. Ahn, J. Choo. "Baseline Correction Using Asymmetrically Reweighted Penalized Least Squares Smoothing". Analyst. 2015. 140(1): 250–257.
25. Y.J. Yu, Q.L. Xia, S. Wang, B. Wang, et al. "Chemometric Strategy for Automatic Chromatographic Peak Detection and Background Drift Correction in Chromatographic Data". J. Chromatogr. A. 2014. 1359: 262–270.
26. Y. Zheng, Y.R. Fan, C. Qiu, Z. Liu, et al. "An Improved Algorithm for Peak Detection in Mass Spectra Based on Continuous Wavelet Transform". Int. J. Mass Spectrom. 2016. 409: 53–58.
27. M. Morhac. "Multidimensional Peak Searching Algorithm for Low-Statistics Nuclear Spectra". Nucl. Instrum. Meth. A. 2007. 581(3): 821–830.
28. C. Yang, Z. He, W. Yu. "Comparison of Public Peak Detection Algorithms for MALDI Mass Spectrometry Data Analysis". BMC Bioinf. 2009. 10: 4.
29. J. Christensen, L. Nørgaard, H. Heimdal, J.G. Pedersen, et al. "Rapid Spectroscopic Analysis of Marzipan-Comparative Instrumentation". J. Near Infrared Spectrosc. 2004. 12(12):63–75.

Supplemental Material

Experimental Details

In this section, more experimental details are described. In section 1.1, we present the processing results of each algorithm in simulated spectra with different SNRs. In section 1.2, we combine the proposed algorithm with other preprocessing methods, and the results are demonstrated.

DOI: 10.1177/0003702819885002

Experiments on Simulated Spectra with Different SNRs

Since there is always some noise in a measured spectrum, it will interfere with baseline correction. Hence, random noise generated by a uniform random number was added to the synthetic spectra to illustrate the applicability of mcaLS in spectra with noise. Meanwhile, in this paper, two kinds of spectra with high and low signal to noise ratios (SNRs) are designed to study the sensitivity of the algorithm to noise. The high and low SNR are 31.73db and 19.42db

Table S1 compares the RMSEs of the four algorithms for the polynomial baseline and exponential baseline under high and low SNRs. In the four experiments, the results of mcaLS all had the minimum RMSE, indicating that mcaLS also performs well in baseline correction of spectra containing noise. From the data presented in Table III, it can be seen that when the SNR of the spectra decreased by 38.8%, the RMSE of the polynomial baseline and exponential baseline only increases by 1.54 and 1.31, respectively, which proves that the algorithm has a stable effect under different SNR.

Table S1. Comparison of the RMSEs of four algorithms under high and low SNRs.

Method	RMSE (High SNR)		RMSE (Low SNR)	
	Quadratic polynomial baseline	Exponential baseline	Quadratic polynomial baseline	Exponential baseline
asLS	1.1753	0.9759	3.0241	2.3202
arPLS	0.6139	0.6175	2.1365	1.9348
SNIP	1.0952	1.1315	2.7630	2.3076
mcaLS	0.5307	0.5185	2.0675	1.8242

The correction results of mcaLS and the error of each channel in two kinds of baseline signals under a high SNR situation are shown in Fig. S1. The proportion of the error of the estimated baseline in the peak regions in the total error can be calculated according to the peak boundaries. When SNR is high, for the mcaLS processing results of the polynomial baseline and exponential baseline, the error in the peak regions accounts for 72.58%and 79.42%of the total error, respectively.

Figure S2 also shows the processing results of mcaLS in the polynomial baseline and exponential baseline under a low SNR situation. The estimated error in the peak regions

DOI: 10.1177/0003702819885002

accounted for 58.74%and.82%of the total errors respectively. This result indicates that the error of mcaLS in estimating the baseline mainly comes from the estimation of the baseline in peak regions.

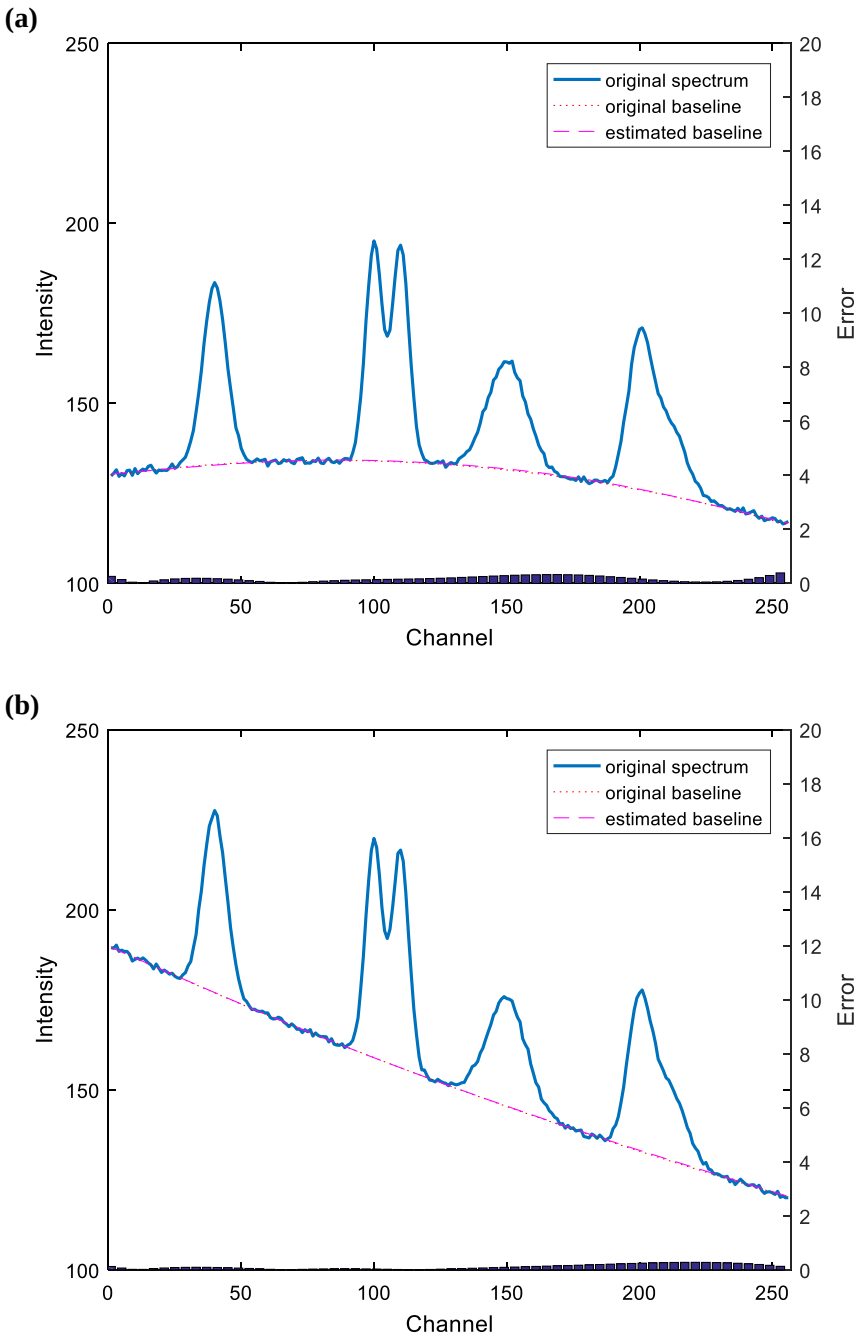


Fig. S1. Baseline estimation error of each channel under a high SNR for a (a) quadratic polynomial baseline and (b) exponential baseline.

DOI: 10.1177/0003702819885002

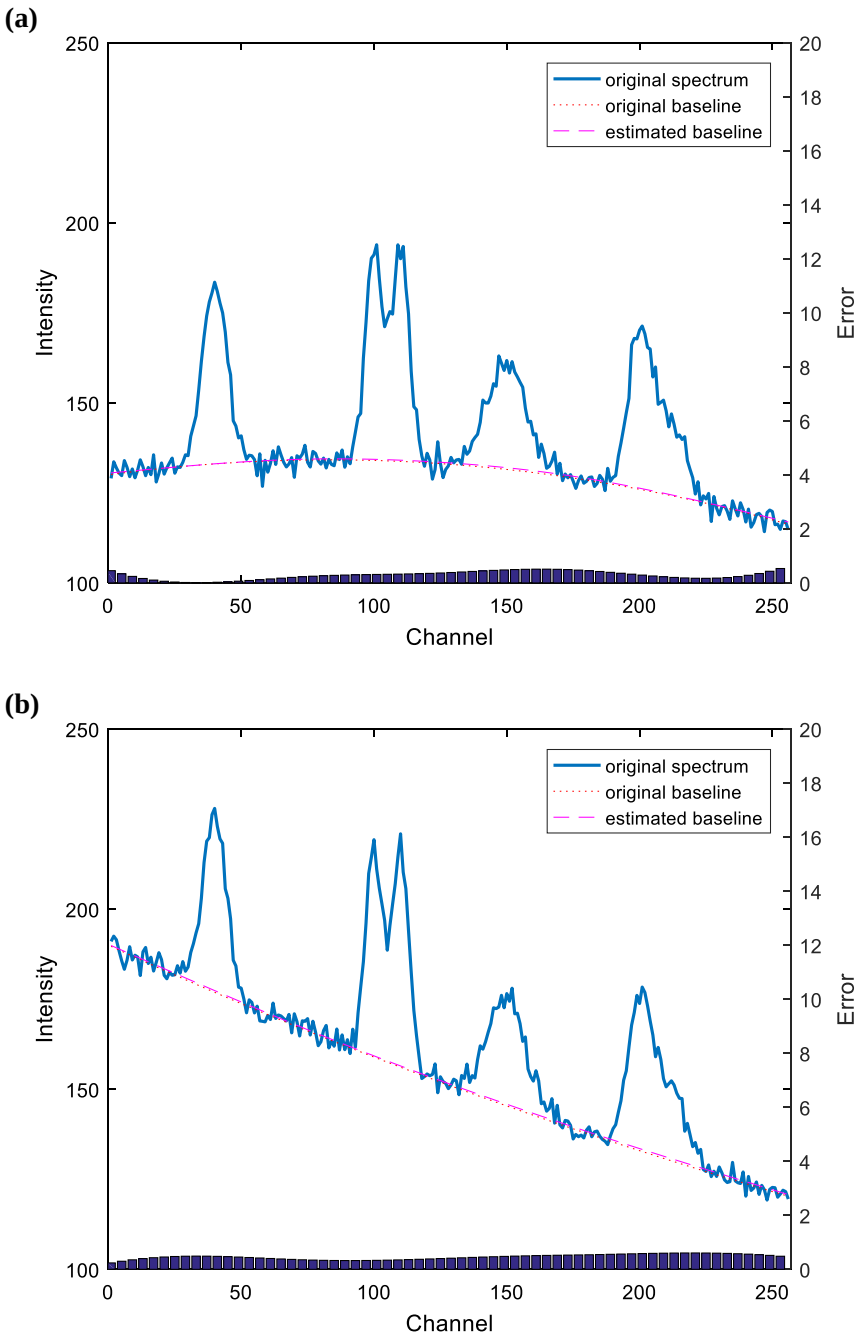


Fig. S2. Baseline estimation error of each channel under a low SNR for a (a) quadratic polynomial baseline and (b) exponential baseline.

Table S2 shows the difference between the characteristic peak heights corrected by mcaLS and the real peak heights under a high SNR and low SNR. After mcaLS was used to correct the two kinds of baselines in the high SNR situation, the maximum error was just 0.3032

(Peak 4) and 0.3092 (Peak 6), respectively. Even under the low SNR situation, the average error between the estimated peak heights and the real peak heights in two baselines reached 0.1556 and 0.2506, which showed good corrected effects. This result indicates that the use of mcaLS in baseline correction will not cause obvious changes of the spectral peak heights, thus it ensures the authenticity of the spectral peaks.

Table S2. Differences between the true and estimated peak heights under high and low SNRs.

Peak	Quadratic polynomial baseline		Exponential baseline	
	High SNR	Low SNR	High SNR	Low SNR
1	0.1558	0.4394	0.1063	0.2191
2	0.1545	0.0907	0.0057	0.1047
3	0.1693	0.0779	0.0144	0.1268
4	0.3032	0.0492	0.1201	0.2607
5	0.1521	0.0989	0.2827	0.3845
6	0.0732	0.1772	0.3092	0.4081

It can be seen from the above analysis that the estimation results of the two kinds of baselines have high accuracy when mcaLS processes spectra under different SNRs. Since the weight of data within spectral peak regions is often close to 0, it is easy for the estimated baseline to deviate from the actual baseline within spectral peak regions. With the reduction of SNR, noise will also have a certain impact on the baseline estimation. Therefore, if the noise in spectra can be effectively suppressed before baseline correction, the accuracy of baseline estimation can be further improved.

Results of the Proposed Algorithm Combined with Other Preprocessing Methods

In calibration optimization of NIR data, it is common to combine several preprocessing methods. Thus, in this section, we combined the mcaLS algorithm with other preprocessing methods and compared the results. The preprocessing methods we selected including: first derivative, second derivative, MSC and SNV. We used these algorithms alone and in combination to process real NIR spectra of corn samples. The corn data set is available from the website:

http://www.eigenvector.com/data/Corn/index.html.

The processing results are shown in Table S3. As seen from Table S3, combining mcaLS with other preprocessing methods can improve the accuracy of the spectral analysis results, especially mcaLS + the first derivative method. Moreover, the processing results of mcaLS + the first derivative method are superior to the individual processing results to a certain extent. It should be noted that the combination of different algorithms does not necessarily produce better results. Therefore, it is necessary to test before combining several preprocessing methods.

Table S3. Comparison of the RMSEP for the different spectral processing methods.

Method	Moisture		Oil		Protein		Starch	
	RMSEP	PC	RMSEP	PC	RMSEP	PC	RMSEP	PC
mcaLS	**0.1967**	10	0.1798	6	0.2808	7	0.6163	7
first derivative	0.2273	12	**0.1349**	7	0.3011	7	0.6107	3
mcaLs+SNV	0.2510	8	0.1857	5	0.3523	7	0.7559	5
mcaLs+MSC	0.2455	8	0.1858	5	0.3201	7	0.6976	6
mcaLS+ first derivative	0.2270	11	0.1355	7	**0.2709**	6	**0.5574**	3
mcaLS+ second derivative	0.3070	4	0.2146	7	0.5332	5	0.8599	2

Source Codes for the Algorithms used in the Paper

This section provides the Matlab source code of the algorithms used in the article, including asLS[1], arPLS[2], SINP[3,4], MCS, SNV and mcaLS proposed in this paper. We are grateful for the source code provided by the authors of the references.

It should be highlighted that the codes for the preprocessing methods (SNV and MCS) used in this paper are from the internet. In accordance with the annotations, the author of the code is Cleiton A. Nunes from UFLA, MG, Brazil. We thank the author for the open source code.

To use this algorithm for baseline correction, the boundary of the peak region should first be determined. As described in the article, there are two cases for this problem. First, the spectral regions can be directly determined if the characteristic peak positions and the calibration of the peak width in the spectrum are known. Gamma-ray spectra, which are commonly used in the field of geochemistry, fit this case because the characteristic energy of the element to be studied is constant and the instrument broadening response function of the detector can be obtained

through experiments. Therefore, the peak region boundaries of this kind of spectrum can be directly determined.

Second, if the spectral characteristic peak positions and peak width calibration are unknown, it is necessary to identify the peak positions and determine the peak regions. The author has listed the research results of many scholars on spectral region estimation in the references.[25–27]

Moreover, the author can also put forward a scheme for peak region estimation and provide the Matlab code. In practical processing, the valleys in the spectrum can be regarded as the boundaries of the peak regions. Therefore, the boundary determination problem can be transformed into the identification of valleys. The "*findvalleys*" function shown below can do this. We are not the original authors of this function. The original author is Tom O'Haver**,** Professor Emeritus, Department of Chemistry and Biochemistry, University of Maryland at College Park. This code is part of his published work. Fig. S4 shows the effect of the function. The valleys due to overlapping peaks can be identified easily according to their amplitudes. All the source codes of the algorithms are described as outlined below. The algorithms are compiled by Matlab, and the main parameters are explained in the annotations.

```
% asLS algorithm (by Eilers,2005)
% Parameters:
% y-- the raw spectrum
% lambda-- the smoothing parameter
% p-- the asymmetric weight
% z-- the estimated baseline
function z = AsLS( y, lambda, p )
m = length(y);
D = diff(speye(m),2);
w = ones(m,1);
for it = 1:10
    W = spdiags(w, 0, m, m);
```

DOI: 10.1177/0003702819885002

```
        % Cholesky decompoition
        C = chol(W + lambda * (D'* D));
        z = C \ (C' \ (w .* y));
        w = p * (y>z) + (1-p) * (y<z);
end

end

% arLS algorithm (by Baek,2014)
%Parameters:
% y-- the raw spectrum
% lambda-- the smoothing parameter
% p-- termination precision
% z-- the estimated baseline
function z = arLS( y, lambda, p )
N = length(y);
D = diff(speye(N),2);
H = lambda * D' * D;
w = ones(N,1);
while true
        W = spdiags(w, 0 ,N, N);
        % Cholesky decomposition
        C = chol(W + H);
        z = C \ ( C' \ (w.*y));
        d = y - z;
        % make d-, and get w^2 with m and s
        dn = d(d<0);
        m = mean(dn);
        s = std(dn);
        wt = 1./(1 + exp(2 * (d-(2*s-m))/s));
        % check exit condiion and backup
```

DOI: 10.1177/0003702819885002

```
        if norm(w-wt)/norm(w) < p, break;
        end
        w = wt;
end

end


% SNIP algorithm (by Ryan,1988)
% Parameters:
% m--the clipping window width
% Signal--the raw spectrum
% PSignal-- the spectrum after baseline correction
function Psignal = SNIP(m,Signal)
N = length(Signal);
Psignal = zeros(1,N);
v = zeros(1,N);
b = zeros(1,N);
w = zeros(1,N);
% LLS operator
for i=1:N
        v(i) = log(log(sqrt(Signal(i)+1)+1)+1);
end
% SNIP procedure
for p=1:m
        for i = p +1 : N - p
                a1 = v(i);
                a2 = (v(i-p)+v(i+p))/2;
                if a1 > a2
                        w(i) = a2;
                else
```

DOI: 10.1177/0003702819885002

```
                w(i) = a1;
            end
        end
        for i=p + 1 : N - p
            v(i) = w(i);
        end
end
% inverse LLS operator
for i=1:N
    b(i) = (exp(exp(v(i))-1)-1).^2-1;
end
% b is the estimated baseline
for i=1:N
    Psignal(i) = Signal(i) - b(i);
end

end
```

```
% mcaLS algorithm (proposed in this article)
% Parameters:
% x-- the raw spectrum
% lambda1-- the smoothing parameter
% lambda2-- the symmetry parameter
% E-- the boundary coefficient matrix
% p-- termination precision
% filter_x -- the spectrum after denoising (If the noise level in the spectrum is low, it can be set to x)
% z-- the estimated baseline
```

DOI: 10.1177/0003702819885002

```
function z = mcaLS(x, lambd1, lambd2, p, E, filter_x)
N = length(x);
D = diff(speye(N),2);
H = lambd1 * D' * D;
w = ones(N,1);
S = lambd2 * E' * E;
i=0;
while true
    W = spdiags(w, 0 ,N, N);
    % Cholesky decomposition
    C = chol(W + H + S );
    z = C \ ( C' \ (w.*x + S * filter_x));
    d = x - z;
    % make d-, and get w^2 with m and s
    dn = d(d<0);
    m = mean(dn);
    s = std(dn);
    wt = 1./(1 + exp(2 * (d-(2*s-m))/s));
    % check exit condiion and backup
    if norm(w-wt)/norm(w) < p || i==180 , break;
    end
    w = wt;
    i=i+1;
end

end
```

DOI: 10.1177/0003702819885002

```
% Standard Normal Variate
%
% [x_snv] = snv(x)
%
% input:
% x (samples x variables) data to preprocess
%
% output:
% x_snv (samples x variables) preprocessed data
%
% By Cleiton A. Nunes
% UFLA,MG,Brazil
function [x_snv] = snv(x)
[m,n]=size(x);
rmean=mean(x,2);
dr=x-repmat(rmean,1,n);
x_snv=dr./repmat(sqrt(sum(dr.^2,2)/(n-1)),1,n);
```

DOI: 10.1177/0003702819885002

```
% Multiplicative Scatter Correction
%
% [x_msc]=msc(x,xref)
%
% input
% x (samples x variables)          spectra to correct
% xref (1 x variables)             reference spectra (in general mean(x) is used)
%
% Output
% x_msc (samples x variables)   corrected spectra
%
% By Cleiton A. Nunes
% UFLA,MG,Brazil
function [x_msc]=msc(x,xref)
```

DOI: 10.1177/0003702819885002

```
[m, n]=size(x);
rs=xref;
cw=ones(1,n);
mz=[];mz=[mz ones(1,n)'];mz=[mz rs'];
[mm,nm]=size(mz);
wmz=mz.*(cw'*ones(1,nm));
wz=x.*(ones(m,1)*cw);
z=wmz'*wmz;
[u,s,v]=svd(z);sd=diag(s)';
cn=10^12;
ms=sd(1)/sqrt(cn);
cs=max(sd,ms );
cz=u*(diag(cs))*v';
zi=inv(cz);
b=zi*wmz'*wz';B=b';
x_msc=x;
p=B(:,1);x_msc=x_msc-(p*ones(1,mm));
p=B(:,2);x_msc=x_msc./(p*ones(mm,1)');
```

```
function V=findvalleys(x,y,SlopeThreshold,AmpThreshold,smoothwidth,peakgroup,smoothtype)
% function
```

DOI: 10.1177/0003702819885002

```
P=findvalleys(x,y,SlopeThreshold,AmpThreshold,smoothwidth,peakgroup,smoothtype)
% Function to locate the valleys (mimnima) in a noisy x-y time series data
% set.    Detects valleys by looking for upward zero-crossings
% in the first derivative that exceed SlopeThreshold.
% Returns list (V) containing valley number and position,
% depth, and width of each valley. Arguments "slopeThreshold",
% "ampThreshold" and "smoothwidth" control sensitivity.
% Higher values will neglect smaller features. "Smoothwidth" is
% the width of the smooth applied before valley detection; larger
% values ignore narrow features. "Peakgroup" is the number points
% around the bottom part of the valley that are fit to a parabola to
% determine the valley vertex (x and y at lowest point) and width.
% The argument "smoothtype" determines the smooth algorithm:
%      If smoothtype=1, rectangular (sliding-average or boxcar)
%      If smoothtype=2, triangular (2 passes of sliding-average)
%      If smoothtype=3, pseudo-Gaussian (3 passes of sliding-average)
% T. C. O'Haver, Version 3.1, June, 2013
%
% Permission is hereby granted, free of charge, to any person obtaining a copy
% of this software and associated documentation files (the "Software"), to deal
% in the Software without restriction, including without limitation the rights
% to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
% copies of the Software, and to permit persons to whom the Software is
% furnished to do so, subject to the following conditions:
%
% The above copyright notice and this permission notice shall be included in
% all copies or substantial portions of the Software.
%
% THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR
% IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
```

```
MERCHANTABILITY,
% FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
% AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
% LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
% OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
% THE SOFTWARE.
if nargin~=7;smoothtype=1;end    % smoothtype=1 if not specified in argument
if smoothtype>3;smoothtype=3;end
if smoothtype<1;smoothtype=1;end
smoothwidth=round(smoothwidth);
peakgroup=round(peakgroup);
d=fastsmooth(deriv(y),smoothwidth,smoothtype);
n=round(peakgroup/2+1);
V=[0 0 0 0 0];
vectorlength=length(y);
peak=1;
AmpTest=AmpThreshold;
for j=smoothwidth:length(y)-smoothwidth,
    if sign(d(j)) < sign (d(j+1)), % Detects zero-crossing
        if d(j+1)-d(j) > SlopeThreshold, % if slope of derivative is larger than SlopeThreshold
            if y(j) > AmpTest,    % if height of valley is larger than AmpThreshold
                xx=zeros(size(peakgroup));yy=zeros(size(peakgroup));
                for k=1:peakgroup, % Create sub-group of points near valley
                    groupindex=j+k-n+1;
                    if groupindex<1, groupindex=1;end
                    if groupindex>vectorlength, groupindex=vectorlength;end
                    xx(k)=x(groupindex);yy(k)=y(groupindex);
```

DOI: 10.1177/0003702819885002

```
                    end
 %                    sizexx=size(xx)
     %                sizeyy=size(yy)
                    [coef,S,MU]=polyfit(xx',yy',2);    % Fit parabola to sub-group with centering
and scaling
                    c1=coef(3);c2=coef(2);c3=coef(1);
                    valleyX=-((MU(2).*c2/(2*c3))-MU(1));       % Compute valley position and
height of fitted parabola
                    valleyY=(c1-(c2*c2/(4*c3)));
                    MeasuredWidth=norm(MU(2).*2.35482/(sqrt(2)*sqrt(-1*c3)));
                    % if the valley is too narrow for least-squares technique to work
                    % well, just use the min value of y in the sub-group of points near valley.
                    if peakgroup<5,
                        valleyY=min(yy);
                        pindex=val2ind(yy,valleyY);
                        valleyX=xx(pindex(1));
                    end
                    % Construct matrix P. One row for each valley
                    % detected, containing the valley number, valley
                    % position (x-value) and valley depth (y-value).
                    if isnan(valleyX) || isnan(valleyY),
                    else
                        V(peak,:) = [round(peak) valleyX valleyY MeasuredWidth 0];
                        peak=peak+1;
                    end
                end
            end
        end
    end
% --------------------------------------------------------------------
function [index,closestval]=val2ind(x,val)
```

DOI: 10.1177/0003702819885002

```
% Returns the index and the value of the element of vector x that is closest to val
% If more than one element is equally close, returns vectors of indicies and values
% Tom O'Haver (toh@umd.edu) October 2006
% Examples: If x=[1 2 4 3 5 9 6 4 5 3 1], then val2ind(x,6)=7 and val2ind(x,5.1)=[5 9]
% [indices values]=val2ind(x,3.3) returns indices = [4 10] and values = [3 3]
dif=abs(x-val);
index=find((dif-min(dif))==0);
closestval=x(index);

function d=deriv(a)
% First derivative of vector using 2-point central difference.
%    T. C. O'Haver, 1988.
n=length(a);
d(1)=a(2)-a(1);
d(n)=a(n)-a(n-1);
for j = 2:n-1;
   d(j)=(a(j+1)-a(j-1)) ./ 2;
end

function SmoothY=fastsmooth(Y,w,type,ends)
% fastbsmooth(Y,w,type,ends) smooths vector Y with smooth
%    of width w. Version 2.0, May 2008.
% The argument "type" determines the smooth type:
%      If type=1, rectangular (sliding-average or boxcar)
%      If type=2, triangular (2 passes of sliding-average)
%      If type=3, pseudo-Gaussian (3 passes of sliding-average)
% The argument "ends" controls how the "ends" of the signal
% (the first w/2 points and the last w/2 points) are handled.
%      If ends=0, the ends are zero.    (In this mode the elapsed
%         time is independent of the smooth width). The fastest.
%      If ends=1, the ends are smoothed with progressively
```

DOI: 10.1177/0003702819885002

```
%         smaller smooths the closer to the end. (In this mode the
%         elapsed time increases with increasing smooth widths).
% fastsmooth(Y,w,type) smooths with ends=0.
% fastsmooth(Y,w) smooths with type=1 and ends=0.
% Example:
% fastsmooth([1 1 1 10 10 10 1 1 1 1],3)= [0 1 4 7 10 7 4 1 1 0]
% fastsmooth([1 1 1 10 10 10 1 1 1 1],3,1,1)= [1 1 4 7 10 7 4 1 1 1]
%    T. C. O'Haver, May, 2008.
if nargin==2, ends=0; type=1; end
if nargin==3, ends=0; end
   switch type
      case 1
          SmoothY=sa(Y,w,ends);
      case 2
          SmoothY=sa(sa(Y,w,ends),w,ends);
      case 3
          SmoothY=sa(sa(sa(Y,w,ends),w,ends),w,ends);
   end

function SmoothY=sa(Y,smoothwidth,ends)
w=round(smoothwidth);
SumPoints=sum(Y(1:w));
s=zeros(size(Y));
halfw=round(w/2);
L=length(Y);
for k=1:L-w,
    s(k+halfw-1)=SumPoints;
    SumPoints=SumPoints-Y(k);
    SumPoints=SumPoints+Y(k+w);
end
s(k+halfw)=sum(Y(L-w+1:L));
```

```
SmoothY=s./w;
% Taper the ends of the signal if ends=1.
   if ends==1,
      startpoint=(smoothwidth + 1)/2;
      SmoothY(1)=(Y(1)+Y(2))./2;
      for k=2:startpoint,
          SmoothY(k)=mean(Y(1:(2*k-1)));
          SmoothY(L-k+1)=mean(Y(L-2*k+2:L));
      end
      SmoothY(L)=(Y(L)+Y(L-1))./2;
   end
% ------------------------------------------------------------
```

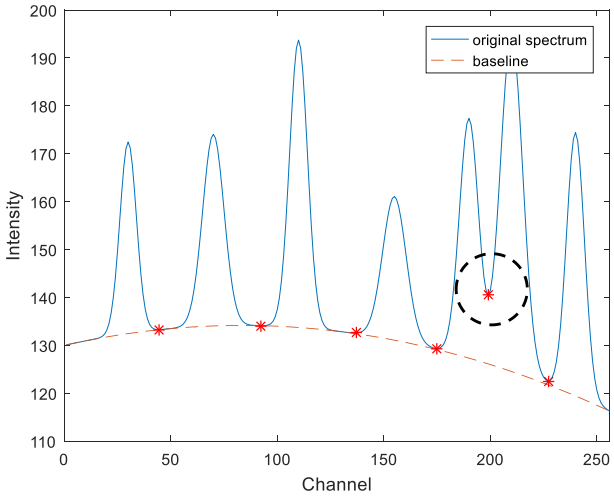


Fig. S4. Processing effect of the *findvalleys* function.