Hindawi Publishing Corporation
EURASIP Journal on Applied Signal Processing
Volume 2006, Article ID 63582, Pages 1–9
DOI 10.1155/ASP/2006/63582

MALDI-TOF Baseline Drift Removal Using Stochastic Bernstein Approximation

Joseph Kolibal[1] and Daniel Howard[2]

[1] *Department of Mathematics, College of Science & Technology, The University of Southern Mississippi, Hattiesburg, MS 39406-0001, USA*
[2] *QinetiQ PLC, Malvern, Worcestershire WR14 3PS, United Kingdom*

Received 7 July 2005; Revised 21 August 2005; Accepted 1 December 2005

Stochastic Bernstein (SB) approximation can tackle the problem of baseline drift correction of instrumentation data. This is demonstrated for spectral data: matrix-assisted laser desorption/ionization time-of-flight mass spectrometry (MALDI-TOF) data. Two SB schemes for removing the baseline drift are presented: iterative and direct. Following an explanation of the origin of the MALDI-TOF baseline drift that sheds light on the inherent difficulty of its removal by chemical means, SB baseline drift removal is illustrated for both proteomics and genomics MALDI-TOF data sets. SB is an elegant signal processing method to obtain a numerically straightforward baseline shift removal method as it includes a free parameter $\sigma(x)$ that can be optimized for different baseline drift removal applications. Therefore, research that determines putative biomarkers from the spectral data might benefit from a sensitivity analysis to the underlying spectral measurement that is made possible by varying the SB free parameter. This can be manually tuned (for constant σ) or tuned with evolutionary computation (for $\sigma(x)$).

1. INTRODUCTION

Each measurement analysis tool for determining the presence and concentration of biomolecules has its particular signal processing challenge. Consider some of these challenges for two of the most powerful tools: microarray analysis and spectral analysis. For example, the proximity of dots in a microarray can cause a degree of correlation between neighboring dots that must be removed with signal processing. With spectral analysis, typical signal processing challenges are (a) baseline drift correction; (b) denoising by smoothing and averaging of signals; (c) peak alignment; and (d) peak identification.

This paper tackles baseline drift correction with algorithms that are based on a recent method of signal processing, stochastic Bernstein (SB) approximation [1]. Although baseline drift correction is illustrated with respect to matrix-assisted laser desorption/ionization time-of-flight (MALDI-TOF) [2] data, our approach has much wider application. Other types of spectral data suffer from baseline drift and, potentially, this technique can also assist with a variety of instrumentation (not necessarily in the bioinformatics domain) that suffers from baseline drift (e.g., [3]).

Consider MALDI-TOF and baseline drift. For instrumental reasons that are not easy to control, multiple MALDI-TOF measurements on the same biological sample can result in curves at different heights. The drifted baselines must be corrected before comparing peak intensities. Section 2 discusses concepts that are specific to baseline drift in MALDI-TOF.

Bernstein functions are the natural extension of the Bernstein polynomials, and they have remarkable monotonicity and convergence properties [4]. Unlike the Bernstein polynomials, the Bernstein functions are more readily computable for large data sets (for large n), and most significantly for the purposes of computing the baseline, they produce infinitely smooth approximations which introduce no spurious false extrema. This results in a robust and efficient algorithm for computing the baseline correction of spectral curves, including the MALDI-TOF spectra. The algorithm is adjustable to user requirements pertaining to the underlying shape of the baseline curve, and is suitable for automatically processing a large number of spectra.

The use of Bernstein functions, in contrast to the more traditional Bernstein polynomials, for approximation offers a free parameter that can be adjusted to provide domain-specific levels of smoothing, and hence of baseline correction. The method is global, but can also be implemented as a windowing method on the data if this should be required. Finally, as explained in Section 4, the method enjoys three implementations: approximation; interpolation; and quasi-interpolation, in regard to generating smooth

representations of data. This alone offers enormous generality and flexibility; however perhaps the most compelling reason for using this approach is that it does not introduce any spurious extrema, unlike higher-order-polynomial-based methods, and thus it does not corrupt the signal.

Classification and comparison of parts of the spectra or the extraction of quantitative information are important to bioinformatics research. Therefore, the removal of the baseline from spectral data should not remove or alter peak information from the spectrum, and it should produce a smooth baseline curve which best represents the average, or mean of the noisy data. An approach to baseline correction using a windowed polynomial interpolation method was introduced and validated in [5]. The algorithm subdivides the data into bins or windows in which the mean of the data is computed. These means are joined through the process of polynomial interpolation to yield a curve which is then adjusted to account for various difficulties which could cause the loss of peak quality, including adaptively resetting the window widths. Finally, the data that is produced is fit using least squares to an exponential curve so as to provide a smooth baseline curve to the spectral data. While the basic concept is simple, there is some algorithmic complexity to this approach, and analytically it is unclear what the baseline curve which is obtained represents.

Traditional approximation by least-squares fit, Fourier analysis, and wavelets are popular choices for the characterization of signals. While these classical techniques can and have been applied to the problem of baseline correction, along with attempts to characterize the baseline using traditional polynomial approximation techniques, an alternative approach, using stochastic approximation methods based on suitable mollifiers built from Bernstein functions, appears to provide a flexible, easily adaptable approach to characterizing the mean behavior of a signal, and hence the complex errors that affect baseline drift.

2. BASELINE DRIFT AND MALDI-TOF

There is little information available in the literature about the origin of the noise and the baseline shift in MALDI spectra. However, baseline drift appears to be related to noise. All of the noise signals in MALDI spectra represent chemical noise (real ions arriving at the detector), while all other noise sources, for example, electronic noise, are at least one order of magnitude less.

Most of these ions seem to (nominally) come from either nonzero position (axially), or are created at nonzero time (relative to the origin of the time scale of the TOF). Thus, these ions arrive in an axial extraction TOF at random times. This causes single-ion signals to merge into each other resulting in an overall rise of the baseline. The baseline shift in printed spectra often actually represents only a lack of resolution that is caused by the binning of the sample pixels. If these spectra are displayed with the maximum time resolution, then many, if not most of the signals in the low-mass range, show significant modulation, sometimes even baseline resolution. In TOF instruments with orthogonal extraction with one-to-three transfer quads preceding the TOF, all such processes are finished before the ions enter the TOF, and accordingly all signals which can be characterized as noise have integer mass differences.

Strong noise and baseline shifts in the low-mass range undoubtedly represent mostly matrix ions, their clusters, and fragments. They increase strongly with the laser fluence (energy per unit area). The background of matrix ions can even be completely suppressed for clean samples with not too low an analyte concentration, for example, at a concentration of 10^{-6} M. The higher fluence required when the cleanliness of the sample and its analyte concentration are low will result in a much stronger background and baseline shift for the low-mass range.

It is a common observation that many analyte signals even in the higher-mass range ride on a type of hump in the baseline. This elevated baseline contains mostly ions of clusters of analyte and matrix. This has been demonstrated in an elegant MS/MS experiment in an ion trap by Krutchinsky and Chait [6] that sheds light on the nature of the chemical noise background. Some of these ions must, obviously, have energy deficit to account for the part of the hump below the analyte mass.

All signals are seen in the spectra and the baseline shift is also included. It represents ions generated in the MALDI process. This limits the possibility for a chemical filtering procedure. This has motivated us to develop a simple signal processing method which can be adapted by the user to correct for the baseline shift in MALDI-TOF spectra.

3. MATHEMATICAL PRESENTATION

In this section, signal processing using stochastic methods built from Bernstein functions [1] is developed further into an iterative method to correct MALDI-TOF baseline drift. Additionally, the novel scheme has a tunable parameter $\sigma(x)$ that can be set to a constant for all x; can be set to different values for different masses of the spectra; or it can be discovered as a continuous function of x using supervised learning from examples of known analyte concentrations in MALDI-TOF spectra or in any other instrumentation domain.

Section 5 illustrates the straightforward application of the new method to both a proteomics and a genomics MALDI-TOF data set. In these cases, however, optimization of σ became unnecessary because the baseline correction provided equivalently acceptable results for constant smoothing.

3.1. Stochastic approximation using Bernstein functions

Consider the function $f(x)$ sampled at points $x_k \in [0.1]$, that is, at $f(x_k) = y_k$. We denote the natural continuum extension of the Bernstein polynomials on the set of data $\{(x_k, y_k)\}$, $k = 1, \ldots, n$, by $K_n(x)$, expressible as the sum

$$K_n(x) = \sum_{k=0}^{n} \frac{y_k}{2} \left[\operatorname{erf}\left(\frac{z_{k+1} - x}{\sqrt{\sigma(x)}}\right) + \operatorname{erf}\left(\frac{x - z_k}{\sqrt{\sigma(x)}}\right) \right], \quad (1)$$

where f is assumed to be piecewise constant in (z_{k-1}, z_k) with value y_k and where $z_0 = -\infty$, $z_k = (x_{k+1} + x_k)/2$ for $k = 1, 2, \ldots, n-1$, and $z_n = \infty$. The smoothing in this case is directly related to the magnitude of the term $\sigma(x) = (2/n)x(1-x)$ in the argument of the error function in (1). When n is large, the smoothing, which is related to the magnitude of the second moment of the Gaussian probability distribution function, is small, and when n is small, the smoothing is large. A more robust model allows for variable smoothing, where $\sigma(x) > 0$. In most cases, it is convenient to take $\sigma(x)$ to be constant throughout the interval. Note that there is no requirement that the data be uniformly spaced.

For simplicity, the constant smoothing model is used to construct the baseline curves in this paper. Also, because we are not interested in creating a finer approximation to the spectral data, the points x at which $K_n(x)$ are evaluated are the same as the input data coordinate values, that is, $K_n(x_j)$, $j = 1, 2, \ldots, n$. For very large data sets, the sums in (1) can also be truncated when the value of $\operatorname{erf}(u)$ is sufficiently small yielding significant reduction in the work required to compute the value of K_n.

The approximation provided by K_n intrinsically consists of a matrix-vector multiply, where $A_{nn} = (a_{jk})$ is the $n \times n$ matrix containing the coefficients

$$a_{jk} = \frac{1}{2}\left[\operatorname{erf}\left(\frac{z_{k+1} - x_j}{\sqrt{\sigma(x)}}\right) + \operatorname{erf}\left(\frac{x_j - z_k}{\sqrt{\sigma(x)}}\right)\right]. \tag{2}$$

Thus, $K_n(x_k) = A_{mn}\mathbf{y}$, where $\mathbf{y} = (y_1, y_2, \ldots, y_n)$ and where A_{mn} is a row-stochastic matrix in which the kth row is generated using (2) for each point x_k, $k = 1, \ldots, m$, at which the function is evaluated. Intrinsically, this amounts to a Gaussian mollifier applied to the data; the advantages of the stochastic formulation become apparent when it is realized that A_{nn}^{-1} is a deconvolution operator on the data, and thus $A_{mn}A_{nn}^{-1}\mathbf{y}$ provides an elegant solution to the interpolation of the data. Choosing σ to be different in A_{mn}, A_{nn} yields a range of data representational forms, ranging from pure smoothing through interpolation to deconvolution. Constructing an approximate inverse to A_{nn} has computational advantages, however most significantly, there are known approximate inverses which allow for interpolation of smooth data, but which become increasingly smoother as the data becomes noisy. This is referred to as the pseudoinverse method.

Increases in computational efficiency can be achieved by restricting the size of the data set over which the sums are taken. This effectively creates a multiblock algorithm. By overlapping, the blocks differentiability across blocks is still maintained, although smoothness (being able to construct an infinitely differentiable baseline curve) is lost. In any event, these are structural components of the algorithm which can be selectively implemented in tradeoffs between efficiency measured in terms of CPU cycles and accuracy.

Experience has shown that implementing any of these devices for improving efficiency can dramatically impact the computation time without substantial effect on the accuracy of smoothness of the resulting approximation. Of greater significance than any of these in regard to the quality of the results is the value of $\sigma(x)$. Choosing the smoothing allows the approximation to be more or less sensitive to the low-frequency oscillations intrinsic to the data curve. Choosing it too small causes the resulting approximation to be sensitive to even the high-frequency oscillations associated with the noise, and while it may seem that this choice is quite difficult, in practice it is very easy to implement effective and usable choices without much concern.

3.2. Constructing smoothing bounding curves to spectral data

The algorithm we propose to construct the baseline curve is based on the approximating property of K_n which results in a family of curves which uniformly approximate the data set, thereby providing an envelope of width ε such that the error in the approximation and the data is always less in magnitude than ε at any point in the domain. This provides a convenient method for averaging. Also importantly, it can be shown that using (1) to approximate the data yields approximation curves that have almost the same area as the piecewise constant data $\hat{f}$ [1], providing an area-weighted mean to the data.

Denote by B_0 the initial approximation to the data set $D_0 = \{(x_k, y_k)\}$, $k = 1, \ldots, n$, by constructing K_n applied to D_0. This initial baseline curve at x has the values $B_0(x)$. Then construct a succession of smooth baseline curves, denoted by B_p, $l = 1, \ldots$, which successively approximate the data, $D_p = \{(x_k, y_k^{(l)})\}_{k=1}^{n}$, on each iteration. At each iteration, the data to be approximated lies below the previous iteration's approximation curve. Thus, we introduce the following algorithm for generating a sequence of baseline curves B_p.

(1) Construct the curve B_0 by constructing the Bernstein approximation K_n to the data set $D_0 = \{(x_i, y_i^{(0)})\}$, $i = 1, 2, \ldots, n$, where $y_i^{(0)} = y_i$.
(2) Obtain the data $D_1 = \{(x_i, y_i^{(1)})\}$, $i = 1, 2, \ldots, n$, where $y^{(1)} = \min(y_i^{(0)}, B_0(x_i))$.
(3) Continue iterating, that is, obtain the data $D_p = \{(x_i, y_i^{(p)})\}$, $i = 1, 2, \ldots, n$, where $y^{(t)} = \min(y_i^{(p-1)}, B_{p-1}(x_i))$.
(4) Stop the iteration when most of the points in D_p are bounded below by B_p.

While there is no criterion for establishing when most of the data lie above the baseline, a cutoff of 98% work well. Stopping the iteration when a specified tolerance is reached, when $\|D_p - D_{p-1}\| < \varepsilon$, for some $\varepsilon > 0$, has been seen to produce oversmoothing of the baselines in some cases, and thus is more difficult to apply. Note that because of the nature of the Bernstein approximation, the limiting baseline curve B_p as p gets large is not the minimum of the data D_0, but instead is the low-frequency curve which best fits, based on the parameter $\sigma(x)$, the lower bound to the data. If there is interest in determining limiting upper-bound curves, these can also be constructed using the same approach.

The dependence of the baseline on the value of σ is illustrated in Figure 1 for some "sample" data generated from the model function consisting of a Gaussian peak at $x = 400$

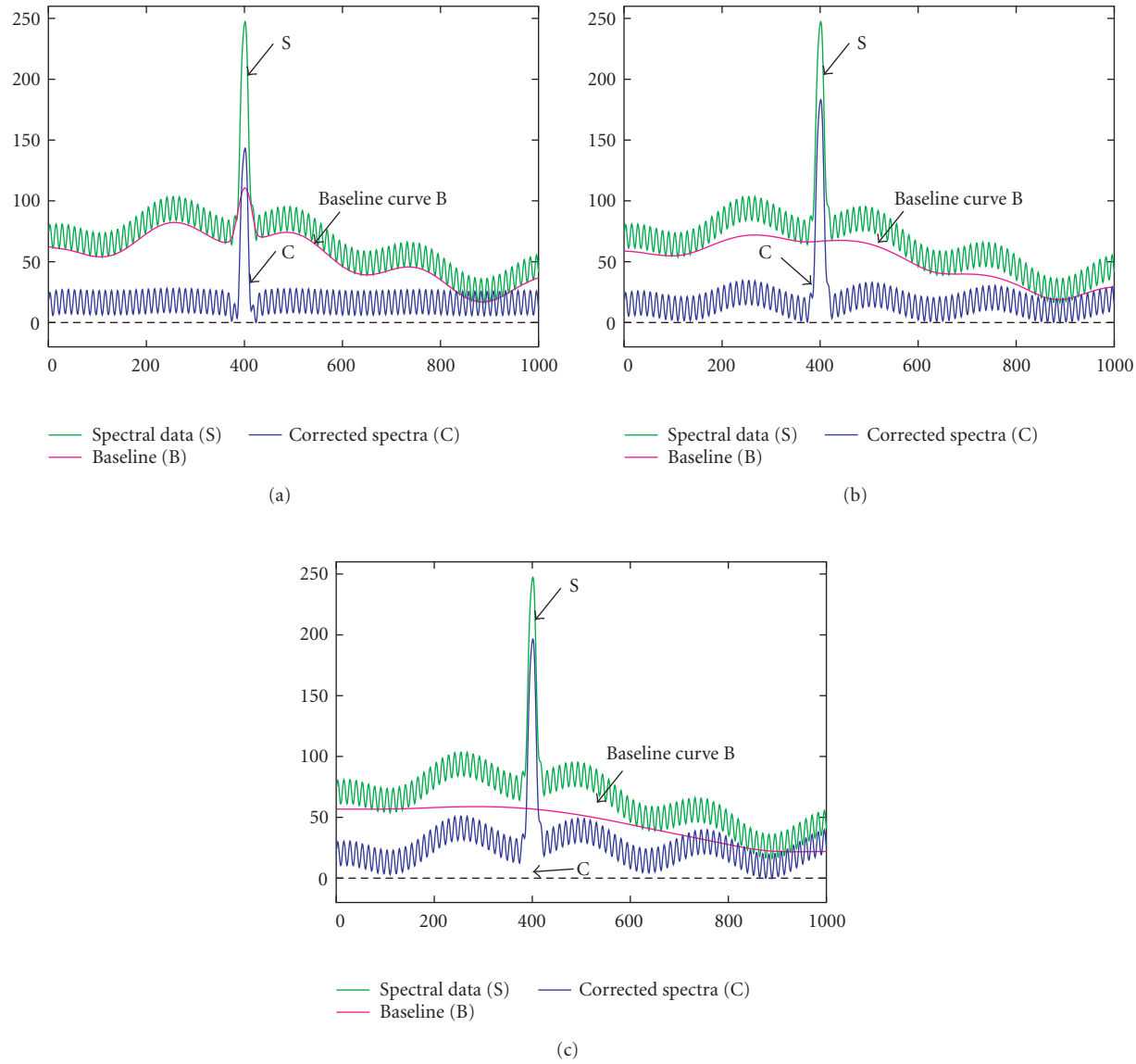

Figure 1: Construction of the corrected spectra using a signal, $s(x) = 180\exp(-0.01(400 - x)^2)$ with underlying harmonic components $h_0 = 60.0$, $h_1(x) = 10\sin(x/2)$, $h_2(x) = 10\cos(x/40)$, $h_3(x) = 25\sin(x/200)$, so that $f(x) = s(x) + h_1(x) + h_2(x) + h_3(x)$. The spectra are labelled S and the corrected spectra with baseline removal are labelled B; (a) $\sigma = 10$, (b) $\sigma = 100$, (c) $\sigma = 1000$.

which is perturbed by sinusoidally oscillating data sampled from three characteristic frequencies, $\sin(x/2)$, $\cos(x/40)$, and $\sin(x/200)$. All of the baseline curves are produced with a cutoff of 98%. The baselines are generated at values of sigma ranging from 10, 100, and 1000 in Figures 1(a), 1(b), and 1(c), respectively. It is obvious that when σ is small, all of the harmonics, except the highest frequency associated with $\sin(x/2)$, are well approximated by the baseline curve. As σ increases, the ability of the curve to respond to the high frequencies is diminished, such that when $\sigma = 1000$, only the lowest harmonic at $\sin(x/200)$ is revealed in the trace of the baseline.

The algorithm produces a succession of baseline curves $B_0, B_1, \ldots, B_m$ which appear to approach a lower-bound curve B for each value of σ. This curve has the property that it is a baseline curve (it is a Bernstein approximation and thus is infinitely smooth) and it lies below all other baseline curves with $p < m$. It is not strictly a lower bound to the data, since at some x_k the values of y_k will exceed the value of the baseline $B_p(x_k)$. This can be seen in all three plots in Figure 1 where there are a few places where the spectral data undershoot the baseline curve by a small amount. Equally, it is not the greatest lower bound to the data, although it approaches this when σ is very small, as seen in the graph in Figure 1(a).

Clearly, using stochastic Bernstein approximation provides a convenient mechanism for computing a set of lowpass filters for the data, but it does more than that, since it can be

combined easily to produce interpolation and deconvolution of the same data, and to do all of these locally through modifications of the structural form of the smoothing by working with $\sigma(x)$. Since the baseline curves are uniformly approximating, they are well behaved. Moreover, under suitable circumstance, it is possible to construct the baseline curve in one iteration, that is, by constructing only one approximant K_n to the data, and we discuss this in greater detail in Section 6.

4. ENGINEERING PRESENTATION

The new method of baseline drift removal is an iterative approach that repeatedly applies the SB approximation. The input signal for the next iteration stage becomes the minimum of the input signal for the current iteration stage and its SB approximation.

An engineering or computer science presentation of the stochastic Bernstein function method is complementary to the mathematical treatment of Section 3. It offers an appreciation for the generality and flexibility of the SB approximation method. The stochastic Bernstein function method (embedded in the iterative process) can be described by pseudocode as follows.

(1) Read the MALDI-TOF data $\{(x_i, y_i)\}$, $i = 0, n-1$ (x_i are the m/z spectral bins and y_i are the spectral intensities).
(2) Convert data coordinates to lie on the unit interval.
(3) Construct the convolution matrix A_{nn}, which depends on the data coordinates x_i and on the value of the smoothing parameter σ. The generator of the row space of A_{nn} is a Bernstein function.
(4) Construct the deconvolution matrix, A_{nn}^{-1}.
(5) Construct the augmented matrix $\widetilde{A}_{mn}$, where $m > n$, using the same generator of the row space.
(6) Evaluate $\widetilde{A}_{mn}A_{nn}^{-1}\mathbf{z}$, to obtain output data $\{z_i\}$, $i = 0, m-1$.
(7) Convert the output data to the world coordinate system to obtain the Bernstein function values at the locations of the output data.

These matrices correspond to the mathematical terms already presented. Note also that both the input and the output data points can be nonuniformly distributed in x, and that they can be unrelated to one another, and are of different size (different number of points).

The pseudocode is for the "interpolation" version of the stochastic Bernstein function method. In this version, the Bernstein function passes exactly through the input data points. The "pseudointerpolation" version of the SB method retains all steps but obtains A_{nn}^{-1} as an approximate inverse and causes the Bernstein function to pass very closely but not exactly through the input data points; with the deviation being larger, the more the data deviates from being locally smooth.

The method applied in this paper is the SB "approximation" version of the method. The Bernstein function does not pass through the input data points. The approximation version of SB does not require steps 3 and 4 of the pseudocode and also replaces A_{nn}^{-1} in step 6 by the identity matrix.

5. RESULTS OF APPLICATION AND ILLUSTRATIONS

The process of finding a baseline curve to the proteomics MALDI spectral data as provided through [5] is illustrated in Figure 2. In this case, the spectral data (labelled S) along with the corrected spectral data (labelled C) is shown for two different values of $\sigma(x)$. Choosing small $\sigma = 100$ results in a limiting baseline curve which still preserves the underlying low-frequency oscillation apparent in the spectral data around the spectral peaks at $x = 5000$ and $x = 8500$. Choosing $\sigma = 10000$, however, results in a significantly smoother limiting baseline curve which yields a corrected spectral curve which is significantly flatter and which is lacking in any of the low-frequency response which characterizes the data in Figure 2(a). Note also that the limiting baseline curve was attained in about 20 iterations, and that there are still a few points, particularly in the range from 3000 to 7000, where corrected data still have negative values. Clearly, it may be desirable to iterate further to eliminate these negative deviations, which can be done, however this exceeds the purposes of this demonstration.

A more detailed examination of Figure 2 is shown in Figure 3 and it shows that there is no loss in the peak spectral information. The baseline curve does not reduce the magnitude of the spectral peaks. The use of maximal smoothing, for example, can be seen to provide a spectral curve which is shifted down by 4000 units at the peak at $x = 5000$, however the magnitude of the peaks remains unchanged before and after the baseline correction. This is because the SB approximation for $\sigma \gg 1$ does not respond to high-frequency oscillations and thus is acting as a lowpass filter only. Note that using a smaller value of the parameter σ (using strong smoothing) causes even the lower-frequency hump from $x = 4000$ to $x = 6000$ to be ignored in the generation of the baseline curve, and thus causes the hump to be incorporated into the spectral data. In comparison, using a larger value for σ allows the SB approximation to pick up the low-frequency values along the hump, yielding a baseline curve which contains this low-frequency oscillation, thus resulting in a spectral curve which is flatter as shown in Figure 3(a).

Although MALDI-TOF is found principally in proteomics, it is also used in genomics. Figure 4 gives an overall appreciation for the baseline correction for a spectra of genomics origin. Figure 5 illustrates the sensitivity to the value of $\sigma(x)$ on this particular data. In these experiments, the sensitivity is not great but in other cases of baseline correction it would be necessary to optimize $\sigma(x)$.

5.1. Remarks

In assessing the design of any algorithm for removal of baseline drift from spectra, such as the SB approximation for MALDI-TOF data, it is important to examine the possible

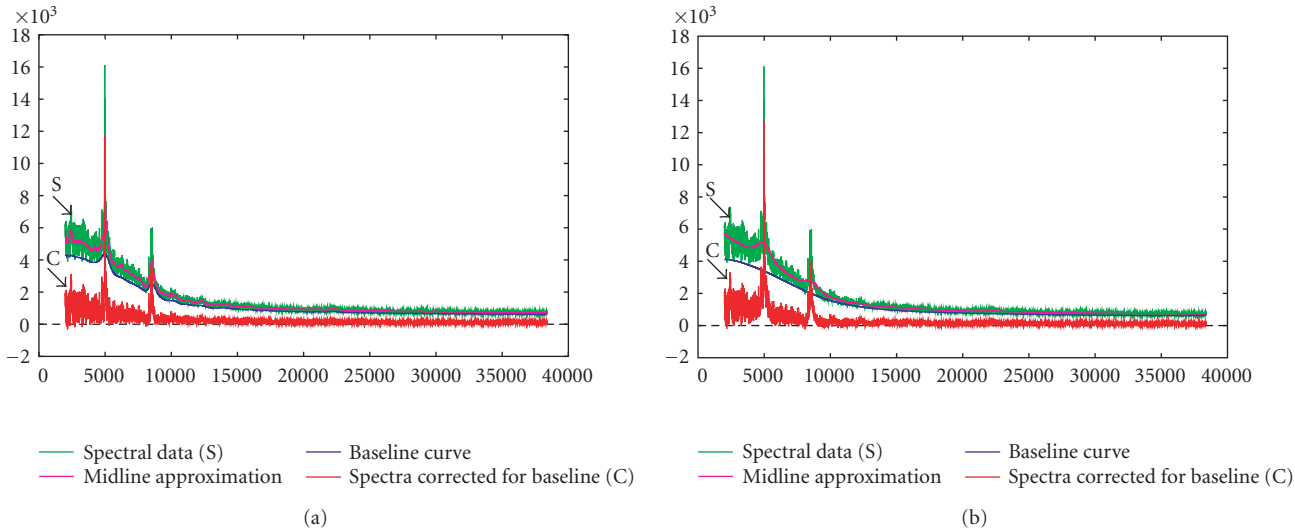

Figure 2: Convergence of SB approximation to 15 000 data point spectra applying min-mean baseline algorithm. (a) The approximations are computed using minimal smoothing as this removes the baseline hump at $x = 5000$ and $x = 8500$. (b) The approximations are computed using strong smoothing as this preserves the baseline hump at $x = 5000$ and $x = 8500$.

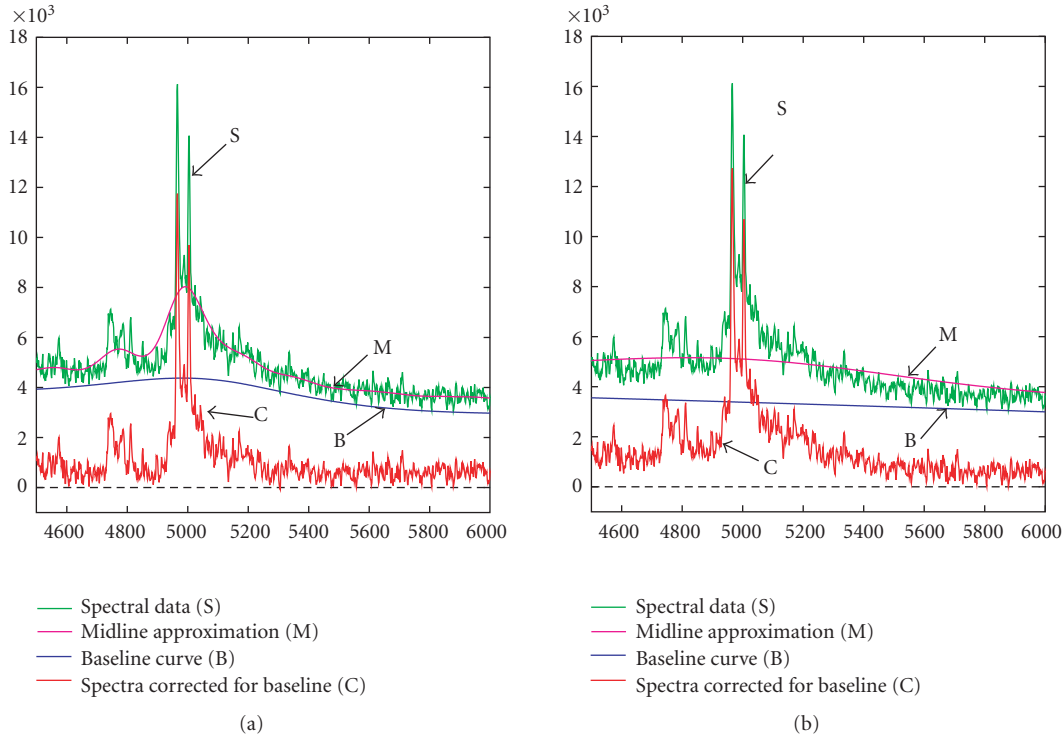

Figure 3: Detail from $x = 4500$ to 6000 for the min-mean baseline corrected spectra shown in Figure 2. The approximations in Figure 3(a) are computed using minimal smoothing and in Figure 3(b) are computed using strong smoothing.

distortion of the signal by the method. Inevitably, every numerical method affects the signal in some manner. A compelling reason for choosing the SB approximation in developing this method, aside from the algorithmic simplicity of the approach, is that it does not introduce any false extrema into the signal. Thus, the SB approximation to a function sampled at a discrete set has the property that the approximant lies between the nodal values at which the function is sampled. With the exception of piecewise linear and piecewise quadratic interpolation by polynomials,

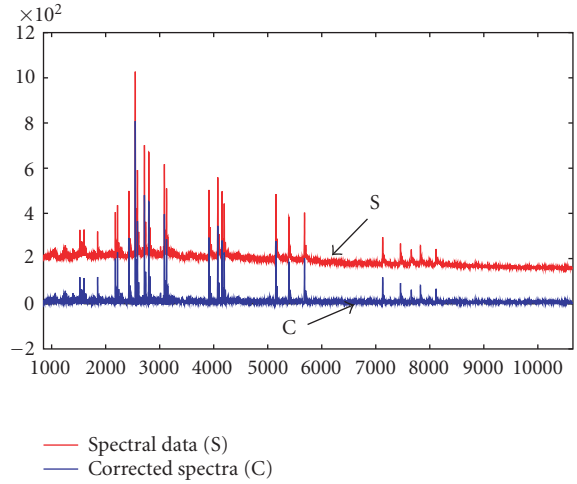

FIGURE 4: Original and baseline corrected MALDI-TOF spectra using the method with $\sigma = 150$.

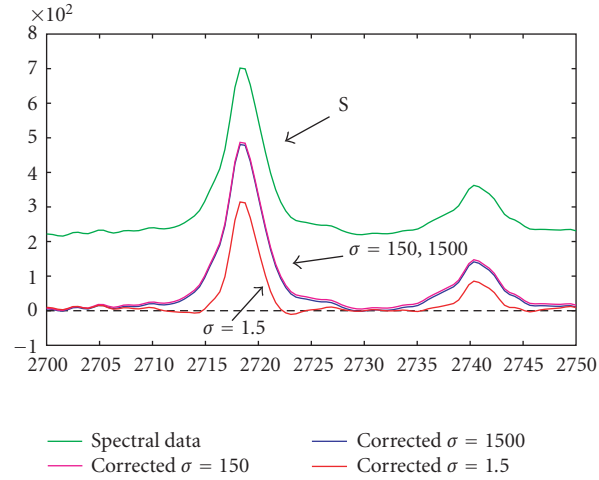

FIGURE 5: Detail from $x = 2700$ to 2751 for the baseline corrected spectra shown in Figure 4, showing SB approximation baseline corrected MALDI-TOF using two different values of the parameter $\sigma(x)$. One of the curves uses $\sigma = 150$ and the other uses $\sigma = 1500$. Note in this case that both methods perform similarly.

this property cannot be attained without the introduction of limiters to prevent overshooting and undershooting between interpolation points. Furthermore, unlike other polynomial approximation methods, the SB approximation can be constructed for even a large number of points in the computational stencil, and unlike the Bernstein polynomials to which the Bernstein functions are related, the properties can be tuned to increase or decrease the smoothing through the choice of the parameter σ and if required to determine this choice with evolutionary computation, for example, genetic programming. This provides control and efficiency.

The efficiency of the algorithm can be increased significantly by computing a baseline correction over sets of data: by restricting the range of the summation in the computational stencil for each output point. Since for baseline correction, each output data point x_k is located at the same x-coordinate as the input value, the sum in the SB approximation can be taken over the range $k - n$ to $k + n$, where n is sufficiently large to ensure that the tail of the sum is insignificant. For σ on the order of about 100, this means including only several hundred values on either side of the output point into the sum. Clearly, this saves significantly with data sets as large as in the example being considered. In these examples, the sums were computed using a truncated sum. In addition, the costly computation of $\mathrm{erf}(u)$ for each value of u in the sum was done only once, and saved to an array, so that for all subsequent computations of K_n, the values were reused. In computing the baseline curves B_j, $j > 0$, the operation consisted of a short-matrix-vector multiply, which is $\mathcal{O}(n^2)$.

6. FINDING THE BASELINE DIRECTLY

The approach described thus far for finding the baseline is an iterative method, requiring the computation of successive approximations to the data sets D_k as described in Section 3. The convergence rate to a usable baseline depends on the spectral content of the data, as well as whether σ is large or small. Typically, it requires anywhere from 10 to up to 100 iterations to find the baseline, and this does not include the effort required to evaluate the baseline using different values of σ. Clearly, the fundamental approach we have described is usable, however in implementing this approach with the more sophisticated functional representational techniques, including pseudointerpolation and windowing combined with adaptive, intelligent algorithms, would require that many baselines be iteratively constructed.

In many cases, it is possible to construct the baseline directly. The reason is that in most cases, the midline approximation provided by the first iteration B_0 is nearly a shifted copy of the baseline curve. Evidently, this is not always the case, and it is possible to devise spectral data which would cause this approach to break down; however for many of the spectral data examined, this approach provides a quick estimate, and thus can be used in these cases to more rapidly characterize the baseline.

The alternative consists of finding the midline curve, and subtracting this from the data. This removes all of the long-wave oscillations, if we add back the minimum value of this curve, we would get a spectrum which has been straightened out, more or less, depending on the value of sigma. The resulting baseline curve is not computed. The values of σ at which we get the same results as computing the baseline curve iteratively would be different, since in the iterative case, smoothing is applied to a partially smoothed data set at each step.

To illustrate the workability of the approach, consider the results of using the mid-mean algorithm to obtain the corrected spectra shown in Figure 6 and compare this to the

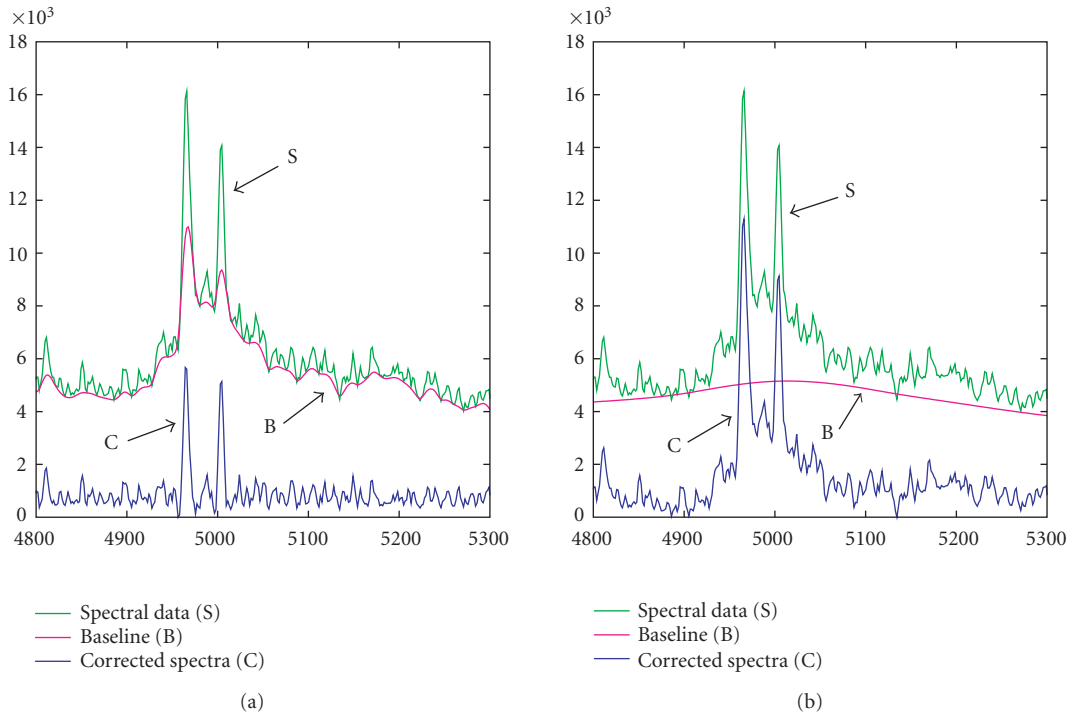

Figure 6: Convergence of the min-mean baseline algorithm (a) using minimal smoothing, $\sigma = 0.1$, and (b) using strong smoothing, $\sigma = 100.0$. The spectra are taken from the same data set as shown in Figure 2.

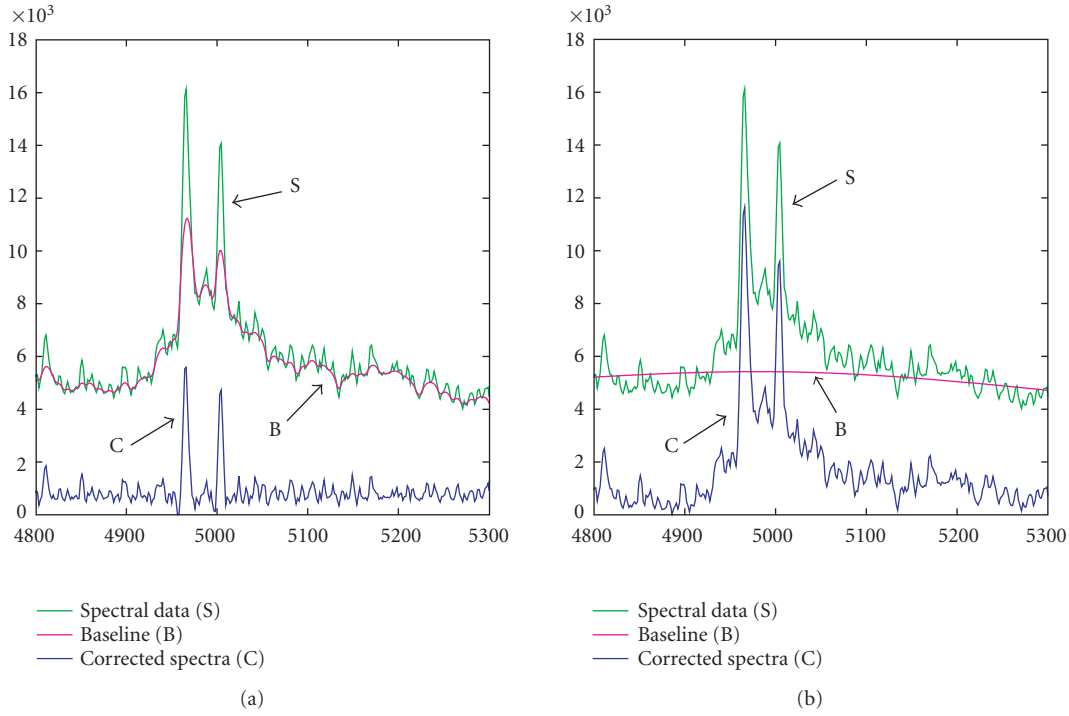

Figure 7: Construction of the corrected spectra using the midline removal (a) using minimal smoothing, $\sigma = 10.0$, and (b) using strong smoothing, that is, $\sigma = 10000.0$. The spectra are taken from the same data set as shown in Figure 2. Note that the baseline curve is not constructed, however the corrected spectra compare well with the results obtained from using the mid-mean algorithm shown in Figure 6.

results shown in Figure 7 for the corrected spectra obtained by using a direct approach. For either case of weak or strong smoothing, the corrected spectra appear very similar, and indeed overlaying these on the same graph would show only negligible differences.

7. CONCLUSIONS AND FUTURE WORK

The application of stochastic Bernstein function approximation can be seen to produce usable families of baseline curves for correcting spectral data bias shift due to low-frequency errors. There are several advantages to this approach, most notably its algorithmic simplicity and robustness. Unlike methods based on interpolation of various means, there is no possibility of any instabilities arising due to the interpolation process, and thus no possibility of generating any spurious oscillations which may affect the signal.

Perhaps the most useful feature of this approach is that the computations can be incorporated into many adaptive algorithms in which the value of σ is optimized with regard to several selection criteria. For constant σ, tuning is simple. More sophisticated analysis may use genetic programming [7] to evolve polynomial terms for the function $\sigma(x)$.

This offers further research opportunities. Is it worthwhile revisiting research that obtains candidate biomarkers and a sample classification from MALDI-TOF data (e.g., [8]) to investigate the sensitivity of results to different amounts of baseline drift removal? Can tuning clarify the nature of chemical noise in different conditions (Section 2)? Finally, by means of supervised-learning, it should be possible to fine tune baseline drift removal for different instrumentation.

The SB method [1] was recently combined with genetic programming [9] and this opportunity is immediately available for problems of baseline drift.

In attempting to optimize the baseline, the use of the direct method for computing the baseline has obvious advantages, and it should be tried before anything else. At worst, it may be necessary to construct it iteratively.

ACKNOWLEDGMENTS

We are grateful to Sequenom Corporation of San Diego, for providing us with MALDI-TOF genomics data. We are also indebted to Professor Franz Hillenkamp from the Institute for Medical Physics and Biophysics at the University of Münster in Germany for furnishing us with the information that is presented in Section 2 of this paper.

REFERENCES

[1] J. Kolibal and C. Saltiel, "Data regularization using stochastic methods," 2005, to appear in *SIAM Journal on Numerical Analysis,* Paper ID is: Manuscript # 063083.

[2] M. Karas and F. Hillenkamp, "Laser desorption ionization of proteins with molecular masses exceeding 10,000 daltons," *Analytical Chemistry*, vol. 60, no. 20, pp. 2299–2301, 1988.

[3] M. A. Ryan, M. G. Buehler, M. L. Horner, et al., *Results from the space shuttle STS-95 electronic nose experiment*, JPL Publication 99-0780, 1999.

[4] G. G. Lorentz, *Bernstein Polynomials*, Chelsea, New Yourk, NY, USA, 1986.

[5] B. Williams, S. Cornett, A. Crecelius, R. Caprioli, B. Dawant, and B. Bodenheimer, "An algorithm for baseline correction of MALDI mass spectra," in *Proceedings of the 43rd ACM Southeast Conference (ACMSE '05)*, Kennesaw, Ga, USA, March 2005.

[6] A. N. Krutchinsky and B. T. Chait, "On the nature of the chemical noise in MALDI mass spectra," *Journal of American Society of Mass Spectrometry*, vol. 13, pp. 129–134, 2002.

[7] J. R. Koza, *Genetic Programming*, MIT Press, Cambridge, Mass, USA, 1992.

[8] H. W. Ressom, R. S. Varghese, E. Orvisky, et al., "Analysis of MALDI-TOF serum profiles for biomarker selection and sample classification," in *Proceedings of IEEE Symposium on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB '05)*, San Diego, Calif, USA, November 2005.

[9] D. Howard and J. Kolibal, "Solution of Differential Equations with Genetic Programming and the Stochastic Bernstein Interpolation," Tech. Rep. BDS-TR-2005-001, Biocomputing Developmental Systems Group, University of Limerick, Limerick, Ireland, June 2005.

Joseph Kolibal received a B.S. degree in chemical engineering from Carnegie Mellon University, an M.S. degree in nuclear engineering from Imperial College, and his D.Phil. degree in numerical analysis from Oxford University. He joined the Mathematics faculty of the University of Southern Mississippi (USM) where he is a Tenured Associate Professor. In 2005 at USM, he developed methods for stochastic Bernstein approximation and interpolation. His research is focused on functional approximation, partial differential equations, and numerical analysis.

Daniel Howard received a B.S. degree in chemical engineering from Lafayette College, an M.S. degree in chemical engineering from Swansea University, and his Ph.D. degree from the Civil Engineering Department of Swansea University. He is a former Research Fellow of Pembroke College and the Numerical Analysis Group of Oxford University. Employed at QinetiQ in the United Kingdom (the former Defence Research Agency), he is a Company Fellow, and he is pursuing research in signal processing, bioinformatics, and evolutionary computation.