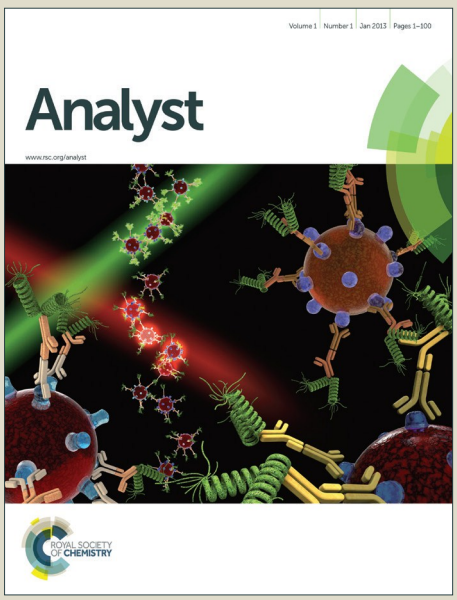
Analyst

Accepted Manuscript



ROYAL SOCIETY OF CHEMISTRY

www.rsc.org/analyst

Journal Name

RSCPublishing

ARTICLE

A Concise Iterative Method with Bezier Technique for Baseline Construction

Cite this: DOI: 10.1039/xoxxoooooox

Y.J. Liu*, X.G. Zhou and Y. D. Yu

Received ooth January 2012,
Accepted ooth January 2012

DOI: 10.1039/xoxxoooooox

www.rsc.org/

A novel approach, coined the Corner-Cutting method (CC for short), is presented in this paper which affords to efficiently construct the baseline for analytical data streams. It was derived from techniques used in computer aided geometric design, a field established to produce curves and surfaces for aviation and automobile industry. This corner-cutting technique provided a very efficient baseline calculation through an iterative process. Furthermore, a terminal condition was developed to make the process fully automated and truly non-parametric. Finally, we employed Bezier curve to convert iterating result into smooth baseline solution. Comparing to other iterative schemes used for baseline detection, our method was significantly efficient, easier to implement, and with broader range of applications.

Introduction

Baseline correction of digitized signals has universal interests with broad range of applications in data processing of analytical instruments. Various methods to correct the drift with signal reception have been developed. Shirley background[1], a baseline construction method published in 1972 for X-Ray Photoemission Spectra, has been extensively used in numerous projects. Since then, other approaches have also been developed to decompose data stream into combination of series of waves, treating the baseline as low frequency oscillation, and apply specific filter or space/spectra decomposition to achieve detection of baseline [2] [3] [4]. There are also other methods that use iteration to reach a solution agreed with main trend of signals [5] [6] as well as methods where baseline is produced by interpolating key data points along traces [7] [8] [9] where key points are determined either through direct space spectra or wavelet analysis. Both commercial and open source software packages to carry out baseline correction are available. Among them, two publicly available algorithms, the CROMWELL[10] and the LIMPIC[8], are widely used during their era, while airPLS[12] and AMIA[6] represents the state-of-the-art performance. In CROMWELL, baseline is constructed through linear interpolation of selective minimum values designated as key points. The LIMPIC uses a wavelet method to locate signal regions void of peaks and constructs the baseline by connecting the resulting data points with straight lines. Our new method is largely inspired and followed the concepts of these two. Both algorithms were used as benchmarks for comparison throughout the paper, however the performance will be evaluated on each. The method we developed improved upon existing approaches by using an iterative process, we termed corner-cutting, to select key points for baseline construction. It proved to be fast, effective, and, at the same time, non-parametric. The whole baseline construction process completes without user input. It benefited from concepts used in Computer Aided Geometric Design, consisting of curve construction and Bezier technique[11]. In the following sections, this new approach was compared with benchmark methods, including open source software (LIMPIC, CROMWELL, airPLS, AIMA) and datasystems from popular commercial manufactures (Thermo, Agilent, Ciphergen), demonstrating its advantage and utility.

Related Works

CROMWELL and LIMPIC

Shirley background[1] is a straightforward method and uses the following formulation (1) to iteratively construct the baseline.

$$I(x) = I^{''}(0)\frac{\int_x^{\infty} I'(t)dt}{\int_0^x I'(t)dt} \qquad (1)$$

However, this method is limited to work with cases where baselines decline with x. For data that go up and down or fluctuate in waves rather than simple decline, it does not work quite well. Alternative method, such as publicly available CROMWELL algorithm, was also developed, which extracts baseline using the minimum value interpolation[10]. This method suffers similar constrain as Shirley background, producing a monotonically decreasing baseline. Another method, the LIMPIC algorithm, uses wavelet techniques to produce a dental baseline[8] which shows more robustness and

broader applications. We compared both methods with our own approach in the following.

The minimum value interpolation carried out in CROMWELL works as follows: take the starting point as the first point of baseline and, then, loop the data from the beginning to the end. Data point which is smaller than or equal to the last point of baseline is taken as the new baseline end point and connected to the previous one. After the entire data stream is processed, it produces a final polyline which is used as the baseline. This approach suffers the same problem as Shirley background subtraction due to the fact that they both assume the baseline monotonically decreasing and, therefore, has limited utility.

The LIMPIC uses wavelet to identify regions in data where no meaningful signal peaks exist. The remaining data points in those regions are connected successively to produce the baseline. This method has been shown to produce better results than CROMWELL and other commercial packages such as APEX and CENTROID[8].

State-of-the-art methods, airPLS[12] and AIMA[6] are also iterative approaches. Both of them are developed in recent years and have been used and cited extensively, and will be used as main comparison to this work. At the same time, we also compared the performance with popular commercial software such as Thermo OMNIC for Raman spectra, Agilent Chemstation for LCMS data, and Ciphergen for MALDI-TOF data.

Above-mentioned methods can generally be categorized into two classes: one that uses iterative steps to build baseline as the Shirley background subtraction[1][5][6] and others that uses key point interpolation[3][8][10]. Our approach described next combined the iterative steps with key point interpolation: using iteration to obtain data points tracking baseline trend and using interpolation to construct the final baseline. In both steps, Bezier techniques were utilized.

Bezier Curve

Bezier techniques provide a geometric-based method for describing and manipulating polynomial curves and surfaces, transforming complicated mathematical concepts into a highly geometric and intuitive form. Following the expert's advice, we have realized that as early as about 20 years ago, Koch and Weber[13] had introduced the idea using Bezier technique to interactively draw baseline. In the paper, the authors explained the advantage of using Bezier curve in hand drawing of baseline: permits the drawing in a manner that allows full control of the line curvatures by the user. We have the same perception of Bezier's benefits and developed an automatic scheme utilizing the Bezier technique to produce smooth baseline. In this section, we review this classical technique first.

In the Bezier method, modeling is primarily carried out with parametric curves. The simple quadratic function, written as a 2D parametric curve, takes the form

$$\mathbf{p}(t) = \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} x(t) \\ y(t) \end{bmatrix} = \begin{bmatrix} t \\ 1 - t + t^2 \end{bmatrix}, t \in R$$

Each coordinate is a function of parameter t, and the real line is the domain of the curve. A concise form with notation for points and vectors can be written as

$$\mathbf{p}(t) = \boldsymbol{a_0} + \boldsymbol{a_1} t + \boldsymbol{a_2} t^2, t \in R$$

where

$$\boldsymbol{a_0} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \boldsymbol{a_1} = \begin{bmatrix} 1 \\ -1 \end{bmatrix}, \boldsymbol{a_2} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

The a_is are coefficients of the curve and $1, t, t^2$ are quadratic monomial basis functions.

The monomial form above is one way to represent a polynomial curve. However, it does not provide the most intuitive way to view the data geometrically. A better formulation comes from the Bernstein basis functions which form the building block for Bezier curves. The quadratic Bezier takes the form

$$\mathbf{p}(t) = \boldsymbol{b_0} B_0^2(t) + \boldsymbol{b_1}\, B_1^2(t) + \boldsymbol{b_2} B_2^2(t), t \in R$$

where

$$\boldsymbol{b_0} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \boldsymbol{b_1} = \begin{bmatrix} 1 \\ -1 \end{bmatrix}, \boldsymbol{b_2} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

representing a group of Bezier control points, and

$$B_0^2(t) = (1-t)^2, B_1^2(t) = 2t(1-t), B_2^2(t) = t^2$$

being the quadratic Bernstein polynomials or basis functions. A common operation is to evaluate Bezier curves for $t \in [0, 1]$, although it is defined for all t over the reals.

A n degree Bezier curve takes the form

$$p(t) = \sum_{i=0}^{n} \boldsymbol{b_i} B_i^n(t) \qquad t \in [0,1]$$

where $B_i^n = C_i^n (1-t)^{n-i} t^i$, being the degree n Bernstein polynomials, and the binomial coefficients defined as $C_n^i = \frac{n!}{(n-i)!i!}$. The polygon formed by the control points $\boldsymbol{b_i}$ is called control polygon.

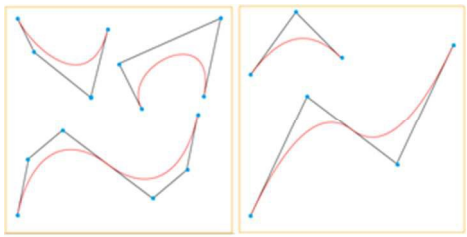


Figure-1 Bezier curves and the control points (blue dots), control polygons (the black polylines connecting control points as vertices are called control polygon in computer aided geometric design).

Figure-1 exhibits a few Bezier curves, demonstrating the relationship between the curves and their control polygons. Bezier curve possesses superior properties, many of which are apparent from Figure-1:

- The curve passes through polygon endpoints.
- The tangents at the ends are parallel to the polygon legs.

These properties afford this method a very efficient interpolation to generate smooth curve from designated key points.

Besides the algebraic construction, Bezier curve can also be built geometrically. It is a procedure to approximate the control polygon into a smooth curve step by step [11]. This "cut away" manner inspired us to develop the corner-cutting baseline construction method described next.

Corner-Cutting (CC) method

Simple Corner-Cutting Algorithm

As with most iterative methods, this new approach constructed baseline through approximation step by step. The key difference was that it was totally non-parametric, meaning no prior assumption needs to be made, and, therefore, was more robust.

Given an array

$$y = \{y_0, y_1, y_2, \cdots, y_n\},$$

representing a data stream with isometric timeline, a data trace as 2-dimensional data points can be written as $\{(x_0, y_0), (x_1, y_1), (x_2, y_2), \cdots, (x_n, y_n)\}$ (where $x_i = i$). For each$(x_i, y_i)(0 < i < n)$, consider the triplet group

$$C_i = \{(x_{i-1}, y_{i-1}), (x_i, y_i), (x_{i+1}, y_{i+1})\},$$

when (x_i, y_i) is above the line connecting point (x_{i-1}, y_{i-1}) and (x_{i+1}, y_{i+1}), i.e.

$$y_i > \frac{y_{i+1} - y_{i-1}}{x_{i+1} - x_{i-1}}(x_i - x_{i-1})$$

C_i is designated as a corner. In geometric term, the corner is a point protuberant from the neighbours, as shown in Figure-2.

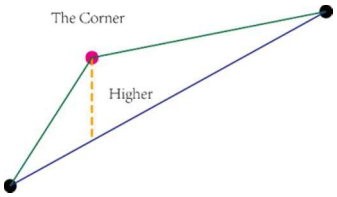


Figure-2 The corner is a point protuberant from the neighbours.

Our incipient strategy is simply eliminating the corners iteratively. Steps of the algorithm are presented in the following algorithm:

Input	**Data stream with length n: $y = [y_0, y_1, \cdots, y_{n-1}, y_n]$**
output	**Baseline $\tilde{y} = [\tilde{y}_0, \tilde{y}_1, \cdots, \tilde{y}_{n-1}, \tilde{y}_n]$**
Step1	Establish a 2-dimension point list $p = [p_0, p_1, \cdots, p_{n-1}, p_n]$, where $p_i = (i, y_i)$.
Step 2	Check every element in p, if p_i is a corner, mark it.
Step 3	Eliminate all the marked elements in p.
Step 4	If p contains no corner point, move to Step 5, otherwise, repeat Step2 and Step 3.
Step 5	Calculate $\tilde{y}_i (i = 0, 1, \cdots, n)$ according to the obtained $p = [(x_{k0}, y_{k0}), (x_{k1}, y_{k1}), \cdots, (x_{km}, y_{km})]$, that is, if $x_{kj} \le i \le x_{kj+1}$, then $\tilde{y}_i = y_{kj} + \frac{y_{kj+1} - y_{kj}}{x_{kj+1} - x_{kj}}(i - x_{kj})$. $\tilde{y} = [\tilde{y}_0, \tilde{y}_1, \cdots, \tilde{y}_{n-1}, \tilde{y}_n]$ represents the baseline constructed.

Algorithm 1

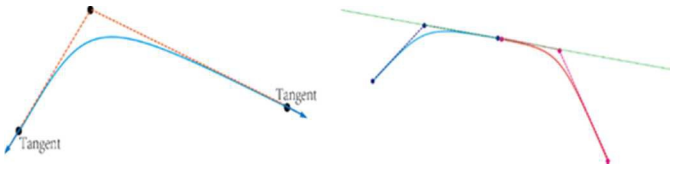
We coined this iterative process Corner-Cutting method or CC in short.

Smooth Baseline Construction

The result from Algorithm 1 is simply a polyline with sharp corners. To produce a smooth baseline, we applied the Bezier technique. This technique is effective to produce elegant smooth curve according to a few discrete points (also concluded in [13]).

As introduced previously, an N-order smooth Bezier curve can be built with N+1 control points. Therefore, 3 data points were needed to construct a 2-order curve with the tangent coincide to control polygon edge at each end (Figure-3, a). To build a continuous smooth curve from a series of control points, we

only needed to ensure the control edges sharing the same end point to lie on the same line (Figure-3, b).



(a) (b)

Figure-3 (a) The tangent coincide to control polygon edge at each end. (b) The control edges sharing the same end point lie on the same line will keep the joint curve smooth.

Our strategy is to add middle point in each control segment from the second to the last but one (Figure-4).

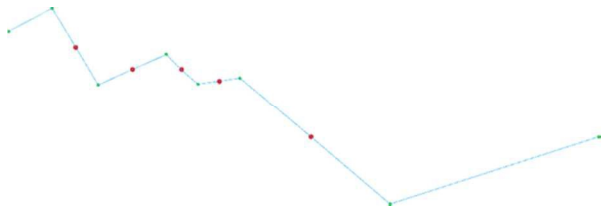


Figure-4 Add middle point in each control segment.

As result, the Bezier curve built with tri-points segments were derivative continuous at connections, and the entire curve was smooth (Figure-5).

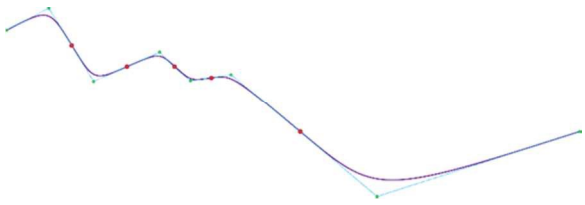


Figure-5 The Bezier curve built tri-points is smooth.

Augmented with the Bezier antialiasing, the revised Algorithm 1 produced a smooth baseline for a model data set as shown in Figure-6.

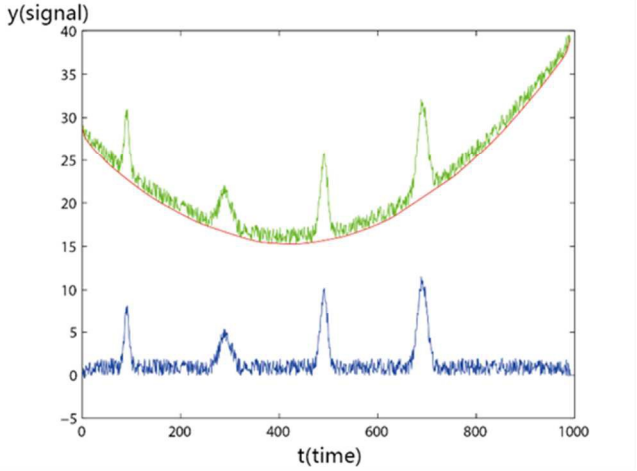
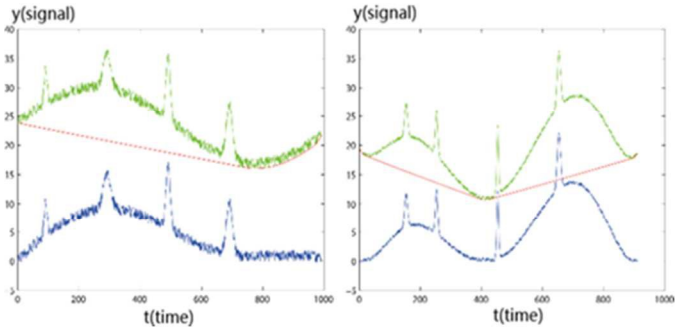


Figure-6 Augmented with the Bezier antialiasing, smooth baseline is obtained.

Terminal Condition

Algorithm 1 with the Bezier smoothing produced a baseline as shown above. However, it still suffered an apparent limitation — always generated a convex curve.



(a) (b)

Figure-7 Algorithm-1 always generates convex baselines (both in (a) and (b): the green curves are original data, red curves are baseline produced by Algorithm-1, blue curves are the results after baseline removal).

This was fine where baseline trend was actually convex. However, it failed if the baseline was concave, fluctuating, or more complex. In those cases, the iteration stopped too late to produce results as shown in Figure-7. To address this problem, we devised a terminal condition where iterative process stops based on area variation.

Given a curve represented by a sequence of 2-dimensional points

$$\boldsymbol{P} = \{(x_1, y_1), (x_2, y_2), \cdots, (x_n, y_n)\}$$

define $A = \frac{1}{2}\sum_{i=0}^{n-1}(x_{i+1} - x_i)(y_{i+1} - y_i)$ as the area of P. This definition describes the calculus of the polyline with P as vertices. In Algorithm 1, the area under curve P decreases with every iteration of corner-cutting as set of points are eliminated. The average reduction in area as result of the elimination was

an indicative of iteration progress. We define the "elimination ratio" (ER) as

$$\mathrm{ER}_i = \frac{A_i - A_{i-1}}{N_i}$$

where A_i and N_i, respectively, represents the area and number of eliminated points at the ith iteration. Intuitively, ER can be described as following.

The elimination ratio represents the size of triangles that get eliminated from each iteration. Higher ER means larger triangles are eliminated. In corner-cutting iteration, high intensity signals get eliminated first in early iterations. As iteration progresses, remaining data points are becoming predominantly baseline signal. Consequently, the portion of area being removed at each corner-cutting becomes more significant in comparing with that of previous iteration, resulting in higher ER value. As such, we chose our terminal condition to be the iteration which resulted in the maximal ER. The ER plot for the data set in Figure-7b is shown in Figure-8.

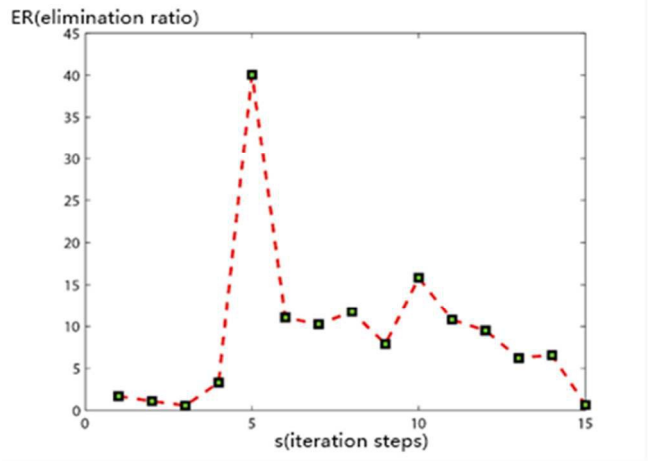


Figure-8 The ER curve for the data set in Figure-7b.

Figure 8 reveals that there are several local maxima. The iteration with the highest ER represents the emergence of baseline and is chosen to establish an effective terminal condition in our algorithm. This enables us to implement a fully-automated process for baseline construction with no user input required. The revised algorithm with the terminal condition is outlined below:

Input Data stream with length n: $\boldsymbol{y} = [\boldsymbol{y_0}, \boldsymbol{y_1}, \cdots, \boldsymbol{y_{n-1}}, \boldsymbol{y_n}]$

output Baseline $\tilde{\boldsymbol{y}} = [\tilde{\boldsymbol{y}}_0, \tilde{\boldsymbol{y}}_1, \cdots, \tilde{\boldsymbol{y}}_{n-1}, \tilde{\boldsymbol{y}}_n]$

Step1 Establish a 2-dimension point list

$p = [p_0, p_1, \cdots, p_{n-1}, p_n]$, where $p_i = (i, y_i)$.

Set an array T with length n and every element 0 as initial value.

Set a list ERlist to record the Elimination Ratio of each loop.

Step 2 Check every element in p, if p_i is a corner, mark it.

Step 3 Eliminate all the marked elements in p, let $T[i] = T[i] + 1$ if p_i is not eliminated.

Step 4 Calculate Elimination Ratio according to the marked eliminated points

$$p = [(x_{k0}, y_{k0}), (x_{k1}, y_{k1}), \cdots, (x_{km}, y_{km})]$$

that is, $\mathrm{ER} = \sum_{i=1}^{m}(x_{k0} - x_{k0-1})(y_{k0} + y_{k0-1}) + (x_{k0+1} - x_{k0})(y_{k0+1} + y_{k0}) - (x_{k0+1} - x_{k0-1})(y_{k0+1} + y_{k0-1})$. Add ER to the ERlist.

Step 5 If p contains no corner point, move to Step 6, otherwise, repeat Step2 to Step 4.

Step 6 If the kth element is the maximum one of ERlist, select the points p_i in p, where $T[i]$ equals to k.

Step 7 Calculate $\tilde{\boldsymbol{y}}_{\boldsymbol{i}}(\boldsymbol{i} = \boldsymbol{0}, \boldsymbol{1}, \cdots, \boldsymbol{n})$ according to the obtained $\boldsymbol{p} = [(\boldsymbol{x_{k0}}, \boldsymbol{y_{k0}}), (\boldsymbol{x_{k1}}, \boldsymbol{y_{k1}}), \cdots, (\boldsymbol{x_{km}}, \boldsymbol{y_{km}})]$, that is, if $\boldsymbol{x_{kj}} \le \boldsymbol{i} \le \boldsymbol{x_{kj+1}}$, then $\tilde{\boldsymbol{y}}_{\boldsymbol{i}} = \boldsymbol{y_{kj}} + \frac{\boldsymbol{y_{kj+1}} - \boldsymbol{y_{kj}}}{\boldsymbol{x_{kj+1}} - \boldsymbol{x_{kj}}}(\boldsymbol{i} - \boldsymbol{x_{kj}})$. $\tilde{\boldsymbol{y}} = [\tilde{\boldsymbol{y}}_0, \tilde{\boldsymbol{y}}_1, \cdots, \tilde{\boldsymbol{y}}_{n-1}, \tilde{\boldsymbol{y}}_n]$ is the baseline acquired.

Algorithm 2

When applied to the undulating data trace in Figure-7b, the algorithm produced a smooth baseline as shown in Figure-9.

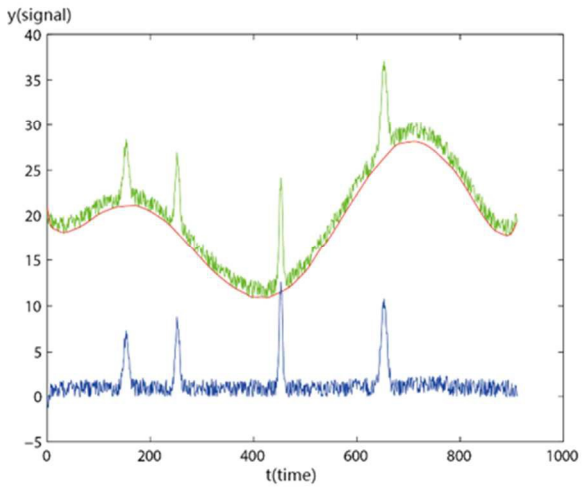


Figure-9 A correct baseline is produced by Algorithm 2.

Summary

Besides methods mentioned above, there are other modified polynomial fitting techniques for baseline building[14 15]. In 2005, Georg Schulze et al. [16] published an excellent survey on

representative baseline detection algorithms. Other methods were also developed over the past couple of years[17 18]. Our new method is essentially an iterative method just like AIMA and airPLS. It starts a loop scanning through the whole trace to reach an expected situation. However, comparing with these previously published methods, the biggest difference is that CC method is derived from geometric view point, no direct optimization criteria is needed to guide the iteration. This property leads to more concise scheme which makes the iterating procedure much more straight forward. The user is even allowed to visually observe the result in each iterating step and intervene with tie progress if desired. Another benefit of this geometric processing is that it performs better for low signal-to-noise ratio data since the higher noise is more likely to be truncated away.

In the following section we will illustrate these features through practical examples.

Evaluation of CC algorithm

In this section, a few different types of spectroscopic and spectrometric data obtained publically were used to demonstrate the superior performance of the CC method for baseline construction. The sources and contents of these data are listed in Table 1:

Data source	*Specific information*
Raman Spectroscopy	Spectra from Romanian Database of Raman Spectroscopy, including Kaliborite, Agardite and Amazonite
LC-MS	These spectra files are downloaded from the website provided by the author of [6]. According to the author, these data are LC-MS subset of the data from 200-600 m/z and 2500-4500 seconds of the spinal cords of 6 wild-type(wt) and 6 FAAH knockout(ko) mice taken from Saghatelian A, Trauger SA, Want EJ, Hawkins EG, Siuzdak G, Cravatt BF: Assignment of endogenous substrates to enzymes by global metabolite profiling. Biochemistry 2004, 43(45):14332-14339.
MALDI-TOF	The data consist of several sets of MALDI-TOF mass spectra containing bacteria protein analysis. For the first part of testing, there are two sets of data with 32 spectra from Bowman bacteria and 80 spectra from Salmonella. For the second part of testing, the details are listed in Table-6.

Table-1 Experimental data used to evaluate CC algorithm

Raman Spectroscopic Data

CROMWELL, LIMPIC, airPLS, AIMA and CC, baseline detection methods as well as the mainstream commercial software OMNIC provided by Thermo were applied to a set of Raman spectroscopic data collected for various minerals and stones with varying degree of undulation, and the results are shown in Figure-10, Figure-11, Figure-12 and Figure-13.

As previously discussed, CROMWELL uses minimum value interpolation to generate baseline and works only when data points trend down. In regions where data points trending upwards, this algorithm failed to produce proper baseline, leading to nearly straight lines parallel to the x-axis for the test data (in Figure 10, 11, 12, and 13). LIMPIC produced somewhat better results in regions where data do not satisfy the monotonically decreasing condition but still did not properly track the baseline. In contrast, the CC algorithm as well as airPLS and AIMA generated smooth curves that track well with the baseline in all cases.

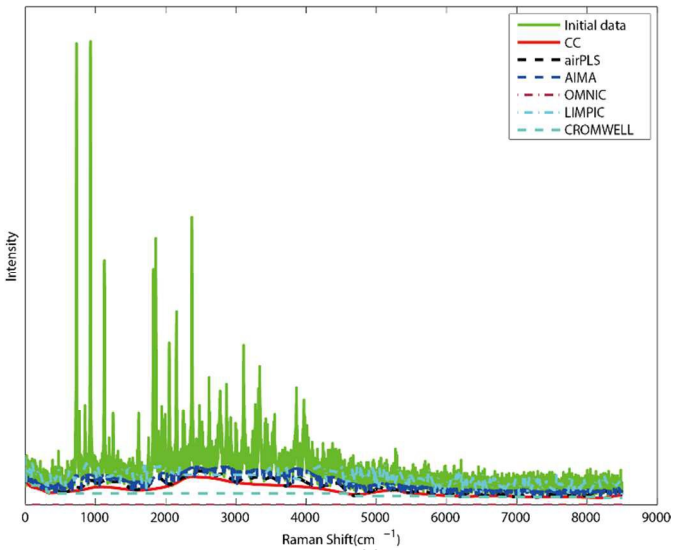


Figure-10 Kaliborite.

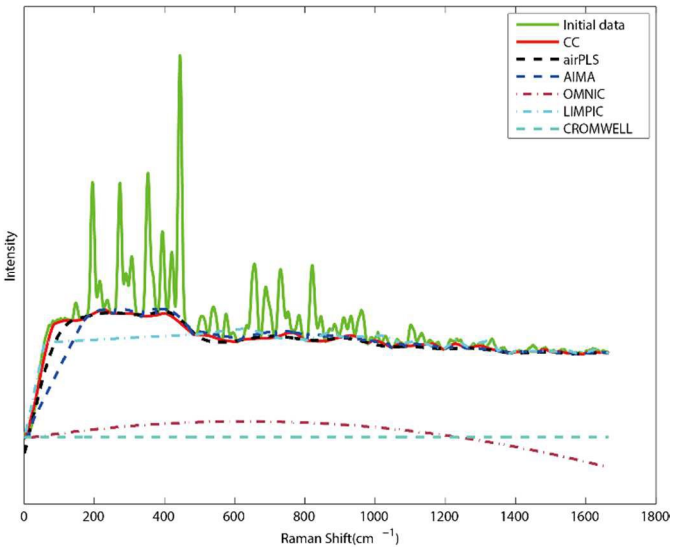


Figure-11 Agardite.

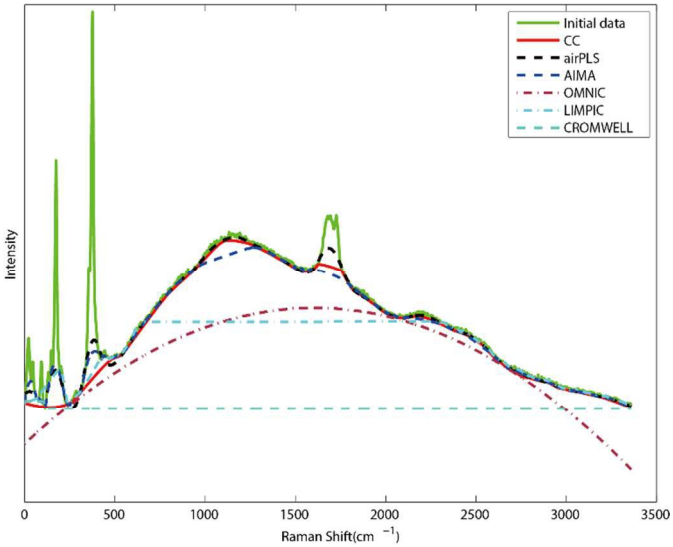


Figure-12 Agardite.

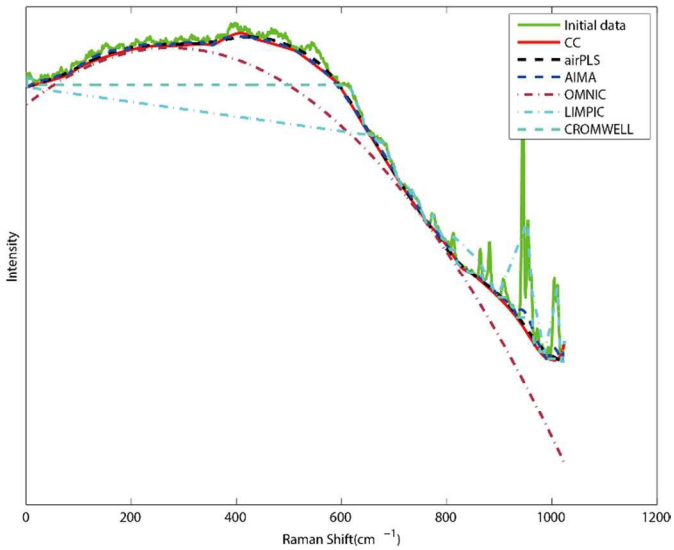


Figure-13 Amazonite.

Quantitate Measurement on Classification Distance in SVM

There is no generally accepted criterion to quantify performance of baseline subtraction. However, an approach based on principle component analysis (PCA) has been previously utilized [6] to evaluate the effectiveness of baseline correction. This approach can be summarized as the following:

1. Choose a set of traces generated from the same source, denoted as set α;
2. Apply baseline removal on α, resulting in processed set, denoted as β;
3. Use PCA on both sets and reduce the dimension of the traces into 2-dimemsion, obtain two planar point sets α' and β';
4. Calculate the area of convex hull of α' and denote as $\mathcal{A}$. Do the same thing on β' and denote the area as $\mathcal{B}$;
5. The ratio of the two areas $\mathcal{B}/\mathcal{A}$ is calculated and used as the evaluator for baseline correction results: smaller ratio indicates improved baseline subtraction result.

The whole process is shown as a schematic in Figure-14.

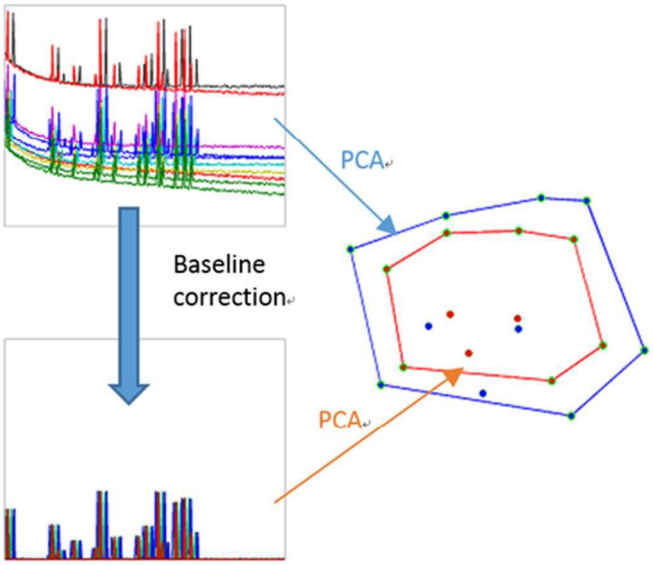


Figure-14 Sketch of previous evaluation method based on PCA.

The rationale behind this process is that if discrepancy between data sets obtained for the same analytics is primarily due to baseline interferences, the effective removal of them ought to increase similarity in analytical results, which should be reflected in a tighter PCA cluster.

The method, however, has an intrinsic flaw—a smaller ratio calculated might not be the result of effective baseline removal. As a case in point, when data sets in the group consists mostly of baseline signals, the baseline removal would result in a near zero subtracted traces for all analytics. The subsequent PCA analysis would cluster well, producing a false optimum testing condition and leading to an erroneous conclusion. To circumvent this problem, we here propose a different approach

to quantify baseline detection result by measuring distance between hyper-planes after performing SVM (Support Vector Machines) classification. The essence of this technique can be epitomized as separating two kinds of point sets by a pair of parallel hyper-planes. The greater the distance between them, the better the classification result and, therefore, better baseline removal.

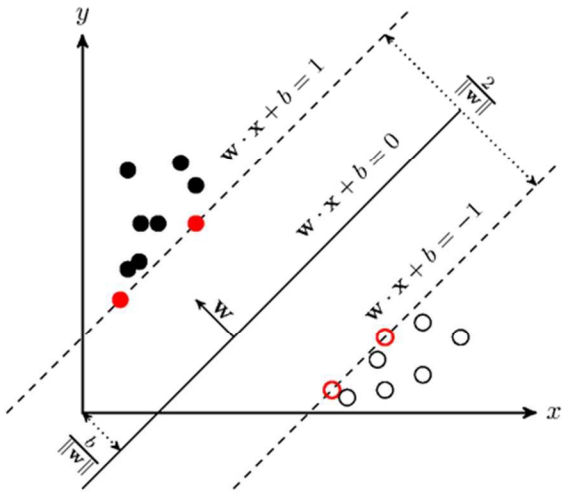


Figure-15 SVM classification method.

The tactic to maximize the margin is one of the keys for successful SVM analysis. The plane $wx + b = 1$ and $wx + b = -1$ shown in above figure are classification hyper-planes, and $2/\|w\|$ is the margin between the two. The aim of SVM analysis on baseline detection result was to increase this margin, the equivalent of minimizing $\|w\|$. The resulting $\|w\|$ was used to quantify the merit of each baseline removal algorithm—the smaller the $\|w\|$, the better the performance.

Comparing with the previous evaluating method based on PCA analysis, the new method takes data from two classes rather than one and avoids the situation when signals are seriously damaged by inappropriate baseline subtraction. For example, just choose a set of traces generated from the same source, if a baseline removal method simply subtracted all the traces to 0, it will get a 0 ratio under the previous method and evaluated as the best detector. That is a misleading result. In contrast, under our method, we would take traces generated from two sources: the same baseline removal driven the two classes of data to exactly one point—zero vector and the $\|w\|$ in SVM classification is ∞, which implied worst performance. The new evaluating criteria, based on amelioration of distance between classification hyper-planes of baseline corrected signals, showed more robustness in discriminating against artefact.

With this approach, we evaluated CROMWELL, LIMPIC, airPLS, AIMA, ChemStation side by side with our own CC baseline subtraction method on two abovementioned data sets, one from an LC-MS and the other from an MALDI-TOF experiment in Table-1. The graphic results for the LC-MS data and MALDI-TOF data are presented in Figure-16 and Figure-17, respectively. On visual inspection, it is difficult to tell which method performs better, especially among CC, airPLS, AIMA, and ChemStation. However, the new evaluating criteria provided a more subjective way to reveal the difference.

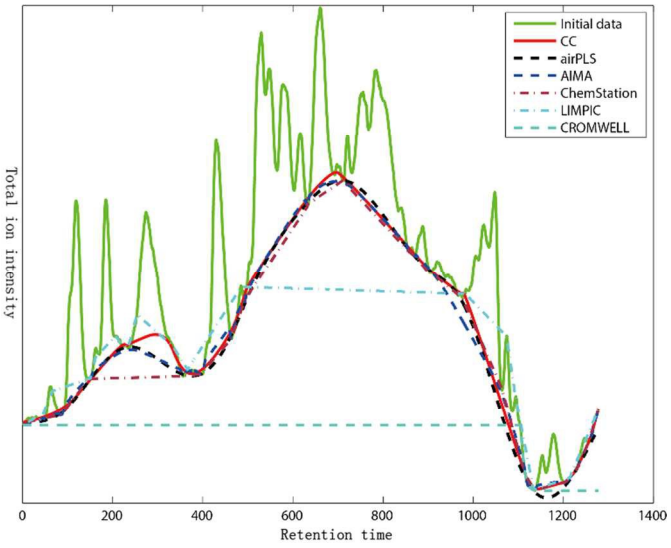


Figure-16 TIC baseline-subtracted with CROMWELL, LIMPIC, airPLS, AIMA, ChemStation and CC method.

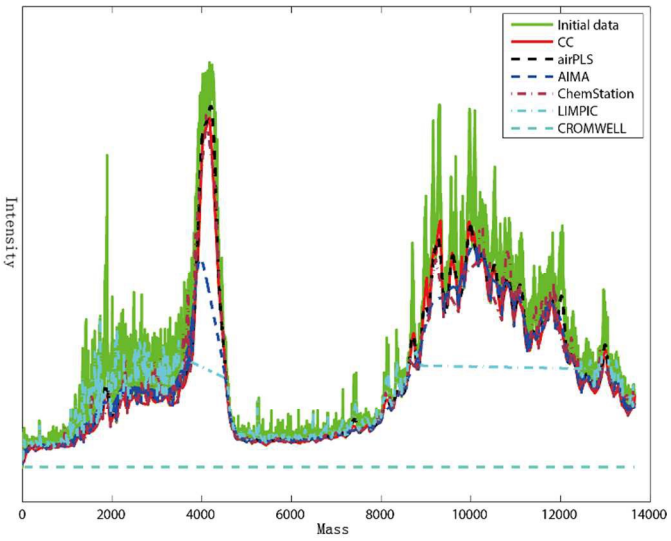


Figure-17 MALDI-TOF mass spectrum baseline-subtracted with CROMWELL, LIMPIC, airPLS, AIMA, ChemStation and CC method.

The SVM analysis results for both data sets are shown in Table-2 and Table-3, respectively.

For the LC-MS experiment, where the total ion current (TIC) were baseline-subtracted with CROMWELL, LIMPIC, airPLS, AIMA, ChemStation and CC method (Figure-16), all these methods produced smaller $\|w\|$ values than that of the original data.The correlation between smaller $\|w\|$ value and improved baseline subtraction is abundantly apparent with the CC method clearly showing the most improvement of all.

Method	$\|w\|$
Original Data	1.40944E-02
CROMWELL	1.00189E-02
LIMPIC	9.66525E-03
airPLS	9.63763E-03
ChemStation	9.53654E-03
AIMA	9.04923E-03
CC	9.02681E-03

Table-2 Experimental data used to evaluate CC algorithm

For MALDI-TOF data, the SVM analysis results are shown in Table-3. The CROMWELL baseline subtraction did not result in any change in SVM analysis. The apparent failure in baseline subtraction is expected since the data trace is not monotone decreasing. In comparison, airPLS, AIMA, and CC produced the closest results with CC perform the best.

Method	$\|w\|$
Original Data	1.69287E-04
CROMWELL	1.69287E-04
LIMPIC	1.19315E-04
ChemStation	1.19202E-04
airPLS	1.17631E-04
AIMA	1.16431E-04
CC	1.16265E-04

Table-3 Experimental data used to evaluate CC algorithm

Further scrutinization

Results from the previous section consistently demonstrated better outcomes with AIMA, airPLS, and CC among all methods. Therefore, it warrants closer comparison of the three to illustrate the merit of CC method in three aspects: speed, effectiveness and better ||w||.

Effectiveness and efficiency

Upon further examination, the CC method seemed to outperform other methods more on notch data. To verify this observation, we generated a batch of traces with random peaks lying on notch with randomly generated baselines.

Using available source code (airPLS in Matlab and AIMA in Java), the tests were done pairwise and visual outcomes are shown in Figure-18 to 25 (CC vs. airPLS), Figure-26 to 33 (CC vs. AIMA). The corresponding computing times are tabulated in Table-4, 5.

	airPLS computing time(**milliseconds**)	**CC** computing time(**milliseconds**)
Figure-18	3.087	0.2600
Figure-19	5.129	0.2600
Figure-20	7.185	0.2370
Figure-21	3.058	0.2580
Figure-22	5.634	0.2680
Figure-23	82.42	0.2710
Figure-24	4.511	0.2920
Figure-25	7.199	0.2950

Table-4 Computing time comparison of airPLS and CC (in Matlab on Linux laptop)

	AIMA computing time(**milliseconds**)	**CC** computing time(**milliseconds**)
Figure-26	9	4
Figure-27	11	4
Figure-28	12	4
Figure-29	14	4
Figure-30	8	2
Figure-31	9	5
Figure-32	12	5
Figure-33	10	3

Table-5 Computing time comparison of AIMA and CC (in Java on Linux laptop)

In these results, CC method had shown more effective baseline removal and produced the result with less time than airPLS and AIMA.

Statistics comparison of ||w||

Since the ||w|| values in Table-2, 3 for AIMA, airPLS and CC were fairly close, further statistical analysis was performed to achieve a more reliable conclusion. This was carried out on a series of combination from 7 classes of MALDI-TOF spectra. The name of these species (abbreviation in parentheses) and corresponding spectra number are listed in Table-6.

Source species	Spectra number
Bowman (B.)	16
Enterococcus faecalis (E.F.)	32
Staphylococcus aureus (S.A.)	16
Atopobium Collins (A.C.)	16
Pseudomonas aeruginosa (P.A.)	32
Enterobacter cloacae (E.C.)	16
Salmonella (S.)	24

Table-6 MALDI-TOF data species and numbers

Spectra of every two species were paired to perform a SVM analysis and produced a ||w|| value. For 7 species, we carried out $C_7^2 = 21$ pairs of analysis. In this analysis, each pair of data was created by randomly choosing certain spectra from the two species. The corresponding outcomes are tabulated in Table-7 and visually presented in Figure-34.

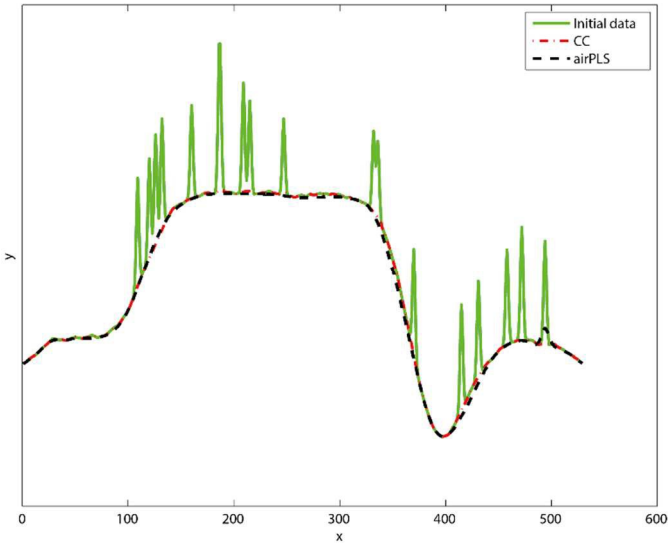


Figure-18 CC in 0.26 ms versus airPLS in 3.087 ms.

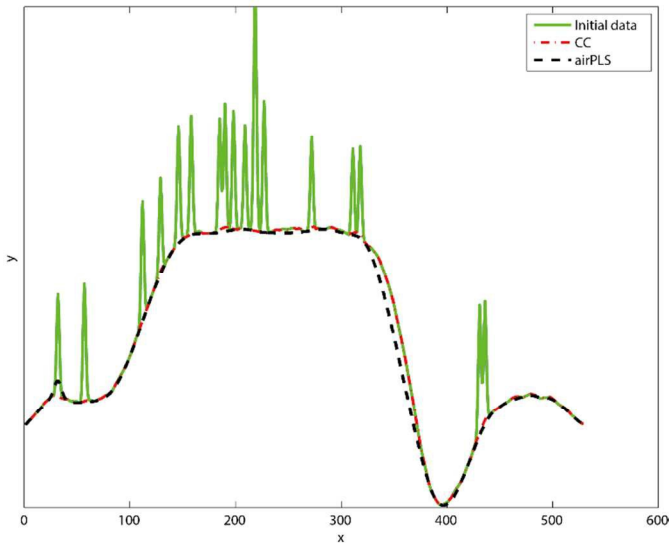


Figure-21 CC in 0.258 ms versus airPLS in 3.058 ms.

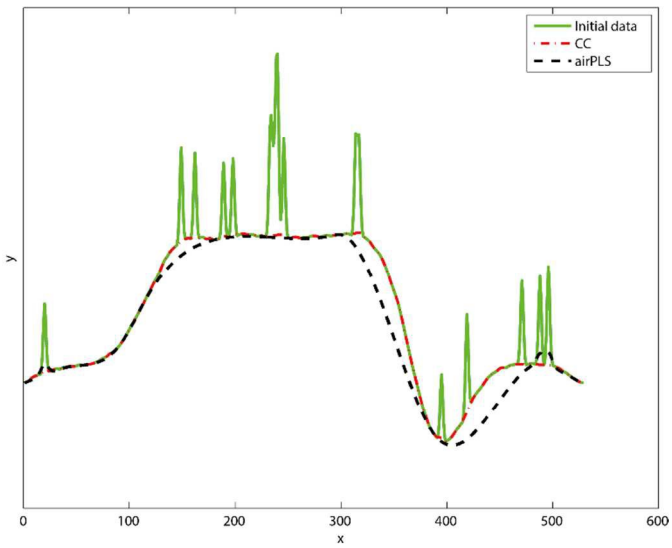


Figure-19 CC in 0.26 ms versus airPLS in 5.129 ms.

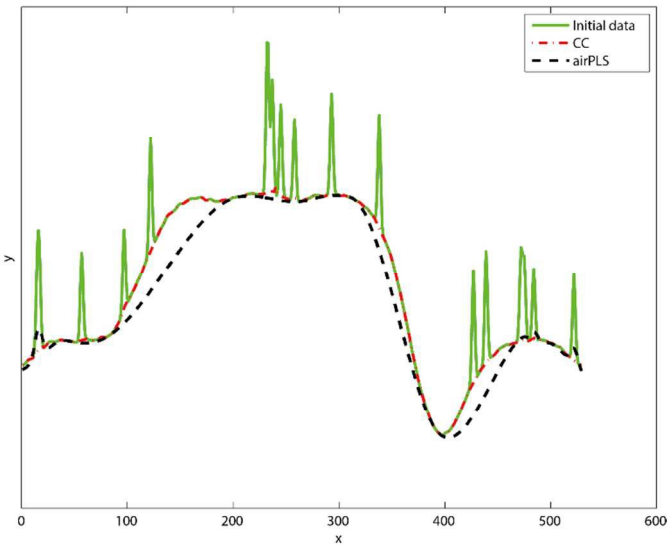


Figure-22 CC in 0.268 ms versus airPLS in 5.634 ms.

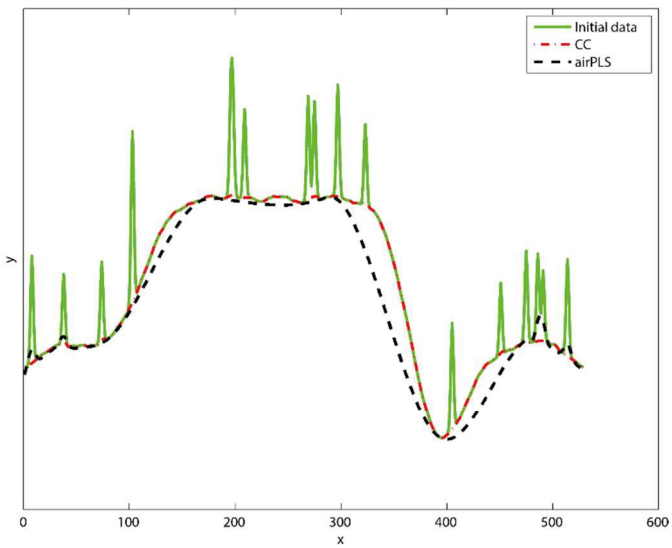


Figure-20 CC in 0.237 ms versus airPLS in 7.185 ms.

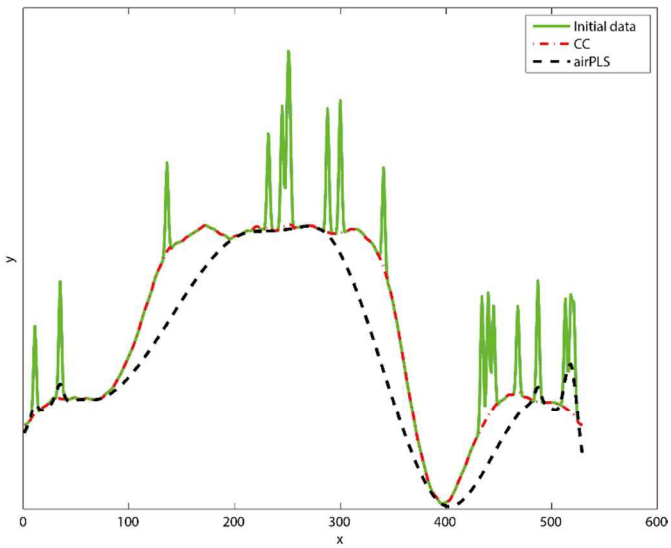


Figure-23 CC in 0.271 ms versus airPLS in 82.42 ms.

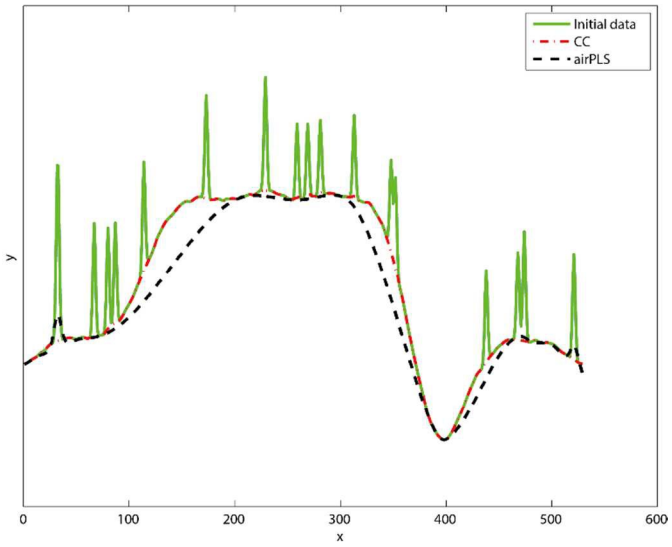


Figure-24 CC in 0.292 ms versus airPLS in 4.511 ms.

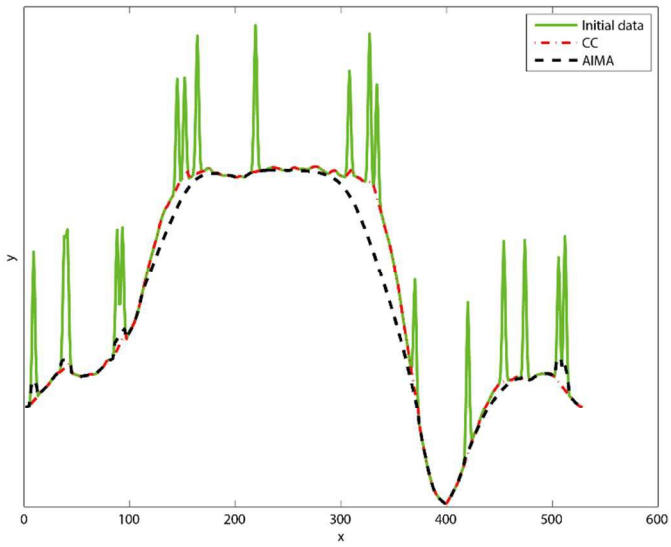


Figure-27 CC in 4 ms versus AIMA in 11 ms.

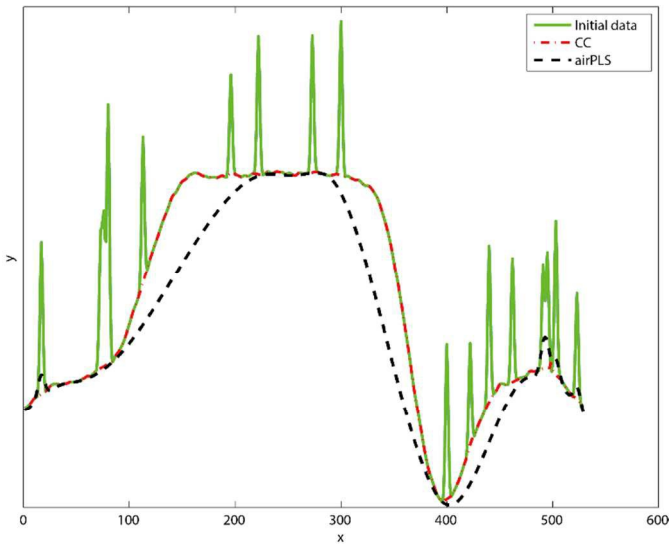


Figure-25 CC in 0.295 ms versus airPLS in 7.199 ms.

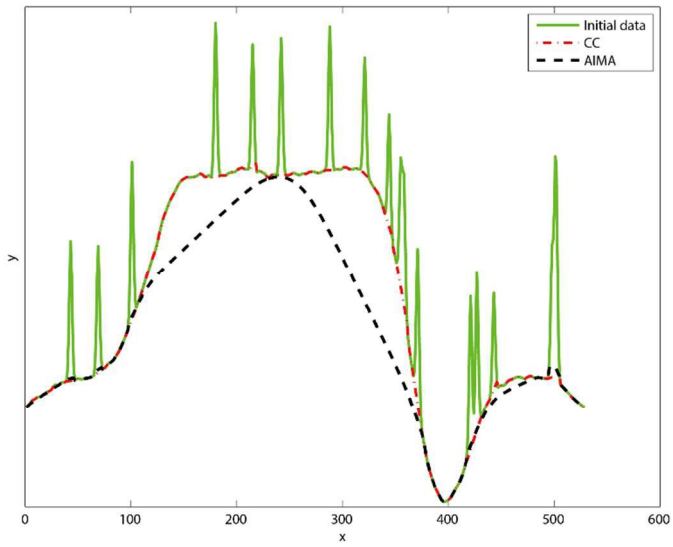


Figure-28 CC in 4 ms versus AIMA in 12 ms.

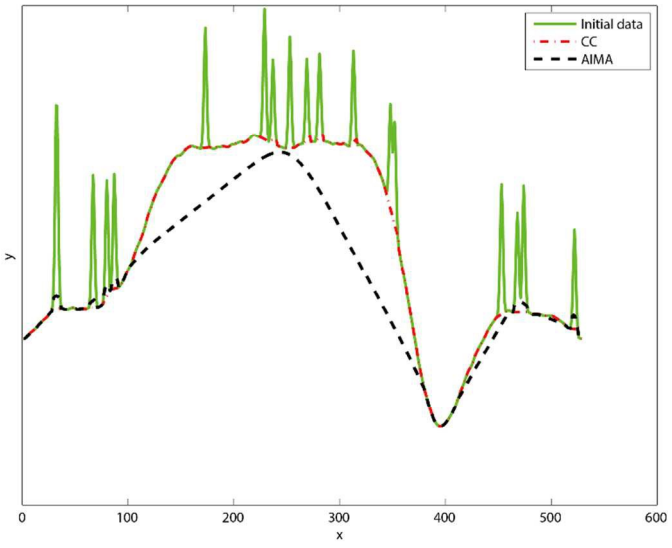


Figure-26 CC in 4 ms versus AIMA in 9 ms.

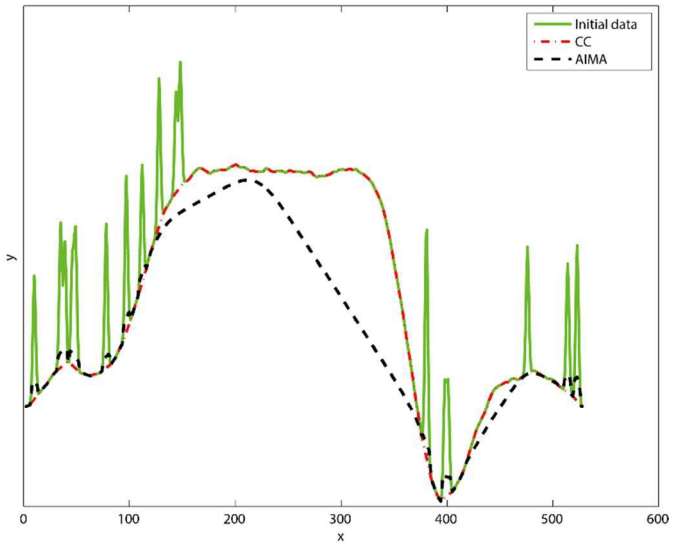


Figure-29 CC in 4 ms versus AIMA in 14 ms.

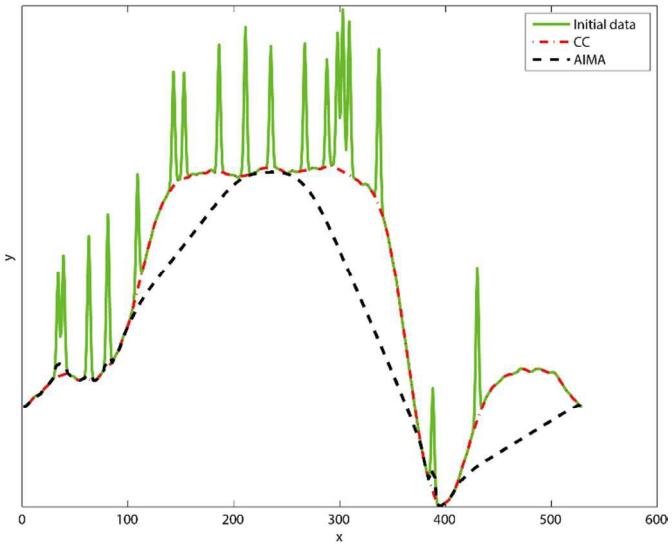


Figure-30 CC in 2 ms versus AIMA in 8 ms.

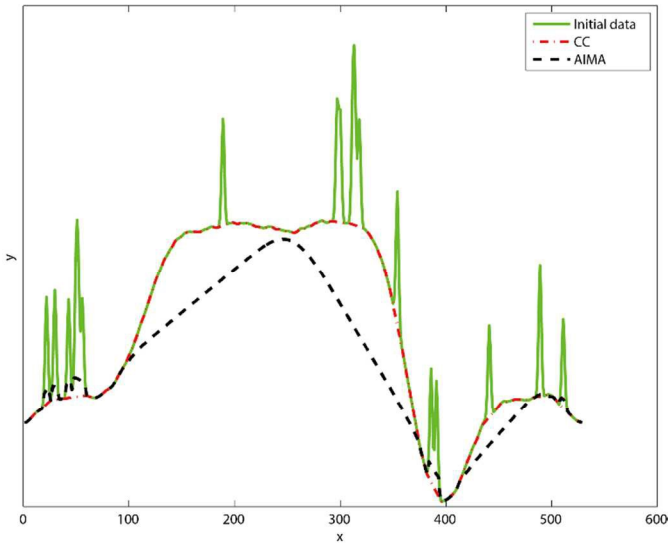


Figure-33 CC in 3 ms versus AIMA in 10 ms.

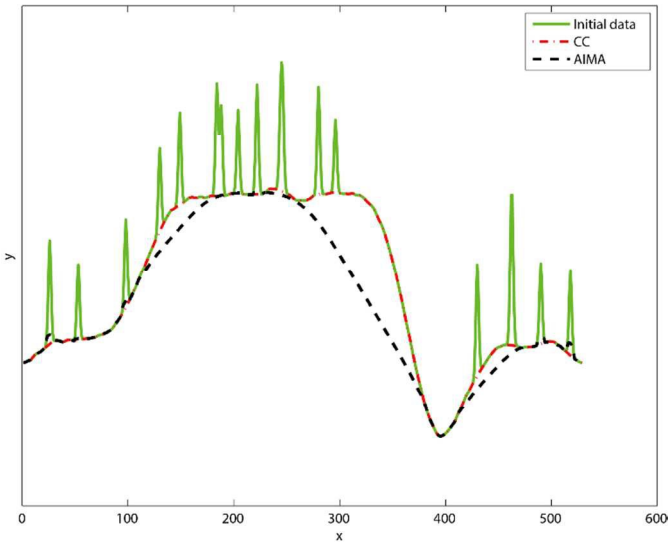


Figure-31 CC in 5 ms versus AIMA in 9 ms.

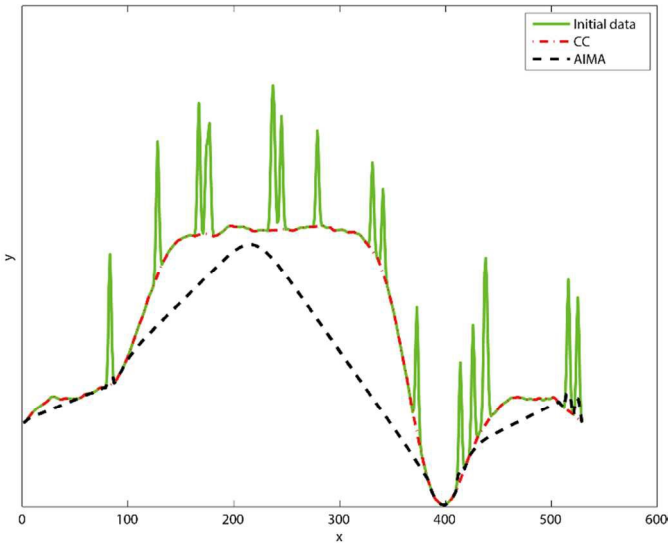


Figure-32 CC in 5 ms versus AIMA in 12 ms.

	Initial	airPLS	AIMA	CC
B.-E.F.	3.12490E-04	2.29129E-04	1.39172E-04	1.37291E-04
B.-S.A.	4.05368E-04	3.70872E-04	1.18902E-04	8.88115E-05
B.-A.C.	1.48133E-04	1.12673E-04	9.76651E-05	8.38890E-05
B.-P.A.	3.76643E-04	2.87664E-04	2.44192E-04	2.18645E-04
B.-E.C.	2.36566E-04	1.96513E-04	1.19439E-04	1.15605E-04
B.-S.	3.13895E-04	2.40417E-04	1.60261E-04	1.60536E-04
E.F.-S.A.	1.24179E-03	6.05932E-04	2.12501E-04	1.98285E-04
E.F.-A.C.	2.79176E-04	1.51732E-04	1.06739E-04	1.17459E-04
E.F.-P.A.	5.44939E-04	2.90802E-04	2.36695E-04	2.28822E-04
E.F.-E.C.	4.78890E-04	3.15165E-04	1.66394E-04	1.97486E-04
E.F.-S.	5.66213E-04	3.04481E-04	2.58421E-04	2.11874E-04
S.A.-A.C.	3.81328E-04	3.31044E-04	1.09815E-04	9.73469E-05
S.A.-P.A.	4.99312E-04	4.44198E-04	2.26434E-04	2.21883E-04
S.A.-E.C.	4.95864E-04	4.76996E-04	1.43877E-04	1.33639E-04
S.A.-S.	5.30115E-04	4.15376E-04	1.65609E-04	1.65672E-04
A.C.-P.A.	3.20499E-04	2.02923E-04	1.87383E-04	1.76502E-04
A.C.-E.C.	3.04329E-04	2.06210E-04	1.22394E-04	1.19477E-04
A.C.-S.	2.78246E-04	1.98075E-04	1.45559E-04	1.25396E-04
P.A.-E.C.	3.63072E-04	2.77125E-04	2.13140E-04	2.05999E-04
P.A.-S.	3.71543E-04	2.88272E-04	2.18886E-04	2.18122E-04
E.C.-S.	5.35720E-04	4.18442E-04	2.36611E-04	2.12983E-04

Table-7 The ||w|| value of each testing result.

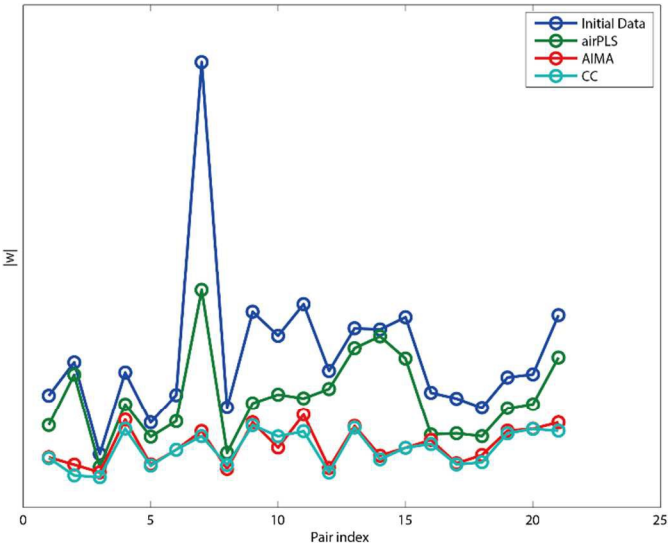


Figure-34 The ||w|| value of 21 groups of data before and after baseline correction.

The result shows that CC consistently outperforms airPLS. In case of AIMA, among 21 tests, CC produced smaller ||w|| in 17 tests. Therefore, it is fair to say that the CC method can more reliably produce better baseline among all methods tested.

Conclusions

Baseline removal plays an important role in digital signal processing of analytical data. It has a strong impact on the accuracy for down-stream processing such as peak detection, period identification, etc.

We examined existing baseline detection techniques and developed a new approach, the Corner-Cutting method that combined iteration with key point interpolation. This new strategy was, non-parametric, efficient and easy to implement. In conjunction with this work, a new evaluation method was also devised to provide quantifiable measurement on the quality of baseline subtraction result. The results of different algorithms when applied to real life analytical data showed that the new method was more accurate and robust. In addition, the local extremes of the ER curve also revealed structural information of the data stream, the comprehension of its mathematical meaning will be our future research direction.

References

[1] D.A. Shirley, Phys. Rev. B 5, 4709-4714 (1972).
[2] J.C. Cobas, M.A. Bernstein, M. Martín-Pastor, P.G. Tahoces, JMR, Vol.183, Issue 1, Pages 145-151(2006).

[3] K. R. Coombes, H. A. Fritsche Jr., C. Clarke, J. Chen, K. A. Baggerly, J. S. Morris, L. Xiao, M. Hung, H. M. Kuerer, Clinical Chemistry, October 2003, vol.49, no.10, 1615-1623.

[4] P. Du, W.A. Kibbe, S.M. Lin, Bioinformatics. 2006 Sep 1;22(17):2059-65. Epub 2006 Jul 4.
[5] F. Gan, G. Ruan, J. Mo, Chemometrics and Intelligent Laboratory Systems, Vol. 82 (2006), pp. 59-65.
[6] B. D. Prakash, Y. C. Wei, Analyst, 2011, 136, 3130.
[7] M. Kempka, J. Sjodahl, A. Bjork, J. Roeraade, Rapid Commun. Mass Spectrom. 2004; 18: 1208-C1212.
[8] D.Mantini, F. Petrucci, D. Pieragostino, P. Del Boccio, M. Di Nicola, C. Di Ilio, G. Federici, P. Sacchetta, S. Comani, A. Urbani, BMC Bioinformatics. 2007 Mar 26; 8:101.

[9] Y. Yasui, D. McLerran, B.L. Adam, M. Winget, M. Thornquist, Z. Feng, J Biomed Biotechnol, 2003(4):242-248.

[10] K.R. Coombes, S. Tsavachidis, J.S. Morris, K.A. Baggerly, M.C. Huang, H.M. Kuerer, Proteomics. 2005 Nov; 5(16):4107-17.

[11] G. Farin, Curves and Surfaces for CAGD: A Practical Guide, 5th edition, 2002, Published by Morgan-Kaufmann, ISBN 1-55860-737-4

[12] Z.-M. Zhang, S. Chen, and Y.-Z. Liang, Analyst, vol. 135, no. 5, pp. 1138-1146, 2010

[13] ALAIN KOCH, JEAN-VICTOR WEBER, Appl. Spectrosc. 1998, 52(7):970
[14] A. Jirasek, G. Schulze, M. M. L. Yu, W. Blades and R. F. B. Turner, Appl. Spectrosc., 2004, 58, 1488–1499

[15] C. A. Lieber and A. Mahadevan-Jansen, Appl. Spectrosc., 2003, 57,1363–1367.

[16] Georg Schulze, Andrew Jirasek, Marcia M. L. Yu, Arnel Lim, Robin F. B. Turner, and Michael W. Blades, Appl. Spectrosc. 59, 545-574 (2005)

[17] A. T. Weakley, D. E.Aston, P. R. Griffiths, Applied Spectroscopy, 2013, 67(10), 1117-1126

[18] Z. Li, D.-J. Zhan, J.-J. Wang, J. Huang, Q.-S. Xu, Z.-M. Zhang, Y.-B. Zheng, Y.-Z. Liang, and H. Wang, Analyst, vol. 138, no. 16, pp. 4483-4492, 2013.