Optimal Choice of Baseline Correction for Multivariate Calibration of Spectra

KRISTIAN HOVDE LILAND,* TRYGVE ALMØY, and BJØRN-HELGE MEVIK

The Norwegian University of Life Sciences, Department of Chemistry, Biotechnology and Food Science, P.O. Box 5003, N-1432 Ås, Norway

Baselines are often chosen by visual inspection of their effect on selected spectra. A more objective procedure for choosing baseline correction algorithms and their parameter values for use in statistical analysis is presented. When the goal of the baseline correction is spectra with a pleasing appearance, visual inspection can be a satisfactory approach. If the spectra are to be used in a statistical analysis, objectivity and reproducibility are essential for good prediction. Variations in baselines from dataset to dataset means we have no guarantee that the best-performing algorithm from one analysis will be the best when applied to a new dataset. This paper focuses on choosing baseline correction algorithms and optimizing their parameter values based on the performance of the quality measure from the given analysis. Results presented in this paper illustrate the potential benefits of the optimization and points out some of the possible pitfalls of baseline correction.

Index Headings: **Baseline correction; Statistical analysis; Multivariate calibration; Mass spectrometry; Raman spectroscopy; Matrix-assisted laser desorption–ionization time-of-flight spectroscopy; MALDI-TOF; Partial least squares regression; PLSR; Asymmetric least squares; ALS; Robust baseline estimation; Rolling ball algorithm; Wavelets; Local medians; Iterative polynomial fitting.**

INTRODUCTION

When taking measurements using mass spectrometry or other techniques producing spectra, the baselines of the spectra should be a flat line at zero (no signal). Often, however, the baseline is not flat; it has a linear or nonlinear addition. This baseline addition typically varies more or less randomly between spectra in a data set, which creates problems for many analysis methods. Many algorithms have been developed for identifying and removing the baselines. Most of them have one or more tuning parameters. Finding the algorithm and parameter settings giving baselines that seem to fit the data best is usually done by visual inspection on selected spectra. This is a highly subjective procedure and can be tedious work.

There is no guarantee that the most visually pleasing baselines are optimal when the corrected spectra are used in a statistical analysis, such as regression or classification. Also, different baseline corrections might be optimal for different data sets and statistical analyses. For a given analysis we propose a systematic way to select the algorithm and parameter settings, in which the goodness of the baseline correction is assessed by applying the statistical analysis to the corrected spectra and calculating a quality measure for the analysis. The optimal algorithm and parameter settings are then the ones giving the best value of the measure. This removes much of the subjectivity, and it results in spectra that are better suited for the analysis.

In this paper we will first present our proposal for a more objective procedure for choosing baseline corrections. After this we will describe briefly a number of baseline algorithms, and finally, the procedure will be applied to two real data sets.

PROCEDURE FOR AN OPTIMAL CHOICE OF BASELINE CORRECTION

We propose the following general procedure for evaluating and choosing the optimal baseline correction for any given statistical analysis:

(1) Limit the parameter spaces: For each baseline algorithm select the levels to be tested for all baseline parameters.
(2) Correct baselines and perform the statistical analyses: For each algorithm and combination of its parameter levels, perform the baseline corrections on the calibration data, do the statistical analysis, and calculate the quality measure(s).
(3) Select and validate the optimal parameter levels: For each baseline algorithm, select the combination of parameter levels giving the best quality measure, as judged by the quality measure(s). Validate the resulting baseline corrections by visual inspection.
(4) Select the baseline algorithm(s) giving the best quality measure, apply the correction on independent validation data, and predict the response using the model(s) from the calibration.

Typically the quality measure chosen is the one most widely used for the type of analysis at hand. For classifications, this could be a cross-validated misclassification rate, and for regressions, it could be the generalized cross-validation or root mean squared error of cross-validation (RMSECV). Many statistical methods have a tuning parameter controlling the complexity of the models, such as the number of components in partial least squares regression (PLSR)[1] or the ridge parameter λ in ridge regression.[2] In such cases, it is best to calculate the quality measure for several values of the tuning parameter and judge the quality of the baseline correction by the resulting vector of values. Sometimes the minimum measured value can be used directly for the assessment, but the complexity of the model implied by the tuning parameter should be taken into account in cases where complexity is important.

In the present paper, we focus on regression, trying to predict one or more continuous response vectors from a matrix of spectra. Because the number of predictor variables is high, the covariance matrices of the predictors become singular. This can be overcome by applying dimension reduction methods such as principal component regression (PCR) or PLSR, or covariance stabilizing methods such as ridge regression. In PCR score vectors, $\mathbf{t}_i$, are linear combinations of the predictor variables that try to span as much information as possible in the predictor variable space, $\mathbf{X}$. Using PLSR the response vector, $\mathbf{y}$, is also taken into account by trying to maximize the covariance between the score vectors and $\mathbf{y}$. In the PLSR algorithm used in

Received 7 October 2009; accepted 22 June 2010.
* Author to whom correspondence should be sent. E-mail: kristian.liland@umb.no.

0003-7028/10/6409-1007$2.00/0

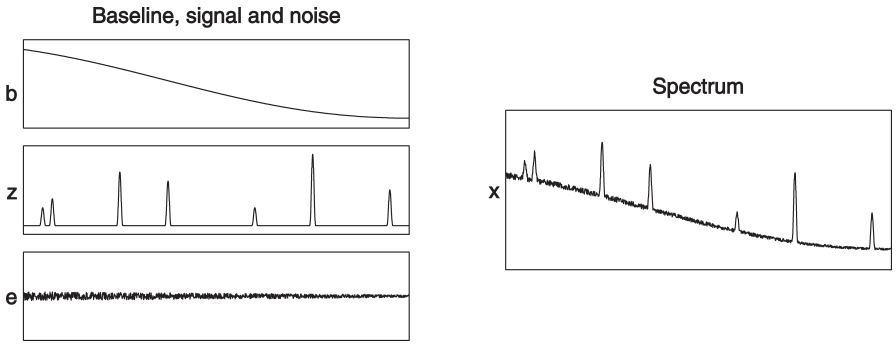

FIG. 1. Sketch of the components of a spectrum from mass spectrometry including **b**: baseline, **z**: signal, and **e**: noise.

this paper predictor variables are centered before the calculation of components starts.

Cross-validation[3] is applied as a tool to find the optimal number of components from the PCR/PLSR model to use for prediction. With cross-validation the data are split into K segments, where $K \leq n$ (the number of samples), and each segment is kept aside once as a validation set that is predicted by the model generated from the remaining $K-1$ segments. A commonly utilized measure of goodness of fit in regression is the before-mentioned RMSECV. This is an estimate of the square root of the expected deviation between the true response and the predicted values, $\theta = \sqrt{E(y-\hat{y})^2}$. For a model containing the k first components this is estimated as:

$$\hat{\theta}_k = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(y_i - \hat{y}_{(i),k}\right)^2} \qquad (1)$$

Here, N is the number of observations, y_i is the ith observation, and $\hat{y}_{(i),k}$ is the prediction of the ith observation based on the k first components, using a model where observation i has been held aside. One can also compute $\hat{\theta}_0$ by substituting $\overline{y}_{(i)}$ for $\hat{y}_{(i),0}$ to compare the different models to using the mean value as a response estimate. It is important to use RMSECV with care when working with the mentioned feature extraction methods because these employ a parameter controlling the complexity. The choice of number of components to include in the model will be a balance between minimizing RMSECV and limiting the model complexity. When assessing the fit of validation data, root mean square error of prediction (RMSEP) is calculated. This is equal to RMSECV only replacing the predictions from cross-validated fits with predictions from the validation data.

Depending on the analysis, there are many different regression and classification methods that can be used in combination with different measures of the goodness of fit. In the following the main focus will be on PLSR in combination with RMSECV and RMSEP.

BASELINE ESTIMATION ALGORITHMS

There exist several definitions of the term "baseline" in the literature. When working with spectrometric data it is common to think of a spectrum **x** as consisting of three components: baseline, **b**; signal, **z**; and noise, **e**. The easiest situation to describe is the one in which the three components are purely additive. This would give the following equation:

$$\mathbf{x} = \mathbf{b} + \mathbf{z} + \mathbf{e} \qquad (2)$$

A sketch of these components is given in Fig. 1. The goal is therefore to estimate **b** so that it can be removed from the equation. This would leave only the pure signal and some noise. Of course the picture is seldom as simple as this. There is often convolution, and there can be other interactions between the components. A common case is the one in which the noise or baseline varies in some linear or nonlinear fashion with the scale and intensity of the signal.

A more algorithm-focused way of defining the baseline is to say that the baseline is the slowly varying curve going through the lower part of the spectra without the jumps of the peaks. In this way we only distinguish between baseline and signal. In practice this is equally useful when the main goal is to identify the baseline and leaves noise as a separate problem to be handled elsewhere.

In most spectra the signal can be viewed as a non-negative phenomenon. Noise, however, can also take negative values in some applications. This is reflected in the different baseline-correction algorithms. Some algorithms place the baseline below the spectrum everywhere, so that no point will get a negative value after subtraction of the baseline. Most of the algorithms we have encountered, however, place the baseline somewhere inside the noise, either quite low or in the center of it. Depending on the measuring techniques and applications, different distributions of noise occur. In the three-component model described, the noise component could for instance be equally spread above and below the baseline.

Another angle to the definition of baselines is to define peaks instead and leave the rest as baseline. Many algorithms exist that solely try to identify peaks or that use identification of peaks as an aid to estimating baselines.

Figure 2 shows a matrix-assisted laser desorption–ionization time-of-flight (MALDI-TOF) spectrum from a mixture of cow, goat, and ewe milk. We have marked some of the different shapes of peaks with rectangles. This shows how peaks from one spectrum can be both narrow, wide, sharp, and smooth. Where to place the baseline in noisy areas, like the one up to the peak at around 8500 m/z in the figure, can depend on the amount of information present. If the area is mostly comprised of noise, the baseline would probably best be placed in the center. If some of the larger fluctuations are real peaks, the baseline might better be placed quite low in the "noise". What looks like a rise in the baseline between 11 000 and 13 000 m/z is to some degree a cluster of partially overlapping peaks,

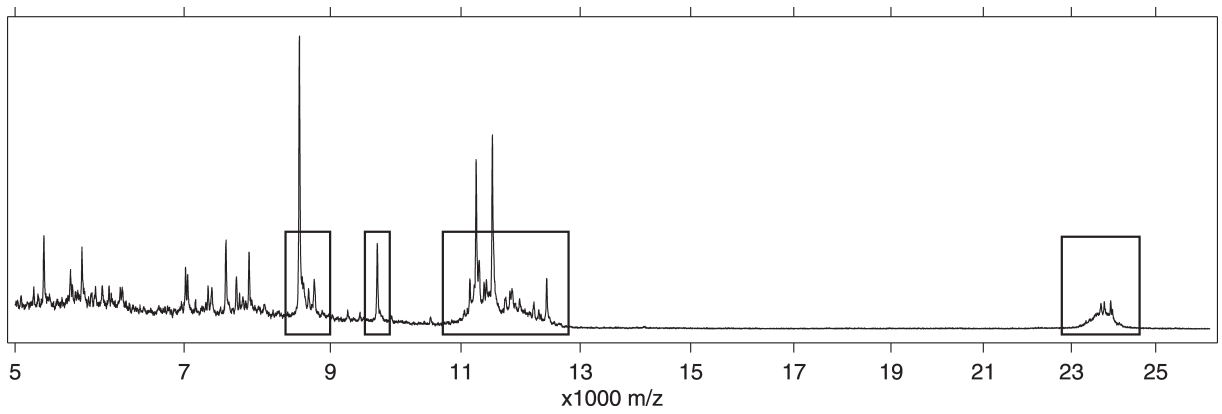

FIG. 2. Some types of peaks in a MALDI-TOF spectrum including narrow and broad peaks and clusters of peaks. The spectrum was recorded from a mixture of cow, goat, and ewe milk.

making the position of the baseline extra important and difficult.

In the following sections, we will describe some of the baseline algorithms available and point out some potential strengths and weaknesses. The focus of this paper is on how to evaluate and choose baseline corrections, and not on the baseline algorithms themselves, so the descriptions will be brief, with basic ideas and some emphasis on the use of parameters.

Asymmetric Least Squares. The asymmetric least squares (ALS) algorithm is described by Eilers and Boelens.[4] It uses a least squares algorithm weighting down predictor variables with positive residuals and smoothing by adding a second-derivative constraint:

$$S = \sum_i w_i (x_i - b_i)^2 + \lambda \sum_i (\Delta^2 b_i) \tag{3}$$

Here x_i is the original spectrum, b_i is the current estimation of the baseline, w_i is the asymmetric weighting of the residuals, and $\Delta^2 b_i$ is the second derivative of the current baseline. There are two parameters: the amount of smoothing λ and the magnitude of the weights, w_i. In Ref. 4. positive residuals are weighted by $w_i = p$ and negative by $w_i = 1 - p$, where $0 \leq p \leq 1$. The signs of the residuals can change from iteration to iteration giving different weights in the next iteration. Usually one iterates until there are no changes in the signs of the residuals. A potential problem with the algorithm can be seen in the exaggerated example in Fig. 3, where too much weight to positive residuals and too little constraint on the second derivative results in filling the peak cluster in such a way that the baseline cuts above the lower parts of the peaks, creating artificial negative peaks in the corrected spectra.

Robust Baseline Estimation. Ruckstuhl et al.[5] have proposed an algorithm called robust baseline estimation (RBE). This is an extension of the LOcally WEighted Scatter plot Smoother (LOWESS), adding more possibilities for weighting. RBE calculates an estimate of the regression curve $\hat{g}(t_i, \hat{\theta})$ where:

$$\hat{\theta}(t_0) = \arg\min_{\theta} \sum_{i=1}^{n_p} \rho(t_i) w(t_i) K\left(\frac{t_i - t_0}{h}\right) \times \{x_i - [\theta_0 + \theta_1 (t_i - t_0)]\}^2 \tag{4}$$

Here t_i are the spectrum abscissas, t_0 is the current abscissa, x_i are the spectrum intensities, n_p is the number of spectrum intensities used, $\rho(t_i)$ are estimated measurement errors along the spectra, $w(t_i)$ are robustness weights, and $K(u)$ is a kernel used for the locally weighted regression. A tricube kernel is employed: $K(u) = [\max(1 - |u|^3, 0)]^3$. As with ALS, RBE uses asymmetric weighting of the observations, here in the form of

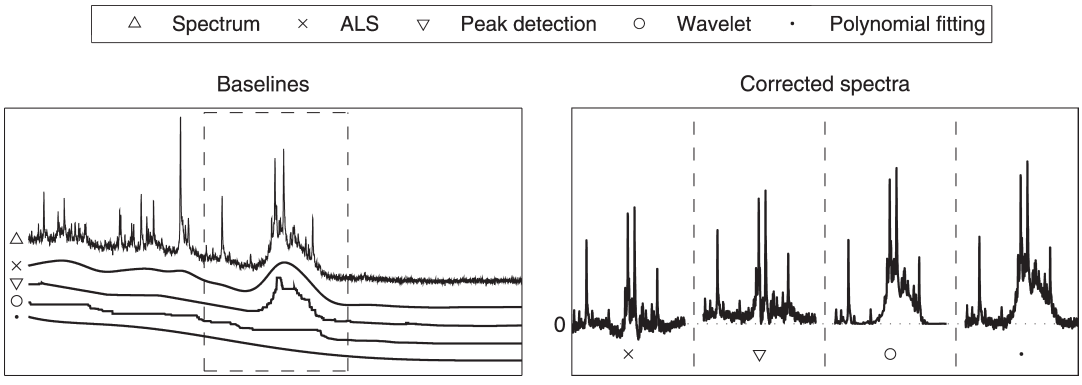

FIG. 3. Spectrum corrected by four baseline-correction algorithms. The estimated baselines are illustrated under the spectrum on the left, while the corrected peak cluster has been magnified in the right part of the figure. (Vertical positions of baselines are arbitrary.)

Tukey's bi-square weights:

$$w(t_i) = \begin{cases} 1 & \text{if } r_i < 0 \\ \left\{\max[1 - (r_i/b)^2, 0]\right\}^2 & \text{otherwise} \end{cases} \quad (5)$$

where $r_i = \sqrt{\rho(t_i)}[x_i - \hat{g}(t_i)]/\hat{\sigma}$ and usually $\hat{\sigma} = \hat{\sigma}_{MAV} = \text{median}[\sqrt{\rho(t_i)}|x_i - \hat{g}(t_i)|]/0.6745$.

There are many parameters that can be altered, but most of these work well in the standard settings proposed by the authors. We choose to use two parameters; n_p giving the number of points to use in the regression and b giving the amount of weighting of positive residuals. If the measurement error along the spectra is unknown, $\rho(t_i)$ can be omitted. Otherwise the association $\text{var}[x(t_i)] = \sigma^2/\rho(t_i)$ is used.

Wavelets. Coombes et al.[6] utilize undecimated discrete wavelet transforms (UDWT) with Daubechie's wavelets to smooth spectra. Wavelets are finite length or fast-decaying oscillations where "mother wavelets" are combined by convolution through scaling and translating to adapt to a signal or spectrum. The process of "rebuilding the spectrum" with wavelets extracts information connected to the unknown function generating the spectrum. Discrete in this sense means that the shift and translation has a limited number of possible values. Daubechie's wavelets is a class of commonly used wavelets recursively generating finer discrete samplings of an implicit mother wavelet function.[7] The undecimated version is shift invariant and has preferable properties when used for denoising.

Starting from the left after smoothing, the baseline follows the spectrum whenever it decreases and remains level if the spectrum increases. This means concavely shaped data will be poorly corrected, and a steep baseline will have a staircase shape as seen in Fig. 3. Required parameters are a threshold for what to regard as noise: t, where in the spectrum to start smoothing, and the block size of the wavelets, L.

Local Medians. Friedrichs[8] has made a so-called model-free algorithm for the removal of baseline artifacts. This finds the median of local windows and smooths these by using Gaussian weights. By choosing the right window width, the median will most often be a part of the baseline, given that the points in the baseline are more frequent than the points in the peaks. The baseline is found by:

$$B(i) = \sum_{j=i-(W/2)+1}^{i+(W/2)} M(j)G(i-j) \quad (6)$$

where $M(j)$ is the median of the window around point j and $G(k)$ is the Gaussian weight of the medians in the local area:

$$\sum_{k=-(W/2)}^{(W/2)-1} G(k) = 1 \quad (7)$$

Here the only parameter needed is the window width W.

The algorithm was developed for NMR spectra where the endpoints are usually of equal intensity so that windows can be wrapped across the spectrum boundaries. For spectra where this assumption does not hold, the authors choose to keep the first full window from 1 to W for the $W/2$ first points and vice versa for the last points. The result is that the first $W/2$ points get the same intensity and the last W/2 points get the same intensity.

We have chosen to make two adjustments to the algorithm to make it more adaptable to other kinds of mass spectra. Because baselines can be quite steep near endpoints in spectra, e.g., in Raman and MALDI-TOF, we use a shrinking window size near the endpoints so that windows remain symmetrical and we avoid the constant baseline artifact. In addition, we have split the window parameter so that median windows, w_m (W in Eq. 6), and smoothing windows, w_s (W in Eq. 7), can have different sizes. This has proved valuable for some types of spectra since smoothing sometimes does not need as wide a window as the medians do. As for the Gaussian weights, we use the normal distribution from $-w_s$ to w_s with mean 0 and standard deviation w_s. Near the endpoints the vector of Gaussian weights is cut and divided by the sum of its remaining elements, making an asymmetric window with higher weights near the current point.

Iterative Polynomial Fitting. Polynomial fitting is one of the most common forms of baseline correction and comes in many different variants. Gan et al.[9] have made an iterative polynomial fitting algorithm that fits an nth degree polynomial to the data. For each iteration it sets the value of all points of the current spectrum that are higher than the polynomial to the value of the polynomial and refits the polynomial. When the change from one iteration to the next becomes smaller than a chosen threshold, the algorithm stops. The final baseline is the last predicted polynomial.

$$x_{new} = \min(x_{predicted}, x_{old}) \quad (8)$$

where x is the spectral value at a point. Lieber and Mahadevan-Jansen[10] have modified this so that, instead of using the minimum of the predicted x and the old predicted x for each iteration, it takes the minimum of the predicted x and the original x:

$$x_{new} = \min(x_{predicted}, x_{original}) \quad (9)$$

The difference might seem insignificant but the effect is that an iteration giving a too low estimate of the baseline ($x_{predicted}$) might be increased in the next iteration in Eq. 9 if the next iteration gives a higher estimate of the baseline ($x_{predicted}$). In Eq. 8 the sequence is monotonically non-increasing as x_{new} in one iteration becomes x_{old} in the following iteration. For both algorithms two parameters are needed: the degree of the polynomial, d_p, and a threshold for the convergence criterion, t. In case one wants to stop the iterations before convergence, one can add a parameter for the maximum number of iterations allowed, m. We use the modified algorithm in Eq. 9 in the rest of this paper. An example of this algorithm is seen in Fig. 3 where a polynomial of degree 12 is used.

Simultaneous Peak Detection and Baseline Correction. Peak detection is a field of its own too large to be handled in this paper. The reason for mentioning it is that Coombes et al.[11] have made algorithms for peak detection and baseline removal. Simple peak finding (SPF) and simultaneous peak detection and baseline correction (SPDBC) try to iteratively locate peaks as consistent deviations from noise and interpolate over the spectra after peak removal. Though some of the parameters have standard values that can be used, the original algorithm employs many parameters: a minimum signal-to-noise ratio, s_n; four window size parameters, l, r, l_w, and r_w; a parameter telling if the baseline should be monotonically decreasing, m;

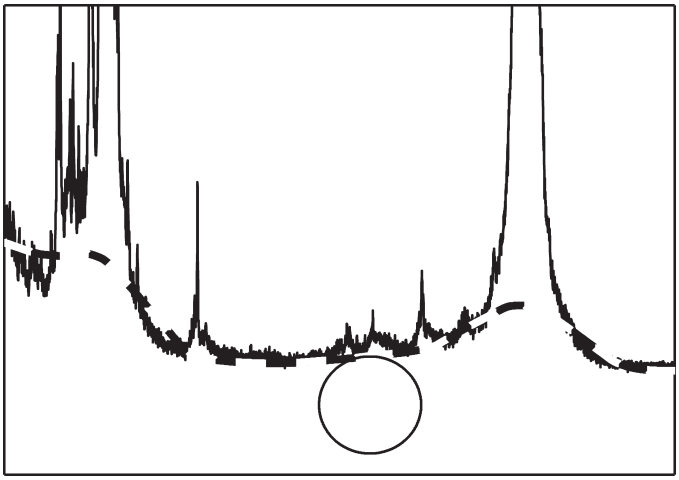

FIG. 4. Illustration of a ball rolling on the underside of a spectrum.

and an optional peak width multiplier, w_m, used for discarding doubtful peaks. l and r indicate the smallest and largest window sizes, smoothly and quadratically increasing from left to right, of the maximum peak width. l_w and r_w are the equivalent parameters for finding minimum and median values in windows in peak removed spectra. Assuming constant window sizes along spectra makes two of the parameters redundant. As can be seen in Fig. 3, the baseline is placed below the spectrum at all points, meaning all noise will be non-negative. If an existing peak in the spectrum is not detected, the estimated baseline might partially or completely follow the shape of the peak, removing much of the peak in the baseline-corrected spectrum. An example of this is seen in Fig. 3.

Rolling Ball. Kneen and Annegarn[12] have described a baseline removal algorithm where one imagines a large ball rolling on the underside of the spectrum. The baseline is simply the trace of the topmost point of the ball. The original algorithm was made for X-ray spectra and has in this paper been simplified to not making the window width change across the spectrum. In three loops the algorithm finds minimum points in local windows, finds maximum points among the minimum points, and smooths by averaging over the maximum points. If one wants to be able to differentiate between window sizes for baseline identification and smoothing, one needs two parameters, w_m and w_s. Otherwise, only one parameter is needed. An illustration of this method is given in Fig. 4.

TABLE I. Raman on fish oil. Optimal parameter values.

Algorithm	Parameters
ALS	$\lambda = 7$, $p = 0.11$
RBE	$n_p = 1400$, $b = 1.5$
Wavelet	$t = 12$, $L = 10$
Local medians	$w_m = 300$, $w_s = 300$
Polynomial fitting	$d_p = 5$, $t = 10^{-4}$, $m = 100$
Peak detection	$s_n = 11$, $l = r = 500$, $l_w = r_w = 100$
Rolling ball	$w_m = 430$, $w_s = 175$

MATERIALS

Two data sets were used in the experiment. These were chosen to span some of the potential applications of optimal choices of baselines. Looking at Figs. 2 and 5 we see that the challenges of the baseline correction algorithms are quite different for the individual types of spectra.

Raman on Fish Oil. The first data set is a set of 45 salmon oil samples extracted from farmed salmon (*Salmo salar*). Raman spectroscopy with a UV laser has been conducted. As a fat indicator the iodine value has been chosen as the response for regression.[13]

Raman spectroscopy is a spectroscopic technique used in condensed-matter physics and chemistry to study vibrational, rotational, and other low-frequency modes in a system. It uses monochromatic light, usually some form of laser, to excite these low-frequency modes. A lens picks up the light and sends it through a monochromator with a filter for removing laser effects. The resulting light can be captured by a charge-coupled device (CCD) detector or some other form of light-recording device. Especially in biological samples using UV lasers, fluorescence is a competing emission process, often orders of magnitude stronger than the Raman scattering. Baseline correction can potentially remove most of this effect.

Spectra were collected using a Raman RXN1 Analyzer. The instrument consisted of a holoprobe transmission holographic spectrograph and a CCD detector with a working temperature of −40 °C. The spectrograph was connected with fiber optics to a Kaiser multi-reaction filtered probe head, and the system was equipped with a 785-nm stabilized external cavity diode laser. All Raman spectra were obtained using a 20-cm-long and 12.5-

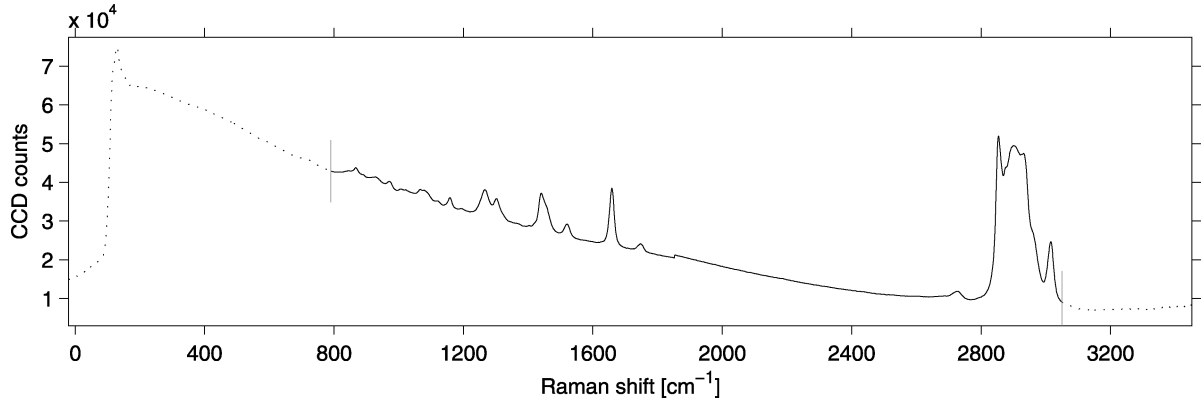

FIG. 5. Raman spectrum from measurements on fish oil. The main rise in the spectrum starting from around 200 and ending at around 3000 cm^{-1} shift comes from fluorescence. This should be corrected for.

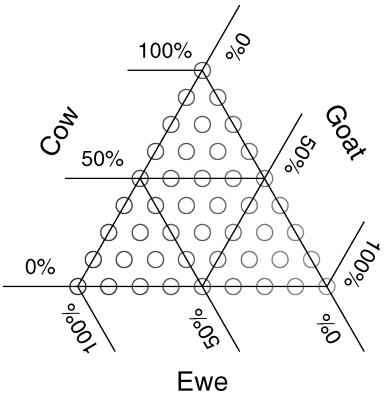

FIG. 6. Simplex lattice mixture design with three mixture components: cow milk (upper corner), goat milk (lower right), and ewe milk (lower left).

mm-o.d. Raman immersion ballprobe (Matrix Solutions, Bothell, WA).

To make comparison to Ref. 13 easier, we have not preprocessed the spectra before baseline correction. Instead normalization is done by dividing each baseline-corrected spectrum by what is called its total intensity: $\sqrt{\Sigma_{i=1}^{N} x_i^2}$. Before baseline correction and normalization we have cut away all wavelengths before 790 cm^{-1} and after 3050 cm^{-1}, because these contain extreme variations, a minimal amount of information, and several bands due to the sapphire probe used.

MALDI-TOF on Milk Mixtures. The second data set is a set of $45 \times 4 \times 2$ raw spectra from a designed experiment on mixtures of cow milk, goat milk, and ewe milk (milk mixtures $\times$ spot replicates $\times$ full technical replicates). The mixture design was a simplex lattice, as shown in Fig. 6. The weighed proportions of each type of milk were used as responses. MALDI-TOF spectra of 21 451 m/z values have been created for each sample, but only the first 6179 values are used for modeling and prediction. Figure 2 shows part of one of these spectra. The data consist of a calibration set (first technical replicate) and a validation set (second technical replicate) created under almost identical conditions from the same bulk milks.[14]

Matrix-assisted laser desorption–ionization is a soft ionization technique that does not shatter molecules as easily as more conventional ionization methods. This allows it to analyze proteins, peptides, sugars, etc. A matrix, in this sense, consists of crystallized molecules mixed into the test samples and placed in spots. The laser is fired at the crystals in the spots, which absorb the energy, protecting the molecules in the analyte. This is thought to ionize the matrix, which in turn ionizes the molecules of the test sample. Most of the molecules get a single charge. Because of a difference in electrical charge in the MALDI-TOF instrument, the charged molecules are drawn against a target with a time-of-flight mass spectrometer. This measures how much time the molecules spend between the spot and the target, producing a spectrum of counts of mass to charge ratios, m/z.

Spectra were collected using an Ultraflex MALDI-TOF spectrometer from Bruker Daltonik GmbH. The UV source was a N_2 laser with a wavelength of 337 nm, a pulse energy of 100 μJ, and a 1-ns pulse width. Tandem TOF is possible, but only ordinary TOF was utilized for the milk samples. The vacuum system consisted of two high vacuum areas ($<8 \times 10^{-7}$ mbar) and three rough vacuum areas (from atmosphere to 10^{-2} mbar). The detectors converting ion current to electrical current were 1000 volt micro-channel plate detectors with millions of holes (Ø = 5–10 μm, length = 0.5–0.8 mm) working as electron multipliers.

There was an error in the MALDI-TOF settings when analyzing the validation data so that a slightly lower laser intensity was applied, resulting in spectra containing more noise than the calibration data. In addition, the protein content of the ewe milk is around twice as high as in the other milk types, resulting in a nonlinearity in the response. This is partially mended by using a square root transformation in the modeling and prediction. The error and the nonlinearity make this a very challenging validation set.

Before applying baseline correction we normalize the spectra by dividing each spectrum by its mean value. Because the milk data are counts of the number of particles having given m/z values, this can be described as a Poisson process. A common way of standardizing Poisson data is to divide by the empirical standard deviation estimated by $\sqrt{x_i}$, which is approximately the same as applying the square root to the spectrum values as $(x_i/\sqrt{x_i}) = \sqrt{x_i}$.[14] A preprocessing step is therefore included before normalization and baseline correction by taking the square root of the original spectrum.

TABLE II. MALDI-TOF on milk mixtures. Optimal parameter values.

Algorithm	Parameters
ALS	$\lambda = 8$, $p = 0.14$
RBE	$n_p = 1750$, $b = 4.5$
Wavelet	$t = 7.5$, $L = 6$
Local medians	$w_m = 425$, $w_s = 50$
Polynomial fitting	$d_p = 5$, $t = 10^{-3}$, $m = 5$
Peak detection	$s_n = 1$, $l = r = 800$, $l_w = r_w = 200$
Rolling ball	$w_m = 400$, $w_s = 200$

EXPERIMENTAL

We will apply cross-validated multivariate regression to baseline-corrected and normalized spectra to find optimal baseline algorithms and corresponding parameter settings for spectra from Raman and MALDI-TOF. As a benchmark we include regression on spectra where no baseline correction has been done.

To get some idea of where to start looking for the best parameters for the different algorithms, visual inspection of baselines can be done. A simple way of searching for the optimal settings is to create a wide grid around a set of parameter settings that visually seem to be a good choice. The grid is adapted iteratively to narrow in on the most promising areas. It is of high importance to inspect the resulting baseline corrections after the optimization has been done to avoid over-fitting and local minima. Unfortunately, faulty baseline corrections can introduce "false information" to the spectra that correlates with the response of the regression or classification. This can usually be avoided by proper validation of spectra and analyses.

Fish Oil. To enable test set validation we split the data set into 30 calibration samples and 15 validation samples before the analysis. This will give higher credibility to the results and show whether the cross-validated prediction error is overly optimistic or if over-fitting has occurred.

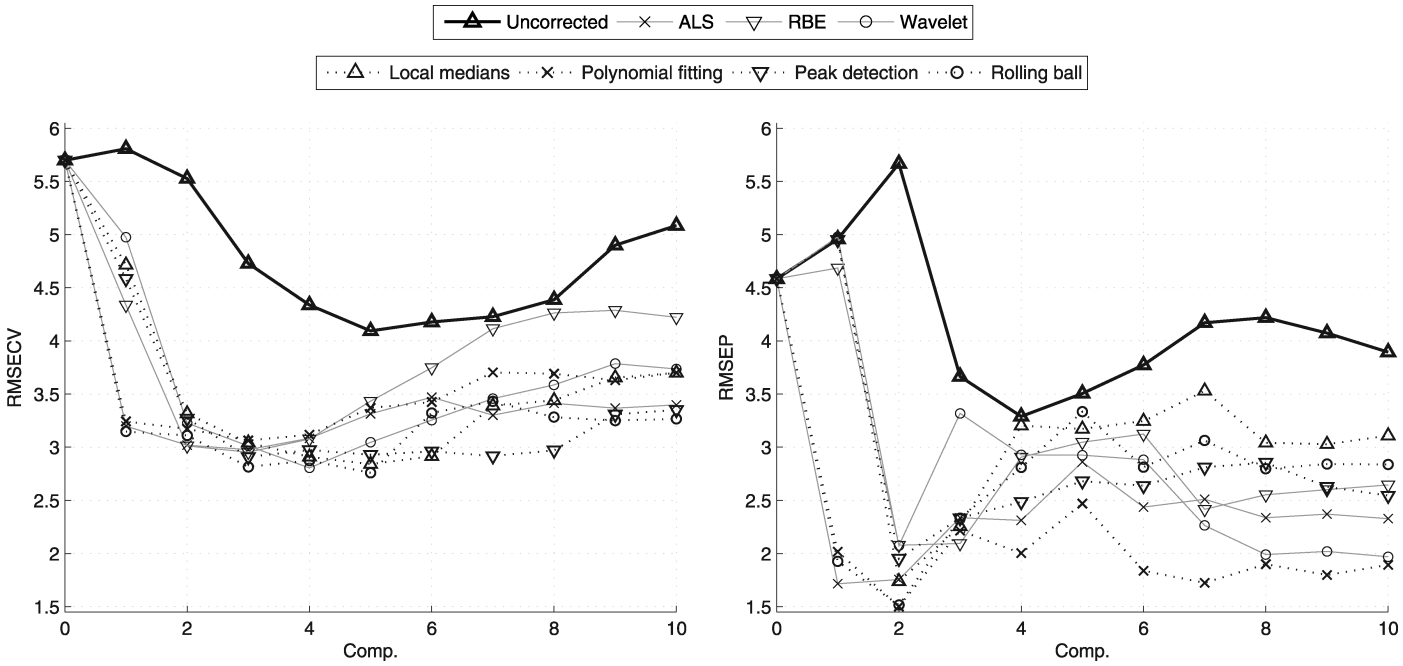

FIG. 7. Fish oil data: RMSECV and RMSEP curves from PLS regression based on original spectra and optimized baseline-corrected spectra. Based on the cross-validation, one component is chosen for prediction of the validation data for the ALS, polynomial fitting, and rolling ball algorithms and two components for the rest of the methods.

For prediction we use PLSR with RMSECV from leave-one-out cross-validation. Leave-one-out cross-validation is used because there are no replicates and the dataset is relatively small. Since the RMSECV values are heavily distorted by the baseline, using the shape of the RMSECV curve from the uncorrected data to choose the number of components to optimize on makes no sense. Instead we observe that the main drop in RMSECV values comes in the first two components using data corrected by most algorithms, with a slight improvement in the third component for some algorithms. Therefore, we choose to optimize on minimum RMSECV up to three components.

The sequence of operations on the calibration spectra becomes: baseline correction, total intensity normalization, cross-validated PLS regression, and RMSECV calculation. The same preprocessing will be applied to the validation data, and RMSEP will be calculated from the predictions.

Milk Mixtures. The calibration data will be used in the optimization while the validation data will be used to check the performance of the optimal parameter settings on new data. For prediction we use three separate PLSR models with RMSECV from cross-validation on raw and corrected spectra to try to see which corrections give the best models. The spectra are comprised of spot replicates in groups of four, meaning that the cross-validation segments must be a multiple of four in size (4, 20, 36, or 60 if we want equal segment length). We choose a segment size of 20, giving nine-fold cross-validation by holding out five groups of spot replicates at a time. As in Liland et al.,[14] we build all models using each spot replicate as a separate measurement to produce robust models, and we use averages over preprocessed and baseline-corrected spot replicates for prediction to obtain stable and precise results.[15]

As mentioned, the choice of number of components is important. One way of limiting the number of components is to look at the RMSECV for the uncorrected spectra, saying that a model that needs a significantly higher number of components to get a good prediction is uninteresting. In our case, the mean RMSECV for the three responses reaches its minimum in the first ten components, but has its largest drop up to two components and a steady decrease to around five components. Because of this, we choose to show the first ten components in our plots and optimize on minimum mean RMSECV for the three responses at five components in accordance with Liland et al.[14]

As there are three responses, we end up with three sets of RMSECV values. Instead of only averaging over these and looking for a common minimum in the optimization, we apply generalized multiplicative analysis of variance (GEMANOVA)[16,17] to look for common trends. To use this procedure we make an array with the number of modes (dimensions) equal to the number of varied parameter settings plus one for the milk types and place the RMSECV values in the array according to their respective level combinations. Fitting a multi-way GEMANOVA model to this array, we can estimate how RMSECV values change along each of the parameters across all the other parameters and milk types. Minimizing RMSECV for each parameter is then reduced to a series of one-dimensional problems that are easily visualized. We can indicate whether higher resolution should be used for a particular parameter value by using a bootstrap procedure.

The sequence of operations on the calibration and validation spectra is the same as for the fish oil data, except that normalization is done by taking the square root and dividing by the spectrum means.

RESULTS

Fish Oil. Following the proposed sequence of calculations from the Experimental section, we have used RMSECV to seek for the optimal parameter settings for the baseline algorithms at

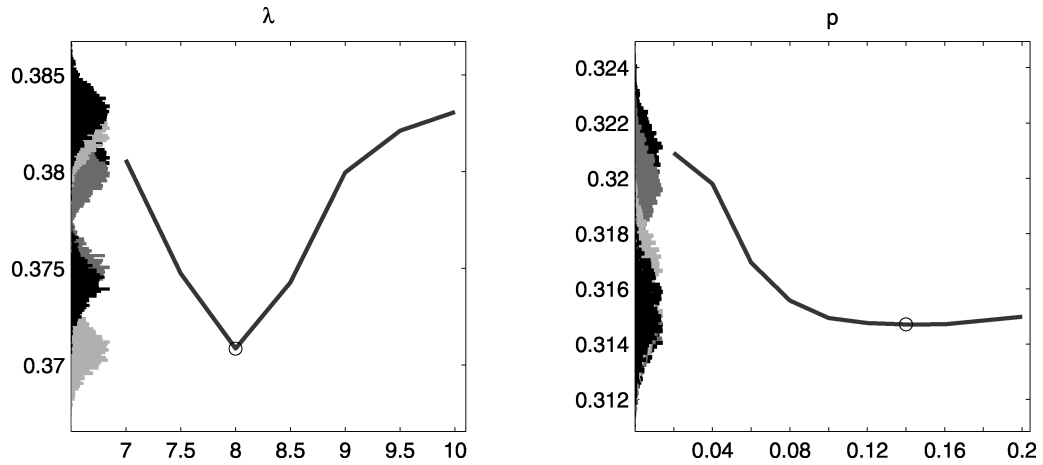

FIG. 8. GEMANOVA loadings for RMSECV based on ALS. The *y*-axis indicates how large the relative difference in RMSECV between parameter values is. Minima are found at $\lambda = 8$ and $p = 0.14$. Estimated P values for the difference from the minima to the closest neighbors are 0.0095 and 0.0205 for λ and 0.476 and 0.504 for p, respectively.

hand (see Table I). When plotting RMSECV and RMSEP values, lower values are better than higher values. A model having low RMSECV/RMSEP values with few components will usually be simpler to interpret than one obtaining the low values with a higher number of components. In addition, we may prefer a model that has stable results around the optimum over one that is fluctuating much. From Figs. 7a and 7b we see that none of the algorithms perform worse than using the uncorrected data. When using only one or two components, several algorithms have an RMSECV that is around half of what can be found for the uncorrected spectra. For the validation data the difference is even larger. We observe that the ALS, polynomial fitting, and rolling ball methods need only one component to reach a satisfactory level of error in the cross-validation, with only minor improvements including two or three components. For the rest of the algorithms the largest drop in cross-validated prediction error is up to two components with small improvements up to three or four components. Aiming for simple and robust models, we choose one component for the three mentioned algorithms and two components for the rest when validating.

Looking at the RMSEP values for the validation data we see the same grouping of algorithms as for the cross-validated calibration data. The number of components chosen for the individual algorithms seems to also be close to optimal for the validation data. All algorithms have a rise in the RMSEP curves one or two components after reaching the minimum RMSEP, though they never reach a higher level than the uncorrected data. If one aims at constructing as simple and easily interpretable a model as possible, it is evident that the choice of baseline correction algorithm is important. In this case choosing the ALS, polynomial fitting, or rolling ball algorithms for baseline correction can reduce the model complexity down to a single PLSR component from as much as five with the uncorrected data. Choosing among these three algorithms is difficult as they perform very similarly, and the small differences we observe may be due to data-dependent random noise affecting the methods slightly differently.

There can be several reasons for the large difference in the number of components needed to obtain satisfactory predictions for the uncorrected data and the corrected data. The main reason, however, is most likely that the PLSR models generated by the corrected data work more or less directly on the Raman part of the spectra, as the fluorescence has been removed by the baseline correction. In contrast, the PLSR models generated by the uncorrected data have to include the less informative fluorescence in the modeling and try to compensate for the lack of baseline correction, incorporating this in the first components. As we see from Fig. 7 this compensation is not sufficient as the model using uncorrected data both needs more components and gives much poorer predictions at its best. Also the compensation for lack of baseline correction might be too specific to the calibration data if there are differences in baselines between calibration and validation data.

Milk Mixtures. Following the proposed sequence of calculations from the Experimental section, we have used RMSECV to seek for the optimal parameter settings for the baseline algorithms at hand (see Table II). Because of the complexity caused by several parameters and milk types, we have applied GEMANOVA as described in the Experimental section to simplify the interpretation. As an example of the resulting optimization curves, Fig. 8 shows the loading vectors from GEMANOVA corresponding to the two parameters λ and p from ALS. We note that the drop in normalized RMSECV values (unit length of RMSECV vector) is relatively large around its minimum for the λ parameter. This is confirmed by the estimated P values for the differences between the minimum and the neighbor values of 0.0095 and 0.0205, respectively. This can indicate that the resolution of the optimization should have been higher around the minimum for this parameter.

Figure 9 shows RMSECV/P values for one to ten components using three separate PLSR models on spectra corrected by the different algorithms. The differences between the baseline-correction algorithms is so small for the calibration data that we have included magnified views of the most interesting areas with Fig. 9. The fits of the three responses seem to be influenced a bit differently by the baseline algorithms. Local medians is generally the most successful algorithm in the calibration, while peak detection and rolling ball are among the worst. The shape of the RMSECV curves differs from ewe to cow and goat. Also, the goat milk concentration seems to be a bit more difficult to predict than

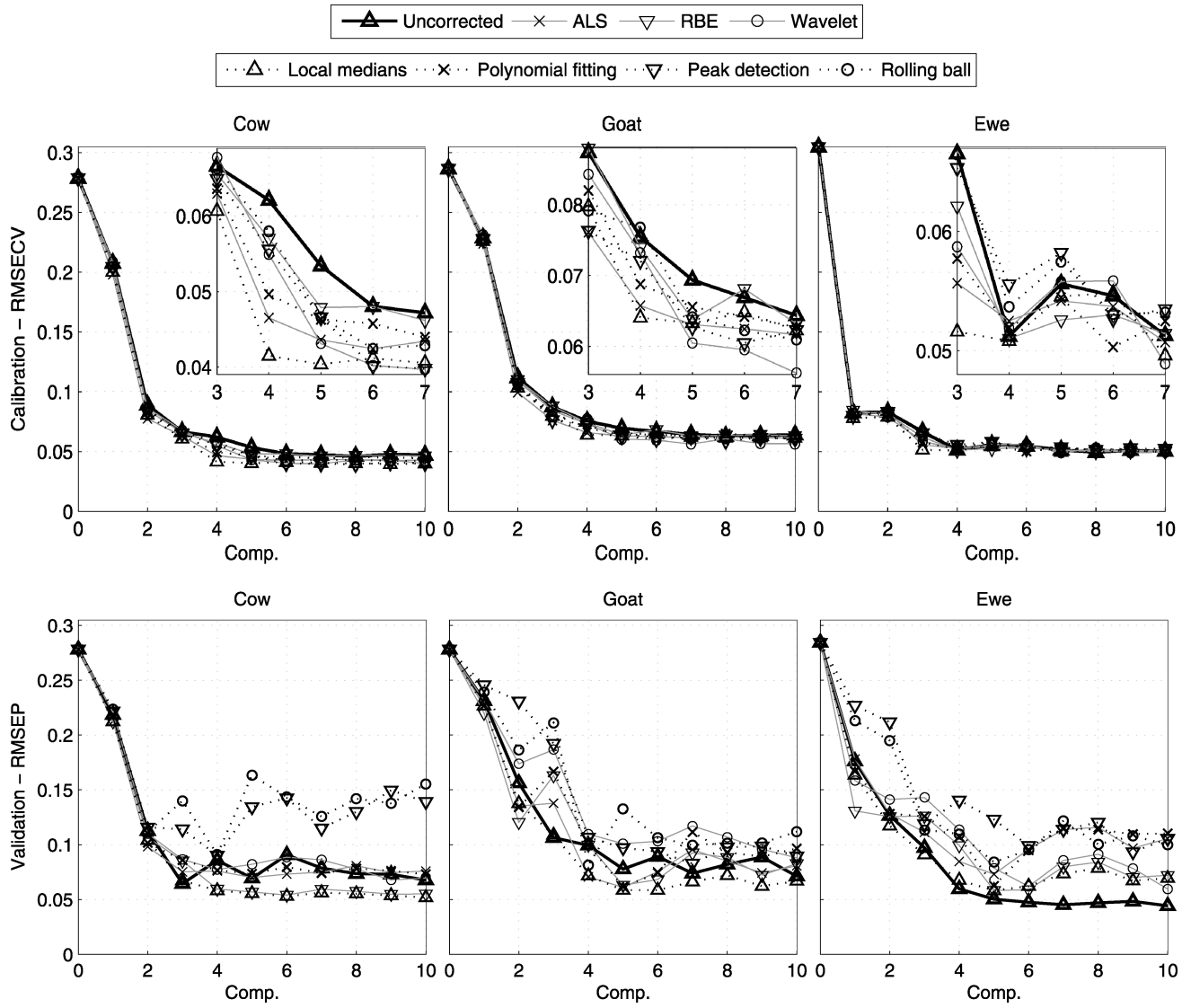

FIG. 9. Milk data: RMSECV and RMSEP curves from PLS regression based on original spectra and optimized baseline-corrected spectra. Based on the cross-validation, 5 components are chosen for prediction of the validation data.

cow. Picking a common number of components for all responses is not easy as there are varying local minima from four to six components for the individual algorithms. As a compromise we choose five components for all methods and responses. Looking at the validation data the first thing we notice is that avoiding baseline correction seems to be the best choice for the ewe response. There are several factors than can be direct or indirect causes of this odd behavior. As mentioned earlier, the validation data have been produced under slightly different conditions, and ewe milk contains around twice the amount of proteins as the other milk types. There might also be information in what is deemed a baseline elevation in the lower m/z range of the spectra relating to proteins that are more abundant in ewe milk than the other milk types, thus reducing the preciseness of the models using baseline-corrected spectra.

The peak detection and rolling ball methods are again consistently among the worst algorithms while local medians is consistently preferable. The algorithm most inconsistent from calibration to validation is wavelets. This can probably be explained to some extent by the fact that it provides an excellent smoothing of the spectra but uses a crude baseline estimation afterwards, possibly resulting in artifacts that do not harm the fitting of the calibration data but make the calibration and validation data even more different, thus harming the

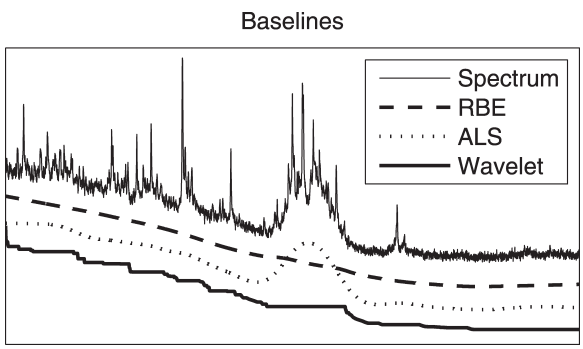

FIG. 10. Illustration of the variation from algorithm to algorithm with regard to optimal baseline correction. (Vertical positions of baselines are arbitrary.)

predictions. The choice of five components for prediction seems to be a good compromise for most algorithms except for the peak detection and rolling ball algorithms.

The milk data shows how important it is to validate the choices of baseline-correction algorithm and parameter settings both visually and through predictions to avoid over-fitting and false correlations produced by the baseline correction. Figure 10 illustrates how different optimal baselines for the different correction algorithms can be.

DISCUSSION

The main focus in this paper has been on optimal choice of baseline correction for spectra used in statistical analysis. From the analysis of the Raman spectra it is evident that there might be substantial rewards from making the correct choice of baseline corrections. Rather than always using your favorite correction algorithm with the traditional settings that give baseline corrections that look nice, optimizing the choice of algorithm and settings can give simpler and more stable models. This in turn means that easier interpretation and increased confidence can be achieved.

The analysis of the MALDI-TOF spectra reminds us that over-fitting and spurious correlations are always lurking in the background, ready to mess up modeling and predictions. Relying solely on automation or old habit is unwise at best, and validation is of utmost importance. Comparing the two data sets we also see that there is no uniformly best algorithm. While the ALS, polynomial fitting, and rolling ball algorithms do a great job on the Raman spectra, the RBE and especially local medians methods are the best choices for the MALDI-TOF spectra in this particular case. At the same time RBE is among the worst performers on the Raman spectra, while rolling ball is among the worst on the MALDI-TOF spectra. The only algorithm consistently failing on our data sets is the wavelets algorithm. However, this algorithm might have advantages on other data sets, e.g., where the strengths of the smoothing algorithm in the wavelets method is important.

There are several possible sources of error when a baseline-correction algorithm leads to poor predictions of validation data. As mentioned in the Results section on milk, the estimated baseline might include elevations to the spectra that contain information that should not be suppressed. This kind of error can be hard to detect and compensate for as most baseline-correction algorithms work on the spectra without having any background knowledge about where information and noise are located. The opposite problem arises if the baseline correction adds "false information" to spectra by correcting in a way that introduces artifacts beneficial for the calibration but not reproducible for the validation. This might be the case for the wavelets algorithm as the calibration data are well fitted while the validation data are badly predicted. A third source of error in the prediction of validation data can occur when the baseline correction of the calibration data has been systematically incomplete in such a way that the modeling has had to compensate for it. In such a case, the prediction model also contains some baseline correction specific to the calibration data. If the validation data are not exactly equal to the calibration data with regard to baselines, this can harm the predictions severely.

It is important to keep in mind that these are just single examples of data sets from two different types of spectra and should not be used as standards for what are the best-performing algorithms in all situations. There might be variations just as large between data sets from one type of measuring technique as there is between the data sets we have chosen, e.g., because of the types of samples used, different conditions when measuring, and settings on the measuring equipment. This paper should rather be used as a guide to looking for the baseline-correction algorithm best suited for a specific situation and as a reminder of some of the potential benefits and pitfalls of baseline correction.

1. H. Martens and T. Næs, *Multivariate Calibration* (John Wiley and Sons, Chichester, UK, 1989).
2. T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference and Prediction* (Springer, New York, 2001).
3. M. Stone, J. Roy. Stat. Soc., Series B: Methodological **36,** 111 (1974).
4. P. H. Eilers, Anal. Chem. **76,** 404 (2004).
5. A. F. Ruckstuhl, M. P. Jacobson, R. W. Field, and J. A. Dodd, J. Quant. Spectrosc. Radiat. Trans. **68,** 179 (2001).
6. K. R. Coombes, S. Tsavachidis, J. S. Morris, K. A. Baggerly, M.-C. Hung, and H. M. Kuerer, Proteomics **5,** 4107 (2005).
7. I. Daubechies, Commun. Pure Applied Math. **41,** 909 (1988).
8. M. Friedrichs, J. Biomol. NMR **5,** 147 (1995).
9. F. Gan, G. Ruan, and J. Mo, Chemom. Intell. Lab. Syst. **82,** 59 (2006).
10. C. Lieber and A. Mahadevan-Jansen, Appl. Spectrosc. **57,** 1363 (2003).
11. K. Coombes, H. Fritsche, C. Clarke, J. Chen, K. Baggerly, J. Morris, L. Xiao, M. Hung, and H. Kuerer, Clin. Chem. **49,** 1615 (2003).
12. M. Kneen and H. Annegarn, Nucl. Instrum. Methods Phys. Res. **82,** 59 (1996).
13. N. K. Afseth, V. H. Segtnan, and J. P. Wold, Appl. Spectrosc. **60,** 1358 (2006).
14. K. H. Liland, B.-H. Mevik, E.-O. Rukke, T. Almøy, and T. Isaksson, Chemom. Intell. Lab. Syst. **99,** 39 (2009).
15. B.-H. Mevik, V. Segtnan, and T. Næs, J. Chemom. **18,** 498 (2004).
16. R. Bro and M. Jakobsen, J. Chemom. **16,** 294 (2002).
17. K. H. Liland and E. M. Færgestad, Chemom. Intell. Lab. Syst. **96,** 172 (2009).