Analyst

Cite this: *Analyst*, 2011, **136**, 3130

www.rsc.org/analyst

PAPER

A fully automated iterative moving averaging (AIMA) technique for baseline correction†

Bhaskaran David Prakash and Yap Chun Wei*

Received 7th October 2010, Accepted 18th May 2011
DOI: 10.1039/c0an00778a

Baseline correction is one of the pre-processing steps in the analysis of metabolite signals from chemometric analytical instruments. Fully automated baseline correction techniques, although more convenient to use, tend to be less accurate than semi-automated baseline correction. A fully automated baseline correction algorithm, the automated iterative moving averaging algorithm (AIMA), is presented and compared with three recently introduced semi-automated algorithms, namely the adaptive iteratively reweighted penalized least squares (airPLS), Asymmetric Least Squares baseline correction (ALS) and a parametric method, using NMR, Raman and HPLC chromatograms. AIMA's potential in increasing the accuracy of multivariate analysis *via* SELTI-TOF and LCMS chromatograms was also assessed. The results show that the AIMA's accuracy is comparable to these semi-automated algorithms and has the advantage of ease of use. An AIMA plug-in for an open source metabolomics analysis tool, MZmine, was also developed. The AIMA plug-in is available at http://padel.nus.edu.sg/software/padelaima.

Introduction

The analysis of metabolite signals from analytical chemical instruments has been used in the study of biological processes in both healthy and disease states and may aid in the identification of disease causes, toxicological progression and biomarkers.[1] These raw signals usually have inherent artifacts such as random noise and baseline. Univariate analysis involves mainly the identification and quantification of the metabolites whereas multivariate analysis such as Principal Component Analysis (PCA) is used to discriminate features between two groups such as diseased patients and healthy patients. For both univariate and multivariate analyses, it is necessary to remove the baseline and minimize the noise of the raw signals to increase the accuracy of the analysis. In multivariate analysis, further pre-processing steps such as normalization and alignment are needed.

Baseline correction is one of the components of the chemometric data pre-processing phase. Manual baseline correction though commonly used in vibrational spectroscopy tends to have bias towards user experience, noise levels and baseline characteristics.[2] Automatic baseline correction can be broadly divided into fully automated[3–7] and semi-automated.[8–15] For fully automated baseline correction, the predominant method is polynomial fitting. Polynomial fitting has been shown to perform badly for low signal to noise and signal to background spectrums[3,4] for Raman spectroscopy. In NMR data, even with commercially available polynomial baseline correction, manual correction might sometimes be necessary.[5] Some variants of polynomial fitting have been shown to be suitable for only broad and smooth baseline deviation.[6]

Semi-automated baseline correction, such as the most recently introduced penalized least squares variant, the adaptive iteratively reweighted penalized least squares (airPLS),[8] generally has better accuracy than the current fully automated baseline correction. One general disadvantage of semi-automated methods is the need to optimize parameters, which can be very time-consuming. Other semi-automated penalized square approaches can result in negative valued regions.[3,9] Although default values for these parameters are available for different signals such as NMR, Raman and HPLC chromatograms, the accuracy of baseline correction depends on the careful optimization of these parameters.

Semi-automated wavelet baseline correction techniques transform the signals into different frequency components, followed by the removal of the varying low-frequency background to finally reconstruct the signal from the wavelet coefficient. This reconstruction results in some loss of spectra information and can cause distortion at some part of the spectra.[3] Wavelet based algorithms assume that the background is well separated in the transformed domain from the signal, which may not be correct for real-world spectra.[10]

Semi-automated derivative methods tend to give poor results in high noise environment[11] and it is difficult to interpret the spectra after correction since the peak shapes are changed.[12]

Pharmaceutical Data Exploration Laboratory, Department of Pharmacy, National University of Singapore, Blk S4, 18 Science Drive 4, 117543, Singapore. E-mail: phayapc@nus.edu.sg; Fax: +65-67791554; Tel: +65-65165971

† Electronic supplementary information (ESI) available: See DOI: 10.1039/c0an00778a

In this study, the objective is to develop a fully automated baseline correction method which has similar accuracy as semi-automated baseline correction methods.

Methods

Automated iterative moving averaging (AIMA) algorithm

The moving average is an univariate spectral filtering method used in chemometrics.[16] The most recent use of the moving average as a baseline correction technique was in a computational tool, LIMPIC,[7] where the baseline was estimated using a simple linear interpolation of the average values of signals with selected segments. AIMA was developed based on this idea of moving average smoother. The algorithm is divided into two steps.

The first involves getting a baseline of a spectrum where the peaks are not maximized. Starting with an array of intensities at equal interval $y = [y_1, y_2, \ldots, y_N]$, the first iteration updates the even intensities as follows:

$$y_{i+1} = \text{minimum}(y_{i+1}, (y_i + y_{i+2})/2), \quad (1)$$

where $i = 1, 3, 5, \ldots, N-3, N-1$.

The next iteration updates the odd intensities as follows:

$$y_{i+1} = \text{minimum}(y_{i+1}, (y_i + y_{i+2})/2), \quad (2)$$

where $i = 2, 4, 6, \ldots, N-4, N-2$.

The following iteration updates the even intensities but leaves out the first and last updates as follows:

$$y_{i+1} = \text{minimum}(y_{i+1}, (y_i + y_{i+2})/2), \quad (3)$$

where $i = 3, 5, \ldots, N-5, N-3$.

In a similar note, the next iteration involves the updating of the odd intensities with the first and last updates being left out as follows:

$$y_{i+1} = \text{minimum}(y_{i+1}, (y_i + y_{i+2})/2), \quad (4)$$

where $i = 4, 6, \ldots, N-6, N-4$.

The stopping criterion is when the first update, i, reaches the floor $(N/2)$, where N is the number of intensities. This is known as the Iterative Averaging (IA) procedure, which will be reused with a slight modification in Step 2.

Next, consecutive segments are formed from the initial intensity array where the first and last intensities in each segment are equal to the corresponding intensity value of the derived intensity array y. The first and last intensity positions of these segments are noted and used to update the initial intensity array with linear interpolated intensity values.

Step 2 involves an iterative procedure to maximize the peak. Fig. 1 shows the maximization of the peak after the first and second calls to the Iterative Averaging Smoothing (IAS) function. Step 2 starts with a call to the IAS function using the output array y from Step 1. The first part of IAS involves a loop creating a copy of y known as y'. In the loop, the first iteration updates the even intensities as follows:

$$y'_{i+1} = (y'_i + y'_{i+2})/2, \quad (5)$$

where $i = 1, 3, 5, \ldots, N-3, N-1$.

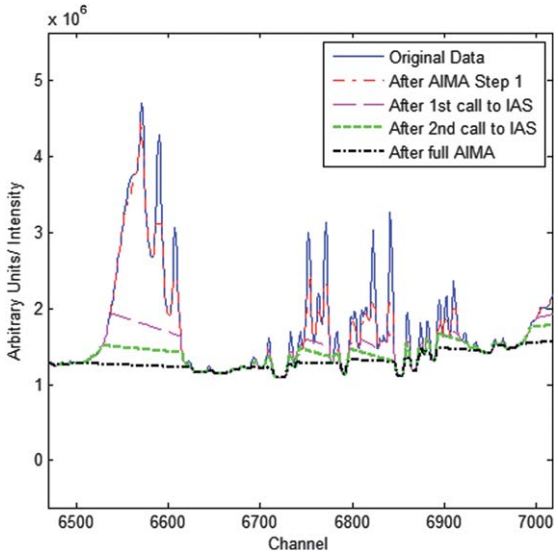

Fig. 1 Zoom in on a sample NMR spectrum from ref. 17 Baseline correction after Step 1, 1st call to IAS function in Step 2, 2nd call to IAS function in Step 2 and after full AIMA (Steps 1 and 2).

The next iteration updates the odd intensities as follows:

$$y'_{i+1} = (y'_i + y'_{i+2})/2, \quad (6)$$

where $i = 2, 4, 6, \ldots, N-4, N-2$.

The following iteration updates the even intensities but leaves out the first and last updates as follows:

$$y'_{i+1} = (y'_i + y'_{i+2})/2, \quad (7)$$

where $i = 3, 5, \ldots, N-5, N-3$.

In a similar note, the next iteration involves the updating of the odd intensities with the first and last updates being left out as follows:

$$y'_{i+1} = (y'_i + y'_{+2})/2, \quad (8)$$

where $i = 4, 6, \ldots, N-6, N-4$.

The stopping criterion is when the first update, i, reaches the floor $(N/2)$, where n is the number of intensities.

Next a new array $y_{\max}$ is created using

$$y_{\max,i} = \max(y'_i, y_i), \quad (9)$$

where $i = 1, 2, \ldots, N$.

Another array $y_{\min}$ is created as

$$y_{\min,i} = \min(y_{\max,i}, y_i), \quad (10)$$

where $i = 1, 2, \ldots, N$.

Using $y_{\min}$ and $y_{\max}$, another array y_{diff} is created using

$$y_{\text{diff},i} = |(y_{\max,i} - y_{\min,i})|, \quad (11)$$

where $i = 1, 2, \ldots, N$.

Then we update every consecutive segment of y', where $y_{\text{diff},i} = 0$ with a linear interpolation using the first intensity, y_f, and the last intensity, y_l, of that particular segment with M as the length of the segment as follows:

$$y'_i = y_f + k(y_f - y_1)/(M) \quad (12)$$

where $k = 0, 1, 2, \ldots, M$.

This is known as IAS.

Suppose the return intensity array of IAS is y', another array y'' is created using

$$y''_i = \min(y_i, y'_i), \quad (13)$$

where $i = 1, 2, \ldots, N$.

Next we get a value A_{abs} using

$$A_{abs} = \sum \left| (y''_i - y_i) \right| \quad (14)$$

Then we repeat the IAS except instead of creating a copy in the start of IAS, y'' is used as y' and y is reused.

Suppose the return intensity array of IAS is a modified y', another array y'' is created using

$$y''_i = \min(y_i, y'_i), \quad (15)$$

where $i = 1, 2, \ldots, N$.

Next we get a value B_{abs} using

$$B_{abs} = \sum \left| (y''_i - y_i) \right| \quad (16)$$

Using A_{abs} and B_{abs}, we get a value

$$C_{abs} = A_{abs}/B_{abs} \quad (17)$$

Step 2 is repeated until the current C_{abs} is less than the previous C_{abs}.

Evaluation of AIMA algorithm

Both simulated and experimental data were used to evaluate and compare the performance and speed of the AIMA algorithm. All data were compared with three other semi-automated baseline correction techniques, airPLS, Asymmetric Least Squares baseline correction (ALS)[13,14] and a parametric baseline correction.[15] airPLS is the most recently introduced baseline correction technique and it is shown to give better accuracy than ALS.[9] Although ALS was shown to have poorer accuracy than airPLS, we decided to include it in our comparison because our set of experimental data was larger than in the earlier study. Thus, it will be interesting to further compare airPLS and ALS on this larger experimental data. Parametric baseline correction[15] is a recent NMR baseline correction method, which has been shown to give better results than a commercial automatic baseline correction function in XWINNMR 3.5.

Simulated data. Simulated data were used because the actual peak heights were known and thus it is possible to compute the baseline correction relative error of the AIMA algorithm. Data were simulated using three different baselines, which are convex curved, concave curved and linear. Pure signals of three Gaussian peaks were used. Each peak varied in intensity. Random noise was also added to the spectrum. Mathematically, the spectrum can be expressed as follows:

$$s(x) = a(x) + b(x) + r(x) \quad (18)$$

where $s(x)$ is the simulated signal, $a(x)$ is pure signal peaks, $b(x)$ is the baseline (either convex, concave or linear), and $r(x)$ is random noise.

A range of noise factor was multiplied by the random noise created to evaluate the ability of the algorithm to perform baseline correction in both high and low noise environment. Values from the sets {0.01, 0.02, …, 1} and {1.1, 0.1, …, 11} were used for low and high noise ranges respectively. For each noise factor the baseline correction was performed with 10 newly generated random noises to minimize bias.

As the simulated data and parameters ($\lambda = 10$, $p = 0.001$ and $d = 2$) were similar to an earlier study using airPLS and ALS, the optimum parameters for both of these methods were obtained from that study.[8] The parametric method required estimation of the standard deviation of noise, σ_P. A systematic search showed that a value of 1×10^3 for this parameter was optimal for the simulated data of low noise and a larger value of 1×10^1 was suitable for high noise data. Fig S1 and S2† of the ESI show a plot of a curved convex baseline spectrum simulated with noise factors of 0.01 and 11 respectively with various baselines estimated using different estimated standard deviations of noise.

Experimental data. Experimental data from HPLC chromatograms, Raman spectra, surface enhanced laser desorption ionization time-of-flight (SELDI-TOF) chromatograms and LC-MS chromatograms were used to show the applicability of the AIMA algorithm in actual datasets. Information about these datasets is shown in Table 1.

For HPLC and Raman spectra, comparison of the three methods for such univariate data was done by measuring the reduction of convex hull of the PCA plots. This is because the compactness and separation in principle components pattern

Table 1 Experimental data used to evaluate AIMA algorithm

Spectra type	Description
HPLC	Eight chromatograms of Red Peony Root,8 which has varying baseline drifts from sample to sample. The Red Peony Root was collected from different producing areas in China, and a standard sample was also bought from the National Institute for control of Pharmaceutical and Biological Products. Two UV spectra per second from 200 nm to 600 nm with a bandwidth of 4 nm resulted in 100 data points in each UV spectrum. The "most peaks rich" wavelength 230 nm was then selected
Raman	Spectra of Prednisone Acetate Tablets (PATs) from 10 different pharmaceutical factories.8 The spectra were measured using a laser of 785 nm wavelength for excitation by BWTEK i-Raman-785 spectrometer with a 2048 elements thermoelectric cooled linear charge-coupled device (TEC-CCD) array and recorded with 5000 ms integration times
SELDI-TOF	One set of chromatograms containing mouse pancreas protein analysis, with 101 spectra from control cells and 80 spectra from cancerous cells21
LC-MS	Subset of the data from 200–600 *m*/*z* and 2500–4500 seconds of the spinal cords of 6 wild-type and 6 FAAH knockout mice22

space would improve clustering and classification results.[8,18] λ values of 30 and 50 which were previously used in airPLS for the HPLC and Raman datasets respectively[8] were assumed to be optimal. For ALS, we performed a grid search using p from the set {0.001, 0.011, …, 0.081, 0.091} and λ from the set {10, 20, …, 490, 500} and determined the optimal parameters that have the minimum reduction in convex hull for both HPLC and Raman datasets. The parametric method required the optimization of the estimated σ_P which was obtained by doing a grid search on the set of values of {1×10^4, 1×10^3, 1×10^2, 1×10^1, 1} and {10, 20, 30, …, C} and assuming the optimal to have the minimum reduction in convex hull for both HPLC and Raman datasets. C is the ceiling of the maximum of the standard deviation of every spectrum and was calculated to be 150 for HPLC datasets and 7250 for the Raman datasets.

For both SELDI-TOF and LCMS spectra analysis, we compared the ability of the algorithms to improve the prediction performance of partial least square (PLS) models. For the SELDTI-TOF data, we selected 64 spectra per class to form a training set for developing PLS models. The remaining spectra were used as a validation set. Root mean squared error of prediction (RMSEP)[19] determined using 10-fold cross-validation (CV) was used to determine the optimum number of latent variables for the PLS models and the optimal parameter combination for airPLS, ALS and parametric methods. Once the optimum PLS model was determined for each algorithm, the prediction performance of these models was assessed by computing the area under the response operating characteristic curve (AUC)[20] using the validation set. The entire process of selecting a training set and validation set, developing, optimizing and validating PLS models was repeated 30 times. For the LCMS data, a similar procedure was used except that the full dataset was used for training and the prediction performance of the optimum PLS model was determined using the cross-validated RMSEP. This is because each class contains only 6 spectra and thus it is not practical to divide the dataset into a training set and validation set. During the optimization of the parameters, we used different λ from the set {10, 20, …, 490, 500} for airPLS and a grid search using p from the set {0.001, 0.011, …, 0.081, 0.091} and λ from the set {10, 20, …, 490, 500} for ALS. For the parametric method, we used σ_P from the union set of {1×10^4, 1×10^3, 1×10^2, 1×10^1, 1} and {10, 20, 30, …, 2700}.

Results

Comparing simulated data using our AIMA and other algorithms

Simulated data using three different baselines are shown in Fig. 2.

The difference between the expected and corrected peak height was calculated for the four algorithms and expressed as a percentage difference to the actual peaks as well as for the overall spectrum. Table 2 shows the percentage error of the different algorithms for individual peak heights and the overall spectrum for the various baselines used. For each row, the top performer is highlighted in bold. Overall, AIMA outperforms all the other three algorithms in high noise environment for all peaks except peak 2 where ALS outperformed AIMA by 1.24%. In the low noise environment, airPLS and AIMA were better than ALS and parametric methods. airPLS tends to work better for convex baseline and AIMA tends to work better for concave baseline. The poorer performance of AIMA on convex baseline may be due to the inherent inflexibility of the fully automated approach

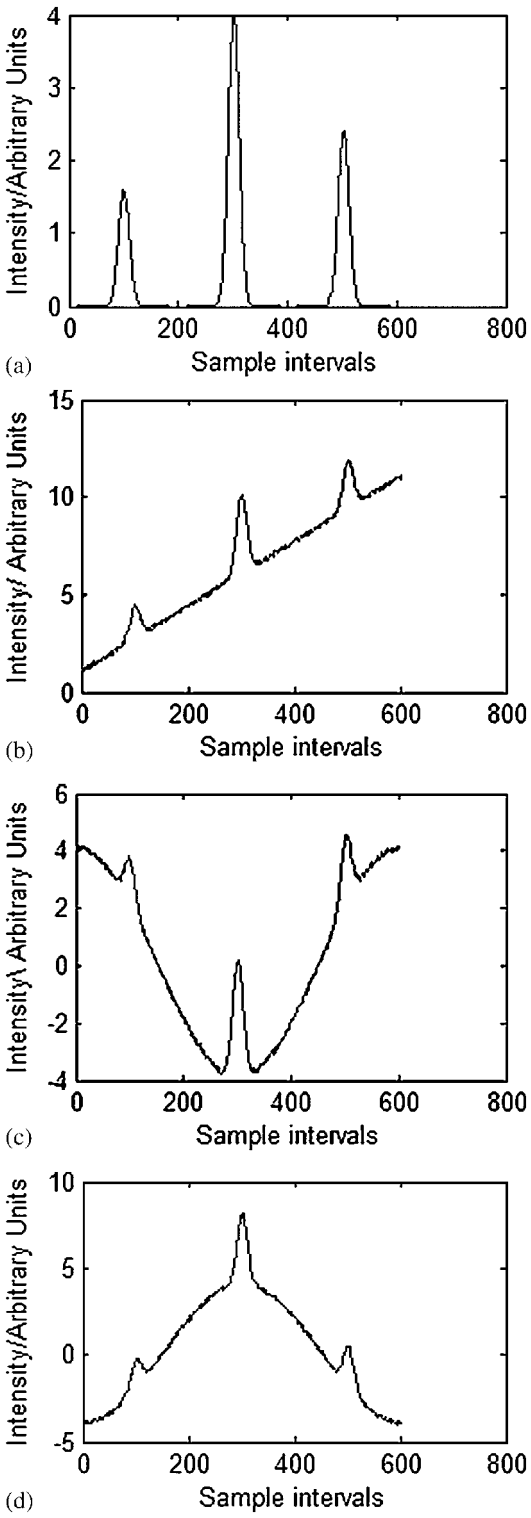

Fig. 2 Simulated data. (a) 3 pure Gaussian signals; (b) pure signal with linear baseline and low random noise; (c) pure signal with convex curved baseline and low random noise; (d) pure signal with concave curved baseline and low random noise.

Table 2 Comparison of percentage error of individual peak heights for all the algorithms and various baselines. For each baseline, we have calculated the percentage error of individual peak heights for both low and high noise for all the algorithms

		Peak 1				Peak 2				Peak 3			
		airPLS	ALS	Parametric	AIMA	airPLS	ALS	Parametric	AIMA	airPLS	ALS	Parametric	AIMA
Concave baseline	Low noise	22.72	52.27	53.00	**9.18**	21.64	56.76	6.78	**3.59**	22.28	55.03	48.82	**6.18**
	High noise	148.40	114.55	134.71	**107.95**	58.57	48.42	55.26	**45.24**	99.61	73.41	94.33	**71.59**
Convex baseline	Low noise	**12.87**	52.28	62.12	14.14	**6.21**	56.75	28.56	7.70	**10.18**	55.03	47.48	11.35
	High noise	155.38	114.56	137.00	**106.61**	72.17	**48.42**	62.81	49.66	105.18	73.41	90.94	**69.91**
Linear baseline	Low noise	**12.44**	52.27	21.50	13.20	10.30	56.77	44.21	**8.69**	**8.45**	55.02	13.94	9.28
	High noise	168.82	114.55	143.00	**109.44**	59.47	48.42	52.96	**42.62**	114.03	73.40	97.96	**72.56**

of AIMA. It is interesting to note that in the spectra where airPLS outperformed AIMA, airPLS outperformed by a maximum error reduction of 1.49%, whereas in the spectra where AIMA outperformed airPLS, AIMA outperformed by a maximum error reduction of 18.05%.

HPLC chromatogram

AIMA was ranked second in terms of the reduction of the convex hull area and was 34.83% behind the top performer, ALS, as shown in Table 3. The much better performance of ALS compared to the other 3 methods suggests that ALS is more suitable for baseline correction of HPLC chromatograms. However, more studies are necessary as the number of HPLC chromatograms used in this study is small.

Raman

Table 4 shows that the performance of the 4 methods is comparable for baseline correction of Raman spectra. AIMA was ranked second in terms of the reduction of the convex hull area and was only 1.72% behind the top performing method, ALS.

SELDI-TOF

The box plot for the AUCs determined using the validation sets for the 30 optimum PLS models of each algorithm is shown in Fig. 3. The parametric method and AIMA had the highest and lowest median AUC respectively. The 3 semi-automated algorithms were able to outperform AIMA through a very careful parameter optimization with a median AUC improvement ranging from 1.13 to 1.17 fold. However, it should be noted that the time taken to optimize these 3 semi-automated algorithms ranged from 28.6 to 197.7 times that required for AIMA.

Table 3 Comparison of percentage reduction in area of convex hull of airPLS, ALS, parametric methods and AIMA

Method	Optimum parameters	Percentage reduction of area of convex hull
ALS	$\lambda = 10, p = 1 \times 10^3$	83.83%
AIMA	N.A.	49.00%
Parametric method	$\sigma_P = 1$	37.77%
airPLS	$\lambda = 30$	35.58%

Table 4 Comparison of percentage reduction in area of convex hull of airPLS, ALS, parametric methods and AIMA

Method	Optimum parameters	Percentage reduction of area of convex hull
ALS	$\lambda = 10, p = 0.031$	99.75%
AIMA	N.A.	98.02%
airPLS	$\lambda = 50$	96.00%
Parametric method	$\sigma_P = 1$	95.59%

LC-MS

The RMSEP of the optimum PLS models for airPLS, ALS, parametric methods and AIMA is given in Table 5. A similar trend is seen as in the SELDI TOF data where the best performer was the parametric method followed by airPLS, ALS and finally AIMA. All the 3 semi-automated algorithms outperform AIMA by 1.03 fold for ALS to 1.23 fold for the parametric method. It is important to note again that the time needed to carefully optimize the parameters of the 3 semi-automated algorithms can be prohibitive compared to AIMA which does not require optimization of parameters.

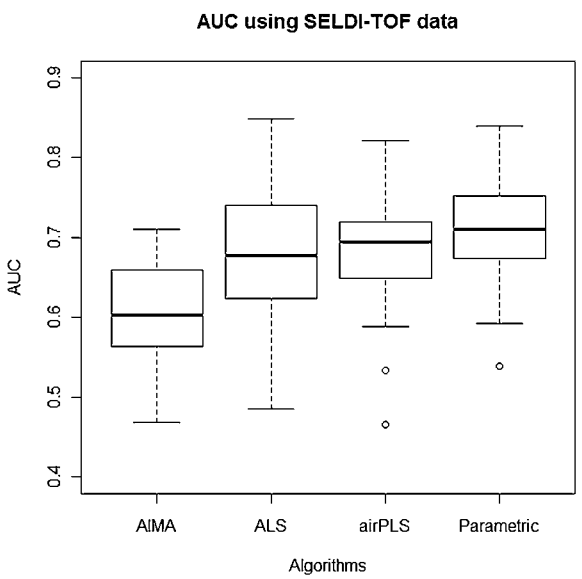

Fig. 3 Box plot of AUC for airPLS, ALS, parametric methods and AIMA using the SELDI-TOF data.

Table 5 Comparison of RMSEP of optimum PLS models for airPLS, ALS, parametric methods and AIMA

Method	Optimum parameters	Optimum PLS latent variables	RMSEP
Parametric method	$\sigma_P = 490$	3	0.445
airPLS	$\lambda = 120$	5	0.461
ALS	$\lambda = 370$, $p = 0.001$	3	0.531
AIMA	N.A.	3	0.549

Table 6 Time taken to process SELDI-TOF data with baseline correction for all parameter combination and PLS optimization of parameter and number of latent variable for airPLS, ALS, parametric methods and AIMA

Algorithm	Time for baseline correction/min	Time for PLS optimization/min	Total time/min
AIMA	5.86	8.33	14.19
airPLS	6.82	398.78	405.6
Parametric	184.49	2621.07	2805.56
ALS	98.92	2957.81	3056.73

Speed

Table 6 shows the time taken in minutes for processing the full SELDI-TOF data to create baseline corrected spectra for all parameter combinations and the optimization of PLS models. Baseline correction was performed on a Duo Core 2.53 GHz Windows Vista Business laptop with 4 GB RAM using Matlab. PLS analysis was performed on a Xeon E5530 2.40 GHz Windows Server 2008 R2 with 40 GB RAM using R. Both the baseline correction followed by PLS was averaged from two runs with very similar timings. AIMAs clear advantage is seen when parameter optimization of the other algorithms is needed as it does not require any optimization. The total time needed for the 3 semi-automated algorithms ranges from 28.6 to 197.7 times that required by AIMA.

MZmine plug-in. An AIMA plug-in for an open source metabolomics analysis tool, MZmine,[23] was implemented and made available at http://padel.nus.edu.sg/software/padelaima. A simple tutorial with screenshots was provided at the website to facilitate installation of our plug-in.

Conclusion

The results in this study show that AIMA is generally comparable to semi-automated algorithms like airPLS, ALS and the parametric algorithm. The AIMA algorithm is a fully automated baseline correction technique whereas other algorithms required optimization of its parameters which would considerably increase the time taken. We acknowledge that further tuning of the parameters for the individual spectrum in each type of spectra was possible. However, individual spectrum parameter optimization would further exponentially increase the computational time for the semi-automated techniques and thus was not done in this study. When processing large datasets, a fully automated algorithm such as AIMA would be desirable as it is not necessary to optimize any parameters. Thus, the AIMA algorithm is a potentially useful baseline correction method for a variety of spectra types.

Acknowledgements

This work was supported by the National University of Singapore (NUS) start-up grant R-148-000-105-133. The authors wish to thank the anonymous reviewers for their comments and suggestions to improve the manuscript.

References

1 H. J. Issaq, Q. N. Van, T. J. Waybright, G. M. Muschik and T. D. Veenstra, *J. Sep. Sci.*, 2009, **32**, 2183.
2 A. Jirasek, G. Schulze, M. M. L. Yu, M. W. Blades and R. F. B. Turner, *Appl. Spectrosc.*, 2004, **58**, 1488.
3 Z. M. Zhang, S. Chen, Y. Z. Liang, Z. X. Liu, Q. M. Zhang, L. X. Ding, F. Ye and H. Zhou, *J. Raman Spectrosc.*, 2009.
4 J. Zhao, H. Lui, D. I. McLean and H. Zeng, *Appl. Spectrosc.*, 2007, **61**, 1225.
5 A. Caligiani, D. Acquotti, G. Palla and V. Bocchi, *Anal. Chim. Acta*, 2007, **585**, 110.
6 D. E. Brown, *J. Magn. Reson., Ser. A*, 1995, **114**, 268.
7 D. Mantini, F. Petrucci, D. Pieragostino, P. Del Boccio, M. Di Nicola, C. Di Ilio, G. Federici, P. Sacchetta, S. Comani and A. Urbani, *BMC Bioinf.*, 2007, **8**, 101.
8 Z. M. Zhang, S. Chen and Y. Z. Liang, *Analyst*, 2010, **135**, 1138.
9 J. Carlos Cobas, M. A. Bernstein, M. Martín-Pastor and P. G. Tahoces, *J. Magn. Reson.*, 2006, **183**, 145.
10 Y. Hu, T. Jiang, A. Shen, W. Li, X. Wang and J. Hu, *Chemom. Intell. Lab. Syst.*, 2007, **85**, 94.
11 A. O'Grady, A. C. Dennis, D. Denvir, J. J. McGarvey and S. E. J. Bell, *Anal. Chem.*, 2001, **73**, 2058.
12 M. N. Leger and A. G. Ryder, *Appl. Spectrosc.*, 2006, **60**, 182.
13 P. H. C. Eilers and H. F. M. Boelens, http://www.science.uva.nl/~hboelens/publications/draftpub/Eilers_2005.pdf, 2005.
14 P. H. C. Eilers, *Anal. Chem.*, 2003, **76**, 404.
15 Y. Xi and D. Rocke, *BMC Bioinf.*, 2008, **9**, 324.
16 R. G. Brereton, in *Encyclopedia of Analytical Science*, ed. W. Paul, T. Alan and P. Colin, Elsevier, Oxford, 2005, p. 51.
17 T. Wang, K. Shao, Q. Chu, Y. Ren, Y. Mu, L. Qu, J. He, C. Jin and B. Xia, *BMC Bioinf.*, 2009, **10**, 83.
18 S. Farashi, M. D. Abolhassani, Y. Salimpour and J. Alirezaie, in *Engineering in Medicine and Biology Society (EMBC), 2010 Annual International Conference of the IEEE*, 2010, p. 6666.
19 B. H. Mevik and H. R. Cederkvist, *J. Chemom.*, 2004, **18**, 422.
20 S. J. Mason and N. E. Graham, *Q. J. R. Meteorol. Soc.*, 2002, **128**, 2145.
21 S. R. Hingorani, E. F. Petricoin, A. Maitra, V. Rajapakse, C. King, M. A. Jacobetz, S. Ross, T. P. Conrads, T. D. Veenstra, B. A. Hitt, Y. Kawaguchi, D. Johann, L. A. Liotta, H. C. Crawford, M. E. Putt, T. Jacks, C. V. Wright, R. H. Hruban, A. M. Lowy and D. A. Tuveson, *Cancer Cell*, 2003, **4**, 437.
22 A. Saghatelian, S. A. Trauger, E. J. Want, E. G. Hawkins, G. Siuzdak and B. F. Cravatt, *Biochemistry*, 2004, **43**, 14332.
23 M. Katajamaa, J. Miettinen and M. Oresic, *Bioinformatics*, 2006, **22**, 634.