

UNIVERSITE CATHOLIQUE DE LOUVAIN
FACULTE DES SCIENCES
ECOLE DE STATISTIQUE, BIOSTATISTIQUE
ET SCIENCES ACTUARIELLES



PRE-PROCESSING OF NMR SPECTRA :
REVIEW AND EVALUATION OF BASELINE CORRECTION,
NORMALIZATION, SCALING AND TRANSFORMATION METHODS

Supervisor : Bernadette GOVAERTS

Readers : Pascal DE TULLIO

Rainer VON SACHS

Manon MARTIN

Mémoire présenté en vue de l'obtention

du Master en statistiques, orientation

biostatistique par :

Rebecca MARION

Septembre 2016

Acknowledgments

First of all, I would like to express my gratitude to Prof. Govaerts for her guidance, feedback and support during this past year and Manon Martin, who helped give feedback at various stages during the preparation of this thesis. I would also like to thank Bibours for his support, wise words and patience, which made the process of writing this thesis more fruitful and rewarding. Lastly, I would like to thank my parents for their encouragement and my brother, whose help and advice in the last two years have been indispensable.

Contents

1	Introduction	4
1.1	Metabolomics	4
1.2	Nuclear Magnetic Resonance (NMR) spectroscopy	4
1.3	Pre-processing	5
1.3.1	Normalization, scaling and transformation	6
1.3.2	Baseline correction	8
1.4	Thesis outline	9
2	Normalization, scaling and transformation (NST) methods in the literature	11
2.1	Introduction	11
2.1.1	Main purposes	11
2.1.2	Biology-related challenges	14
2.1.3	Chapter overview	17
2.2	Normalization methods	18
2.2.1	Constant Sum Normalization (CSN)	18
2.2.2	Creatinine Normalization	20
2.2.3	Linear Baseline Normalization (Normalization by Median Intensity)	21
2.2.4	Probabilistic Quotient Normalization (PQN)	23
2.2.5	Histogram Matching (HM)	25
2.2.6	Variance Stabilizing Normalization (VSN)	29
2.2.7	Other methods	34
2.3	Scaling methods	34
2.3.1	Autoscaling and Pareto Scaling	34
2.4	Transformation methods	36
2.4.1	Quantile Normalization (QN)	36
2.5	Pre-normalization step (Hochrein et al., 2015)	39
2.5.1	Pre-selection of unregulated features	39
2.6	Conclusion	44
3	Implementation and evaluation of certain NST methods	46
3.1	Data	46
3.1.1	Data sets	46
3.1.2	Data pre-processing	47
3.2	Methods applied	47
3.3	Performance measures	48
3.3.1	Variance structure	49
3.3.2	Clustering	50

3.3.3	Classification	51
3.4	Results	53
3.4.1	Variance structure	53
3.4.2	Clustering	56
3.4.3	Classification	56
3.5	Conclusion	58
4	Baseline correction methods in the literature	59
4.1	Iterative penalized least squares methods	61
4.1.1	Asymmetric Least Squares (AsLS)	61
4.1.2	Adaptive Iteratively Reweighted Penalized Least Squares (airPLS)	65
4.1.3	Asymmetrically Reweighted Penalized Least Squares (arPLS)	68
4.1.4	Penalized Parametric Smoothing Model (PSM)	73
4.2	Noise and signal classification methods	75
4.2.1	Automatic Baseline Correction algorithm (ABC)	77
4.2.2	Fully Automatic Baseline Correction algorithm (FABC)	78
4.2.3	Chenomx algorithm	80
4.2.4	Golotvin and Williams (2000)'s algorithm	81
4.2.5	BaselineCorrector	83
4.2.6	Automatic Two-sided Exponential Baseline correction algorithm (ATEB)	85
4.3	Conclusion	86
5	Implementation and evaluation of certain baseline correction methods	88
5.1	Simulation of spectra	88
5.1.1	Method of simulation	88
5.1.2	Parameter choices	95
5.1.3	Simulated spectra	99
5.2	Application of baseline correction methods	101
5.2.1	Choice of default method parameters	101
5.2.2	Performance Measures	102
5.3	Results	103
5.3.1	Root mean square error (RMSE)	104
5.3.2	Mean error	109
5.3.3	Other measures	111
5.4	Conclusion	111
6	Conclusion	112

6.1	Normalization, scaling and/or transformation (NST) methods	112
6.1.1	Conclusions	112
6.1.2	Further directions	114
6.2	Baseline correction methods	114
6.2.1	Conclusions	114
6.2.2	Further directions	116
6.3	Concluding remarks	117
7	Annex	118
7.1	NST evaluation methods in the literature	118
7.1.1	Data used	118
7.1.2	Evaluation criteria	121
7.2	Implementation and evaluation of certain NST methods	136
7.2.1	Application of the pre-normalization step to Linear Baseline Normalization, PQN and VSN	136
7.2.2	Performance measures	137
7.2.3	PCA plots	141
7.3	Implementation and evaluation of certain baseline correction methods	160
7.3.1	Additional boxplots	160

1 Introduction

1.1 Metabolomics

Metabolomics is a field of study that analyzes the fluctuation of metabolites – products or byproducts of metabolism, such as carbohydrates or amino acids. Endogenous metabolites, which are metabolites produced by the host (not introduced by some external substance), are of particular interest because they could communicate interesting information about biological processes related to disease. Indeed, metabolomic profiles “provide information at a level that is closer to the phenotype” (Euceda et al., 2015), making it possible to study the relationship between metabolite excretion and certain pathologies, leading to an improved diagnosis of patients.

1.2 Nuclear Magnetic Resonance (NMR) spectroscopy

Metabolomics data is typically collected from bio-fluids such as urine or blood. One particular method of determining the metabolic composition of bio-fluid samples is through Nuclear Magnetic Resonance (NMR) spectroscopy. For NMR spectroscopy, samples are put into a magnetic field where different radio frequency (RF) pulses are applied. Nuclei excited by the RF pulses absorb energy and then re-emit it so that they can return to a state of equilibrium. Differences in the resonant frequencies of the re-emitted energy can give information about the molecular structure of the sample in question (Euceda et al., 2015).

At the time of NMR data acquisition, data is output in the form of a free induction decay (FID) signal. During pre-processing, this signal must be transformed using the Fourier transform, resulting in a frequency spectrum (see the examples in Figure 1¹). Each feature in this spectrum represents a resonant frequency relative to a standard in the magnetic field used during data acquisition. Different metabolites resonate at different frequencies, so features in the spectrum can be associated with metabolites. The total intensity of the features associated with a given metabolite is directly proportional to the concentration of the metabolite in the sample.

¹Image credited to John C. Edwards, Ph.D, accessed on August 16, 2016 at <http://www.process-nmr.com/nmr1.htm>

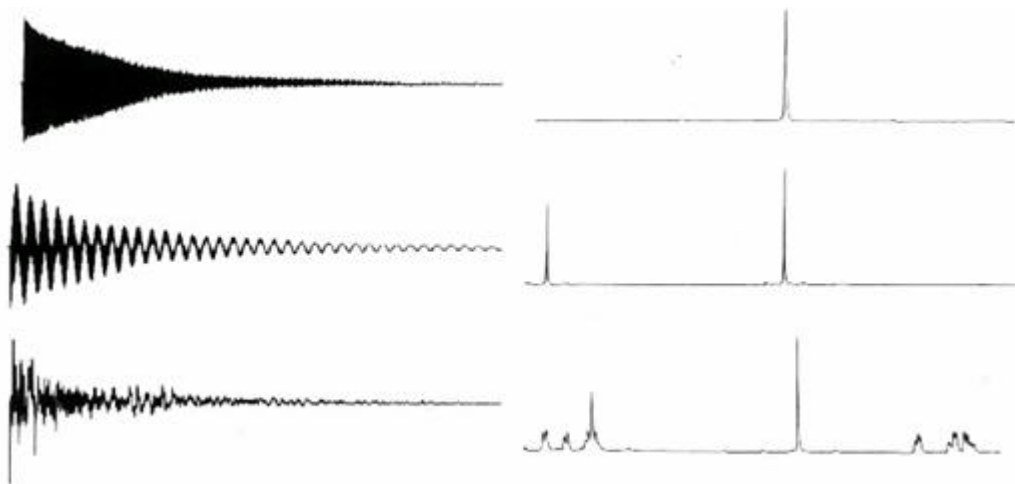


Figure 1: Three examples of FID signals transformed into frequency spectra

NMR spectroscopy is relatively simple and inexpensive, with a sample preparation procedure that requires very little time and investment. In contrast to other data acquisition methods, such as Mass Spectrometry (MS), the samples used to produce NMR spectra can be re-used because sample preparation is non-destructive. While NMR spectroscopy has low sensitivity, meaning that it is impossible to detect metabolites of low abundance, it has the advantage of being reproducible and highly quantitative (Euceda et al., 2015). Indeed, because concentration is directly proportional to area under the curve, there is no need to estimate a calibration model in order to interpret NMR spectra.

1.3 Pre-processing

The pre-processing of the raw NMR data before analysis is an essential step that can dramatically impact the quality of the subsequent multivariate analyses. A variety of different pre-processing steps are generally applied before analysis, including baseline correction, peak alignment, normalization, scaling and transformation, among others. The metabolomics team at the Université catholique de Louvain has developed an R package called **SOAP** that executes a total of 14 different pre-processing steps to raw NMR data. The different steps are displayed in Figure 2 and are described in Rousseau (2011).

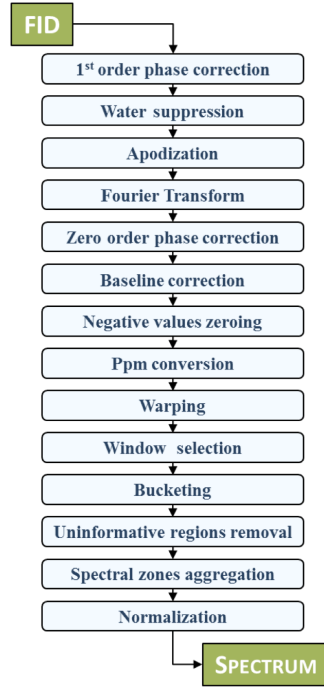


Figure 2: SOAP workflow

For this thesis, two particular pre-processing steps will be studied:

1. normalization, scaling and transformation (step 14) and
2. baseline correction (step 6)

1.3.1 Normalization, scaling and transformation

Spectra can vary for reasons unrelated to metabolic processes or particular phenotypes. While a single urine sample measured on different days or diluted to different levels should produce identical spectra, in practice, these different conditions can yield different spectral profiles (see Figures 3 and 4 respectively). This increased variation is not biologically relevant and should be removed before proceeding with multivariate analyses.

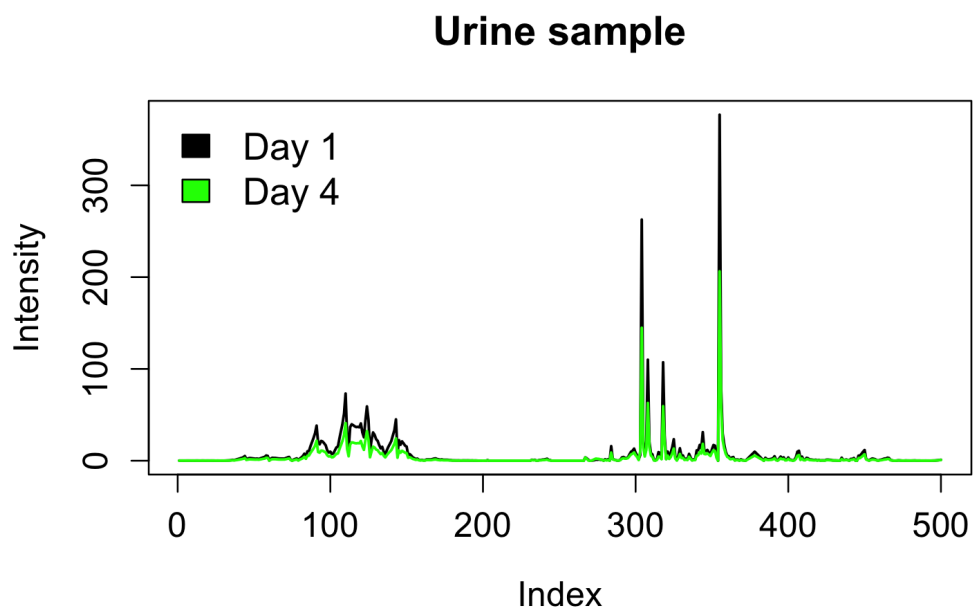


Figure 3: Spectra of a single urine sample on different days

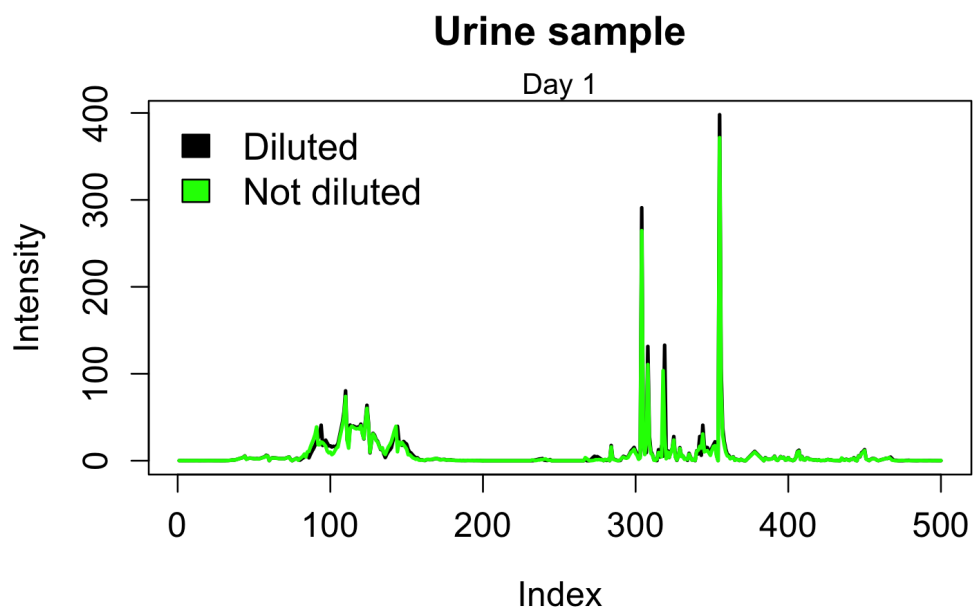


Figure 4: Spectra of a single urine sample with different dilution levels

The goal of normalization, scaling and transformation (NST) is to improve the quality of data by removing artificial variability between spectra. If the variability between spectra is simply a problem of scale differences, a simple solution would be to divide each spectrum by the appropriate scaling factor. Unfortunately, other factors can contribute to the artificial variability between spectra, such as the intensity-dependent bias and variance associated with NMR spectroscopy, making the task more difficult.

Increasing the similarity between spectra is not necessarily sufficient for improving data quality. NST methods must also avoid introducing new artificial artifacts to the data. This means that they should not decrease the contrast between noise and signal and should, if applicable, improve classification performance.

1.3.2 Baseline correction

The baselines of NMR spectra are often distorted to some extent. This distortion artificially alters the intensities of spectral peaks, resulting in biased estimates of the concentration of metabolites. This is problematic because erroneous peak intensities and metabolite concentration estimations can greatly decrease the quality of the data for multivariate analysis. For example, the accuracy of subject classification (healthy vs sick subjects) could be decreased or false bio-markers could be discovered.

The goal of baseline correction is to estimate the true baseline and then subtract it from the spectrum so that the corrected spectrum has a flat baseline at zero intensity. This task is especially difficult in signal regions because overlapping peaks make the true baseline “disappear.” In the example below (Figure 5), it would be difficult to visually identify the true baseline in the region between features 500 and 700. This task is even more difficult for an algorithm, which must construct a baseline based only on the data available.

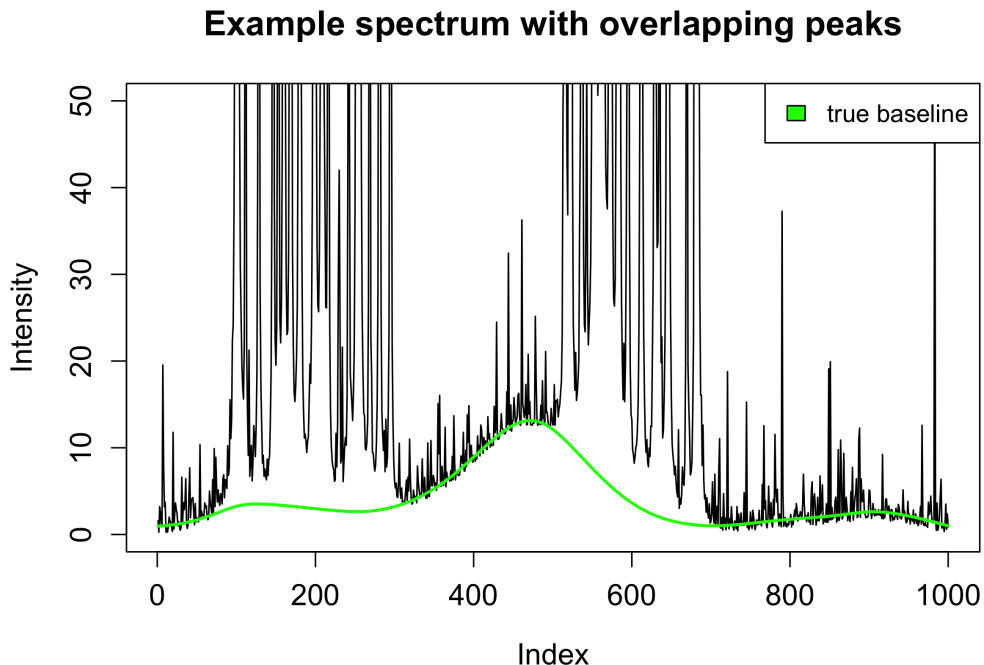


Figure 5: Simulated spectrum with baseline distortion

The fact that, for real spectra, the true baseline is never truly known is also problematic because it makes it difficult to evaluate the performance of baseline correction methods and, if relevant, determine appropriate algorithm parameters. For this reason, most baseline correction methods evaluated in the literature are applied to simulated data sets.

1.4 Thesis outline

This thesis seeks to summarize and evaluate baseline correction and NST methods proposed in the literature. The following chapters are organized as follows:

Normalization, transformation and scaling methods from the literature will be presented and discussed in Chapter 2. In Chapter 3, a selection of these methods will be tested on real spectral data sets (urine and blood serum).

In Chapter 4, various baseline correction methods from the literature will

be presented and discussed. The most promising of these methods will be applied to simulated spectral data sets and evaluated in Chapter 5.

In the concluding chapter (Chapter 6), the results presented in Chapters 3 and 5 will be compared to the performance of methods as reported in the literature (Chapters 4 and 2). Directions for further research will also be suggested.

2 Normalization, scaling and transformation (NST) methods in the literature

2.1 Introduction

Let x_{ij} be the spectral intensity of feature j in spectrum i , with $i : 1, \dots, n$ and $j : 1, \dots, p$. Here we define **normalization** methods as methods that aim to reduce inter-sample differences by transforming each spectrum i entirely. **Scaling** methods adjust the variance of each feature j in all spectra. **Transformation** methods involve a transformation based on both the spectrum i and the feature j . In this section, normalization, scaling and transformation methods will be referred to as NST methods.

2.1.1 Main purposes

Normalization, scaling and transformation (NST) is the last pre-processing step in SOAP and is preceded by spectral zone aggregation (see Figure 2). NST methods serve multiple purposes, most typically:

- Reducing overall differences between spectra obtained from similar samples (technical reproducibility, comparability of spectra)
- Reducing intensity-dependent bias, variance and differences between spectra
- Preserving and/or increasing contrast between noise and signal
- Removing dilution effects (urine spectra)
- Improving classification performance

These tasks aim to reduce or remove irrelevant, technical variation between spectra, resulting in spectra that whose differences can be attributed to metabolic variation only.

Reducing overall differences between similar samples

Technical reproducibility

Due to technical imperfections, two separate NMR measures of the same sample can yield slightly different results (see Figure 6). Thus, it is reasonable to expect that some of the differences between non-identical samples could be attributed to this technical variation. One goal of NST is thus to

improve technical reproducibility by eliminating or reducing the variation in the data resulting from technical error. Perfect technical reproducibility is achieved when spectra from different NMR measures of a single sample are identical.

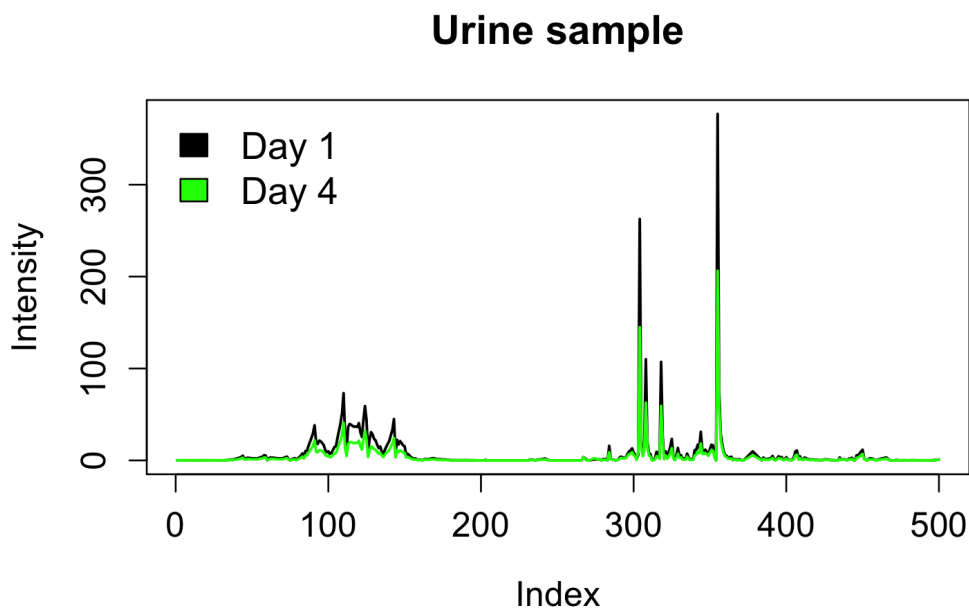


Figure 6: A single urine sample measured on two different days

Comparability of spectra

Peak intensities measured in each NMR spectra are not necessarily expressed in the same scale, so bringing different spectra onto the same scale is also important for increasing the comparability of spectra. In Figure 7, a single urine sample is represented 5 times, each color representing the original spectrum times some scaling factor. Ideally, an NST method should make all of these spectra the same, as seen in the second panel of the figure.

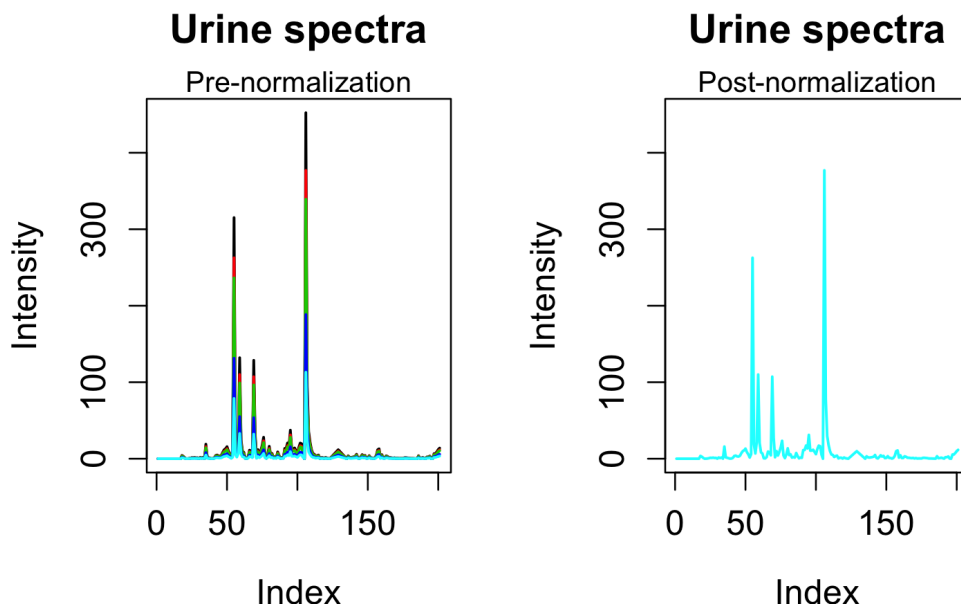


Figure 7: A single urine spectrum multiplied by different scaling factors

Reducing intensity-dependent bias, variance and differences between spectra

The bias and variance of NMR spectral intensities are known to be intensity dependent (Kohl et al., 2012; Parsons et al., 2007). The magnitude of differences between corresponding features in different spectra can also depend on intensity (Kohl et al., 2012). After the application of NST methods, the bias, variance and differences between spectra should be independent of intensity.

Preserving and/or increasing contrast between noise and signal

Ideally, normalizing, scaling and/or transforming spectra should increase or at the very least preserve the original signal to noise ratio (SNR) of the spectra. This is particularly important for the identification of biomarkers, as interesting low-intensity signal features can sometimes be overshadowed by higher-intensity noise features, making it difficult to distinguish noise from signal in multivariate analyses.

Removing dilution effects (urine spectra)

In urine samples, the concentration of metabolites depends on the amount of

water in the urine: the concentration of metabolites decreases with increases in the total volume of water in the urine sample. The dilution of the sample being of no biological interest, it is important to eliminate the effect of dilution before proceeding to data analysis. The quality of NST depends thus on its ability to remove dilution effects.

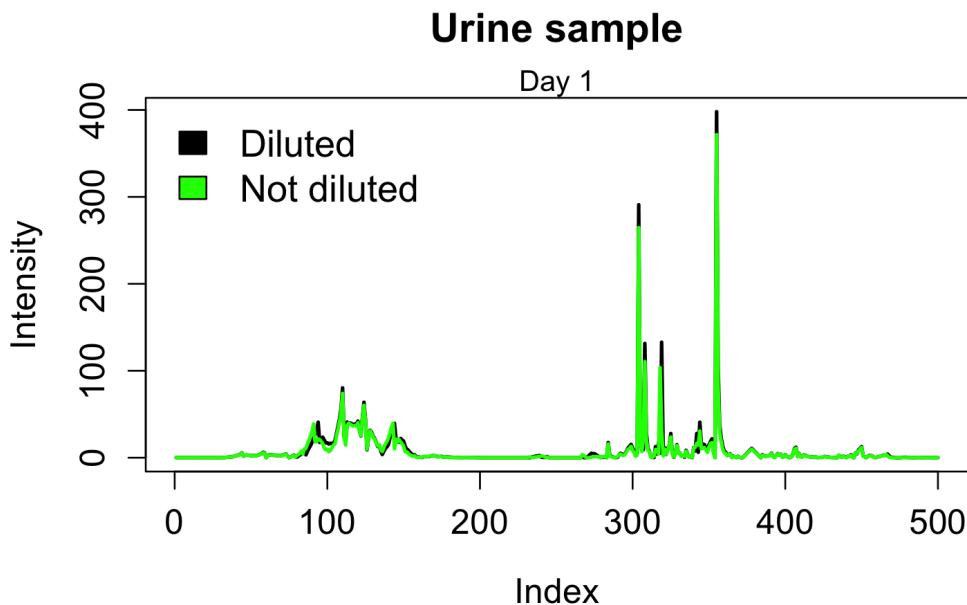


Figure 8: A single urine sample with different dilution levels

Classification performance

If the ultimate goal of data pre-processing is the accurate classification of samples, it seems natural to favor an NST method that improves classification performance.

2.1.2 Biology-related challenges

The tasks outlined in 2.1.1 serve to reduce or remove the artificial spectral variance induced by the technical process of NMR acquisition. Achieving these objectives can be complicated by the particular character of the metabolite regulation associated with a given sample.

Metabolite regulation

In this thesis, the following definitions are used for the terms “regulated” and “unregulated” metabolites:

- A given metabolite is said to be “regulated” when, in reaction to some external factor, the concentration of that metabolite increases or decreases relative to some reference concentration.
- “Unregulated” metabolites are those metabolites whose concentration does not change in reaction to some external factor.

A metabolic process is “self-averaging” if an increase in the concentration of certain metabolites is counterbalanced by a decrease in concentration of other metabolites (and vice versa) (Hochrein et al., 2015). As a result, the average total spectral intensity remains similar across different subject groups (Hochrein et al., 2015). A simplistic example of this is depicted in Figure 9, where a decrease in intensity of the red peak is met with an increase in intensity of the other peaks in black (i.e. the metabolite associated with the red peak increases in concentration and thus the concentration of the other metabolites decreases).

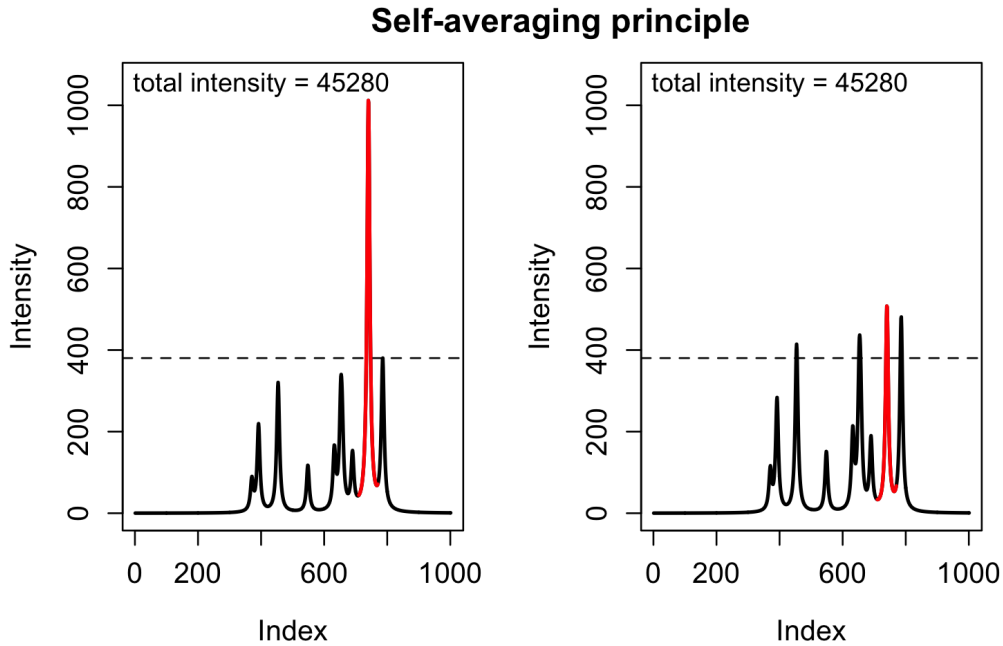


Figure 9: Simulation of the self-averaging principle (pre-normalization)

A metabolic process is “imbalanced” (the opposite of “self-averaging”) if an

increase in the concentration of certain metabolites is not counterbalanced by a decrease in concentration of other metabolites (and vice versa).

Challenges posed by metabolite regulation

The “self-averaging” principle is rarely applicable to metabolomics data (Hochrein et al., 2015; Sysi-Aho et al., 2007), meaning that metabolite regulation is often “imbalanced” (i.e. the increase in concentration of certain regulated metabolites is not counterbalanced by a decrease in concentration of the other regulated metabolites). Indeed, samples from different exposure groups can have very similar spectral intensities for features associated with unregulated metabolites while having dramatically different spectral intensities for features associated with regulated metabolites (see Sysi-Aho et al. (2007) for an example). This results in average total spectral intensities that are dramatically different.

NST methods that rely on the self-averaging hypothesis can distort spectral intensities in cases where the hypothesis is not validated. Indeed, applying certain NST methods to spectra with highly different metabolite regulation can result in the distortion of unregulated metabolites or the reduction of interesting biological variation in the spectrum. Thus, while it is generally desirable to increase the comparability between spectra, spectra with vastly different metabolite regulation should not necessarily be very similar after normalization, scaling and/or transformation.

Suppose that the black peaks in Figure 10 represent metabolites that are “unregulated” (their concentration does not change in reaction to an external factor of interest) and the red peak represents a metabolite that is “regulated” (its concentration changes in reaction to the factor). Because the black peaks are “unregulated”, it is normal to expect that they should be almost identical from patient to patient (see row 1 of Figure 10). The red peak, however, might vary substantially, depending on a patient’s exposure to the factor of interest.

After normalization, the black peaks should ideally remain almost identical. For the normalization method applied here, however, this is clearly not the case: the black peaks in spectrum 2 appear to be higher than in spectrum 1. While in reality these black peaks represent “unregulated” metabolites, the distortion created by normalization would suggest that they are perhaps “regulated”, with intensities that depend on the patient’s exposure to the factor of interest.

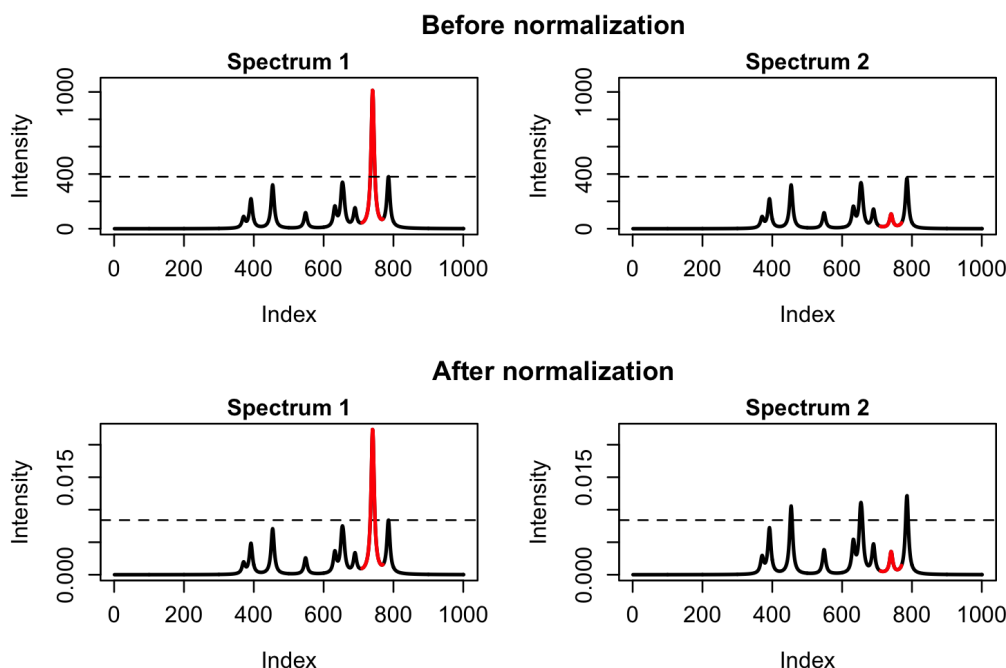


Figure 10: Simulated spectra distorted after normalization

The greater the variability of regulated metabolites in the sample, the more difficult it becomes to optimally normalize, scale and/or transform its spectrum (Hochrein et al., 2015), especially if metabolic regulation is imbalanced (i.e. not “self-averaging”).

Note that, because metabolite regulation poses challenges for many NST methods, the terms “regulated” and “unregulated”, as well as the concepts of “self-averaging” and “imbalanced regulation”, will be frequently referred to in the following sections.

2.1.3 Chapter overview

In the following sections, the most promising NST methods for NMR data will be discussed, each time including the theoretical motivation and principals, the implementation procedure and a discussion of the method in question (based on the quality of its theoretical foundation as well as its performance as reported in the literature). Note that while Constant Sum Normalization (CSN), Normalization to an Internal Standard (NIS) and Pareto and Autoscaling are generally considered non optimal in the literature, they will be

described below because they are often used in current practice.

2.2 Normalization methods

2.2.1 Constant Sum Normalization (CSN)

Constant Sum Normalization (Li et al., 2015), also known as Integral Normalization (Dieterle et al., 2006) or Normalization to Total Intensity (Wang et al., 2013), is one of the most common normalization procedures proposed in the literature and used in practice. This method is used alone or in combination with other normalization methods (such as Probabilistic Quotient Normalization (Section 2.2.4) or Variance Stabilizing Normalization (Section 2.2.6)).

Theoretical motivation and principals

The primary goal of this normalization method is to make the total intensity of each spectrum (i.e. the sum of all intensities in each spectrum) the same in order to reduce the overall differences between spectra. This method assumes that the average total intensities across spectra are relatively similar (Hochrein et al., 2015). In other words, metabolite regulation should be self-averaging (see Section 2.1.2).

Implementation

Let t_i be the total intensity of spectrum i , which is calculated as the sum of all features j in the spectrum.

$$t_i = \sum_{j=1}^p x_{ij}$$

Each feature j in spectrum i is then divided by the total intensity of the spectrum such that the normalized intensity x_{ij}^* becomes

$$x_{ij}^* = \frac{x_{ij}}{t_i}$$

Discussion

When the total intensity of each spectrum is about the same, CSN has been shown to be a relatively effective method (Wang et al., 2013). However, several authors have argued that the total intensities of spectra are usually

not very similar in practice, meaning that CSN can lead to undesired spectral distortion (Li et al., 2015; Dieterle et al., 2006; Hochrein et al., 2015). Hochrein et al. (2015) cite the example of kidney failure, where the total intensity of spectra for kidney failure patients is typically much greater than for healthy patients.

It suffices to have one extreme peak in a spectrum for CSN to distort that spectrum. A toy example is depicted in Figure 11, where the total intensity of the first spectrum is much greater than the second spectrum due to differences in the concentration of a certain regulated metabolite (in red). Suppose that the peaks in black are associated with unregulated metabolites that are present in almost identical concentrations for all patients. We would thus hope that these peaks remain similar after normalization. After implementing CSN, however, the black peaks are no longer the same intensity in the two spectra.

The danger of this is that unregulated metabolites (peaks in black, Figure 11) could be identified as regulated metabolites because the variance of their spectral intensities would be artificially increased by CSN. Regulated metabolites are associated with high-variance spectral intensities whereas unregulated metabolites are associated with low-variance spectral intensities (Hochrein et al., 2015).

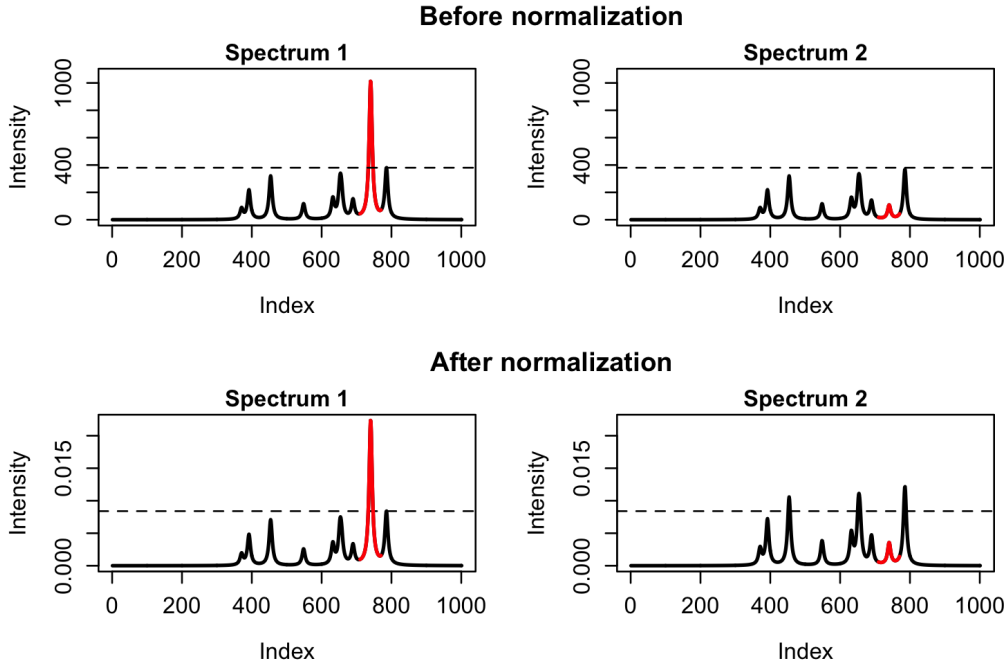


Figure 11: Simulated spectra distorted after CSN

2.2.2 Creatinine Normalization

Normalization to an Internal Standard (NIS) (Wang et al., 2013) includes any method that relies on an internal standard to normalize. One special case of NIS is Creatinine Normalization, which is commonly used for NMR analysis of urine samples.

Theoretical motivation and principals

The concentration of metabolites in urine samples depends on the amount of water in the urine because adding water decreases the concentration of the metabolites. The quantity of creatinine excreted in urine is assumed to be constant from one sample to another, so the spectral intensity of the creatinine peak should be about the same in all spectra. Any differences in the spectral intensity of the creatinine peak between samples suggest a possible dilution problem. Dilution affects the concentration of all metabolites, so it can be removed by dividing all spectral intensities by a single dilution factor. For Creatinine Normalization, all features are divided by the total intensity of the creatinine peak. After applying Creatinine Normalization, the creatinine peaks in all spectra have the same intensity.

Implementation

The area under the creatinine peak is calculated by summing the intensity of all features j associated with the peak.

$$t_i^{creat} = \sum_{j \in creat} x_{ij}$$

Each feature in a given spectrum i is normalized by dividing it by the total intensity of the creatinine peak.

$$x_{ij}^* = \frac{x_{ij}}{t_i^{creat}}$$

Discussion

Creatinine Normalization can be an effective method for ensuring that the contrast between noise and signal is preserved because it does not decrease this contrast (Kohl et al., 2012). However, Kohl et al. (2012) have shown that Creatinine Normalization is one of the worst methods for increasing the comparability of spectra.

Dieterle et al. (2006) have also suggested that Creatinine Normalization poses several problems. If certain metabolites in the sample overlap with the creatinine peak in the spectrum, the calculation of the area under the creatinine peak will be biased. The location (i.e. chemical shift) of the creatinine peak also depends on the pH of the sample. Finally, the assumption of constant creatinine excretion is not always validated in practice, as the excretion of creatinine depends on the metabolomic processes taking place (Dieterle et al., 2006) and the health and age of the patient in question (Li et al., 2015).

2.2.3 Linear Baseline Normalization (Normalization by Median Intensity)

Linear Baseline Normalization (Bolstad et al., 2003), which comes from the field of gene expression, has also been applied to NMR data by Kohl et al. (2012) and Hochrein et al. (2015).

Theoretical motivation and principals

It is assumed that the observed intensity of each feature in a given spectrum is linearly related to a reference spectrum by a single constant factor. Dividing each feature in a spectrum by this constant factor brings each spectrum

onto the same scale as the reference spectrum and thus the other normalized spectra.

Implementation

Let the median intensity of spectrum i be

$$x_{i,med} = median_{1 \leq j \leq p}(x_{ij}).$$

Let c be a constant by which each spectra is multiplied. The normalized spectrum i is thus given by

$$\mathbf{x}_i^* = c \frac{\mathbf{x}_i}{x_{i,med}},$$

Because the same value c is used for each spectrum, this constant does not actually affect the normalization between spectra and can be ignored if desired.

Kohl et al. (2012) and Hochrein et al. (2015) use the median intensity of a certain reference spectrum $\mathbf{z}$ as the constant c . Let the reference spectrum $\mathbf{z}$ be defined as a vector of medians across all spectra at each feature j .

$$\mathbf{z} = (median_{1 \leq i \leq n}(x_{i1}), \dots, median_{1 \leq i \leq n}(x_{ip}))$$

$$\mathbf{z} = (z_1, \dots, z_p)$$

The constant c is given by

$$c = median_{1 \leq j \leq p}(z_j).$$

Thus, each spectrum $\mathbf{x}_i$ is normalized by dividing it by $\frac{x_{i,med}}{c}$, the quotient of its median intensity and the median intensity of the reference spectrum.

Discussion

Kohl et al. (2012) argue that the assumption that spectra are linearly related may be an oversimplification. Nonetheless, Linear Baseline Normalization has been shown to perform well in practice. Kohl et al. (2012) have shown that this method improves classification performance and performs consistently for a wide variety of metabolites. It is also one of the best methods for preserving the contrast between noise and signal (Kohl et al., 2012).

Hochrein et al. (2015) have demonstrated that Linear Baseline Normalization is able to accurately estimate spectral intensities in different spectra when a single varying regulated metabolite (for example, glucose) is present. This was demonstrated in an experiment in which the concentrations of eight

metabolites in each spectrum were increased and decreased in relatively equal proportions and an additional metabolite (glucose) was included in increasing concentrations (violating the self-averaging principle). Estimation bias for the eight metabolites was evaluated in the presence of different glucose concentrations (see the Annex, Section 7.1.2, for more details).

When many regulated metabolites create an imbalance in metabolite concentrations, however, Linear Baseline Normalization fails to accurately estimate spectral intensities in spectra. Nonetheless, for this type of situation, the performance of the method can be improved by implementing the pre-normalization step proposed by Hochrein et al. (2015) (see Section 2.5).

2.2.4 Probabilistic Quotient Normalization (PQN)

Probabilistic Quotient Normalization (Dieterle et al., 2006) is generally preceded by Constant Sum Normalization.

Theoretical motivation and principals

PQN is based on the assumption that scaling phenomena affect all metabolites in a spectrum whereas biological fluctuations only influence certain metabolites. If it is possible to determine a global scaling factor for a given spectrum, that spectrum can be normalized without distorting the intensities associated with biological fluctuations.

Similar to Linear Baseline Normalization (see Section 2.2.3), this method assumes that the relationship between spectra is linear, meaning that spectra differ only by some constant factor. Instead of dividing each spectrum i by its median intensity, $x_{i,med}$, as is the case with Linear Baseline Normalization, each feature j in spectrum i is divided by the most probable factor of difference (the “most probable quotient”) between that spectrum and some reference spectrum. The “most probable quotient” for a given spectrum is the median value of the vector of quotients calculated by dividing the spectrum in question by the reference spectrum.

Implementation

In order to calculate the most probable quotient for a given spectrum i , a reference spectrum $\mathbf{z}$ can first be calculated in the same way as for Linear Baseline Normalization (see Section 2.2.3):

$$\mathbf{z} = (\text{median}_{1 \leq i \leq n}(x_{i1}), \dots, \text{median}_{1 \leq i \leq n}(x_{ip}))$$

While other choices of reference spectrum are possible, the authors argue that the actual choice of the reference spectrum has little impact on the results (Dieterle et al., 2006).

The quotient spectrum, $\mathbf{Q}_i$, is calculated by dividing each feature in spectrum i by the corresponding feature in $\mathbf{z}$.

$$Q_{ij} = \frac{x_{ij}}{z_j}$$

$$\mathbf{Q}_i = (Q_{i1}, \dots, Q_{ip})$$

The median of the quotient spectrum is identified as the most probable quotient, Q_i^{prob} :

$$Q_i^{prob} = \text{median}_{1 \leq j \leq p}(Q_{ij}).$$

Finally, all features in spectrum i are divided by Q_i^{prob} , yielding the normalized spectrum $\mathbf{x}_i^*$.

$$\mathbf{x}_i^* = \frac{\mathbf{x}_i}{Q_i^{prob}} \quad (1)$$

Discussion

Because the most probable quotient is defined as the median quotient, extreme quotients (resulting from single metabolites taking extreme values relative to the reference spectrum) are essentially ignored, meaning that they do not cause PQN to distort the spectrum. Dieterle et al. (2006) have demonstrated that PQN is robust to individual intensity increases if 50% or less of the features have been systematically increased. Dieterle et al. (2006) suggest that in real spectra, only 0 to 20% of features demonstrate systematic variation of this type.

Kohl et al. (2012) have demonstrated that PQN is one of the best methods for preserving the contrast between signal and noise while also being one of the best for reducing overall inter-sample differences (tied with Linear Baseline Normalization, 2.2.3). Wang et al. (2013) have also shown that PQN yields relatively good reproducibility, especially when the total intensities of the spectra are highly variable. PQN tends to increase the reproducibility of small peaks even when these peaks are in the presence of very large neighboring peaks.

When many regulated metabolites are fluctuating across spectra, and the self-averaging principle is violated, PQN fails (Li et al., 2015; Hochrein et al., 2015). However, if PQN is preceded by the pre-normalization step proposed by Hochrein et al. (2015) (see Section 2.5), it yields accurate intensity estimates even in the presence of severely imbalanced regulation.

Once again, if the assumption of linearity between the reference spectrum and each of the spectra in a given data set does not hold, this method performs less optimally.

2.2.5 Histogram Matching (HM)

Theoretical motivation and principals

Many normalization methods are sensitive to high intensity peaks, which can sometimes lead to distortion of the lower intensity peaks. For example, in the case of CSN, the factor used to normalize each spectrum depends on the total intensity of the spectrum in question. While low intensity peaks have little influence on the total intensity of the spectrum, a single high intensity peak can drastically change the total intensity, resulting in dramatically different results (see Figure 11).

In Histogram Matching (Torgrip et al., 2008), high intensity peaks are not given disproportionate importance. Each spectrum in the data set, including some reference spectrum, is transformed into a histogram with B intensity intervals of the same length. The height of the histogram at each interval represents the number of features j with an intensity in the interval in question.

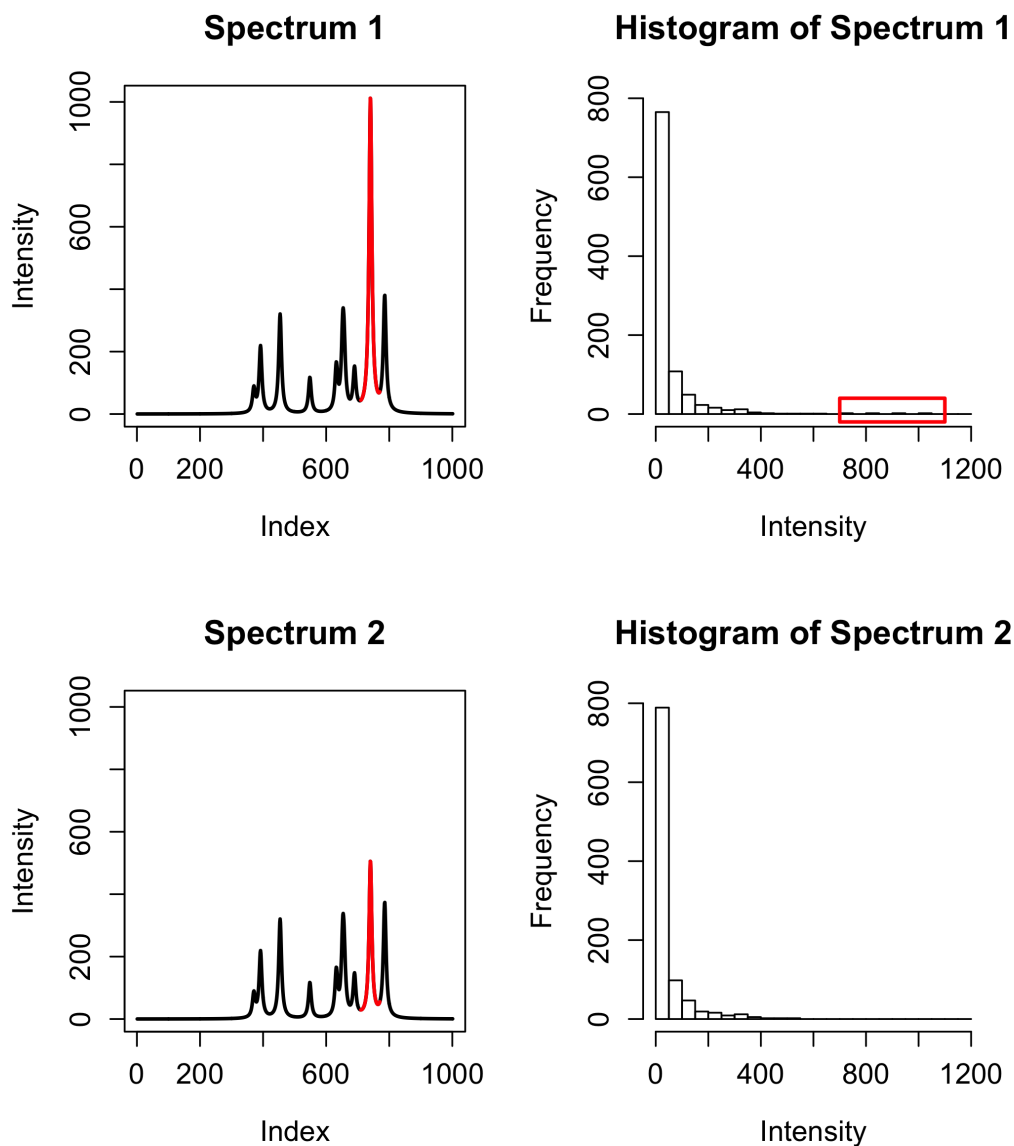


Figure 12: Comparison of histograms for two simulated spectra

In Figure 12, all black peaks in spectrum 1 and 2 are identical and the red peak is dramatically different between the two spectra. However, when transforming each spectrum into a histogram, this dramatic difference seems to disappear. The high intensity peak in spectrum 1 is translated into histogram bars in the high intensity range (see the the red box in the top right panel), but these bars do not dramatically change the overall appearance of the two

histograms, which remain very similar. NST methods based on spectral histograms can thus reduce the influence of high intensity peaks and prevent unwanted distortion (in contrast to methods such as CSN).

The distribution of intensities for a given spectrum and the distribution of intensities for a reference spectrum are assumed to be linearly related by a constant factor. The constant factor for a given spectrum i is the factor that, when multiplied by spectrum i , minimizes the distance between the histogram of spectrum i and the histogram of the reference spectrum. Because features are transformed into counts, intensities that occur rarely (for example, a single high intensity peak) have a small proportion of the total counts and thus little overall importance during minimization (see the “Implementation” section below). Intensities that occur across many features j have much greater overall importance.

Note that because low intensity noise features make up a large proportion of the spectrum, these noise features must be removed from the spectrum prior to creating the histograms.

Implementation

Let S_i be the standard deviation of spectrum i . All features j for which $x_{ij} < 5S_i$ are removed from $\mathbf{x}_i$, yielding the subset $\tilde{\mathbf{x}}_i$. Note that the number of elements in each subsetting spectrum, p_i , is potentially different for each spectrum.

$$\tilde{\mathbf{x}}_i = (\tilde{x}_{i1}, \dots, \tilde{x}_{ip_i})$$

Next, the authors suggest transforming each spectrum i using the function

$$g(\tilde{x}_{ij}) = \log_2(\tilde{x}_{ij} + 1).$$

The constant 1 is added to $\tilde{x}_{ij}$ so that the transformed values fall in the interval $[0, \infty)$. This is necessary because $\log_2(\tilde{x}_{ij}) < 0$ for $\tilde{x}_{ij} < 1$.

Let $g(\tilde{\mathbf{x}}_i)$ be the vector of transformed intensities $(g(\tilde{x}_{i1}), \dots, g(\tilde{x}_{ip_i}))$. Let $I = \{i_1, \dots, i_B\}$ be a set of intervals i_b with $b:1, \dots, B$ and $H(\tilde{\mathbf{x}}_i, I)$ be a histogram operator that transforms the spectrum $g(\tilde{\mathbf{x}}_i)$ into a vector where each element b represents the number of features with an intensity within intensity interval i_b . A single element from the vector $H(\tilde{\mathbf{x}}_i, I)$ is denoted $H_b(\tilde{\mathbf{x}}_i, I)$. Finally, let m be the index of the spectrum with the median total

intensity. The estimated dilution factor $\hat{\alpha}_i$ is given by

$$\begin{aligned}\hat{\alpha}_i &= \arg \min_{\alpha_i} \|H(g(\tilde{\mathbf{x}}_m), I) - H(g(\alpha_i \tilde{\mathbf{x}}_i), I)\|_2 \\ &= \arg \min_{\alpha_i} \sum_{b=1}^B (H_b(g(\tilde{\mathbf{x}}_m), I) - H_b(g(\alpha_i \tilde{\mathbf{x}}_i), I))^2\end{aligned}\tag{2}$$

The normalized spectrum i is thus given by

$$\mathbf{x}_i^* = \hat{\alpha}_i \mathbf{x}_i.$$

Discussion

The greatest advantage of HM is that high concentration metabolites do not distort the normalization. Torgrip et al. (2008) have shown that Histogram Matching performs significantly better than CSN and Probabilistic Quotient Normalization in terms of correcting for a dilution factor (about 10% less error). No other authors have evaluated its performance.

HM is essentially a method that searches for a dilution factor that minimizes the differences between two distributions. Other statistical methods may be able to perform this task more optimally, perhaps by defining the distributions more continuously (rather than relying on the discretization of intensities into user-defined intensity intervals).

It is also possible that the relationship between the two distributions is not linear, meaning that the estimated dilution factor may induce bias. In addition, the performance of this method depends on the successful removal of low-intensity noise features, which requires the definition of an intensity threshold. This threshold could be problematic if too many signal features are removed. Indeed, the threshold proposed by Torgrip et al. (2008), $5S_i$, might be too high, resulting in the removal of signal features (see Figure 13).

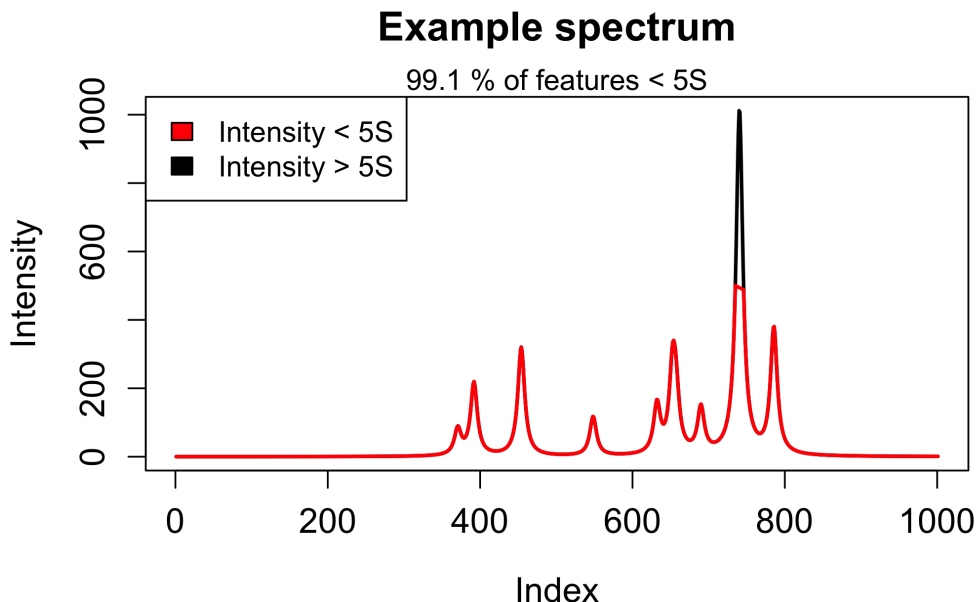


Figure 13: Simulated spectrum

Lastly, the length of the intervals used undoubtedly influences the results of this method, but Torgrip et al. (2008) do not propose any particular interval size. The development of a method for creating optimal intervals thus remains to be explored.

2.2.6 Variance Stabilizing Normalization (VSN)

Variance Stabilizing Normalization is a method originally applied to gene expression data (Durbin et al., 2002; Durbin and Rocke, 2003; Lin et al., 2008), which was applied to NMR datasets by Parsons et al. (2007) and Purohit et al. (2004). VSN is typically preceded by CSN.

Theoretical motivation and principals

Unlike the other normalization methods, VSN corrects for both heteroscedasticity and overall differences between spectra. Correcting for heteroscedasticity is important because in non-normalized spectra, variance tends to increase with increases in intensity. By transforming each spectrum, VSN renders the variance constant across all intensities, all the while bringing each spectrum onto the same scale.

Model for observed intensities

The observed intensities x_{ij} of feature j in spectrum i can be described by the following equation:

$$x_{ij} = \beta_{ij} + \mu_{ij} \exp(\eta) + \epsilon \quad (3)$$

where β_{ij} is the offset of the baseline from zero, μ_{ij} is the true intensity, η is a multiplicative error term and ϵ is an additive error term (measurement errors). Both η and ϵ are assumed to be i.i.d. with distribution $N(0, \sigma_\eta^2)$ and $N(0, \sigma_\epsilon^2)$ respectively. If the spectrum has been baseline corrected, $\beta_{ij} = 0$.

Need for variance stabilization

The variance of x_{ij} is a function of its expected value. The expected value u_{ij} and variance v_{ij} of x_{ij} are developed based on Equation 3:

$$\begin{aligned} u_{ij} &= E(x_{ij}) = E(\beta_{ij} + \mu_{ij} \exp(\eta) + \epsilon) \\ &= \beta_{ij} + \mu_{ij} m_\eta \end{aligned} \quad (4)$$

where $m_\eta = E(\exp(\eta)) = \exp(\sigma_\eta^2/2)$ and

$$\begin{aligned} v_{ij} &= V(x_{ij}) = V(\beta_{ij} + \mu_{ij} \exp(\eta) + \epsilon) \\ &= \mu_{ij}^2 V_\eta + \sigma_\epsilon^2, \end{aligned} \quad (5)$$

where $V_\eta = V(\exp(\eta)) = \exp(\sigma_\eta^2) (\exp(\sigma_\eta^2) - 1)$ and $\sigma_\epsilon^2 = V(\epsilon)$.

Using Equation 5, we can rewrite the true intensity μ_{ij} in terms of the expected value u_{ij} :

$$\mu_{ij} = \frac{u_{ij} - \beta_{ij}}{m_\eta}.$$

If we replace μ_{ij} in equation 5 with $\frac{u_{ij} - \beta_{ij}}{m_\eta}$, it becomes apparent that the variance v_{ij} is a function of the expected value u_{ij} .

$$v_{ij}(u_{ij}) = \left(\frac{u_{ij} - \beta_{ij}}{m_\eta} \right)^2 V_\eta + \sigma_\epsilon^2$$

Variance stabilizing transformation

When the variance is a function of the expected value, noted by $v_{ij}(u_{ij})$, a variance-stabilizing transformation $h(t)$ is given by

$$h(t) = \int_{-\infty}^t \frac{1}{\sqrt{v_{ij}(u_{ij})}} du_{ij}.$$

The solution to this integral is

$$h(x_{ij}) = \ln \left(x_{ij} - \beta_{ij} + \sqrt{(x_{ij} - \beta_{ij})^2 + \frac{\sigma_\epsilon^2}{\sigma_\eta^2}} \right).$$

If we replace $x_{ij} - \beta_{ij}$ with z_{ij} and $\frac{\sigma_\epsilon^2}{\sigma_\eta^2}$ with λ , this becomes

$$h_{ij} = h(z_{ij}) = \ln \left(z_{ij} + \sqrt{z_{ij}^2 + \lambda} \right).$$

Note that z_{ij} represents the intensity of feature j after baseline correction of spectrum i .

The parameter λ is estimated using calibration spectra (see the “Implementation” section below). Applying the same parameter λ to each transformation renders the different spectra comparable and stabilizes the variance of each spectrum so that $V(x_{ij})$ does not depend on $E(x_{ij})$. For small intensities, the transformation function is approximately linear and for large intensities, the transformation function is a log function.

Optimization of λ

According to Parsons et al. (2007), the optimal parameter λ is the one that minimizes $S(h, \lambda)$.

$$S(h, \lambda) = \sum_{i=1}^n SSE_i = \sum_{i=1}^n \sum_{j=1}^p \left(h_{ij} - \frac{1}{n} \sum_{i=1}^n h_{ij} \right)^2,$$

where $h_{ij} = h(z_{ij})$.

However, due to transformation, the scale of $S(h, \lambda)$ for different λ values is not comparable. For this reason, each transformed intensity $h(x_{ij})$ must be divided by the p^{th} root of the Jacobian of the transformed intensity, as suggested by Durbin and Rocke (2003). The Jacobian is written

$$J(\mathbf{h}_{i,\lambda}) = \prod_{j=1}^p \left| \frac{\partial \mathbf{h}_{ij,\lambda}}{\partial z_{ij}} \right| = \prod_{j=1}^p \left| \frac{1}{\sqrt{z_{ij}^2 + \lambda}} \right|$$

The transformed, Jacobian-corrected intensities w_{ij} are thus

$$w_{ij} = h_{ij} J^{-1/p}(\mathbf{h}_{i,\lambda}) = h_{ij} \left(\prod_{j=1}^p \left| \sqrt{z_{ij}^2 + \lambda} \right| \right)^{1/p}$$

and the function $S(w, \lambda)$ is written

$$S(w, \lambda) = \sum_{i=1}^n SSE_i = \sum_{i=1}^n \sum_{j=1}^p \left(w_{ij} - \frac{1}{n} \sum_{i=1}^n w_{ij} \right)^2 \quad (6)$$

Parsons et al. (2007) used a modified formula for $J^{-1/p}(\mathbf{h}_{i,\lambda})$, which is computationally more stable:

$$\begin{aligned} J^{-1/p}(\mathbf{h}_{i,\lambda}) &= \exp \left(\ln \left(\left(\prod_{j=1}^p |\sqrt{z_{ij}^2 + \lambda}| \right)^{1/p} \right) \right) \\ &= \exp \left(\frac{\sum_{j=1}^p \ln \left(\sqrt{z_{ij}^2 + \lambda} \right)}{p} \right) \end{aligned} \quad (7)$$

Each transformed, Jacobian-corrected intensity w_{ij} is thus

$$w_{ij} = h_{ij} J^{-1/p}(\mathbf{h}_{i,\lambda}) = h_{ij} \exp \left(\frac{\sum_{j=1}^p \ln \left(\sqrt{z_{ij}^2 + \lambda} \right)}{p} \right)$$

Implementation

The calibration samples used to estimate λ can be collected in one of two ways. Purohit et al. (2004) suggest dividing a single biological sample into n separate samples and collecting NMR data for each sample i using the same protocol as for all other samples. Parsons et al. (2007) suggest combining small quantities of multiple biological samples (including at least one sample from each class in the data set) into a homogenous mixture and taking n samples from this mixture (again using the same protocol as for the other samples). The calibration spectra should then be pre-processed in the same way as the other spectra in the data set. The only difference between these technical replicates should be due to technical variation. Parsons et al. (2007) use $n = 5$ total calibration samples.

The estimate for λ is the value that minimizes $S(w, \lambda)$ (Equation 6). In $S(w, \lambda)$, both the Jacobian and the transformation depend on the value of λ , so λ must be estimated using a non-linear minimization procedure (Parsons et al. (2007) estimate λ using the Nelder-Mead unconstrained non-linear minimization routine in Matlab).

Each feature j in the normalized spectrum $\mathbf{x}_i^*$ is thus

$$x_{ij}^* = \ln \left(x_{ij} - \beta_{ij} + \sqrt{(x_{ij} - \beta_{ij})^2 + \hat{\lambda}} \right).$$

Discussion

The greatest advantage of VSN is that it addresses the issue of heteroscedasticity in addition to the issue of inter-sample variation. Indeed, Kohl et al. (2012) demonstrated that VSN, unlike the other normalization methods, renders the standard deviation constant over the entire range of spectral intensities. It is also one of the best methods for reducing intensity-dependent differences between spectra and overall inter-sample differences (Kohl et al., 2012).

VSN very slightly reduces the contrast between noise and signal. Nonetheless, it maintains enough contrast so that signal can be distinguished from noise, and it is one of the best methods for biomarker identification (Li et al., 2015) and overall classification performance (Kohl et al., 2012). Indeed, the classification performance of VSN is only surpassed by Quantile Normalization (see Section 2.4.1) (Kohl et al., 2012).

In the presence of imbalanced metabolite regulation resulting from multiple metabolites (i.e. violation of the self-averaging principle), VSN accurately estimates intensities if preceded by Hochrein et al. (2015)’s pre-normalization step (see Section 2.5), as is the case for Linear Baseline Normalization (2.2.3) and PQN (2.2.4). If only one regulated metabolite upsets the balance of metabolite concentrations, VSN is effective if that metabolite varies by small degrees between spectra. This method is less accurate when a single metabolite varies by large degrees between spectra.

One of the greatest disadvantages of VSN is the requirement of calibration spectra for the estimation of the transformation parameter λ . Calibration samples must be included in the experimental design and measured along with the other samples in order to apply this method.

Certain biofluids come in small quantities, making it difficult to divide a single sample into n calibration samples, as was suggested by Purohit et al. (2004). The timing of sample collection could also make the mixing of multiple samples (the method proposed by Parsons et al. (2007)) impossible. The composition of blood evolves with time, meaning that a sample collected at one time has different properties several hours later. If the calibration samples are created by mixing blood samples only after all samples have been received, the metabolic profiles of the mixture will likely be corrupted.

2.2.7 Other methods

Normalization methods other than those outlined above exist but have been shown to have sub-optimal performance in Kohl et al. (2012) and/or Hochrein et al. (2015). These methods include Cyclic Loess Normalization, Li-Wong (or Non-Linear Baseline) Normalization and Contrast Normalization.

Another normalization method is Cubic Splines Normalization. This method, proposed by Workman et al. (2002), was originally used in the field of gene expression analysis and was later applied to NMR spectra by Kohl et al. (2012) and Hochrein et al. (2015). It was not described here because it does not seem particularly interesting for NMR spectra.

2.3 Scaling methods

2.3.1 Autoscaling and Pareto Scaling

Theoretical motivation and principals

The goal of Autoscaling and Pareto Scaling is to make the variance of each feature j less variable across all spectra, giving all features equal importance. Pareto Scaling has a less dramatic normalizing effect because it uses the square root of the estimated standard deviation as the scaling factor (rather than the estimated standard deviation). Using a smaller scaling factor makes it possible scale features without inflating their measurement error too much.

Implementation

Autoscaling and Pareto scaling are applied to all features j whose standard deviation across all spectra, S_j , is greater than 0. For Autoscaling, feature j in each spectrum i is centered around the mean $\bar{x}_j$ and then divided by the standard deviation of feature j across all spectra, S_j . The new intensity x_{ij}^* is thus

$$x_{ij}^* = \frac{x_{ij} - \bar{x}_j}{S_j}$$

where $S_j = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (x_{ij} - \bar{x}_j)^2}$ and $\bar{x}_j = \frac{1}{n} \sum_{i=1}^n x_{ij}$. Pareto scaling is similar to Autoscaling, except that the centered feature j from spectrum i is divided by the square root of the standard deviation S_j :

$$x_{ij}^* = \frac{x_{ij} - \bar{x}_j}{\sqrt{S_j}}$$

Discussion

Autoscaling and Pareto Scaling have been shown to heavily distort spectra. They drastically reduce the contrast between noise and signal, hardly reduce intensity-dependent differences between spectra and substantially increase the standard deviation of features (Kohl et al., 2012). Classification performance and biomarker identification can also be unsatisfactory when using these methods, especially in the case of Pareto Scaling (Li et al., 2015).

Autoscaling in particular tends to increase the influence of noisy variables (and thus the influence of measurement error). While Pareto Scaling inflates this influence to a lesser extent, it still remains sensitive to large fluctuations in the data (Smolinska et al., 2012).

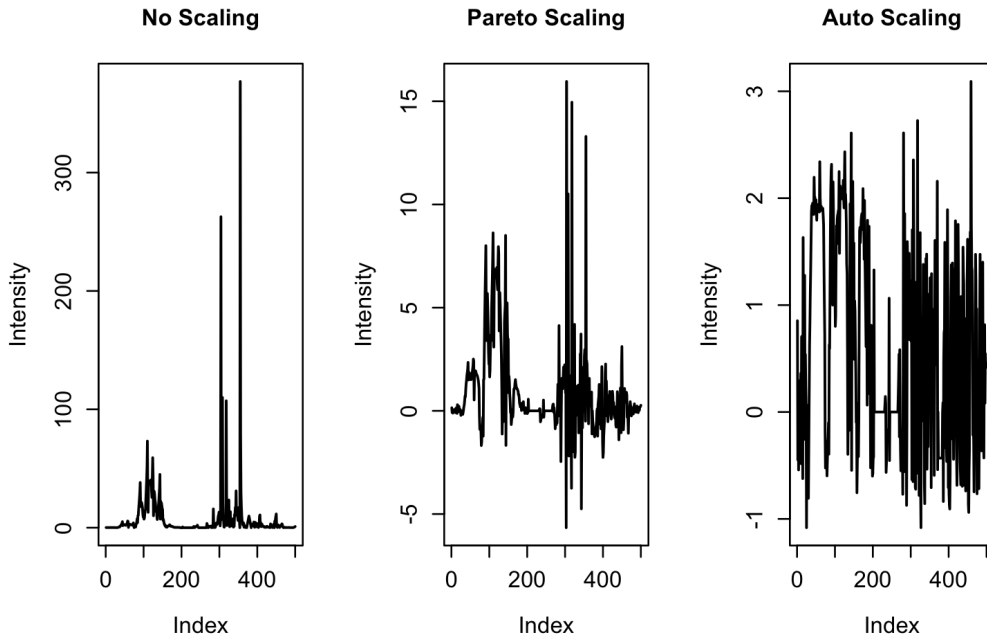


Figure 14: Example of a non-scaled, Pareto-scaled and Autoscaled urine spectrum

2.4 Transformation methods

2.4.1 Quantile Normalization (QN)

Quantile Normalization is a transformation method that was introduced by Bolstad et al. (2003) for the pre-processing of gene expression data. It was later applied to NMR spectra by Kohl et al. (2012), Wang et al. (2013) and Hochrein et al. (2015).

Theoretical motivation and principals

For Quantile Normalization, spectral intensities are transformed so that each spectrum has exactly the same statistical distribution of intensities. As a result, overall differences between spectra are reduced.

Implementation

First, the intensities of each spectrum i are ordered from smallest to largest, with $x_{i(1)}$ being the smallest intensity in spectrum i and $x_{i(p)}$ being the largest. Thus, a vector including the intensities of rank (j) from each spectrum is given by

$$\mathbf{x}_{(j)} = \{x_{1(j)}, \dots, x_{n(j)}\}.$$

The average of all intensities of rank (j) is given by

$$\bar{x}_{(j)} = \frac{1}{n} \sum_{i=1}^n x_{i(j)}$$

$$\bar{\mathbf{x}} = (\bar{x}_{(1)}, \dots, \bar{x}_{(p)})$$

The principal of QN is to replace the intensities of rank (j) in each spectrum i by the same value: $\bar{x}_{(j)}$, the average of all intensities of rank (j) .

$$x_{i(j)}^* = \bar{x}_{(j)}$$

Then, the intensities in each spectrum are put back in their original order. Note that the set of intensity values in each transformed spectrum is the same but that these values are potentially attached to different features. After transformation, all spectra have the same minimum, maximum, mean and median intensities.

Figures 15 and 16 demonstrate how QN is performed. In Figure 15, the vector of averages, $\bar{\mathbf{x}}$, is plotted against the rank of each average. This distribution is used to normalize all of the spectra in a given data set. Each feature of

rank (j) is replaced with the mean intensity for that rank, $\bar{x}_{(j)}$. For example, in each spectrum, the feature j of rank 497 is replaced by the mean intensity 74.94 (left panel). As another example, the feature j of rank 460 is replaced by the mean intensity 13.58 (right panel).

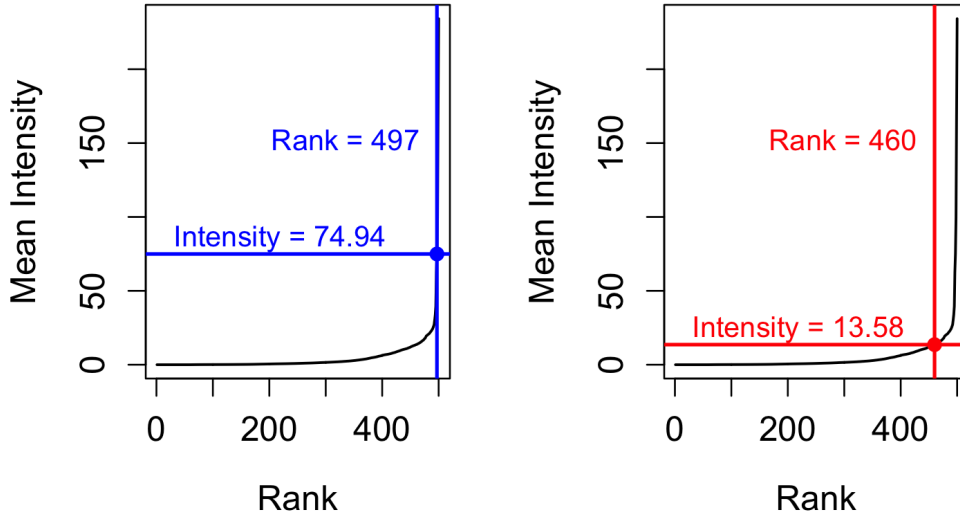


Figure 15: Plot of $(\bar{x}_{(1)}, \dots, \bar{x}_{(p)})$ vs rank (j)

Figure 16 shows two urine spectra before (top panel) and after (bottom panel) application of QN. The feature of rank 497 is feature 318 in the first spectrum and feature 400 in the second spectrum. Both of these features have been replaced by the mean intensity 74.94 (in blue). The feature of rank 460 is feature 131 in the first spectrum and feature 116 in the second spectrum. Both of these features have been replaced by the mean intensity 13.58 (in red). Each feature in each spectrum is replaced in this way, resulting in n transformed spectra.

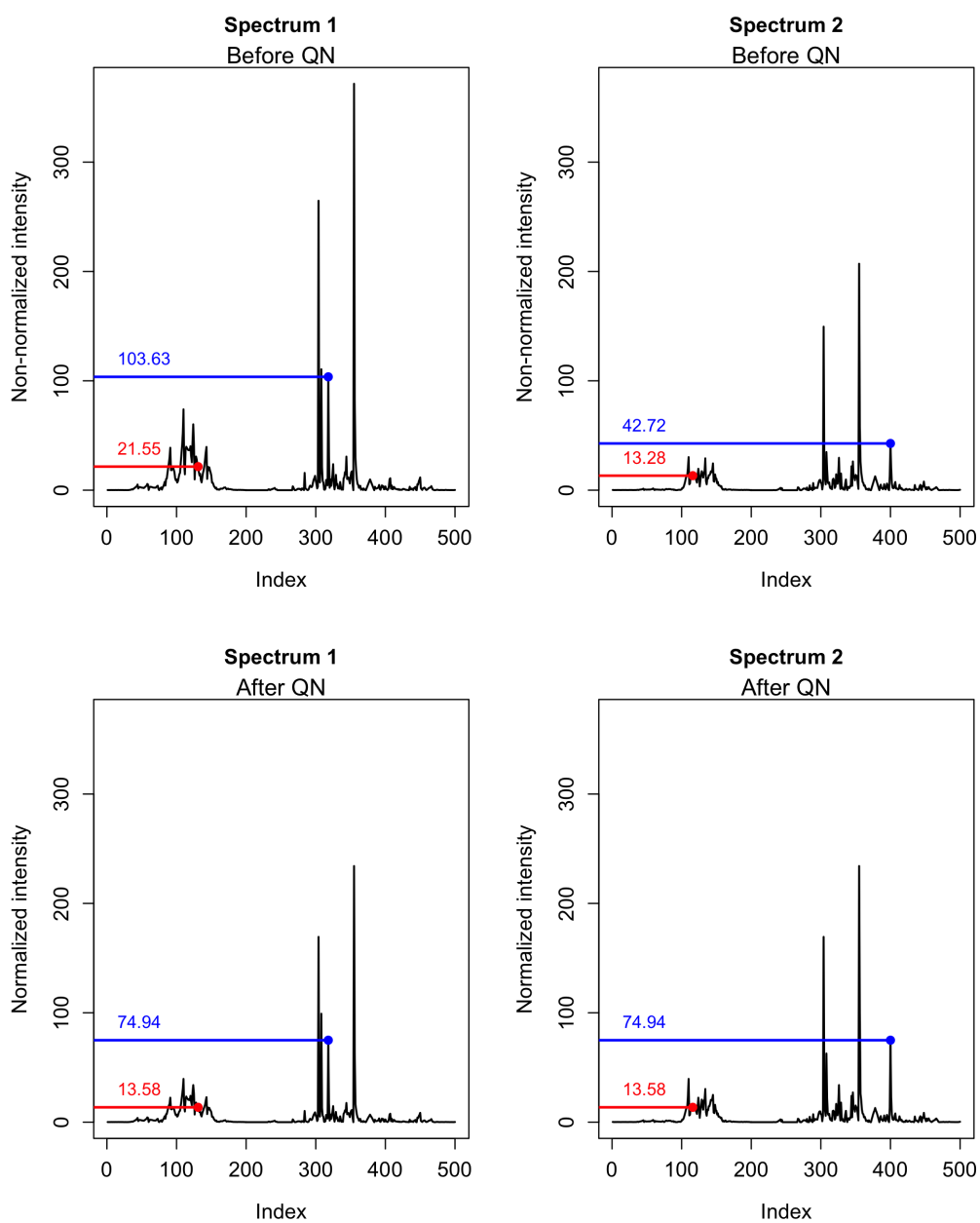


Figure 16: Examples of Quantile Normalized spectra

Discussion

QN is one of the best methods for the reduction of overall differences between samples, with highly consistent results for a large variety of metabolites (Kohl

et al., 2012). It also has the best classification performance for large data sets ($n \geq 50$) (Kohl et al., 2012).

While it is the best method for reducing intensity-dependent differences between spectra, it does not satisfactorily correct for intensity-dependent variance (Kohl et al., 2012). Another disadvantage is that its accuracy depends on the properties of the spectra in question: QN fails when multiple regulated metabolites violate the self-averaging principle in each spectrum (Hochrein et al., 2015; Li et al., 2015). It also distorts spectral intensities when a single regulated metabolite is highly variable across spectra (Hochrein et al., 2015).²

2.5 Pre-normalization step (Hochrein et al., 2015)

2.5.1 Pre-selection of unregulated features

Theoretical motivation and principals

As explained above, Linear Baseline Normalization, Probabilistic Quotient Normalization and Variance Stabilizing Normalization fail to yield accurate intensity estimates when the regulation of multiple metabolites violates the self-averaging principle. This accuracy can be improved by applying a pre-normalization step proposed by Hochrein et al. (2015).

The goal of this pre-normalization step is to reduce the set of features used to determine normalization and/or transformation parameters to a smaller subset including only “unregulated” features (see Section 2.1.2 for the definition of “regulated” and “unregulated” metabolites). The variance of regulated features is assumed to be larger than the variance of unregulated features. Thus, the variance of features can be used to filter out regulated features from the spectra.

Hochrein et al. (2015) propose two methods for determining the most optimal subset of features to use to estimate parameters: selection by p-value and selection by visual validation.

Selection by p-value

The exact proportion of features to remove from the data set is determined

²Unlike for Linear Baseline Normalization, PQN and VSN, however, this poor performance in the presence of severely imbalanced regulation cannot be improved using Hochrein et al. (2015)’s pre-normalization step (see Section 2.5).

by iteratively removing high-variance features and performing a Shapiro-Wilk normality test at each iteration. For spectra with only unregulated features, the total intensity of each spectrum is assumed to follow a normal distribution, in which case the p-value of the Shapiro-Wilk test should be greater than 0.05. The proportion of features that should be removed from the original data set is the smallest proportion for which the p-value of the corresponding Shapiro-Wilk test is greater than 0.05 (i.e. the hypothesis of normality is not rejected).

Implementation

First, the variance of each feature across all spectra, S_j^2 , is calculated as follows:

$$S_j^2 = \frac{1}{n-1} \sum_{i=1}^n (x_{ij} - \bar{x}_j)^2$$

All zero variance features are removed from each spectra and the remaining features are then ranked from largest to smallest in terms of their variance, S_j^2 .

Let π_k be the k^{th} element from the vector of proportions $\boldsymbol{\pi} = \{0, 0.05, 0.10, \dots, 0.90\}$, $k : 1, \dots, 19$. For each iteration k , $\pi_k \times (100)\%$ of the features with the highest estimated variance are removed from the original data set since these features are potentially associated with “regulated” metabolites, and thus are not desired. Let J_k be the set of indexes corresponding to the features remaining in the data set at iteration k . t_{ik} , the total intensity of a given spectrum i at iteration k , is then calculated as follows:

$$t_{ik} = \sum_{j \in J_k} x_{ij}$$

The Shapiro-Wilk test is then applied to the vector of total intensities $\mathbf{t}_k = (t_{1k}, \dots, t_{nk})$ and the p-value is stored. The smallest value π_k with a p-value greater than 0.05 is selected as the proportion π of features to remove, and the proportion of features to keep is thus one minus this selected value π_k . The authors suggest removing no more than 90% of the features in each spectra in order to ensure stable computation of the normalization and/or transformation parameters.

Discussion

The p-values generated using the method described above do not necessarily increase as the proportion π_k increases. For example, in Table 1, all p-values

greater than 0.05 are marked in bold (for p-values greater than 0.05, we conclude that the hypothesis of normality is not rejected). The smallest proportion π_k for which the hypothesis of normality is not rejected is $\pi_k = 0$. This is not the proportion that would be selected by visual validation (see “Selection by visual validation” below and Figure 17). The selection of the proportion π based on p-values can be unstable, making it difficult to determine which value π is the most optimal.

Table 1: Example of p-values for each proportion π_k (urine data)

% features removed ($\pi_k \times 100\%$)	p-value
0	0.0605
5	0.0811
10	0.0499
15	0.1017
20	0.0982
25	0.0208
30	0.0201
35	0.0179
40	0.0507
45	0.0653
50	0.0368
55	0.0251
60	0.0051
65	0.0043
70	0.0128
75	0.0025
80	0.0028
85	0.0026
90	0.0027

Selection by visual validation

The proportion π can also be determined by visual validation. Suppose there is some reference spectrum $\mathbf{z}$ used to calculate the normalization or transformation parameters (i.e. the median spectrum for Linear Baseline and PQN, or the mean spectrum of the calibration samples for VSN). A certain scaling factor c_{ij} can be calculated by dividing the intensity of a given feature j in a given spectrum i by the intensity of that same feature in

the reference spectrum:

$$c_{ij} = \frac{x_{ij}}{z_j}$$

$$\mathbf{c}_i = (c_{i1}, \dots, c_{ip})$$

Let the standard deviation, S_{c_i} , of the scaling factors $c_{i1}, \dots, c_{ip}$ for a given spectrum i be denoted

$$S_{c_i} = \sqrt{\frac{1}{p-1} \sum_{j=1}^p (c_{ij} - \bar{c}_i)^2}$$

If, for a given spectrum i , the standard deviation of all of these scaling factors is relatively small, it is likely that the features in the spectrum are for the most part unregulated. The scaling factor variation for the entire set of spectra is summarized as the mean of the standard deviations S_{c_i} , the “mean sample-wise standard deviation (*mswsd*).”

$$mswsd = \frac{1}{n} \sum_{i=1}^n S_{c_i}$$

By displaying the *mswsd* as a function of the proportion of features removed from the data set (or the proportion of features remaining), it is possible to assess the convergence of the *mswsd* to a constant value. If the *mswsd* appears to converge to a constant, then the correct proportion of features to remove is the smallest proportion attaining that constant.

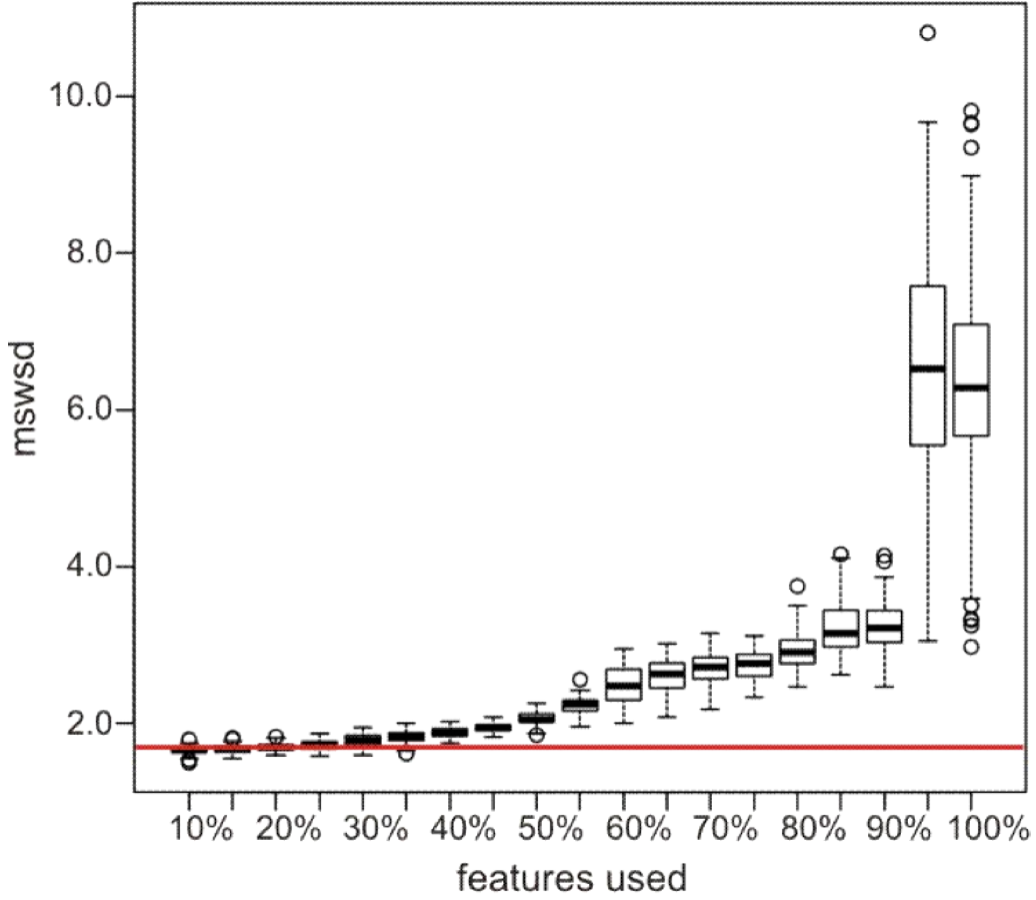


Figure 17: Example of the convergence of the $mswsd$ (Hochrein et al., 2015) where “% features used” = $(1 - \pi_k) \times 100\%$

Implementation

First, a reference spectrum $\mathbf{z}$ is selected. The choice of reference spectrum depends on the NST method in question. For Linear Baseline Normalization and PQN, the reference spectrum would be the median spectrum and for VSN it would be the mean spectrum of the calibration samples.

For each iteration k , $\pi_k \times (100)\%$ of features with the highest estimated variance are removed from the the reference spectrum and the original data set. Then scaling factors are calculated for each of the remaining features in the reduced data set. These calculations are stored in a matrix $n \times p_k$ of scaling factors, $\mathbf{C}_k$, where p_k is the total number of features remaining at iteration k ($p_k = p(1 - \pi_k)$) and where each row contains the scaling factors $c_{i1}, \dots, c_{ip_k}$ for a given spectrum i .

Rather than calculating the *mswsd* once for each proportion π_k , it is calculated a total of 100 times per proportion based on random subsets of features from $\mathbf{C}_k$. This requires the introduction of an additional loop inside of the k loop: For each iteration $m : 1, \dots, 100$, 66% of the features in $\mathbf{C}_k$ are selected at random and the *mswsd* is calculated based on this subset. This process is repeated 100 times for each iteration k .

Boxplots of the *mswsd* values for each proportion π_k make it possible to assess the variability of the *mswsd* resulting from the random selection of features and to visually evaluate the convergence of the *mswsd* values (see Figure 17). The ideal proportion π_k is the smallest proportion for which the median *mswsd* reaches the convergence constant.

Discussion

The selection of the proportion π_k using visual validation seems to be more robust than selection by p-value, but the final choice of π_k requires the intervention of the user (and thus cannot be made automatically).

2.6 Conclusion

The table below (Figure 18) summarizes the strengths and weaknesses of the various normalization, scaling and transformation methods described above. The rows represent the different criteria used in the literature to evaluate the methods (see Section 2.1.1 for an explanation of these criteria). The column “Publication” lists the publications in which these evaluation criteria were used to assess different methods.

If calibration spectra are available, VSN would be a good normalization method to apply. For all criteria for which VSN has been evaluated, VSN has been shown to be one of the best methods.

The other methods must be selected depending on the type of spectra in question and the goals of the analysis. QN is a promising method for improving classification performance, increasing spectral comparability and reducing intensity-dependent differences, but it should be avoided if samples contain many fluctuating metabolites (highly imbalanced regulation). PQN is one of the best methods for reducing overall differences between spectra, reducing intensity-dependent bias and preserving the signal to noise ratio, but it might not be the best method to apply to urine data, and it only improves classification for small data sets.

CSN is rarely among the best normalization methods. While it is one of the best methods for ensuring reproducibility, it ranks poorly for all other criteria. CSN is particularly ineffective for urine spectra suffering from dilution effects.

Pareto and Autoscaling should generally be avoided as they are by far the worst methods described here (in terms of increasing spectral comparability, reducing intensity-dependent differences between spectra and variance, preserving the signal to noise ratio and improving classification).

Category	Subcategory	Best	Worst	Publication	Not evaluated
Reducing overall differences between spectra	Comparability	PQN, QN, VSN	CN, Autoscaling, Pareto Scaling	Kohl et al (2012)	HM
	Reproducibility	CSN, PQN		Wang et al (2013)	HM, Linear Baseline, QN, VSN, NIS, Autoscaling, Pareto Scaling
Reducing intensity-dependent differences between spectra		QN, VSN	Autoscaling, Pareto Scaling	Kohl et al (2012)	HM
Reducing intensity-dependent bias	Unbalanced regulation (due to one metabolite)	Linear Baseline, PQN, QN*, VSN*		Hochrein et al (2015)	HM, CN, Autoscaling, Pareto Scaling
	Unbalanced regulation (due to many metabolites)	Linear Baseline [△] , PQN [△] , VSN [△]	Cubic Splines	Hochrein et al (2015)	HM, QN, CN, Autoscaling, Pareto Scaling
Reducing intensity-dependent variance		VSN	Autoscaling, Pareto Scaling	Kohl et al (2012)	HM
Preserving and/or increasing contrast between noise and signal		CN, PQN, Linear Baseline, VSN	Autoscaling, Pareto Scaling	Kohl et al (2012)	CSN, HM
Removing dilution effects (urine data)		HM	PQN, CSN	Torgrip et al (2008)	Linear Baseline, QN, VSN, CN, Autoscaling, Pareto Scaling
Improving classification performance		VSN, PQN [△] , QN [°]	Autoscaling, Pareto Scaling	Kohl et al (2012)	HM

* Only for small fold changes, [△] only with pre-normalization step, [△] only for small data sets (n = 40 or less), [°] only for large data sets (n = 40 or more)

CN = Creatinine Normalization, CSN = Constant Sum Normalization, HM = Histogram Matching, PQN = Probabilistic Quotient Normalization, QN = Quantile Normalization, VSN = Variance Stabilizing Normalization

Figure 18: Summary of NST Methods

3 Implementation and evaluation of certain NST methods

In this chapter, a selection of the NST methods described in Chapter 2 will be applied to real data sets and evaluated.

3.1 Data

3.1.1 Data sets

Two data sets will be used to evaluate the performance of the different NST methods.

Urine data

The first data set is a set of urine spectra from human subjects. One urine sample was collected from each of three subjects, each subject being labeled S1, S2 or S3. Each urine sample was homogenized before being split into eight samples. Half of these samples were diluted with water (labeled D1) and the other half were left as is (labeled D0). All samples were then frozen. For each of four consecutive days (J1, J2, J3, J4), one diluted and non-diluted sample from each patient was thawed and then analyzed to generate ^1H NMR spectra.

A total of 24 urine spectra were generated. Because each original urine sample was simply split into eight parts, all spectra for a given subject should ideally be exactly the same. However, different experimental factors could create artificial differences between spectra from the same patient: the dilution of the sample, the day of measure, the freezing and thawing process, the measurement itself, etc.

Serum data

The second data set is a set of blood serum spectra from human subjects. In this study, four different subjects gave blood on eight different days (labeled J1, J2, ..., J8), yielding a total of 32 different samples. Each sample was measured using NMR spectroscopy on the day of its collection.

A total of 32 NMR spectra were generated, one for each of the 32 blood samples. Ideally, spectra associated with a given patient should be very similar (the blood itself changing very little between days), and the spectra from different patients should be different.

3.1.2 Data pre-processing

Both data sets were pre-processed using the **SOAP** package in R (all parameters set at their default values). For the serum data, all pre-processing steps before normalization were applied. For the urine data, all steps except warping and normalization were performed. Note that the serum spectrum for subject 4 on day 3 was removed from the data set because it was identified as an outlier (as was done in Féraud et al. (2015)).

3.2 Methods applied

Methods

The NST methods tested include Constant Sum Normalization (CSN, see Section 2.2.1), Linear Baseline Normalization (see Section 2.2.3), Probabilistic Quotient Normalization (PQN, see Section 2.2.4), Quantile Normalization (QN, see Section 2.4.1) and Variance Stabilizing Normalization (VSN, see Section 2.2.6). Note that VSN was not applied to the serum data set due to a lack of technical replicates.

Linear Baseline Normalization, PQN and VSN were also performed with the pre-normalization step proposed by Hochrein et al. (2015) (see Section 2.5). Implementation details can be found in the Annex (see Section 7.2.1).

Choice of parameters for Hochrein et al. (2015)’s pre-normalization step

Based on the plots of *mswsd* versus proportion of high-variance features removed, π_k , the parameter π was selected for both the urine and serum data sets.

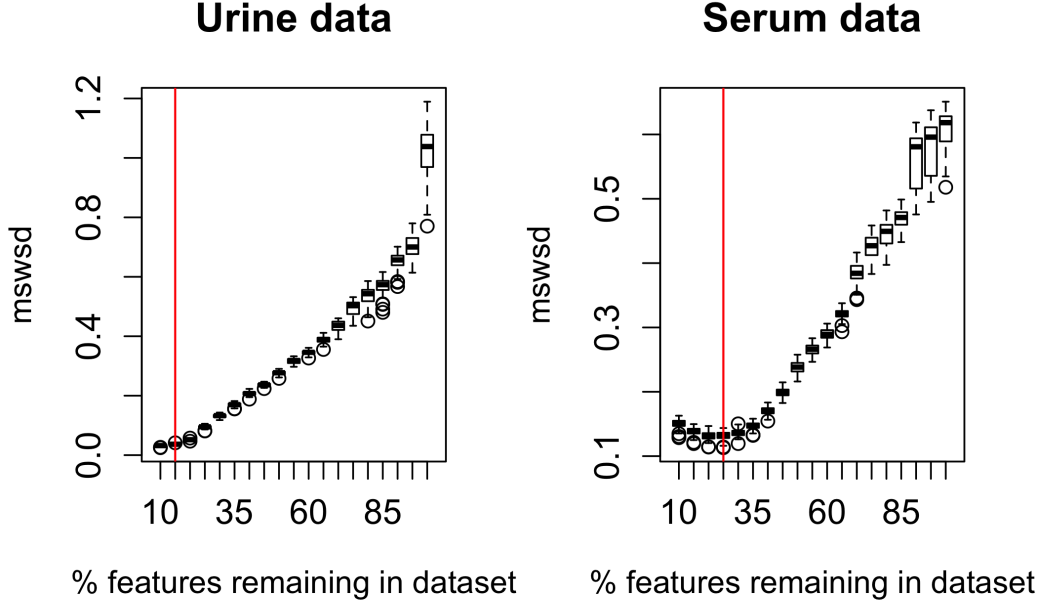


Figure 19: Plots of the *mswsd* for the urine and serum data sets

The *mswsd* seems to be near minimal when only 15% (red line, panel 1) of the original features in the urine data set are used ($\pi = 0.85$) and when only 25% (red line, panel 2) of the original features in the serum data set are used ($\pi = 0.75$). While it is not clear that the *mswsd* converges for the urine data, Hochrein et al. (2015)’s pre-normalization step was applied to determine whether it would improve the quality of NST nonetheless. Note that $\pi = 0.85$ was selected for the urine data set because it seemed to yield an *mswsd* about equal to the *mswsd* for $\pi = 0.90$ while allowing the retention of more features.

NST parameters for PQN, Linear Baseline Normalization and VSN were selected based on each of the reduced data sets (see Section 7.2.1 for details). Then these NST methods were applied to each of the original data sets.

3.3 Performance measures

The performance measures presented in the literature (see the Annex, Section 7.1) are not always well justified and in some cases are overly complex or even

theoretically flawed. For the evaluation of the NST methods tested here, none of the evaluation methods proposed in the literature were used.

The performance measures used here are those used by the UCL MBX team (Université catholique de Louvain, Belgium) and were calculated using the `MXBUCL` package in R³. They aim to evaluate the ability of an NST method to reduce overall differences between similar spectra (the first main purpose of NST listed in Section 2.1.1). In this case, spectra of samples from the same subject should be as similar as possible and spectra of samples from different subjects should be as different as possible. This is assessed by evaluating (after NST) the quality of

1. the variance structure (PCA and group inertia),
2. the clustering of the data into clusters by subject (Dunn Index and Davies-Bouldin Index) and
3. the classification of the data into the subject groups (Rand Index, Adjusted Rand Index, Partial Least Squares Discriminatory Analysis)

The ability of NST methods to achieve the other main objectives in Section 2.1.1 – the reduction of intensity-dependent bias, variance and differences between spectra, the preservation and/or increase of the contrast between noise and signal, the removal of dilution effects and the improvement of classification performance – is not assessed in this study. (Note that “classification performance” as described in Section 2.1.1 is different than the classification referred to in this section. For the former, the data sets used contained spectra for healthy and sick subjects. For the latter, the data sets used contained spectra for healthy subjects only and thus classification performance serves the purpose of assessing the overall reduction of differences between spectra for the same subject (the first main purpose of NST listed in Section 2.1.1).

3.3.1 Variance structure

Principal Component Analysis

Principal Component Analysis (PCA) is used here to assess the variance structure of the data after NST. PCA is a statistical technique that reduces a centered $n \times p$ data matrix $\mathbf{X} = (\mathbf{x}_1, \dots, \mathbf{x}_p)$ composed of n spectra and p

³available at <https://github.com/manonmartin/MBXUCL>

variables into a matrix $\mathbf{V}^* = (\mathbf{v}_1, \dots, \mathbf{v}_{p^*})$ of a lower dimension ($p^* < p$). The rows of $\mathbf{V}^*$ are known as the “scores”.

PCA is used here to identify the percentage of variance explained by the first principal components and to visually evaluate the variance structure of the data after NST. See the Annex (Section 7.2.2) for more details concerning PCA.

Percentage of variance explained

The two data sets considered here were generated based on samples coming from $K = 3$ subjects (urine data) and $K = 4$ subjects (serum data). Ideally, all spectra associated with the same subject should be identical after NST. Thus, all variation in the data after NST should be explained by the first $K - 1$ principal components (the first two components for the urine data and the first three for the serum data). The best NST method is the one with the highest percentage of variance explained by the first $K - 1$ principal components.

Visualization of the variance structure

The PCA scores can also be plotted to assess the variance structure of the data after NST. Ideally, score vectors for the same subject should be closely clustered together and far from the other subject clusters.

Inertia between and within groups

Measures of inertia between and within groups (in this case, inter- and intra-subject) were also used to assess the variance structure of the data after NST. The total inertia of a matrix is a sum of the between-group inertia and within-group inertia. The best NST method is the one with the largest between-group inertia and the smallest within-group inertia. For more information concerning the calculation of between- and within-group inertia, refer to the Annex (Section 7.2.2).

3.3.2 Clustering

Let $\mathbf{C} = \{C_1, \dots, C_K\}$ be a partition of a data set into K clusters, where each spectra in the data set belongs to one of the clusters C_k . Within a given cluster C_k , a given spectra i is denoted $\mathbf{x}_{ik}$, $i : 1, \dots, |C_k|$. $|C_k|$ is the total number of spectra in a given cluster C_k .

Ideally, different clusters should be distant and spectra within a same cluster should be close.

The clustering algorithm used for the evaluation of NST methods was the K-Means algorithm with Euclidean distance as the measure of distance. This algorithm clusters spectra into one of K groups, where K is a number input by the user. Here K was set to be equal to the number of subjects in each data set ($K = 3$ for the urine data, $K = 4$ for the serum data).

The performance measures used to evaluate the quality of clustering and classification were the Dunn Index (DI), Davies-Bouldin Index (DBI), the Rand Index (RI) and the Adjusted Rand Index (ARI). These four measures are explained in Sections 3.3.2 - 3.3.3. (For additional information, please refer to Féraud et al. (2015).)

Dunn Index (DI)

The Dunn Index is a measure of the quality of a given clustering partition, and it should be maximized. This index is maximized when the smallest distance between spectra belonging to two different clusters is large and the largest intra-cluster distance is small. For more information concerning the calculation of the Dunn Index, refer to the Annex (Section 7.2.2).

Davies-Bouldin Index (DBI)

The Davies-Bouldin Index is also a measure of the quality of a given clustering partition but, in contrast to the Dunn Index, should be minimized. This index is minimized when clusters that are close to each other have little intra-cluster variation. For more information concerning the calculation of the Davies-Bouldin Index, refer to the Annex (Section 7.2.2).

3.3.3 Classification

Rand Index (RI)

The Rand Index (RI) is used to evaluate the quality of classification and is related to accuracy. Suppose that $\mathbf{C}^1 = \{C_1^1, \dots, C_{K_1}^1\}$ is a partition of a data set created during unsupervised clustering (the number of classes and the class label for each spectrum is unknown) and $\mathbf{C}^2 = \{C_1^2, \dots, C_{K_2}^2\}$ is the true partition of the data set according to the known classes.

The Rand Index takes values between 0 and 1 and should be maximized. A Rand Index of 1 indicates perfect agreement between the unsupervised partition $\mathbf{C}^1$ and the true partition $\mathbf{C}^2$.

For more information concerning the calculation of the Rand Index, refer to the Annex (Section 7.2.2).

Adjusted Rand Index (ARI)

Two partitions can be in agreement for a given pair of spectra by chance, optimistically biasing the Rand Index. An alternative version of the Rand Index, the Adjusted Rand Index (ARI), was created to correct for chance grouping of spectra. Like the Rand Index, this index is used to evaluate the quality of classification and should be maximized.

The ARI is calculated as follows:

$$ARI_{C^1, C^2} = \frac{Index - ExpectedIndex}{MaxIndex - ExpectedIndex} \quad (8)$$

where *Index* is the Rand Index.

More details concerning the Adjusted Rand Index can be found in Santos et al. (2009).

Partial Least Squares Discriminatory Analysis (PLS-DA)

Partial Least Squares (PLS) is a type of modeling that characterizes the relationship between a matrix of explanatory variables and a matrix of response variables. PLS is similar to PCA in the sense that it reduces the number of dimensions of the original matrix of explanatory variables $\mathbf{X}$. For PLS, only the remaining dimensions (called “latent” variables (LVs)) are used to build the model.

The total number of latent variables used must be input by the user based on her visual evaluation of cross-validation results. For this study, the most optimal number of latent variables for each NST method was selected based on visual inspection of graphs of the coefficient of determination (R^2) and the root mean square error of prediction ($RMSEP$) for various numbers of latent variables. The number of latent variables selected was the smallest number for which the R^2 reached a plateau for all subject classes and for which the $RMSEP$ remained low.

The quality of classification using a PLS model on normalized, scaled and/or transformed data was summarized by two performance measures: the multiple coefficient of determination of the model used to classify (R^2) and the cross-validated R^2 (Q^2). The best NST method should produce a maximally high Q^2 (close to 1) and a maximally high R^2 (close to 1). In order to rank methods according to these two criteria, the product of R^2 and Q^2 was calculated for each method. Methods with a high $R^2 \times Q^2$ value have high values for both measures at once.

For more information concerning PLS and the calculation of the R^2 and the Q^2 , refer to the Annex (Section 7.2.2).

3.4 Results

In this section, the performance of various NST methods applied to the urine and serum data sets will be analyzed. Results for performance measures related to variance structure (Section 3.4.1), clustering (Section 3.4.2) and classification (Section 3.4.3) will be presented in the form of graphs and tables.

3.4.1 Variance structure

Urine data

For the urine data set (see Table 2), PQN (with and without the pre-normalization step), Linear Baseline Normalization (with and without pre-normalization) and CSN dramatically increased the between-group inertia compared to non-normalization. They also maintained a variance structure that can be largely explained by the first two principal components. Indeed, these methods yielded a percentage of cumulative variance explained by the first two principal components that was close to the cumulative variance for non-normalization (90.2 %, which was the highest). Thus, compared to non-normalization, these methods improved the variance structure of the data without increasing complexity. Of these 5 methods, the top three in terms of the inertia measures were Linear Baseline Normalization (without pre-normalization), PQN (without pre-normalization) and CSN.

Table 2: Urine data

Method	BI	WI	% cum var (2 PC)	Dunn	DB	Rand	AdjRand
None	62.89	37.11	90.2	0.87	0.65	0.7	0.34
CSN	89.03	10.97	89.4	0.91	0.62	1	1
Pareto	59.85	40.15	82.5	0.77	0.71	0.7	0.34
Autoscaling	50.91	49.09	76	0.67	0.77	0.7	0.34
QN	86.67	13.33	86.8	0.99	0.64	1	1
PQN	88.35	11.65	89	0.91	0.57	1	1
PQN (Hochrein)	87.81	12.19	90.5	0.34	0.8	0.71	0.41
Linear Baseline	89.64	10.36	90.1	1.01	0.59	1	1
Linear Baseline (Hochrein)	87.09	12.91	90.2	0.38	0.86	0.71	0.41
VSN (Hochrein)	77.79	22.21	82.9	0.3	0.77	0.71	0.41

Serum data

The best method for the serum data was PQN combined with the pre-normalization step (labeled “PQN (Hochrein)”), which had higher between-group inertia than PQN alone (see Table 3). Linear Baseline with pre-normalization was second best for between-group inertia, also performing better than Linear Baseline alone. The difference in performance with and without the pre-normalization step was fairly dramatic for Linear Baseline Normalization: between-group inertia without pre-normalization was 73.57 compared to 90.18 with pre-normalization. This dramatic improvement can be seen in Figures 20 and 21, where the different subjects (coded with different colors) overlap in the first figure (without pre-normalization) and are clearly separated in the second figure (with pre-normalization).

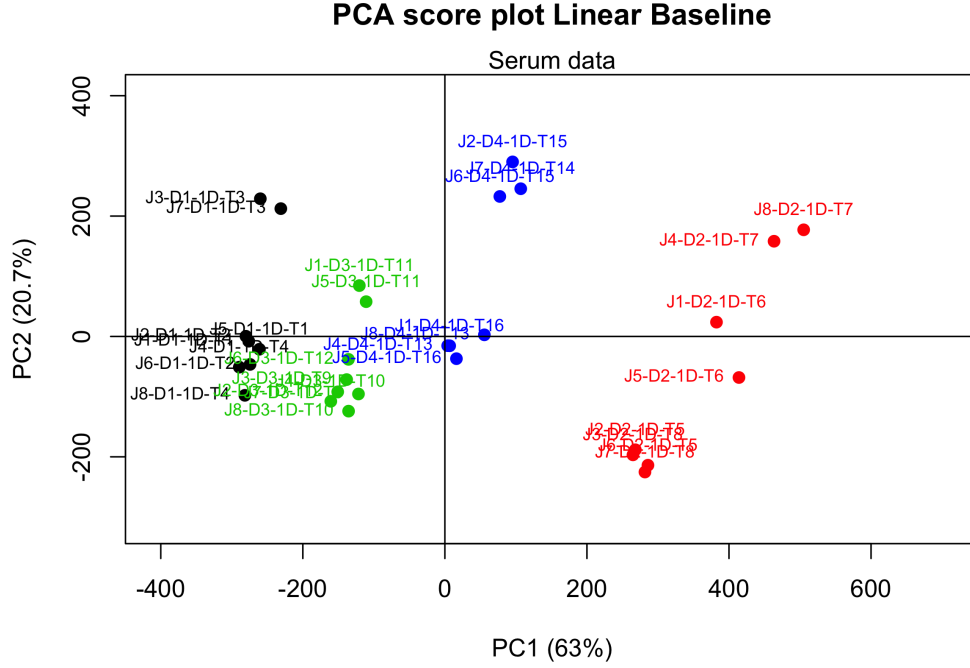


Figure 20: Linear Baseline without Hochrein et al. (2015)’s pre-normalization step

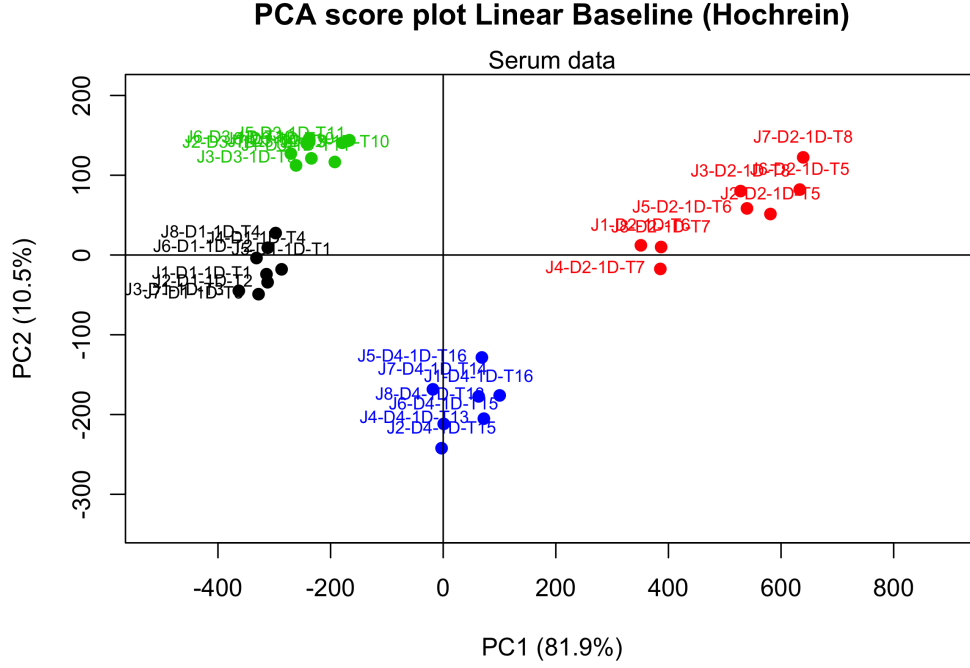


Figure 21: Linear Baseline with Hochrein et al. (2015)’s pre-normalization step

PQN without the pre-normalization step ranked third in terms of between-group inertia, suggesting that PQN can perform well with and without the pre-normalization step.

Table 3: Serum data

Method	BI	WI	% cum var (3 PC)	Dunn	DB	Rand	AdjRand
None	65	35	95.9	0.42	0.91	0.75	0.4
CSN	88.25	11.75	93.3	0.18	1.06	0.85	0.62
Pareto	60.07	39.93	90.9	0.26	1.48	0.82	0.51
Autoscaling	31.9	68.1	71.7	0.58	1.42	0.7	0.29
QN	82.06	17.94	84.8	0.6	0.9	1	1
PQN	88.79	11.21	93.8	0.48	0.86	0.84	0.6
PQN (Hochrein)	90.93	9.07	95.5	0.45	0.7	1	1
Linear Baseline	73.57	26.43	93.7	0.59	0.93	0.77	0.43
Linear Baseline (Hochrein)	90.18	9.82	95.4	0.5	0.69	1	1

Both data sets

For both data sets, Pareto and Autoscaling worsened the variance structure of the data: they increased within-group inertia and decreased the percent

of cumulative variance explained by the first two or three principal components.

PCA scores plots for all methods (Principal Components 1-4) can be found in the Annex (Section 7.2.3, Figures 58 - 93).

3.4.2 Clustering

Urine data

For the urine data, the best NST methods for improving clustering quality were CSN, PQN and Linear Baseline Normalization. These methods were in the top three for both the Dunn Index and the Davies-Bouldin Index (see Table 2).

Serum data

For the serum data there was much less agreement between the two indexes, making it difficult to assess which methods ranked best (see Table 3).

3.4.3 Classification

Urine data

For the urine data set, the best methods were CSN, QN, PQN and Linear Baseline, which all had Rand and Adjusted Rand Indexes of 1 (the maximum) (see Table 2). The other methods, including all methods tested with the pre-normalization step, yielded much weaker measures of classification performance.

For the PLS-DA classification, these same four NST methods (CSN, QN, PQN and Linear Baseline) were consistently ranked in the top four (see Table 4). These results seem to validate the Rand and Adjusted Rand Indexes calculated for these methods.

Table 4: PLS-DA for the urine data

Method	# LVs	Subject 1			Subject 2			Subject 3		
		R2	Q2	R2xQ2	R2	Q2	R2xQ2	R2	Q2	R2xQ2
None	3	0.892	0.818	0.729	0.9	0.814	0.733	0.917	0.877	0.804
CSN	2	0.998	0.997	0.995	0.996	0.994	0.99	0.998	0.997	0.995
Pareto	3	0.894	0.851	0.761	0.902	0.855	0.771	0.917	0.832	0.763
Autoscaling	3	0.91	0.789	0.718	0.906	0.859	0.778	0.929	0.863	0.802
QN	2	0.999	0.996	0.995	0.998	0.997	0.995	1	0.999	0.999
PQN	2	0.992	0.99	0.982	0.993	0.989	0.983	0.998	0.996	0.994
PQN (Hochrein)	2	0.961	0.95	0.912	0.943	0.926	0.873	0.989	0.983	0.972
Linear Baseline	2	0.995	0.992	0.987	0.995	0.992	0.987	0.997	0.996	0.993
Linear Baseline (Hochrein)	2	0.953	0.933	0.89	0.93	0.897	0.834	0.986	0.981	0.967
VSN (Hochrein)	2	0.993	0.978	0.971	0.952	0.915	0.871	0.971	0.963	0.935

Serum data

For the serum data set, the methods that most clearly improved classification were QN, PQN (with the pre-normalization step) and Linear Baseline (with the pre-normalization step). These methods had a Rand and Adjusted Rand index of 1 (see Table 3). QN, which did not dramatically improve the variance structure, ranked among the top three methods for classification, suggesting that maximizing between-group inertia is not necessarily a pre-requisite to good classification results.

In terms of the PLS-DA results, non-normalization was worst or second worst for all subjects. PQN was ranked as first or second for all subjects, suggesting that it was the best overall method for that data set. This finding is not in total accordance with the Rand and Adjusted Rand indexes, for which PQN was only the 5th best method. The Rand and Adjusted Rand indexes seem to favor PQN with the pre-normalization step over PQN without pre-normalization, whereas PQN without pre-normalization is the best method for PLS-DA.

Table 5: PLS-DA for the serum data

Method	# LVs	Subject 1			Subject 2			Subject 3			Subject 4		
		R2	Q2	R2xQ2	R2	Q2	R2xQ2	R2	Q2	R2xQ2	R2	Q2	R2xQ2
None	4	0.876	0.769	0.674	0.963	0.915	0.882	0.919	0.828	0.76	0.923	0.846	0.781
CSN	3	0.925	0.844	0.781	0.957	0.89	0.852	0.919	0.825	0.758	0.941	0.873	0.821
Pareto	4	0.938	0.869	0.815	0.976	0.942	0.919	0.981	0.928	0.91	0.951	0.857	0.815
Autoscaling	4	0.907	0.769	0.697	0.973	0.907	0.882	0.977	0.871	0.851	0.938	0.804	0.755
QN	3	0.974	0.955	0.93	0.964	0.943	0.91	0.968	0.955	0.924	0.955	0.918	0.876
PQN	4	0.979	0.937	0.917	0.989	0.961	0.95	0.972	0.94	0.914	0.969	0.914	0.886
PQN (Hochrein)	3	0.929	0.859	0.798	0.978	0.947	0.926	0.921	0.884	0.814	0.972	0.92	0.894
Linear Baseline	4	0.976	0.938	0.915	0.973	0.947	0.921	0.94	0.894	0.84	0.962	0.937	0.902
Linear Baseline (Hochrein)	3	0.922	0.845	0.779	0.969	0.949	0.92	0.915	0.874	0.8	0.958	0.902	0.865

3.5 Conclusion

Overall, PQN seems to be a reasonable choice for both the urine and serum data sets, giving it the advantage of wide applicability.

The methods applied to the serum data set in particular clearly benefited from Hochrein et al. (2015)’s pre-normalization step while the methods applied to the urine data set did not. PQN and Linear Baseline (both with the pre-normalization step) performed well for the serum data set. For all evaluation criteria, with the exception of those associated with PLS-DA measures, these methods had the best performance. QN could also be a useful method. While it did not yield favorable results for measures related to variance structure, it was one of the best methods for classification (according to the Rand and Adjusted Rand indexes).

For the urine data set in particular, CSN, PQN, Linear Baseline Normalization and QN performed well with respect to all criteria, with the exception of QN, which was only particularly strong for classification. The fact that CSN and PQN performed well on the urine data set is in accordance with Wang et al. (2013), where CSN and PQN were found to improve reproducibility when applied to urine samples similar to those used in this study (Linear Baseline Normalization and QN were not tested).

4 Baseline correction methods in the literature

The true baseline of NMR spectra is often distorted and offset from zero. Baseline distortion can result from a variety of technical imperfections: corruption of the first few data points in the free induction decay (FID) spectrum, “dead time” between NMR pulsations, lack of linearity in the filter-phase response, equipment instabilities and poor suppression of macromolecules such as water or proteins (Golotvin and Williams, 2000; Chang et al., 2007; Xi and Rocke, 2008; Wang et al., 2012). The Fourier transform could also contribute to baseline distortion and offset due to its discrete nature. Some baseline distortion can be avoided by adjusting parameters during the acquisition of NMR data (see Tang (1994) for more details).

One important characteristic of NMR spectra is that peaks often overlap. This makes the estimation of the baseline especially difficult. For example, in Figure 22, the true baseline (in green) would be difficult to identify because overlapping peaks make the baseline seem higher than it really is.

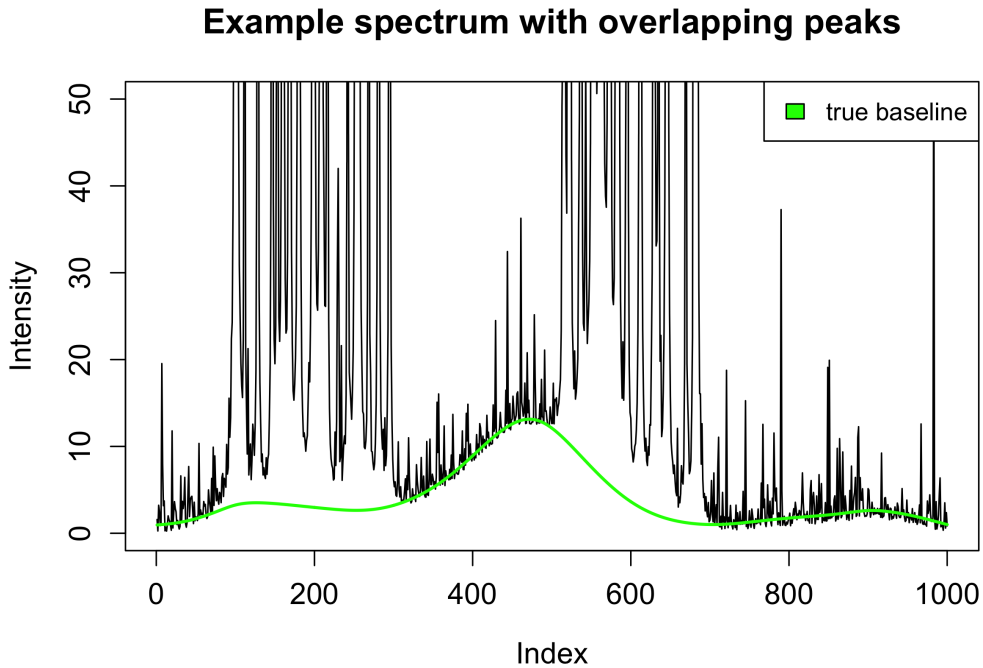


Figure 22: Simulated spectrum with overlapping peaks

After the acquisition of NMR data, a baseline correction method must be applied in order to straighten the baseline and bring it back to zero. Let x_{ij} be the spectral intensity of feature j in spectrum i , with $i : 1, \dots, n$ and $j : 1, \dots, p$. For all methods described here, the baseline corrected spectrum $\mathbf{x}_i^*$ is calculated by subtracting the estimated baseline $\hat{\mathbf{z}}_i$ from the spectrum $\mathbf{x}_i$.

$$\mathbf{x}_i^* = \mathbf{x}_i - \hat{\mathbf{z}}_i$$

In Figure 23, an estimated baseline $\hat{\mathbf{z}}_i$ is depicted in the left panel and is then subtracted from the spectrum $\mathbf{x}_i$ to yield the baseline corrected spectrum $\mathbf{x}_i^*$ in the right panel.

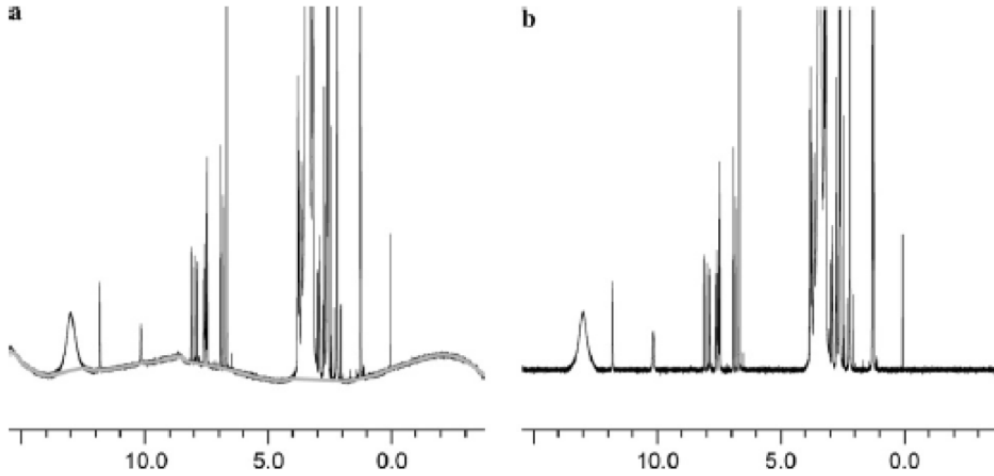


Figure 23: NMR spectrum before and after baseline correction (Cobas et al., 2006)

Baseline correction is preceded by zero order phase correction and followed by negative value zeroing in **SOAP** (see Figure 2). Two major categories of baseline correction methods for NMR data are presented in the literature and will be described in this chapter: iterative penalized least squares methods and noise and signal classification methods. These methods are applied to spectra in the frequency domain.

4.1 Iterative penalized least squares methods

Iterative penalized least squares methods are non-parametric methods that estimate a baseline using weights that are updated iteratively. The theoretical motivation as well as the algorithm for three iterative penalized least squares methods (AsLS, airPLS and arPLS) will be provided in this section. These three methods are the methods tested in Chapter 5. A fourth algorithm (PSM) is presented in lesser detail because it seems to have the worst performance of the four iterative penalized least squares methods.

4.1.1 Asymmetric Least Squares (AsLS)

Theoretical motivation

Eilers and Boelens (2005) introduced an iterative baseline correction method based on a penalized least squares function (see Equation 9) with two terms: a least squares term and a Whittaker roughness penalty. The least squares term represents the overall difference between the candidate baseline $\mathbf{z}_i$ and the original spectrum $\mathbf{x}_i$. The Whittaker roughness term penalizes large jumps in consecutive features. The penalized least squares function is given by

$$S(\mathbf{z}_i) = \sum_{j=1}^p (x_{ij} - z_{ij})^2 + \lambda \sum_{j=1}^p (\Delta^2 z_{ij})^2 \quad (9)$$

where $\mathbf{x}_i$ is the spectrum to be corrected, $\mathbf{z}_i$ is a candidate baseline for $\mathbf{x}_i$, λ is a roughness penalty and the second-order difference $\Delta^2 z_{ij}$ is defined as

$$\begin{aligned} \Delta^2 z_{ij} &= (z_{ij} - z_{i,j-1}) - (z_{i,j-1} - z_{i,j-2}) \\ &= z_{ij} - 2z_{i,j-1} + z_{i,j-2}. \end{aligned} \quad (10)$$

When Equation 9 is minimized, the least squares term tends to bring the candidate baseline as close as possible to the original spectrum while the roughness term prevents the least squares term from pulling the candidate baseline up into the peaks.

The parameter λ represents the trade-off between the two terms. If λ is large, more importance is given to the minimization of the roughness term (the candidate baseline is pulled down, see Figure 24 in blue), whereas small values of λ give priority to the minimization of the least squares term (the candidate baseline is pulled up, see Figure 24 in red).

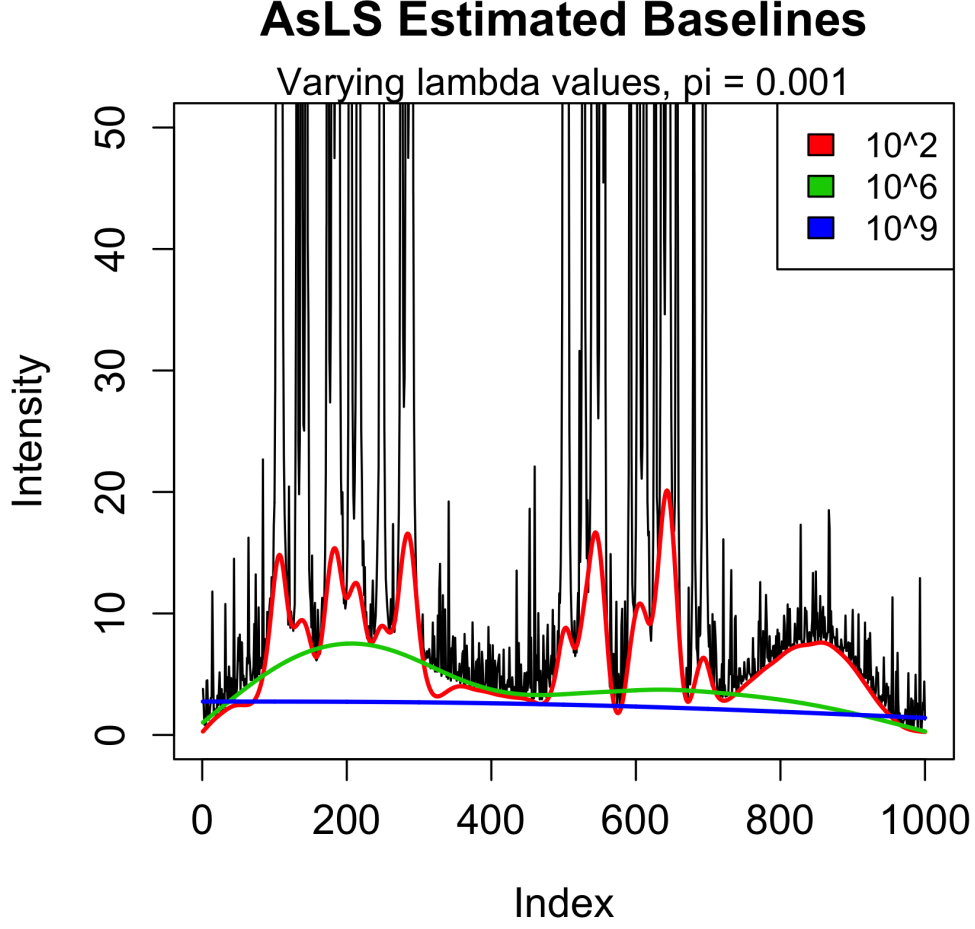


Figure 24: AsLS baseline estimation for different λ values

In AsLS, Equation 9 is generalized to include a vector of weights $\mathbf{w}_i$ which specifies a weight for each feature j of spectrum i (the definition of these weights will be explained in the “Algorithm” section). The addition of these weights make the first term an asymmetric least squares term.

$$S(\mathbf{z}_i) = \sum_{j=1}^p w_{ij} (x_{ij} - z_{ij})^2 + \lambda \sum_{j=1}^p (\Delta^2 z_{ij})^2 \quad (11)$$

In matrix notation, this is represented as follows:

$$S(\mathbf{z}_i) = (\mathbf{x}_i - \mathbf{z}_i)^\top W_i (\mathbf{x}_i - \mathbf{z}_i) + \lambda \mathbf{z}_i^\top D^\top D \mathbf{z}_i$$

where the matrix D is given by

$$D = \begin{pmatrix} 1 & -2 & 1 & 0 & \dots & 0 & 0 & 0 \\ 0 & 1 & -2 & 1 & \dots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & \dots & 1 & -2 & 1 \end{pmatrix}$$

and the matrix of weights W_i is given by

$$W_i = \mathbf{w}_i I_p = \begin{pmatrix} w_{i1} & 0 & 0 & 0 \\ 0 & w_{i2} & 0 & 0 \\ 0 & 0 & \ddots & \vdots \\ 0 & 0 & \dots & w_{ip} \end{pmatrix}.$$

An estimator for the candidate baseline is found by deriving this equation relative to the candidate baseline $\mathbf{z}_i$, setting it to zero and isolating $\mathbf{z}_i$.

$$\frac{\partial S_i}{\partial \mathbf{z}_i^\top} = -2W_i (\mathbf{x}_i - \mathbf{z}_i) + 2\lambda D^\top D \mathbf{z}_i = 0,$$

$$\hat{\mathbf{z}}_i = (W_i + \lambda D^\top D)^{-1} W_i \mathbf{x}_i.$$

In AsLS, the weights for the current iteration t are defined as

$$w_{ij}^t = \begin{cases} \pi & \text{where } x_{ij} > \hat{z}_{ij}^{t-1} \\ 1 - \pi & \text{where } x_{ij} \leq \hat{z}_{ij}^{t-1} \end{cases}.$$

A small value of π is generally used so that large weight is given to features of the candidate baseline that are greater than or equal to the features of the spectrum $\mathbf{x}_i$. If certain features in the candidate baseline are greater than the corresponding features in the spectrum, these features will be negative after baseline correction, which is undesirable. Large weights are thus used to discourage this. See Figure 25 for a demonstration of the effect of different π values.

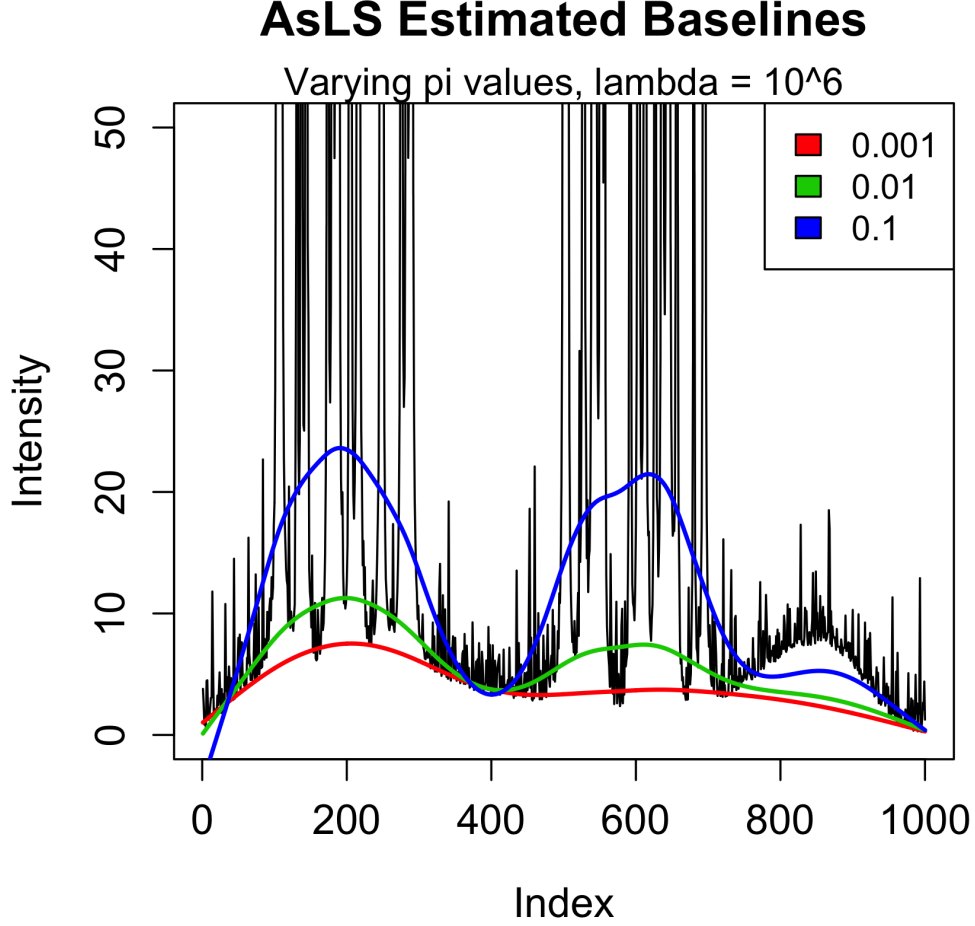


Figure 25: AsLS baseline estimation for different π values

Algorithm

The weights as defined above are updated at each iteration. For a given spectrum, the weights for the current iteration t depend on the candidate baseline from the last iteration, $\hat{\mathbf{z}}^{t-1}$. If, at the previous iteration, the candidate baseline at feature j was greater than or equal to feature j in the spectrum ($z_j^{t-1} \geq x_j$), the current iteration will give a greater weight to this feature in the least squares term ($w^t = 1 - \pi$, where π is small). As a result, the minimization of $x_j - \hat{z}_j$ will be given greater importance.

The iterative procedure is summarized below:

Algorithm 1: AsLS

Data: a single spectrum $\mathbf{x}$

Result: estimated baseline $\hat{\mathbf{z}}$ for that spectrum

```

1  $p = \text{length}(\mathbf{x})$ 
2  $\mathbf{E} = \text{diag}(p)$  // Create diagonal matrix of length  $p$ 
3  $\mathbf{D} = \text{diff}(\mathbf{E}, \text{differences}=2)$  // Create matrix  $\mathbf{D}$  of second-order differences
4  $\mathbf{w}^t = (1, \dots, 1)$ 
5 while  $\sum_{j=1}^p |w_j^t - w_j^{t-1}| \neq 0$  do
6    $\mathbf{W}^{t-1} = \text{diag}(\mathbf{w}^t)$  // Create diagonal matrix whose diagonal is  $\mathbf{w}^t$ 
7    $\hat{\mathbf{z}}^t = (\mathbf{W}^{t-1} + \lambda \mathbf{D}^\top \mathbf{D})^{-1} \mathbf{W}^{t-1} \mathbf{x}$  // Estimate candidate baseline
8    $\mathbf{w}^t = \pi \mathbb{I}(\mathbf{x} > \hat{\mathbf{z}}^t) + (1 - \pi) \mathbb{I}(\mathbf{x} < \hat{\mathbf{z}}^t)$  // Update weights
9 end
10 return  $\hat{\mathbf{z}}^t$ 

```

A threshold can also be set for the maximum number of iterations before the algorithm is stopped. Eilers and Boelens (2005) argue that a threshold of 10 iterations is sufficient.

4.1.2 Adaptive Iteratively Reweighted Penalized Least Squares (airPLS)

Theoretical motivation

Adaptive iteratively reweighted penalized least squares (airPLS) was proposed by Zhang et al. (2010) and is different from AsLS only in terms of the weight function. The weight function in airPLS gives increasingly greater weight to features below the candidate baseline as the number of iterations increases. Note that the parameter π used for AsLS is not necessary for airPLS.

Let $\mathbf{d}_i^{t-1} = \mathbf{x}_i - \hat{\mathbf{z}}_i^{t-1}$ be the vector of difference between the original spectrum and the baseline proposed at the end of the previous iteration, iteration $t - 1$. Positive or null features in $\mathbf{d}_i^{t-1}$ correspond to features in $\mathbf{x}_i$ with intensities greater than or equal to the candidate baseline ($x_{ij} \geq z_j$), whereas negative features in $\mathbf{d}_i^{t-1}$ correspond to features below the candidate baseline ($x_{ij} < z_j$). A second vector, $\tilde{\mathbf{d}}_i^{t-1}$, is created by setting all positive features in $\mathbf{d}_i^{t-1}$ to zero. The resulting vector contains negative values where features in $\mathbf{x}_i$ are below the candidate baseline and zeroes everywhere else.

The weights for the current iteration t are calculated as follows:

$$w_{ij}^t = \begin{cases} 0 & \text{where } x_{ij} \geq \hat{z}_{ij}^{t-1} \\ \exp \left(t \left[|x_{ij} - \hat{z}_{ij}^{t-1}| / \sum_{j=1}^p |\tilde{d}_{ij}^{t-1}| \right] \right) & \text{where } x_{ij} < \hat{z}_{ij}^{t-1} \end{cases} \quad (12)$$

Weights for features below the candidate baseline $x_{ij} < \hat{z}_{ij}^{t-1}$

In the second line of Equation 12, the numerator in the term in brackets represents the absolute difference between a given feature j in $\mathbf{x}_i$ and the corresponding feature in the candidate baseline from the previous iteration $\hat{\mathbf{z}}^{t-1}$. The greater the difference, the more weight that feature will receive in the least squares term for the current iteration. This weight is normalized by the sum of the absolute value of all negative differences (denominator).

The term in brackets is multiplied by the iteration number, meaning that the weight increases with each new iteration for all features below the candidate baseline - regardless of their distance from the candidate baseline. However, if a particular feature is much farther below the baseline than the others, this weight is further magnified (term in brackets).

Weights for features above the candidate baseline ($x_{ij} \geq \hat{z}_{ij}^{t-1}$)

If a feature j is greater than the corresponding feature in the candidate baseline at the previous iteration $(t - 1)$, the weight for the current iteration t will be set to zero (see Equation 12). This means that, for the current iteration t , feature j in the estimated candidate baseline will be a linear interpolation between the closest weighted features before and after feature j (“weighted” meaning that the weight is non-null). The value of the closest weighted features before and after feature j will thus determine the value of feature j in the estimated baseline. These features are in turn dependent on the weight function in the second line of Equation 12.

In Figure 26, the first four iterations of airPLS are shown.

airPLS

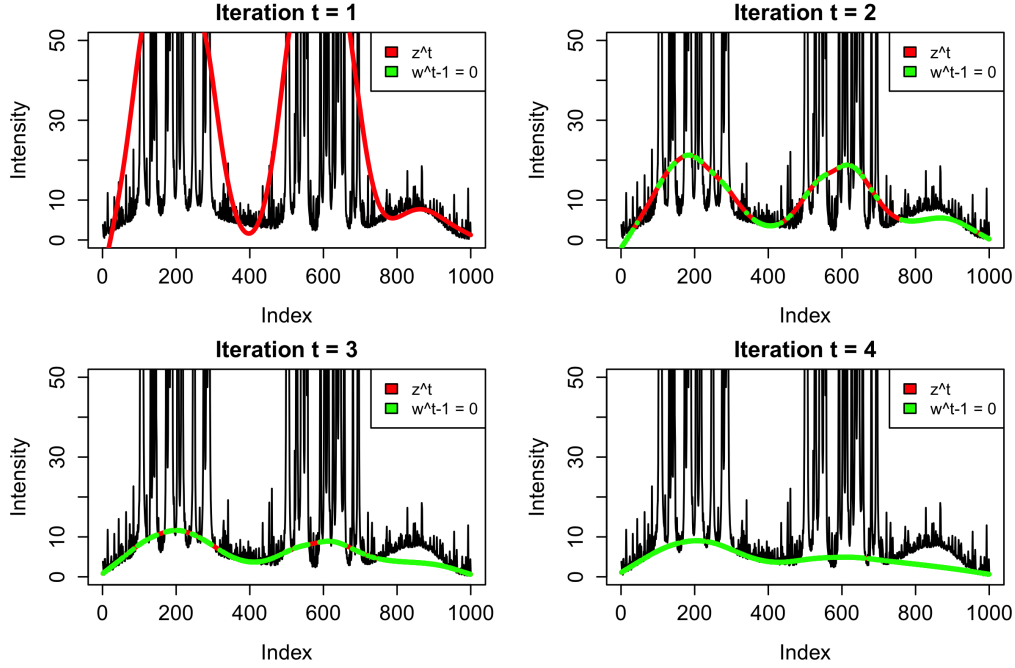


Figure 26: Example of the first 4 iterations of airPLS (applied to a simulated spectrum)

Algorithm

The algorithm for airPLS is the same as for AsLS except that the convergence criterion and the weights are defined differently.

Line 8 in the AsLS algorithm (see Algorithm 1) is replaced with:

```

1  $\mathbf{d}^{t-1} = \mathbf{x} - \hat{\mathbf{z}}^{t-1}$ 
2 for  $j \in 1, \dots, p$  do
3    $\tilde{d}_j^{t-1} = d_j^{t-1} \mathbb{I}(x_j < \hat{z}_j^{t-1})$ 
4 end
5 for  $j \in 1, \dots, p$  do
6    $w_j^t = \exp \left( t \left[ |x_j - \hat{z}_j^{t-1}| / \sum_{j=1}^p |\tilde{d}_j^{t-1}| \right] \right) \mathbb{I}(x_j < \hat{z}_j^{t-1})$ 
7 end
8  $\mathbf{d}^t = \mathbf{x} - \hat{\mathbf{z}}^t$ 

```

The while condition in line 5 is replaced with:

$$\sum_{j=1}^p |d_j^t| \geq 0.001 \sum_{j=1}^p |x_j|$$

As is the case for AsLS, a maximum threshold for the number of iterations can also be used to stop the looping process.

In Figure 27, the baseline estimated by airPLS is shown for different λ values (second panel). The first panel represents the baseline estimated by AsLS for the same λ values and is shown here for the purpose of comparison (the first panel is identical to Figure 24). Both methods seem to yield similar results for the same λ values.

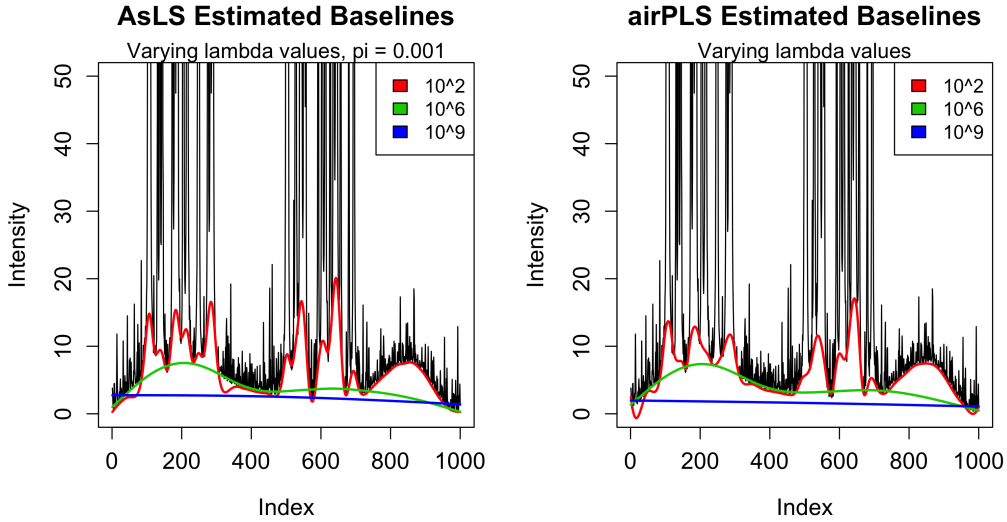


Figure 27: AsLS and airPLS baseline estimation for different λ values

4.1.3 Asymmetrically Reweighted Penalized Least Squares (arPLS)

Theoretical motivation

According to Baek et al. (2015), both AsLS and airPLS tend to underestimate the baseline in noisy regions, resulting in artificial curvature in noise regions (see Figure 28) and potentially overestimated peak intensities (see Figure 29). This underestimation of the baseline results from the fact that, in noisy regions, strong weights (close to 1) are given to any features below the baseline, whereas little to no weight is applied to noisy features above

the baseline. Indeed, for AsLS, all features above the candidate baseline are given a very small weight (π) and for airPLS, all features above the candidate baseline are given a weight of zero. Because the baseline in noisy regions is pulled down by strong penalties, more features end up above the baseline than below it. (This tendency was also observed in Liu et al. (2014a) and Liu et al. (2014b).) In reality, the noise should be randomly distributed both above and below the baseline.

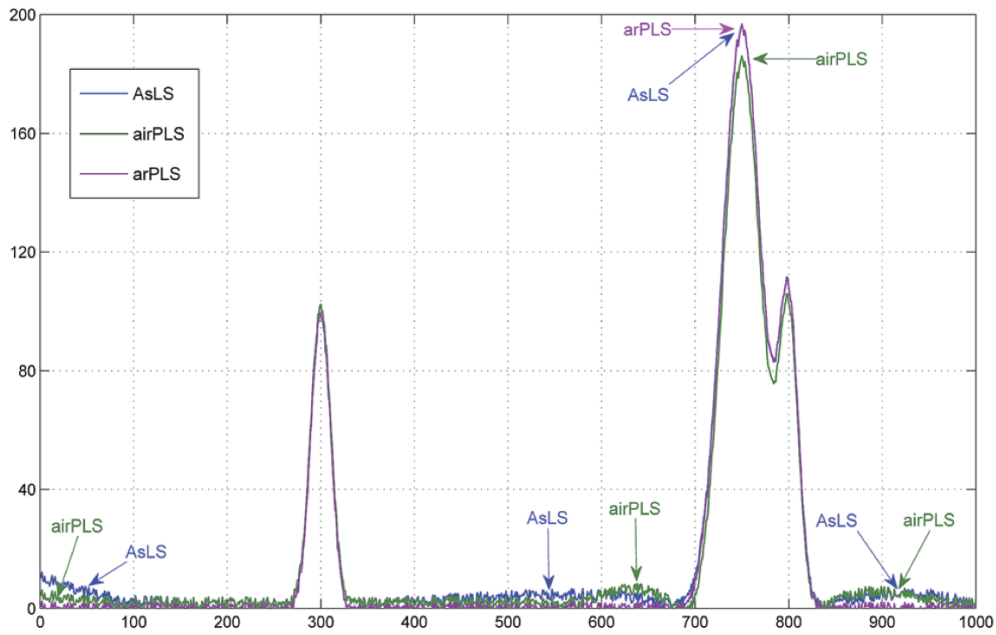


Figure 28: Baseline corrected spectra in low noise (Baek et al., 2015)

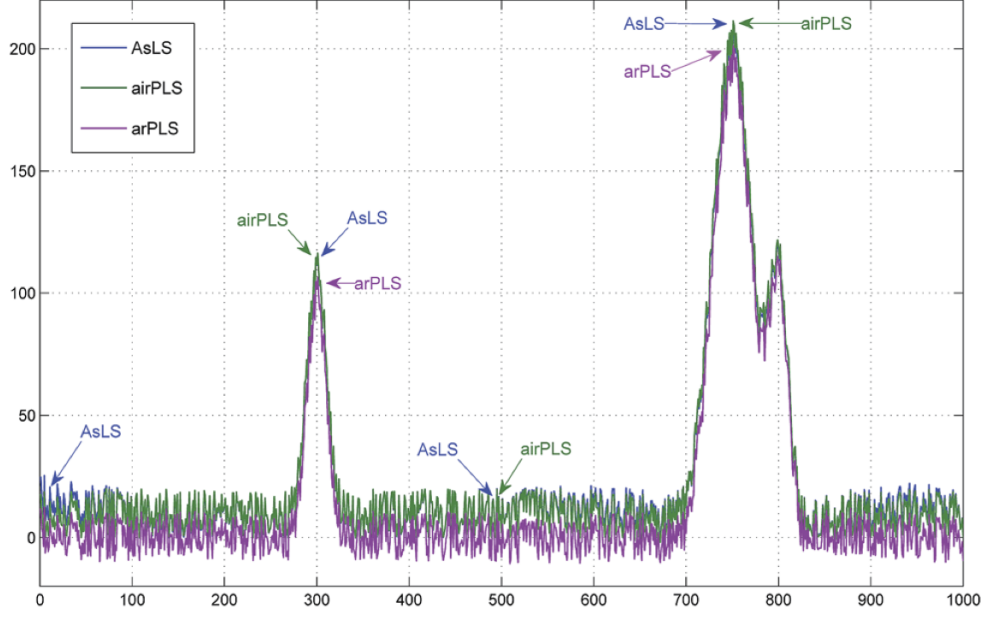


Figure 29: Baseline corrected spectra in high noise (Baek et al., 2015)

Asymmetrically reweighted penalized least squares (arPLS) is a method proposed by Baek et al. (2015) that improves the weighting scheme for AsLS and airPLS. Weights for a given feature are only set to 0 if the feature's intensity is much greater than the baseline intensity. All features within a certain range of the candidate baseline receive similar weights, meaning that, in noisy regions, the weights for features below the baseline (which pull the candidate baseline down) are counterbalanced by the weights for features above the baseline (which pull the candidate baseline up). Thus, the proposed weighting scheme allows features in noisy regions to be evenly distributed around the baseline.

The function used to assign weights is the logistic function, which gradually decreases the weight given to features as signal increases.

Let $\mathbf{d}_i^{*t-1}$ be a subset of $\mathbf{d}_i^{t-1}$ including only those features in $\mathbf{d}_i^{t-1}$ that are negative. $m_{\mathbf{d}_i^*}^{t-1}$ is the sample mean of the subset $\mathbf{d}_i^{*t-1}$ and $S_{\mathbf{d}_i^*}^{t-1}$ is the sample standard deviation. The weight function is defined as

$$w_{ij}^t = \begin{cases} \text{logistic}\left(d_{ij}^{t-1}, m_{\mathbf{d}_i^*}^{t-1}, S_{\mathbf{d}_i^*}^{t-1}\right) & \text{where } x_{ij} \geq \hat{z}_{ij}^{t-1} \\ 1 & \text{where } x_{ij} < \hat{z}_{ij}^{t-1} \end{cases}$$

where the logistic function is defined as

$$\text{logistic}(d_{ij}, m_{\mathbf{d}_i^*}, S_{\mathbf{d}_i^*}) = \frac{1}{1 + \exp\left(\frac{2(d_{ij} - (-m_{\mathbf{d}_i^*} + 2S_{\mathbf{d}_i^*}))}{S_{\mathbf{d}_i^*}}\right)}$$

In Figure 30, weight is plotted as a function of the difference d between a given feature in the original spectrum and the corresponding feature in the candidate baseline. The mean of all negative differences ($m_{\mathbf{d}_i^*}$, noted m in the figure) determines the range of values d for which the weight function is about equal to one. For example, if the mean $m = -3$, all features below the candidate baseline and all features up to $-m = 3$ units above the candidate baseline receive a weight of approximately one. The standard deviation of all negative differences ($s_{\mathbf{d}_i^*}$, noted s in the figure) affects the slope of the weight function for differences $d > -m$. For small values S (left panel), the drop to zero weight is faster than for larger values of S (right panel).

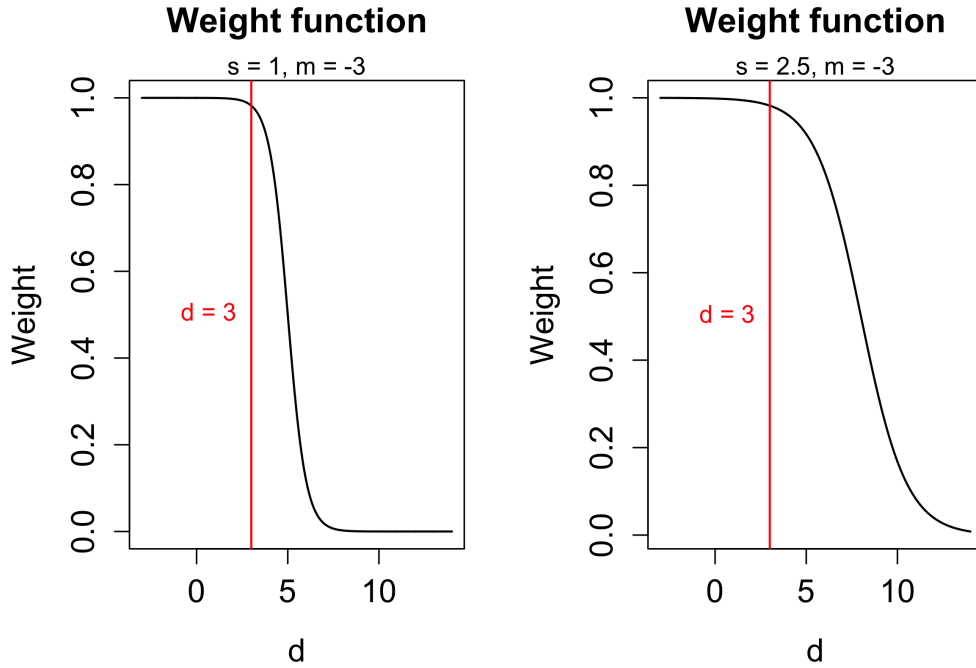


Figure 30: Logistic weight function for arPLS

Algorithm

The algorithm for arPLS is the same as for AsLS and airPLS except that the convergence criterion and the weights are defined differently.

Line 8 in the AsLS algorithm is replaced with:

```

1  $\mathbf{d}^{t-1} = \mathbf{x} - \hat{\mathbf{z}}^{t-1}$ 
2 for  $j \in 1, \dots, p$  do
3   if  $d_j^{t-1} < 0$  then
4      $\mathbf{d}^{*t-1} = \mathbf{d}^{*t-1} \cup d_j^{t-1}$ 
5   end
6 end
7  $m_{\mathbf{d}^*}^{t-1} = \text{mean}(\mathbf{d}^{*t-1})$ 
8  $S_{\mathbf{d}^*}^{t-1} = \text{sd}(\mathbf{d}^{*t-1})$ 
9 for  $j \in 1, \dots, p$  do
10   $w_j^t = \text{logistic}(d_j^{t-1}, m_{\mathbf{d}^*}^{t-1}, S_{\mathbf{d}^*}^{t-1}) \mathbb{I}(x_j \geq \hat{z}_j^{t-1}) + 1\mathbb{I}(x_j < \hat{z}_j^{t-1})$ 
11 end

```

The while condition in line 5 is replaced with:

$$\frac{\|\mathbf{w}^{t-1} - \mathbf{w}^t\|_2}{\|\mathbf{w}^{t-1}\|_2} \geq \text{ratio}$$

where

$$\frac{\|\mathbf{w}^{t-1} - \mathbf{w}^t\|_2}{\|\mathbf{w}^{t-1}\|_2} = \frac{\sqrt{\sum_{j=1}^p (w_j^{t-1} - w_j^t)^2}}{\sqrt{\sum_{j=1}^p (w_j^{t-1})^2}}$$

In their simulation study, (Baek et al., 2015) used a ratio of 10^{-6} .

In Figure 31, the baseline estimated by arPLS is shown for different λ values (third panel). The first and second panels represent the baselines estimated by AsLS and airPLS for the same λ values and are shown here for the purpose of comparison (the first two panels are identical to Figure 27). λ seems to have a different effect on baseline estimation for arPLS than for AsLS and airPLS.

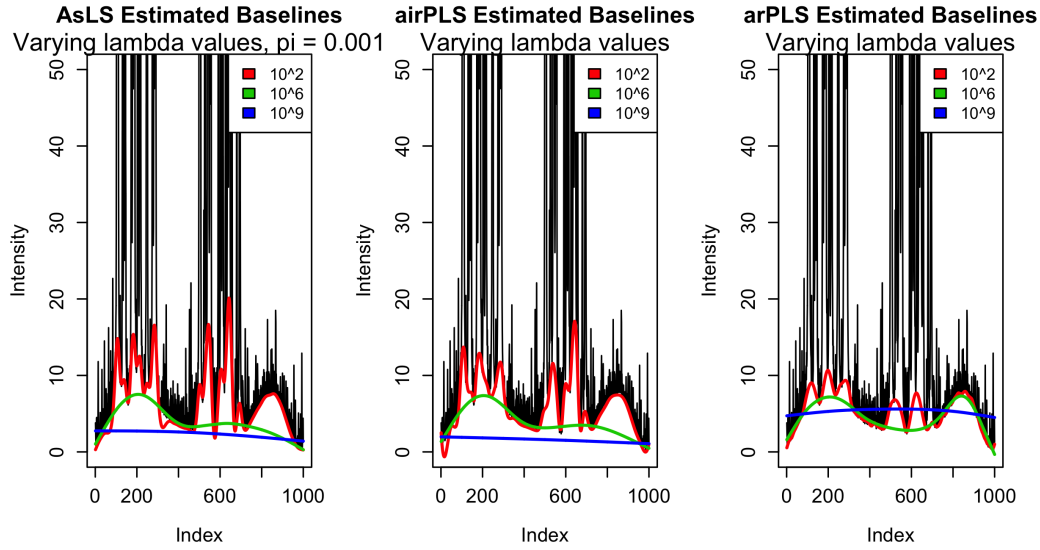


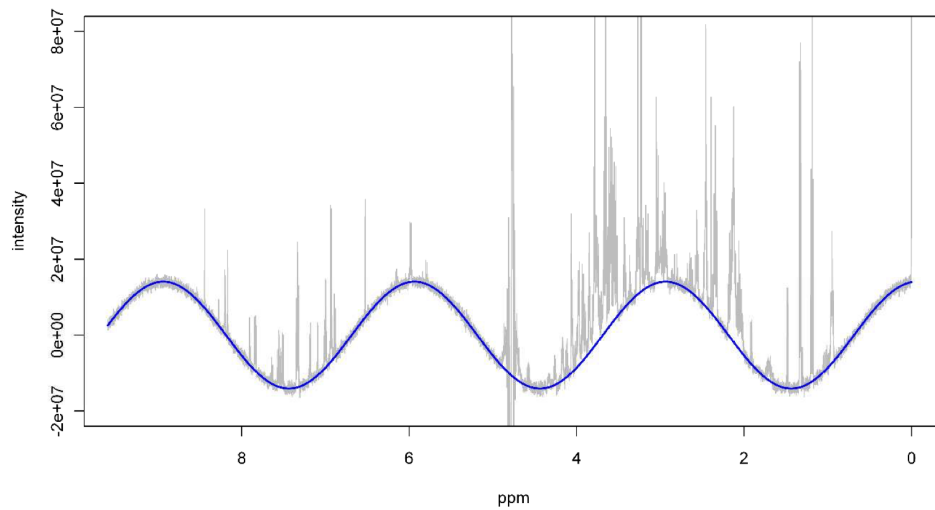
Figure 31: AsLS, airPLS and arPLS baseline estimation for different λ values

4.1.4 Penalized Parametric Smoothing Model (PSM)

Xi and Rocke (2008) proposed a Penalized Parametric Smoothing Model (PSM) method for baseline correction that also depends on the optimization of a score function. The score function includes a smoothness penalty and a negativity penalty, whose relative importance is modulated by two parameters. Unlike the other iterative penalized least squares methods, the parameters in Penalized Parametric Smoothing can be determined automatically.

The disadvantage of this fully automatic method is that the quality of baseline correction has been shown to be worse than other methods. Indeed, Wang et al. (2012) found that the mean percent baseline error for PSM was 24.9%, compared to 5.2% for the BaselineCorrector method (see 4.2.5 for an explanation of the BaselineCorrector method). A visual inspection of the baseline estimated by PSM also suggests that PSM is less effective than the BaselineCorrector algorithm (see Figures 32 and 33).

(a) original sine-wave baseline



(b) baseline estimated by BaselineCorrector

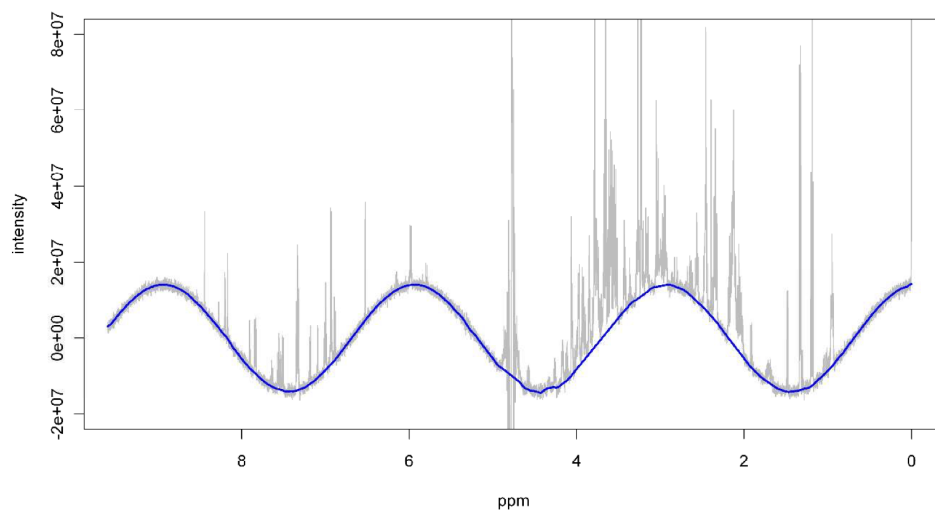


Figure 32: Known baseline and baseline estimated by BaselineCorrector (Wang et al., 2012)

(e) baseline estimated by PSM

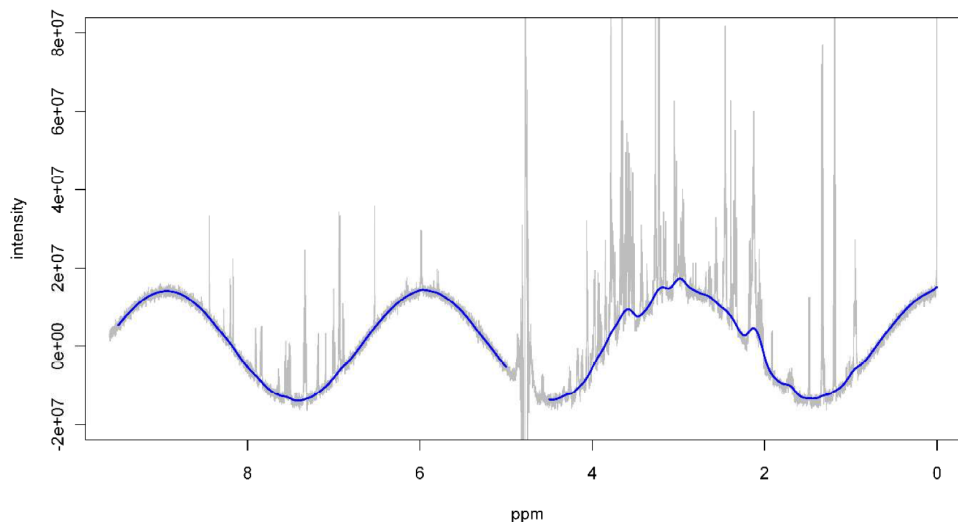


Figure 33: Baseline estimated by PSM (Wang et al., 2012)

4.2 Noise and signal classification methods

Various methods of NMR baseline correction proposed in the literature depend on the classification of features as noise or signal before the baseline is estimated (Dietrich et al., 1991; Golotvin and Williams, 2000; Cobas et al., 2006; Chang et al., 2007; Wang et al., 2012; Liu et al., 2014a). In general, these methods consist of the following steps:

1. Removal of baseline rolling, a phenomenon where the baseline of a spectrum suffers from gradual sloping. An example of this is shown in Figure 34.
2. Classification of features in the spectrum as noise or signal using on an intensity-based threshold
3. Estimation of the baseline using the noise features only (using smoothing or a regression model)

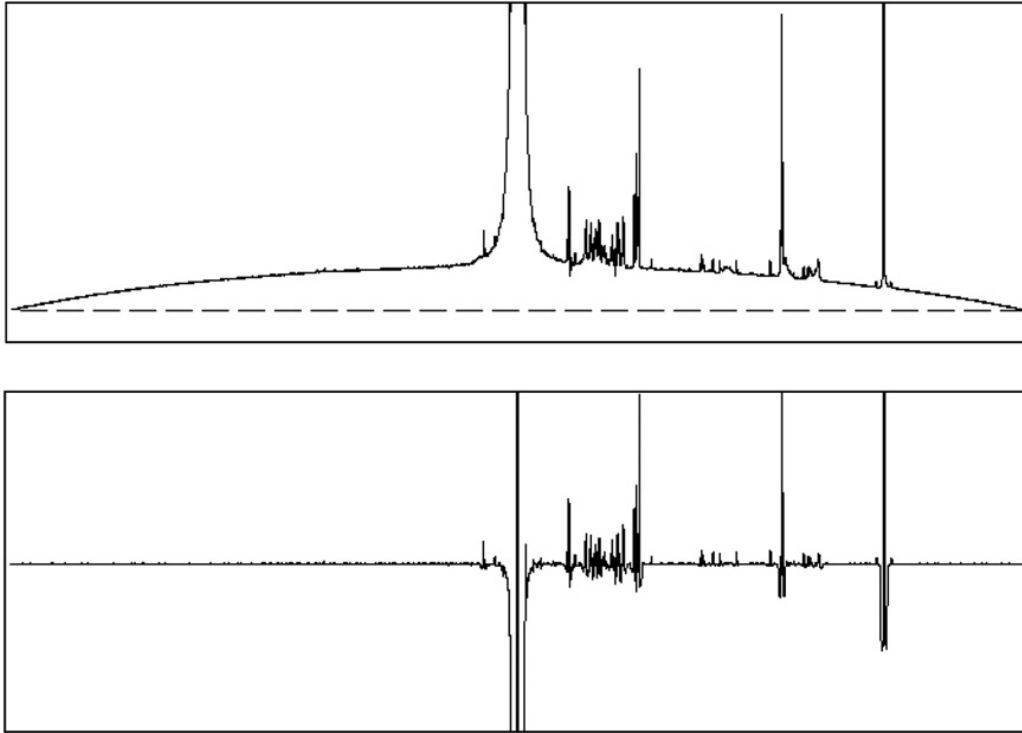


Figure 34: NMR spectrum with baseline rolling (top panel) and after removal of baseline rolling (bottom panel) (Chang et al., 2007)

Before the removal of baseline rolling, the observed intensities of features are biased. This is problematic because noise and signal classification methods depend on intensity-based thresholds for classifying features. If the intensities of features are biased, the classification of features will be less accurate, leading to lower quality baseline estimations.

Note that Golotvin and Williams (2000)’s algorithm, Wang et al. (2012)’s BaselineCorrector algorithm and the Automatic Two-sided Exponential Baseline correction algorithm (ATEB) introduced by Liu et al. (2014a) are different from the other algorithms in that they do not remove baseline rolling before classifying features. In addition, ATEB uses signal features, rather than noise features, to estimate the baseline, smoothing down the peaks in an iterative process.

All methods described here have been applied to NMR data. None of the algorithms from this category were tested for this thesis because of their various weaknesses. For this reason and for the sake of conciseness, the structure of this section will be different than the structure in Section 4.1.

Rather than presenting the theoretical motivations and algorithms for each method, a brief outline (“Description”) and discussion (“Discussion”) of each method will be given.

4.2.1 Automatic Baseline Correction algorithm (ABC)

Description

In ABC (Dietrich et al., 1991), a rolling baseline is removed by taking the first derivative of the spectrum. This derivative is approximated by replacing each feature with the difference between that feature and the next one. Let x'_{ij} , be the differentiated feature j in spectrum i . Then,

$$x'_{ij} = x_{ij} - x_{ij+1}$$

In practice, the derivative can also be calculated by replacing each feature with the difference between that feature and the previous one (this is the more common approach).

$$x'_{ij} = x_{ij} - x_{ij-1}$$

Dietrich et al. (1991) do not provide any justification for their particular calculation of the derivative.

As a result, gradual increases and decreases in the baseline are reduced or eliminated, bringing the baseline down near zero. This type of approximation tends to induce extra noise (Cobas et al., 2006), so this step is preceded by a smoothing of the entire spectrum (using a moving average filter, for example).

Noise features are identified by iteration: the sample mean $\bar{x}'$ and standard deviation S' of the differentiated spectrum is estimated and features with intensities greater than $\bar{x}' + 3S'$ are removed (labeled as signal). This process is repeated until no new features are excluded, the remaining features being classified as noise features. Additionally, if a signal feature is between two noise features, it is reclassified as noise, and if a noise feature is between two signal features, it is reclassified as signal.

A fifth degree polynomial is then fit to only those features in the original spectrum that were classified as noise. This polynomial fit serves as the estimated baseline $\hat{z}$.

Discussion

The use of a fifth degree polynomial for fitting a baseline seems inappropriate considering that the baselines of NMR spectra generally do not take the shape of a polynomial (Cobas et al., 2006). Because the polynomial is fit to the noise features only, the estimated baseline in noise-sparse regions could also be highly distorted.

Lastly, the initial smoothing step depends on a user-defined window length that can affect the quality of the baseline correction.

4.2.2 Fully Automatic Baseline Correction algorithm (FABC)

Description

The features in FABC (Cobas et al., 2006) are classified in exactly the same way as in the ABC algorithm. The differences between the two algorithms lie in the first and last steps listed in Section 4.2.

The approximation of the first derivative is completed using a Continuous Wavelet Transform, which makes it possible to approximate the derivative without introducing additional noise. Because the preliminary smoothing step is no longer necessary, the user-defined parameter in ABC is not necessary for FABC. However, a scaling parameter is introduced in the Continuous Wavelet Transform to modulate the signal to noise ratio, so the actual number of user-defined parameters remains the same.

In the last step, the baseline is estimated by weighted penalized least squares (using the same general function defined in Equation 11 from Section 4.1.1). Note that while the identification of noise and signal features is done through iteration (as was the case for ABC), the penalized least squares term is not updated iteratively. The weights w_{ij} are only defined once (see the paragraph below for more details).

The score function in Equation 11 (reprinted below for convenience) is minimized to estimate the candidate baseline.

$$S(\mathbf{z}_i) = \sum_{j=1}^p w_{ij} (x_{ij} - z_{ij})^2 + \lambda \sum_{j=1}^p (\Delta^2 z_{ij})^2$$

The first term is a weighted least squares term. For FABC, the weight w_{ij} is set to 0 for signal features and 1 for noise features, meaning that the baseline

for signal features is a linear interpolation between the closest noise feature before and after the signal feature. The parameter λ allows the user to define the trade-off between roughness and bias. The baseline that minimizes this function is the estimated baseline $\hat{\mathbf{z}}_i$.

Discussion

The two user-defined parameters in FABC are subject to bias in the sense that they must be determined through visual inspection of the resulting baseline. The performance of this method is thus variable because it depends on user experience and evaluation.

In reality, it is likely that a baseline that is curvy in noise regions is also curvy in signal regions. While FABC is able to estimate curved baselines in noise regions, it is limited to linear interpolation in signal regions. Thus, it fails to capture the entire behavior of the baseline.

For example, in the simulated spectrum in Figure 35, the first signal region has very little baseline distortion (true baseline marked in green), so linear interpolation (in red) comes very close to the true baseline. However, when the baseline distortion is more pronounced, such as in the second signal region, linear interpolation does not match the true baseline as well.

Linear interpolation in signal regions

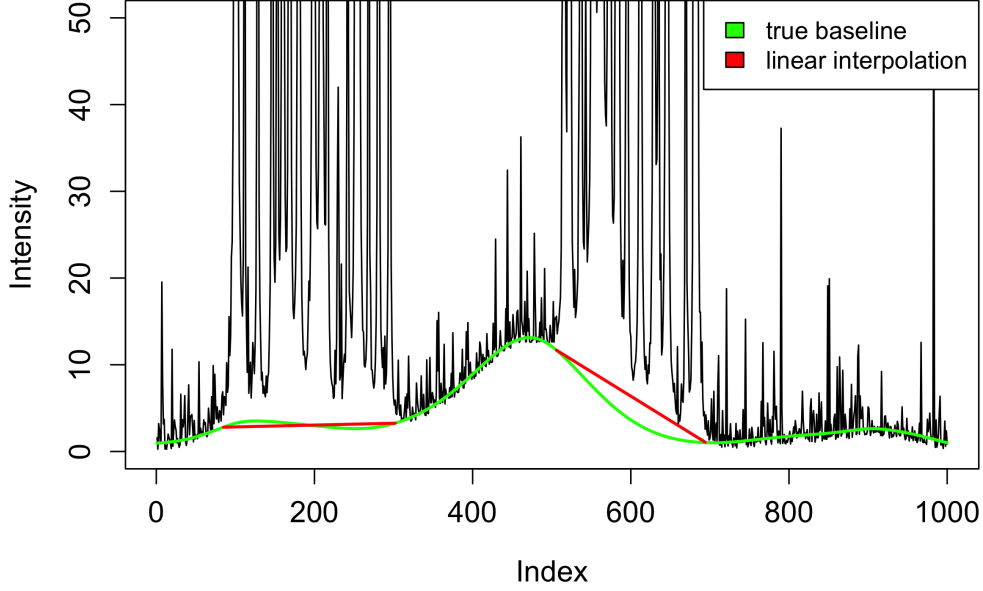


Figure 35: Comparison of the true baseline and linear interpolation of the baseline in signal regions (simulated spectrum)

4.2.3 Chenomx algorithm

Description

The Chenomx algorithm (Chang et al., 2007) first reduces baseline rolling using a high pass filter that removes low frequency parts of the signal. The resulting spectrum $\tilde{x}_i$ is then divided into equal-length segments, the size of which is determined by the user. A user-defined percentage of segments with the lowest frequencies are pooled, forming a sample of noise features. The standard deviation of this sample is estimated, representing the standard deviation of noise, S_{noise} .

A given feature j is then classified as signal if the absolute value of the feature is greater than two times the standard deviation of noise:

$$|\tilde{x}_{ij}| > 2 \times S_{noise} \Rightarrow \tilde{x}_{ij} \in \text{signal}$$

Several minor adjustments are then made to the classifications so that signal

features are not isolated or incorrectly classified as noise. The estimated baseline in areas with noise features is equal to the intensity of the noise features, meaning that, after baseline correction, noise features have zero intensity. The baseline for signal features is a linear interpolation between the closest noise feature before and after the signal. If desired, a more complex fit using splines can be applied.

Discussion

While baseline distortion is effectively reduced in the first step of this algorithm, the performance of the algorithm still depends on the choice of segment window length. What's more, the algorithm depends on one user-defined parameter and several default parameters that might not be well adapted to all types of NMR spectra.

4.2.4 Golotvin and Williams (2000)'s algorithm

Description

This algorithm begins by estimating the standard deviation of noise and then using this standard deviation to classify features. Each spectrum is divided into $L = 32$ segments of length p (the authors give no motivation for the proposed value of L). A given segment l of spectrum i is written $\mathbf{x}_{il} = (x_{il1}, \dots, x_{ilp})$. For each segment $\mathbf{x}_{il}$, the sample standard deviation, S_{il} , is estimated and stored in a vector $\mathbf{S}_i$. The estimated standard deviation of the noise, S_{noise} , is then defined as the smallest standard deviation in $\mathbf{S}_i$.

Each feature x_{ij} is classified based on the intensities of features in a window around it. Let the window centered around x_{ij} be given by $\mathbf{w}_{ij} = (x_{ij-K}, \dots, x_{ij}, \dots, x_{ij+K})$, where K is a user-defined parameter for the window length.

Features are considered to be noise if the difference between the maximum and minimum intensity in their corresponding window is less than the standard deviation of noise times some factor c :

$$\max(x_{ij-K}, \dots, x_{ij+K}) - \min(x_{ij-K}, \dots, x_{ij+K}) < c \times S_{noise} \Rightarrow x_{ij} \in noise$$

The baseline $\hat{\mathbf{z}}_i$ is estimated by smoothing the noise features and connecting baseline segments with linear interpolation.

Discussion

This method of classification is potentially problematic because there is no preliminary reduction of baseline distortion. Suppose that an observed feature x_{ij} is composed of its true intensity μ_{ij} and some baseline offset β_{ij} :

$$x_{ij} = \mu_{ij} + \beta_{ij}$$

If the true intensity of each feature in a given window is small but the baseline offset β_{ij} of each feature is highly variable (heavy baseline distortion), the observed intensities of each x_{ij} will be artificially variable. As a result, it is more likely that the difference between the maximum and minimum intensity in that window be large, possibly resulting in the miss-classification of noise features as signal features. This possibility is even more likely if the window is too large. Thus, the extent of baseline distortion and the width K of the window $\mathbf{w}_{ij}$ could impact the quality of feature classification.

In Figure 36 for example, the mound between feature 800 and 1000 is composed entirely of noise features but could be confused as a signal region because of the heavy baseline distortion.

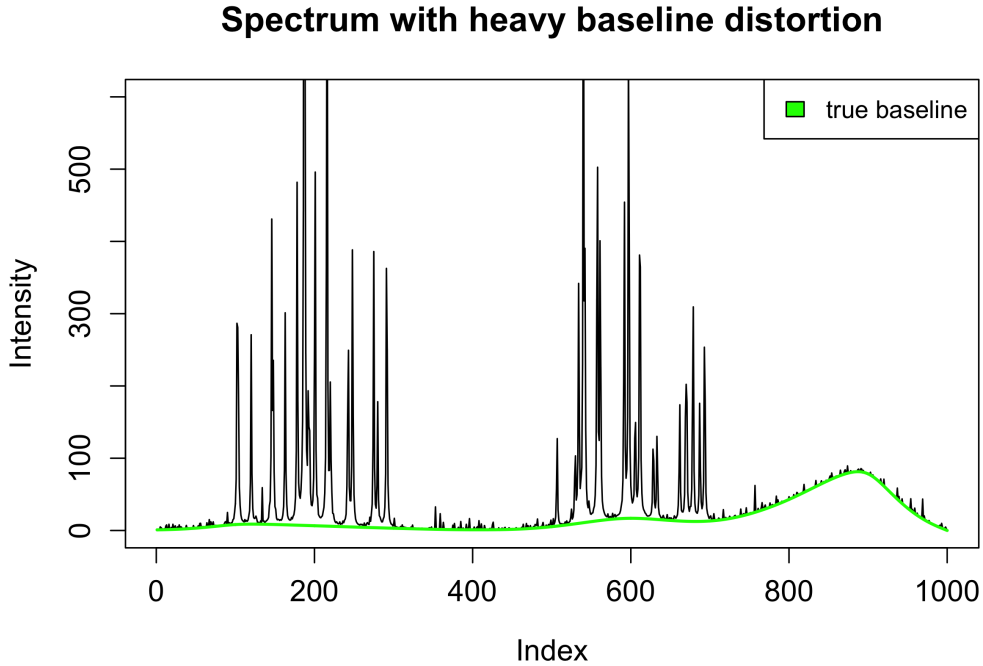


Figure 36: Simulated spectrum with a heavily distorted baseline

The fact that the number of segments L is fixed at 32 could make this algo-

rithm less widely applicable. The number of features p in an NMR spectrum can vary considerably, meaning that the widths of the 32 segments depend on p . In light of this, it seems more appropriate to fix the width of the segments rather than fixing the number of segments.

As with FABC (see Section 4.2.2), linear interpolation is likely an oversimplification of the baseline and could induce bias. Another drawback is that this algorithm relies on three different user-defined parameters, meaning that baseline correction results depend on user experience and, to some extent, guess-work.

4.2.5 BaselineCorrector

Description

As with Golotvin and Williams (2000)’s algorithm, BaselineCorrector (Wang et al., 2012) does not perform any preliminary reduction of baseline rolling. Instead, the algorithm estimates the standard deviation of noise based on an iterative procedure.

Let $\mathbf{w}_{iw} = (x_{ij}, \dots, x_{ij+2K+1})$ be a window of features, where $w : 1, \dots, W$ is the window index and K is the half-width of the window (Wang et al. (2012) suggest that K be set to 20). The first window, $\mathbf{w}_{i1}$, includes features $1, \dots, 2K + 1$ from spectrum i . The second window, $\mathbf{w}_{i2}$, is shifted one feature to the right of the first window and includes features $2, \dots, 2K + 1 + 1$ from spectrum i . A total of $W = p - 2K - 1$ windows are created by sliding a window of length $2K + 1$ to the right in this way. The last window, $\mathbf{w}_{iW}$ includes features $p - 2K, \dots, p$ from spectrum i .

First, the standard deviation of features in each sliding window $\mathbf{w}_{iw}$ is estimated, where the standard deviation for window $\mathbf{w}_{iw}$ is noted by S_{iw} . Before beginning the iterative procedure ($t = 0$), all window standard deviations are stored in a vector $\mathbf{S}_i^0 = (S_{i1}, \dots, S_{iW})$. At the first iteration, $t = 1$, the median of the standard deviations, m^1 , is calculated:

$$m^1 = \text{median}_{1 < w < W}(S_{iw}^0)$$

Standard deviations S_{iw} are only kept in $\mathbf{S}_i^1$ if their standard deviation is less than two times m^1 .

More generally, the median standard deviation for a given iteration t is calculated,

$$m^t = \text{median}_{1 < w < W}(S_{iw}^{t-1}),$$

and thus,

$$S_{iw}^t < 2 \times m^t \Rightarrow S_{iw}^t \in \mathbf{S}_i^{t+1}$$

This iterative procedure continues until m^t/m^{t-1} is greater than or equal to some convergence ratio, at which time m^t is kept as the approximate standard deviation of noise, S_{noise} . Wang et al. (2012) do not suggest a specific ratio.

Next, each window $\mathbf{w}_{iw}$ is classified as a noise or signal window based on S_{noise} . All windows with $S_{iw} \leq c \times S_{noise}$ are classified as noise windows, with all other windows being classified as signal windows.

Once the windows have been classified, the features can be classified. The $b = 7$ central features in each signal window are classified as signal. All features in the noise windows are classified as noise, unless they are one of the 7 central features of a signal window.

The original spectrum is modified by replacing features in signal regions with a linear interpolation between the intensity of the closest noise feature before and after the signal region. This modified spectrum is then smoothed using a moving average filter with a window of $M = 121$ features, yielding the estimated baseline $\hat{\mathbf{z}}_i$.

Discussion

As with Golotvin and Williams (2000)'s algorithm, the correct classification of features depends on the baseline distortion of the spectrum and the size of the window used to calculate standard deviations (see Section 4.2.4). Noisy regions with heavy baseline distortion are likely to be classified as signal due to artificially variable intensities. This is especially problematic because signal features are replaced by linear interpolations between the noise features right before and after the signal region. This can lead to unsatisfactory results (see Figure 37).

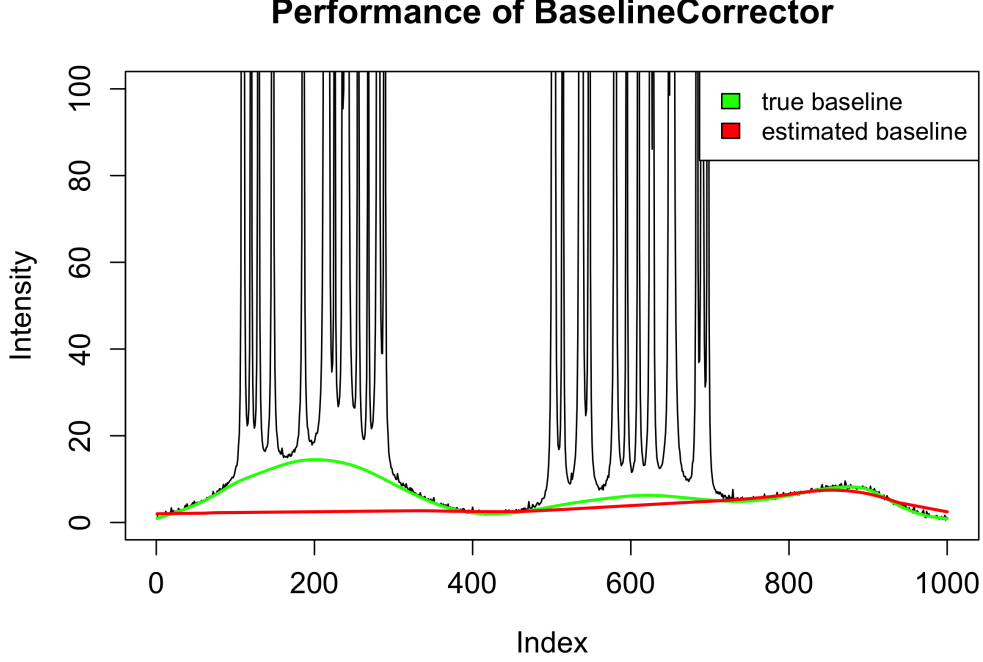


Figure 37: Performance of BaselineCorrector on a simulated spectrum with a heavily distorted baseline

Like the other methods in this category, BaselineCorrector requires the input of many user-defined parameters. While the authors have shown that a window width K between 15 and 20 features and a constant c of 1.1 yields good baseline estimation in a variety of situations, they provide no justification for the choice of the parameters b and M . Indeed, the author’s choice of $M = 121$ could be too large.

4.2.6 Automatic Two-sided Exponential Baseline correction algorithm (ATEB)

In the ATEB algorithm (Liu et al., 2014a), the sample mean $\bar{x}_i$ and sample standard deviation S_i are calculated for each spectrum $\mathbf{x}_i$. A feature j is considered to be noise if its intensity is less than or equal to the sample mean minus three times the standard deviation:

$$x_{ij} \leq \bar{x}_i - 3 \times S_i \Rightarrow x_{ij} \in \text{noise}$$

All features above this threshold are considered to be signal. In contrast to the other algorithms, the features identified as signal are used to determine the baseline. These features are iteratively smoothed down using a two-sided exponential smoother. The baseline is corrected by subtracting the noise features and the smoothed signal features from the original spectrum. As a result, features classified as noise have zero intensity after baseline correction.

Discussion

As is the case for BaselineCorrector (see Section 4.2.5) and Golotvin and Williams (2000)’s algorithm (see Section 4.2.4), this algorithm does not perform a preliminary reduction of baseline distortion, meaning that the quality of the classification of signal and noise features depends on the severity of the baseline distortion in the spectrum.

4.3 Conclusion

While iterative penalized least squares methods depend on all features in the spectrum (some features to a greater extent than others depending on the different weights), noise and signal classification methods depend on only a subset of the features (noise or signal features).

One disadvantage of noise and signal classification methods is that, when estimating the baseline, the baseline distortion in signal regions cannot be taken into account (baselines are estimated based on noise regions only).⁴ Rather than estimating the form of the baseline in signal regions, most methods (FABC, Chenomx, Golotvin and Williams (2000)’s algorithm, BaselineCorrector) perform a linear interpolation between the closest noise feature before and after the signal region, which can be inadequate in cases where there is heavy baseline distortion.

Most methods also rely on several user-defined parameters (ABC, FABC, Chenomx, Golotvin and Williams (2000)’s algorithm, BaselineCorrector), making the performance of the methods dependent on user experience and judgment. Finally, some methods do not remove the rolling baseline before classifying features (Golotvin and Williams (2000)’s algorithm, BaselineCor-

⁴The one exception is ATEB, which depends on signal features rather than noise features. However, because no preliminary reduction of baseline rolling is performed in ATEB, the correct classification of features is sensitive to the degree of baseline distortion in the spectrum.

rector, ATEB), meaning that the quality of classification depends on the extent of baseline distortion present in the spectra.

Iterative penalized least squares methods can yield better results in signal regions with distorted baselines, but depend on at least one user-defined parameter: λ , the trade off between smoothness and fidelity to the original spectrum. Eilers and Boelens (2005) propose a cross-validation approach for selecting parameter values, but this method is not fully automatic. In general, λ is selected by visual validation of the corrected spectra, making the correction of the baseline sensitive to the user’s interpretation of where the true baseline should be. While one method has been proposed that requires no user-defined parameters (PSM), this method has been shown to perform poorly compared to others.

Table 6: Iterative Penalized Least Squares Methods

Name	Abbreviation	Authors	Year
Asymmetric Least Squares	AsLS	Eilers and Boelens	2005
Adaptive Iteratively Reweighted Penalized Least Squares	airPLS	Zhang et al	2010
Assymmetrically Reweighted Penalized Least Squares	arPLS	Baek et al	2015
Penalized Parametric Smoothing Model	PSM	Xi and Rocke	2008

Table 7: Noise and Signal Classification Methods

Name	Abbreviation	Authors	Year
BaselineCorrector		Wang et al	2012
Automatic Baseline Correction algorithm	ABC	Dietrich et al	1991
Fully Automatic Baseline Correction algorithm	FABC	Cobas et al	2006
Golotvin and Williams		Golotvin and Williams	2000
Chenomx algorithm		Chang et al	2007
Automatic Two-sided Exponential Baseline correction algorithm	ATEB	Liu et al	2014

5 Implementation and evaluation of certain baseline correction methods

Based on a preliminary evaluation of the baseline correction methods proposed in the literature (see Chapter 4), the most promising algorithms were selected for further evaluation: AsLS (see Section 4.1.1), airPLS (see Section 4.1.2) and arPLS (see Section 4.1.3). No noise and signal classification methods were tested for several reasons. For one thing, none of the noise and signal classification methods adequately estimate baselines when there is baseline distortion in signal regions. In addition, many methods from this category require several user-defined parameters, making them less generalizable, and some methods poorly classify features when there is significant baseline distortion.

In this chapter, AsLS, airPLS and arPLS will be applied to different simulated data sets and evaluated according to different criteria. First, the methods used for simulating spectra will be described (see Sections 5.1.1, 5.1.2 and 5.1.3), followed by an explanation of the implementation and evaluation process (see Section 5.2). Finally, the results of the simulation study will be presented and analyzed (see Section 5.3).

5.1 Simulation of spectra

5.1.1 Method of simulation

Each simulated spectrum was composed of a vector of noise, $\mathbf{x}^{noise}$, a vector of peaks, $\mathbf{x}^{peak}$, and a vector of curvature, $\mathbf{x}^{curve}$. These three vectors were summed to create the simulated spectrum, $\mathbf{x}$.

$$\mathbf{x} = \mathbf{x}^{peak} + \mathbf{x}^{curve} + \mathbf{x}^{noise}$$

Peaks

The peaks $m : 1, \dots, M$ in each simulated spectrum were created by summing M individual peak vectors of length p . According to Holmes (2013), peaks in NMR spectra have a Lorentzian shape (see Figure 38). Let h be the peak height, w be the peak width (i.e. the width at height $h/2$) and j_0 be the peak center. The j^{th} feature in the m^{th} peak vector can be calculated using

the formula for Lorentzian peaks (Holmes, 2013):

$$x_{j,m}^{peak} = \frac{h_m w_m^2}{w_m^2 + 4(j_{0,m} - j_m)^2}$$

where j is the index of the j^{th} feature. For each peak vector generated, different randomly generated values were used for the peak height h , peak width w and peak center j_0 .

The spectrum for peak m is thus

$$\mathbf{x}_m^{peak} = (x_{1,m}^{peak}, \dots, x_{p,m}^{peak})$$

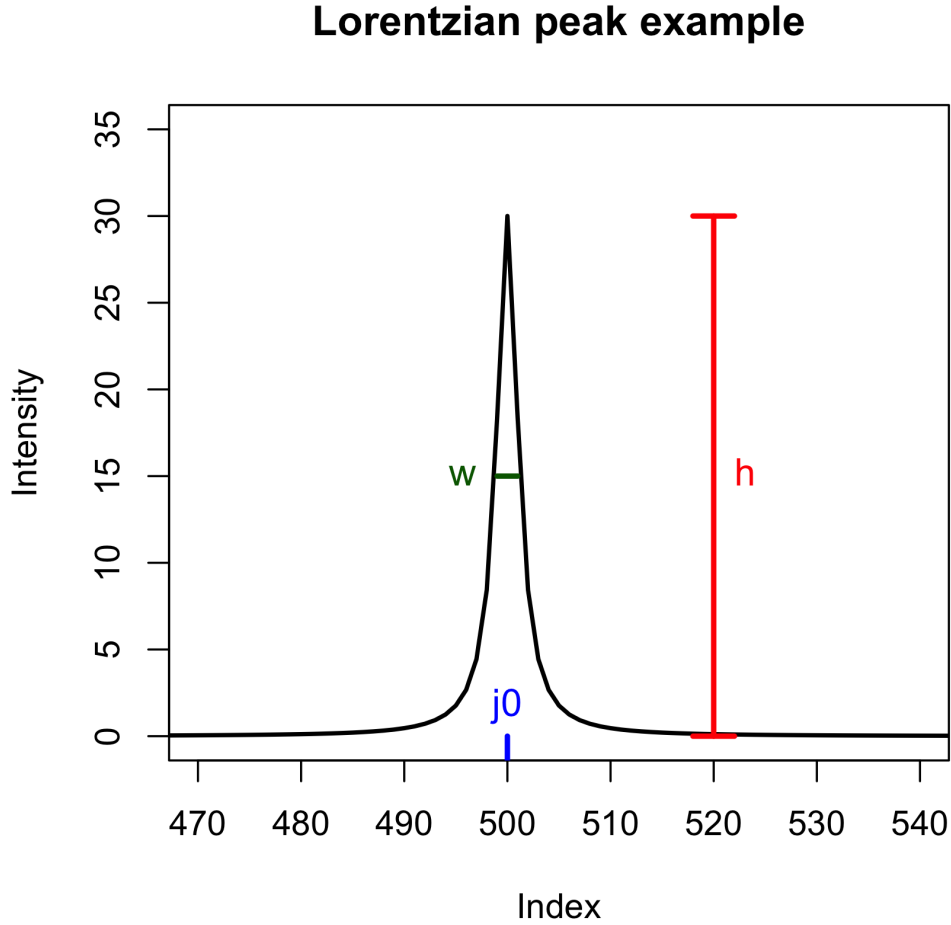


Figure 38: Lorentzian peak example

The peak heights and widths were generated from log-normal distributions and the peak centers were generated from uniform distributions:

$$\begin{aligned} h_1, \dots, h_M &\sim LN(\mu_H, \sigma_H) \\ w_1, \dots, w_M &\sim LN(\mu_W, \sigma_W) \\ j_{0,1}, \dots, j_{0,\frac{M}{2}} &\sim Unif(a_1, b_1) \\ j_{0,(\frac{M}{2}+1)}, \dots, j_{0,M} &\sim Unif(a_2, b_2) \end{aligned}$$

The log-normal distribution was selected for peak heights because, in typical NMR spectra, there are many low-intensity peaks and relatively few high-intensity peaks, meaning that the distribution of feature intensities is skewed to the right (as is the case for the log-normal distribution). The log-normal distribution also has a positive domain, which is advantageous because spectral intensities should be greater than or equal to zero.

Two uniform distributions were used so that there would be two separate regions of peaks in each simulated spectrum.

After generating the peak parameters, M peak vectors $\mathbf{x}_m^{peak}$ were calculated and summed to yield a final peak vector:

$$\mathbf{x}^{peak} = \sum_{m=1}^M \mathbf{x}_m^{peak}$$

An example of a peak vector is shown in Figure 39.

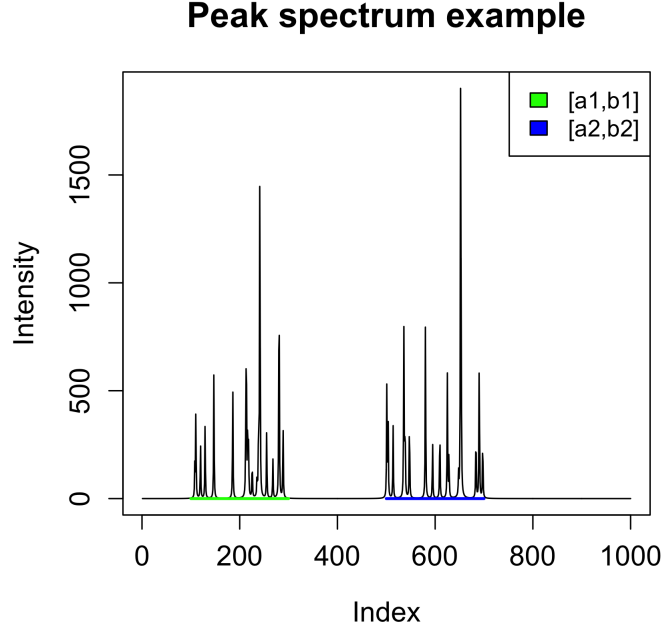


Figure 39: Example of a simulated peak spectrum with two different signal regions

Note that n peak spectra were generated and then used for each curve and noise parameter setting (see Section 5.1.2 for more information concerning the parameter settings). Thus, for each different parameter setting, the set of peak spectra remained the same.

Curve

A curve spectrum was created to simulate a curved baseline. The curve spectrum was generated by smoothing a sequence of p random curve heights c from a log-normal distribution (for the same reasons presented in the “Peaks” section).

$$c_1, \dots, c_p \sim LN(\mu_C, \sigma_C)$$

The smoothing function `smooth.spline` in R was used to smooth the sequence, with the parameter `spar` set to 0.95 (heavy smoothing) (see the first panel of Figure 40 for an example).

The resulting smoothed line $\mathbf{x}$ was shifted down so that the smallest value was set to 0, yielding a vector $\mathbf{x}^{shift}$ (see the second panel of Figure 40). Then the left and right ends of $\mathbf{x}^{shift}$ were adjusted so that the first and last

feature in the spectrum were equal to 0. The first features $j = 1, \dots, 100$ were adjusted as follows:

$$x_j^{curve} = x_j^{shift}(j - 1)\frac{1}{99}$$

The last features $j = 901, \dots, 1000$ were adjusted as follows:

$$x_j^{curve} = x_j^{shift}(1 - (j - 901)\frac{1}{99})$$

Finally, to smooth any potential edges resulting from the adjustment of the first and last features, the smoothing function `smooth.spline` was used again, this time with the `spar` parameter set to 0.5 (see the third panel of Figure 40).

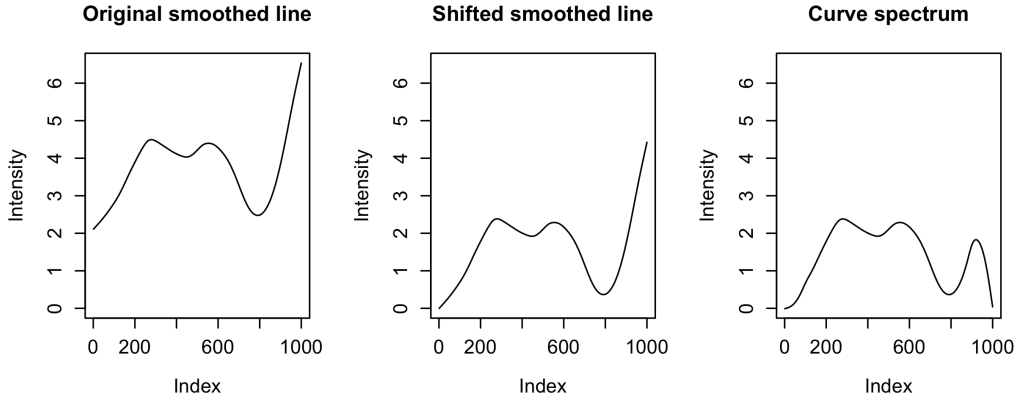


Figure 40: Example simulation of a curve spectrum

Noise

The noise spectrum was created by generating p random numbers from a log-normal distribution.

$$n_1, \dots, n_p \sim LN(\mu_N, \sigma_N)$$

The log-normal distribution was chosen under the assumption that noise in NMR spectra is mostly low intensity, with only a few higher intensity noise features. However, the log-normal distribution is restricted to the positive domain. Because of this, these generated numbers were centered around their expected value so that noise would be distributed above and below 0.

$$x_j^{noise} = n_j - \mu_N$$

$$\mathbf{x}^{noise} = (x_1^{noise}, \dots, x_p^{noise})$$

This was done so that, when adding the noise spectrum $\mathbf{x}^{noise}$ to the curve spectrum $\mathbf{x}^{curve}$, the curve passes directly through the noise (rather than below it). In Figure 41, the left panel shows the baseline position when the noise is not centered, and the right panel shows the baseline position when the noise is centered.

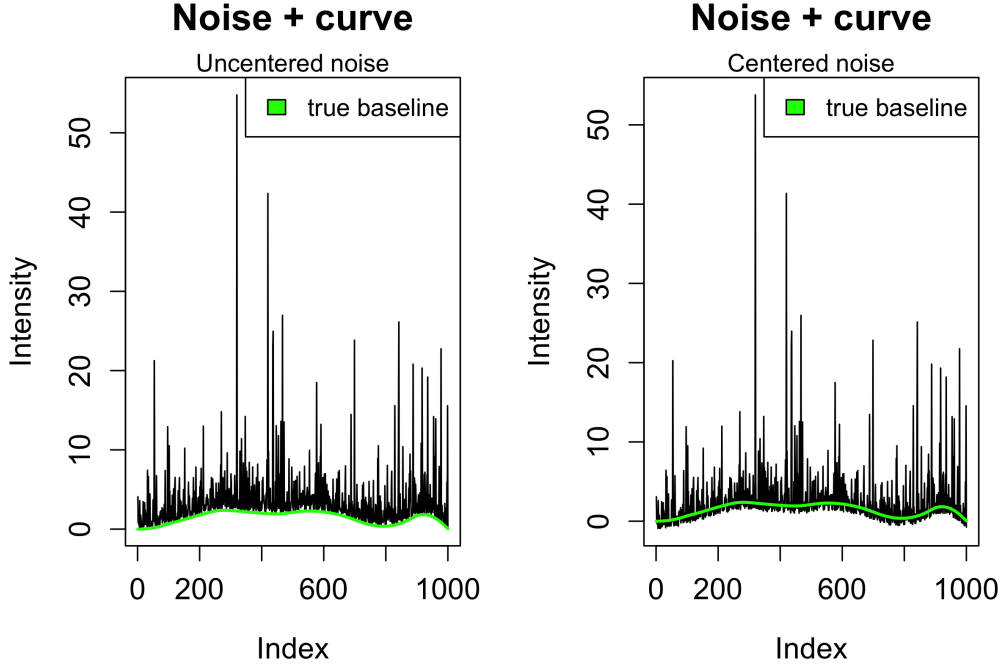


Figure 41: Summed noise and curve spectra with and without noise centering

An example of a noise spectrum is shown in Figure 42.

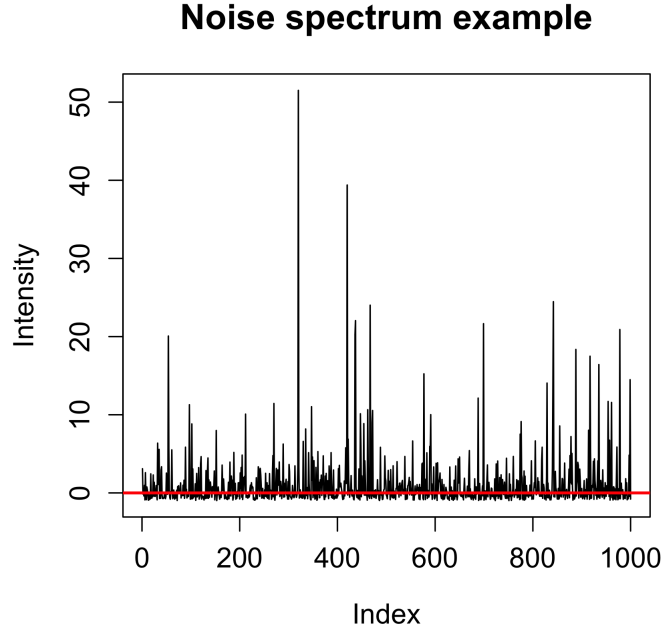


Figure 42: Example of a simulated noise spectrum

Final spectrum

The final spectrum $\mathbf{x}$ was created by summing the peak vector $\mathbf{x}^{peak}$, the curve vector $\mathbf{x}^{curve}$ and the noise vector $\mathbf{x}^{noise}$:

$$\mathbf{x} = \mathbf{x}^{peak} + \mathbf{x}^{curve} + \mathbf{x}^{noise}$$

Figure 43 represents an example of a simulated spectrum. The true baseline is shown in green.

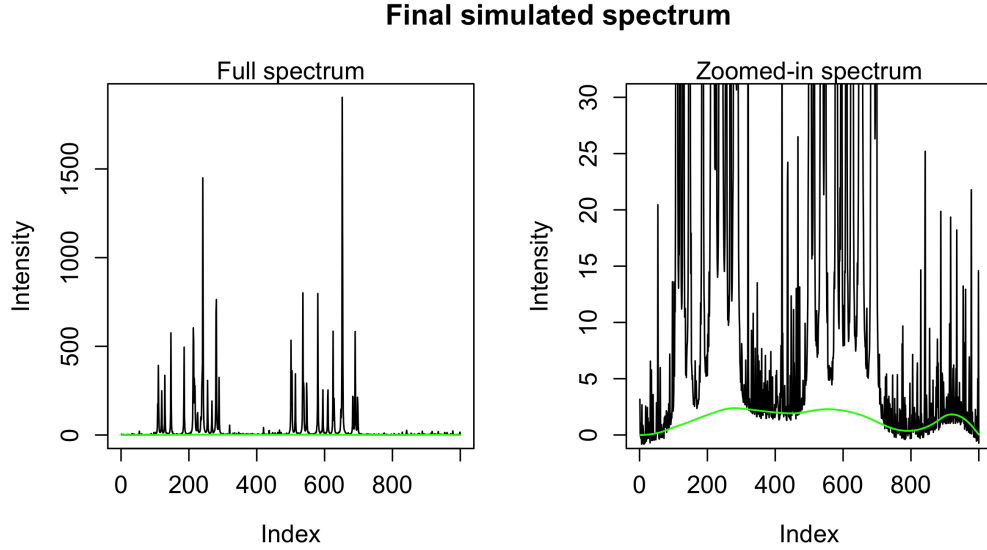


Figure 43: Final spectrum $\mathbf{x} = \mathbf{x}^{peak} + \mathbf{x}^{curve} + \mathbf{x}^{noise}$ (with and without zoom)

5.1.2 Parameter choices

Fixed values

Each simulated spectrum was composed of $p = 1000$ features and $M = 40$ peaks. Different values for the number of peaks M were evaluated before selecting $M = 40$. For $M = 50$ and $M = 80$, dramatic differences were observed between spectra generated under the same conditions (same parameters). The range of intensities, in particular, was highly variable. This can be explained by the fact that peak regions were more crowded, making it likely that peaks overlap, which dramatically increased peak intensities (overlapping peaks are summed). With $M = 40$, spectra generated under the same conditions were visually similar and had similar ranges of intensities.

A total of $n = 500$ different spectra were generated for each parameter setting (see Section 5.1.2 for more information concerning the parameter settings).

Distribution parameters

The baseline correction methods tested here were applied to spectra with varying curve and noise heights but similar peaks (similar peak heights,

widths and locations). In total, nine parameter settings (one for each combination of curve and noise parameters) were used to generate the spectra.

Curve and noise height

The curve height C and the noise height N were considered to be random variables from the log-normal distribution. Their parameters μ_C and μ_N were set to 0 and their parameters σ_C and σ_N were selected such that the expected values $E[C]$ and $E[N]$ would be equal to some proportion α of the expected value of the peak height, $E[H]$.

$$E[C] = \alpha_C E[H] \quad (13)$$

$$E[N] = \alpha_N E[H] \quad (14)$$

The expected value of C and N is given by

$$E[C] = \exp(\mu_C + \sigma_C^2/2) = \exp(0 + \sigma_C^2/2) = \exp(\sigma_C^2/2)$$

$$E[N] = \exp(\mu_N + \sigma_N^2/2) = \exp(0 + \sigma_N^2/2) = \exp(\sigma_N^2/2)$$

Note that μ_C and μ_N were set to 0 by design. The expected value of H is

$$E[H] = \exp(\mu_H + \sigma_H^2/2)$$

Equations 13 and 14 can be rewritten using the formulas for $E[C]$, $E[N]$ and $E[H]$ as follows:

$$\exp(\sigma_C^2/2) = \alpha_C \exp(\mu_H + \sigma_H^2/2) \quad (15)$$

$$\exp(\sigma_N^2/2) = \alpha_N \exp(\mu_H + \sigma_H^2/2) \quad (16)$$

By isolating σ_C and σ_N , we have

$$\sigma_C = \sqrt{2 (\log(\alpha_C) + \mu_H + \sigma_H^2/2)}$$

$$\sigma_N = \sqrt{2 (\log(\alpha_N) + \mu_H + \sigma_H^2/2)}$$

σ_C and σ_N are thus a function of α , μ_H and σ_H .

Because σ_C and σ_N must be positive and non-zero, the choice of μ_H and σ_H is restricted (see Figure 44). Indeed,

$$0 < \sqrt{2(\log(\alpha) + \mu_H + \sigma_H^2/2)}$$

$$-\log(\alpha) < \mu_H + \sigma_H^2/2$$

The proportions α were chosen such that $\alpha_C = (0.01, 0.02, 0.04)$ and $\alpha_N = (0.0025, 0.005, 0.01)$. In other words, the expected curve height was set to 1%, 2% or 4% of the expected peak height, and the noise height was set to 0.25%, 0.5% or 1% of the expected peak height.

Three values for σ_C and σ_N were calculated based on these α_C and α_N values and the values selected for μ_H and σ_H (see the following section for more details):

$$\sigma = \sqrt{2\log(\alpha) + \mu_H + \sigma_H^2/2}$$

This calculation yielded $\sigma_C = (1.70, 2.07, 2.38)$ and $\sigma_N = (0.33, 1.22, 1.70)$.

Peak height, width and location

Peak height H was considered to be a random variable from the log-normal distribution. The limits of the domain for μ_H and σ_H depend on the smallest α value, in this case $\alpha_N = 0.0025$. Thus, μ_H and σ_H must satisfy the condition

$$5.99 < \mu_H + \sigma_H^2/2$$

Ideally, the peak heights and thus total intensity and maximum intensity should be comparable for spectra with the same curve and noise parameters. For this reason, σ_H was set to be small ($\sigma_H = 0.7$). Due to domain constraints, the parameter μ_H was limited to relatively high values (μ_H was set equal to 5.8). The pair of parameters chosen is plotted as a point in Figure 44. Any other pair of parameters within the restricted domain could have been chosen.

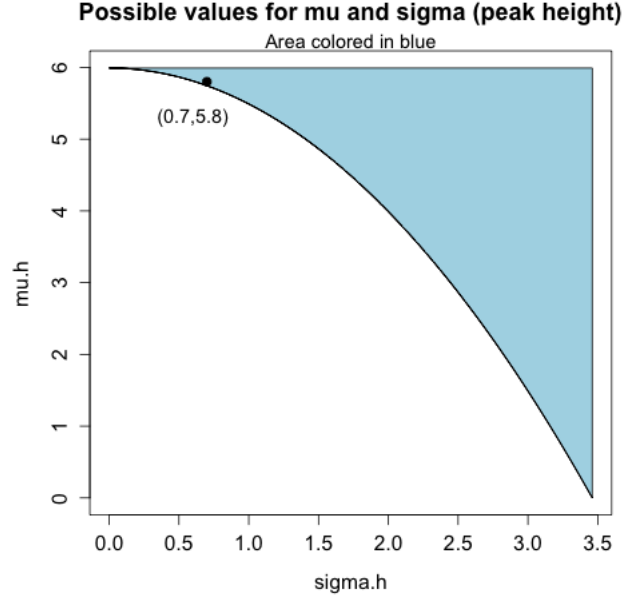


Figure 44: Domain of μ_H and σ_H

The peak width W was also considered to be a random variable from the log-normal distribution. The mean μ_W and standard deviation σ_W of this variable were set to be a certain proportion of the mean and standard deviation of the peak heights. The proportion $\frac{1}{50}$ was arbitrarily selected. Thus, the parameter $\mu_W = \frac{1}{50}(5.8) = 0.116$ and the parameter $\sigma_W = \frac{1}{50}(0.7) = 0.014$.

The values used for the peak centers $j_{0,m}$ were randomly generated from two different uniform distributions with parameters belonging to the domain $[1, 1000]$:

$$j_{0,1}, \dots, j_{0,20} \sim Unif(a_1 = 100, b_1 = 300)$$

$$j_{0,21}, \dots, j_{0,40} \sim Unif(a_2 = 500, b_2 = 700)$$

Summary of distribution parameters

Table 8: Variable parameters

Vector	Variable		Distribution
Peaks	Centers (1-20)	j_0	$\text{Unif}(a_1 = 100, b_1 = 300)$
	Centers (21-40)	j_0	$\text{Unif}(a_2 = 500, b_2 = 700)$
	Height	h	$\text{LN}(\mu_H = 5.8, \sigma_H = 0.7)$
	Width	w	$\text{LN}(\mu_W = 0.116, \sigma_W = 0.014)$
Curve	Height	c	$\text{LN}(\mu_C = 0, \sigma_C = 1.70 \text{ or } 2.07 \text{ or } 2.38)$
Noise	Height	n	$\text{LN}(\mu_N = 0, \sigma_N = 0.33 \text{ or } 1.22 \text{ or } 1.70)$

Note that ideally the log-normal parameters should be chosen based on a log-normal fit to real spectral data sets, but this was outside of the scope of this thesis.

5.1.3 Simulated spectra

Let $l : 1, \dots, L$ be the index indicating the parameter settings (i.e. the level of noise and curvature) for a given simulated spectrum. All combinations of the parameters α_C and α_N were tested, yielding a total of $L = 9$ different parameter settings. Typical spectra are provided in Figures 45 - 48.

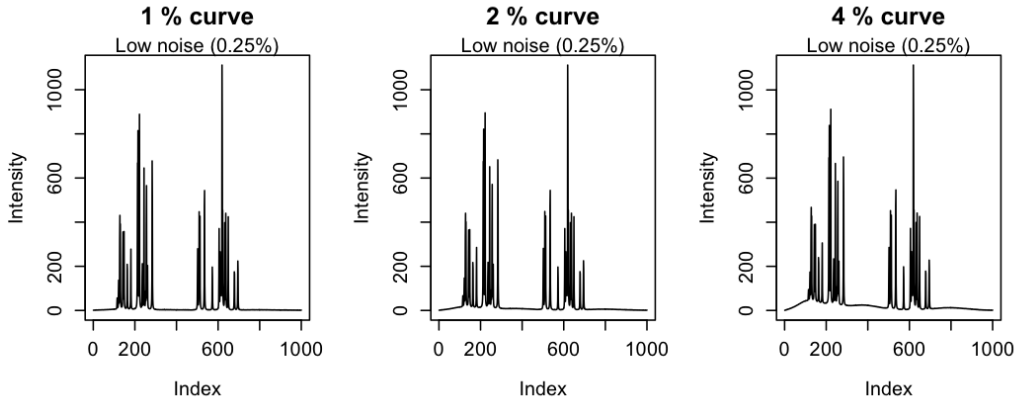


Figure 45: Examples of spectra with different curvature

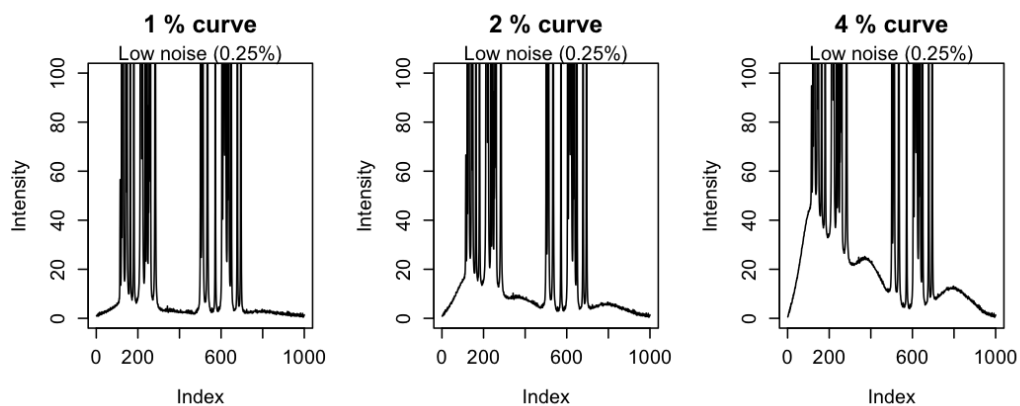


Figure 46: Examples of spectra with different curvature (zoomed)

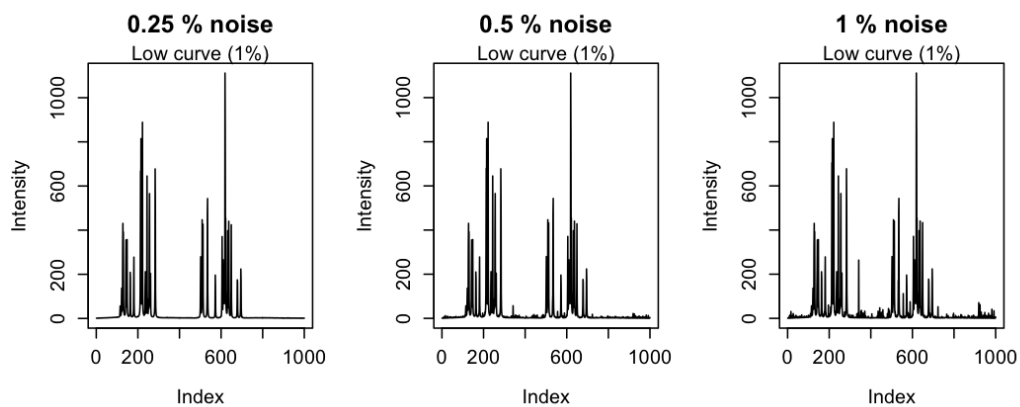


Figure 47: Examples of spectra with different noise levels

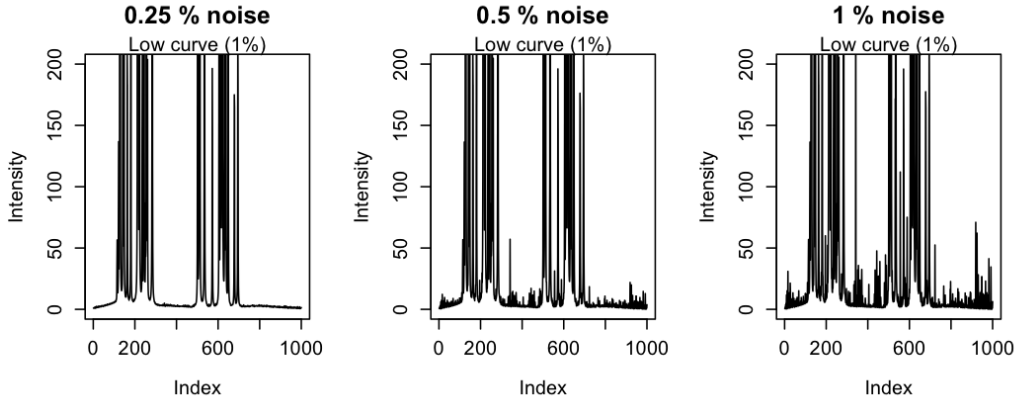


Figure 48: Examples of spectra with different noise levels (zoomed)

Let $i : 1, \dots, n$ represent the simulation number. A total of $n = 500$ simulations were run for each parameter setting l , resulting in $L \times n = 4500$ total simulated spectra. A given simulated spectrum i from condition l is noted

$$\mathbf{x}_{il} = (x_{il,1}, \dots, x_{il,1000})$$

5.2 Application of baseline correction methods

Three baseline correction methods were tested here: AsLS, airPLS and arPLS.

5.2.1 Choice of default method parameters

AsLS depends on two user-defined parameters (roughness parameter λ and weight update parameter π) and airPLS and arPLS depend on one user-defined parameter (roughness parameter λ). In practice, it is easier to choose default parameters that can be applied in a variety of situations.

For this reason, default parameters were identified for each method based on its performance when applied to spectra generated from the average parameter setting: $\alpha_C = 0.02$ (2% curve) and $\alpha_N = 0.005$ (0.5% noise). Eilers and Boelens (2005) suggest that $0.001 \leq \pi \leq 0.1$ and $10^2 \leq \lambda \leq 10^9$ are generally good ranges for parameter values. Here we tested the π values 0.001, 0.01 and 0.1 and the λ values $10^2, 10^3, 10^4, 10^5, 10^6, 10^7, 10^8, 10^9$. For AsLS, all combinations of π and λ values were tested.

Let the index k represent a particular parameter choice for a given method (or combination of parameters, in the case of AsLS), with $k : 1, \dots, K_m$. The total number of parameter combinations tested for AsLS, noted K_1 , was 24. For airPLS and arPLS, $K_2 = K_3 = 8$ total parameters.

The estimated baseline for a given spectrum i , condition l , parameter choice k and method m is noted

$$\hat{\mathbf{z}}_{ilm} = (\hat{z}_{ilm,1}, \dots, \hat{z}_{ilm,1000})$$

The true baseline for a given spectrum i and condition l is noted

$$\mathbf{z}_{il} = \mathbf{x}_{il}^{curve} = (x_{il,1}^{curve}, \dots, x_{il,1000}^{curve})$$

The estimated baseline corrected spectrum for a given spectrum i , condition l , parameter choice k and method m is noted

$$\hat{\mathbf{x}}_{ilm}^* = \mathbf{x}_{il} - \hat{\mathbf{z}}_{ilm}$$

The true baseline corrected spectrum for a given spectrum i and condition l is noted

$$\mathbf{x}_{il}^* = \mathbf{x}_{il} - \mathbf{z}_{il} = \mathbf{x}_{il} - \mathbf{x}_{il}^{curve} = \mathbf{x}_{il}^{peak} + \mathbf{x}_{il}^{noise}$$

The parameter choice k chosen as the default for each method m was the one with the smallest average root mean square error, which is given by:

$$\frac{1}{n} \sum_{i=1}^n RMSE_{ilm} = \frac{1}{n} \sum_{i=1}^n \sqrt{\frac{1}{p} \sum_{j=1}^p (\hat{x}_{ijlkm}^* - x_{ijl}^*)^2}$$

Based on this criterion, the default λ and π values for AsLS were set to 10^6 and 0.001 respectively. The optimal λ value was 10^6 for airPLS and 10^5 for arPLS. After setting the default parameters for the different methods, each method was applied to all 4500 simulated spectra.

5.2.2 Performance Measures

For each spectrum i , condition l and method m , four different performance measures were calculated based on the estimated baseline corrected spectrum $\hat{\mathbf{x}}_{ilm}^*$ and the true baseline corrected spectrum $\mathbf{x}_{il}^*$. The performance

measures used were the root mean square error (RMSE), the mean absolute error (MAE), the maximum error (Max Error) and the mean error (Mean Error):

$$\begin{aligned}
RMSE_{ilm} &= \sqrt{\frac{1}{p} \sum_{j=1}^p (\hat{x}_{ijlm}^* - x_{ijl}^*)^2} \\
MAE_{ilm} &= \frac{1}{p} \sum_{j=1}^p |\hat{x}_{ijlm}^* - x_{ijl}^*| \\
Max\ Error_{ilm} &= \max_{1 \leq j \leq p} (\hat{\mathbf{x}}_{ilm}^* - \mathbf{x}_{il}^*) \\
Mean\ Error_{ilm} &= \frac{1}{p} \sum_{j=1}^p (\hat{x}_{ijlm}^* - x_{ijl}^*)
\end{aligned}$$

The RMSE, MAE, Max Error and Mean Error for each spectrum i , condition l and method m were divided by the mean intensity of the true baseline-corrected spectrum i , $\bar{x}_i^*$, yielding the normalized measures $\widetilde{RMSE}_{ilm}$, $\widetilde{MAE}_{ilm}$, $\widetilde{Max\ Error}_{ilm}$ and $\widetilde{Mean\ Error}_{ilm}$

$$\begin{aligned}
\widetilde{RMSE}_{ilm} &= \frac{RMSE_{ilm}}{\bar{x}_i^*} \\
\widetilde{MAE}_{ilm} &= \frac{MAE_{ilm}}{\bar{x}_i^*} \\
\widetilde{Max\ Error}_{ilm} &= \frac{Max\ Error_{ilm}}{\bar{x}_i^*} \\
\widetilde{Mean\ Error}_{ilm} &= \frac{Mean\ Error_{ilm}}{\bar{x}_i^*}
\end{aligned}$$

This was done to facilitate the comparison of performance measures for different spectra.

5.3 Results

Boxplots of the normalized performance measures are presented here for each method m and condition l . The distributions and medians of the different methods are compared and analyzed.

5.3.1 Root mean square error (RMSE)

The distributions of normalized RMSE values ($\widetilde{RMSE}$) for each method were highly asymmetrical, so the normalized RMSE values were log10-transformed before being plotted. In Figure 49, it is clear that as the percent of curvature increases, the RMSE increases for all methods tested. In addition, the variability of RMSE values increases. This can be explained by the fact that the standard deviation of curve heights was related to the percentage of curvature by design (see Section 5.1.2). The side-effect of this design choice is that curve heights become more and more variable as the percentage of curvature increases, resulting in curve shapes that are less and less similar and thus baseline fits that are more and more variable.

For noise levels between 0.5 and 1% and for all levels of curvature, arPLS is the best method, having a median RMSE that is lower than the other two methods. For a noise level of 0.25%, arPLS performs about the same as the other methods. This result seems to validate Baek et al. (2015)'s claim that AsLS and airPLS perform less well than arPLS when noise is high. Interestingly, when curvature is low, increasing the noise seems to actually improve the performance of arPLS while it does not affect the performance of AsLS and airPLS.

airPLS seems to be second best to arPLS for most conditions of noise and curvature. airPLS is better than AsLS for curve levels of 1 and 2%, no matter the noise level. These two methods seem to have about the same median RMSE when the curve level is 4%.

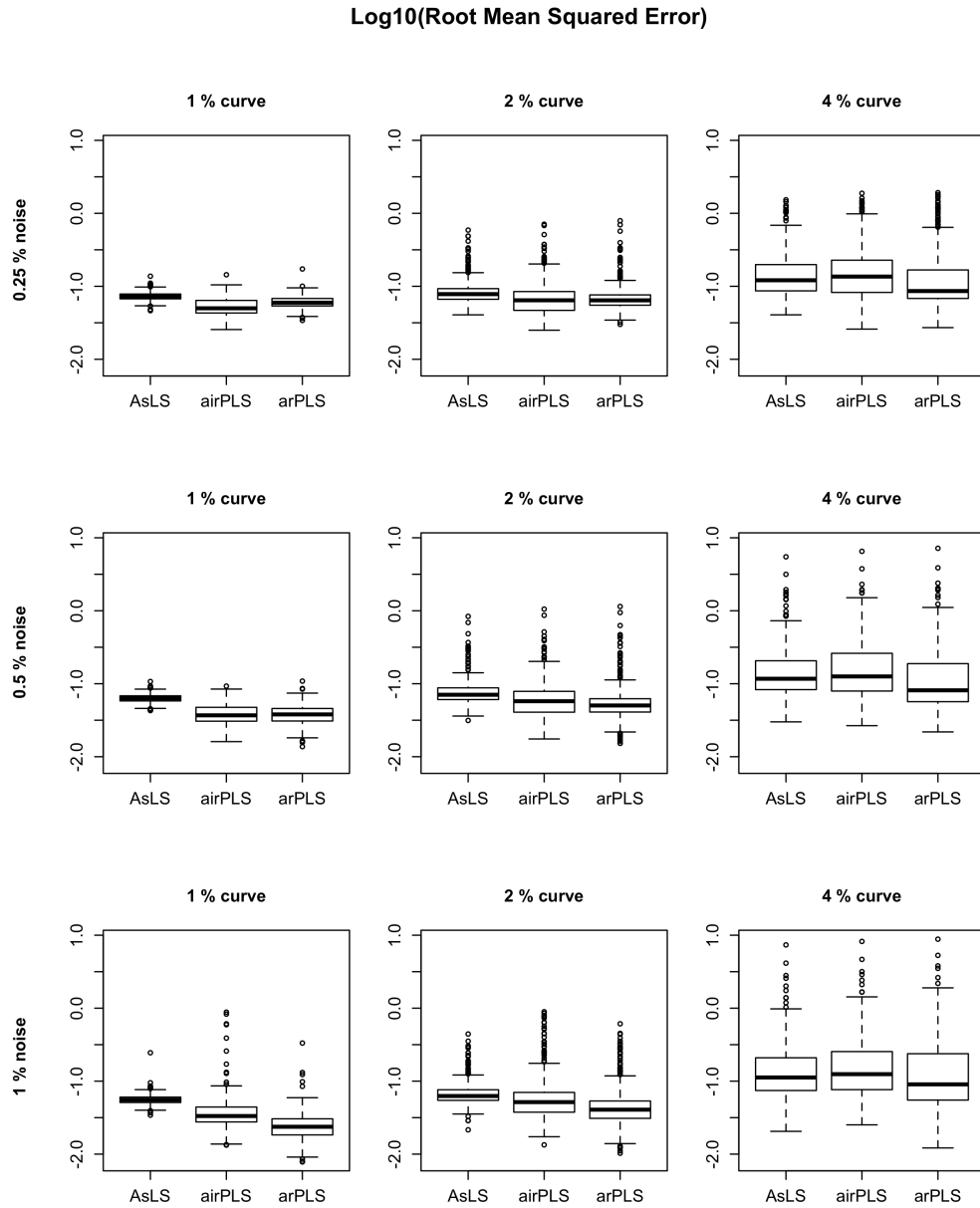


Figure 49: Normalized, log-transformed RMSE for different parameter settings

In Figure 50, the spectrum associated with the median RMSE for arPLS is plotted (median RMSE under the condition of 1% noise and 1% curve). The true baseline and the fitted baselines for AsLS, airPLS and arPLS are also

represented in this plot. All three methods seem to estimate similar baselines in signal regions, but arPLS is clearly the closest to the true baseline in noise regions.

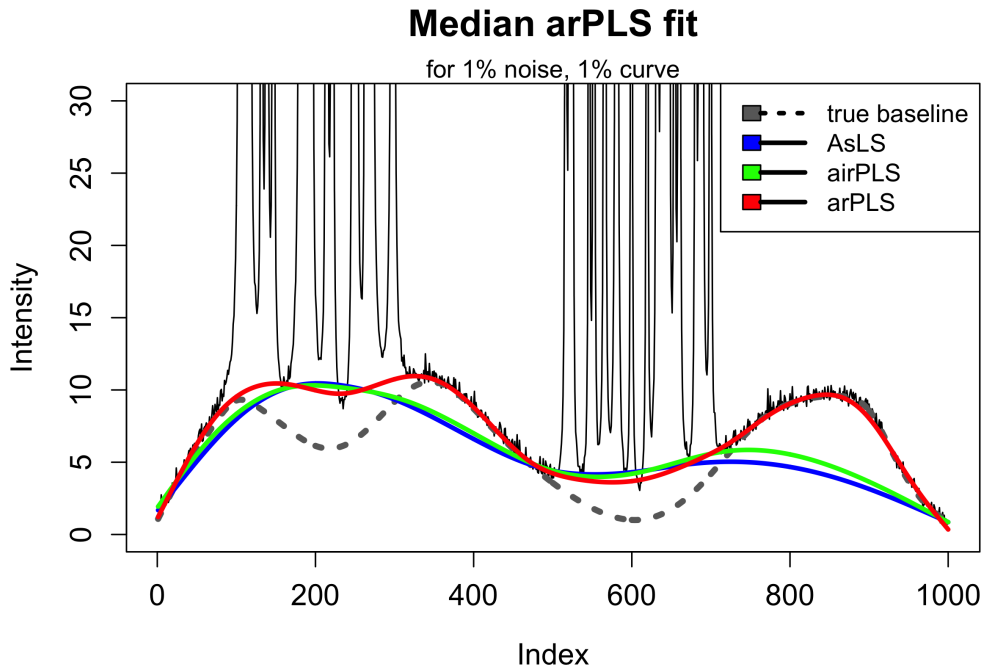


Figure 50: Median arPLS fit compared to AsLS and airPLS ($\alpha_N = 0.01$, $\alpha_C = 0.01$)

In Figure 51, the spectrum associated with the median RMSE for arPLS for spectra with 1% noise and 2% curve is plotted. Once again, all three methods seem to estimate similar baselines in signal regions. The only real difference between methods is that arPLS correctly estimates the baseline in the noise region on the far right of the spectrum whereas the other two methods underestimate it.

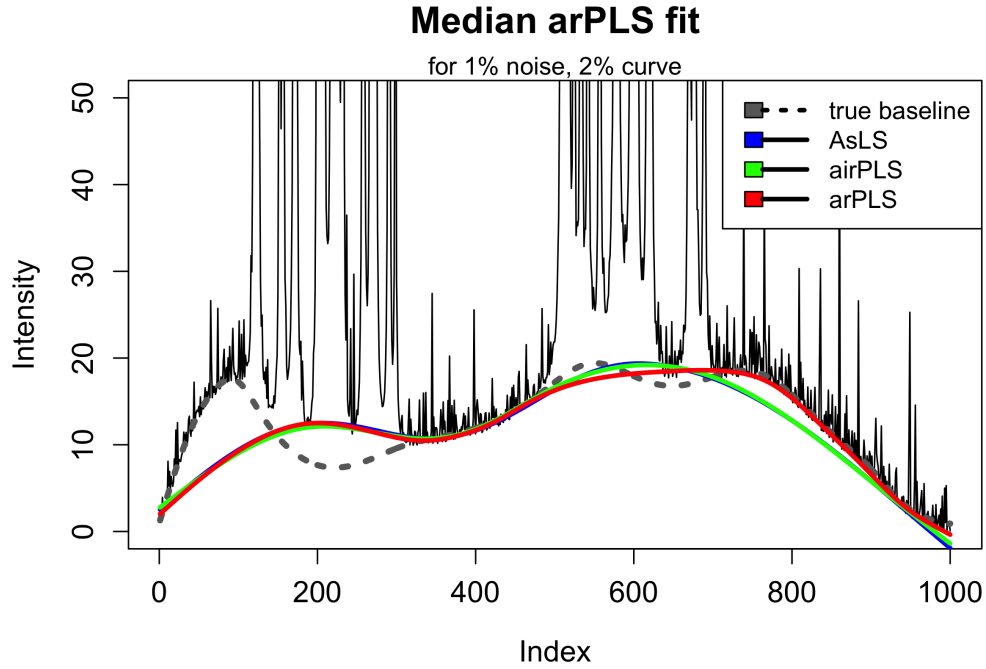


Figure 51: Median arPLS fit compared to AsLS and airPLS ($\alpha_N = 0.01$, $\alpha_C = 0.02$)

In Figure 52, the spectrum associated with the median RMSE for arPLS for spectra with 1% noise and 4% curve is plotted. All three methods yield similar baselines in signal regions but arPLS estimates a baseline closer to the true baseline in the noise regions between index 300 - 500 and 700 - 800.

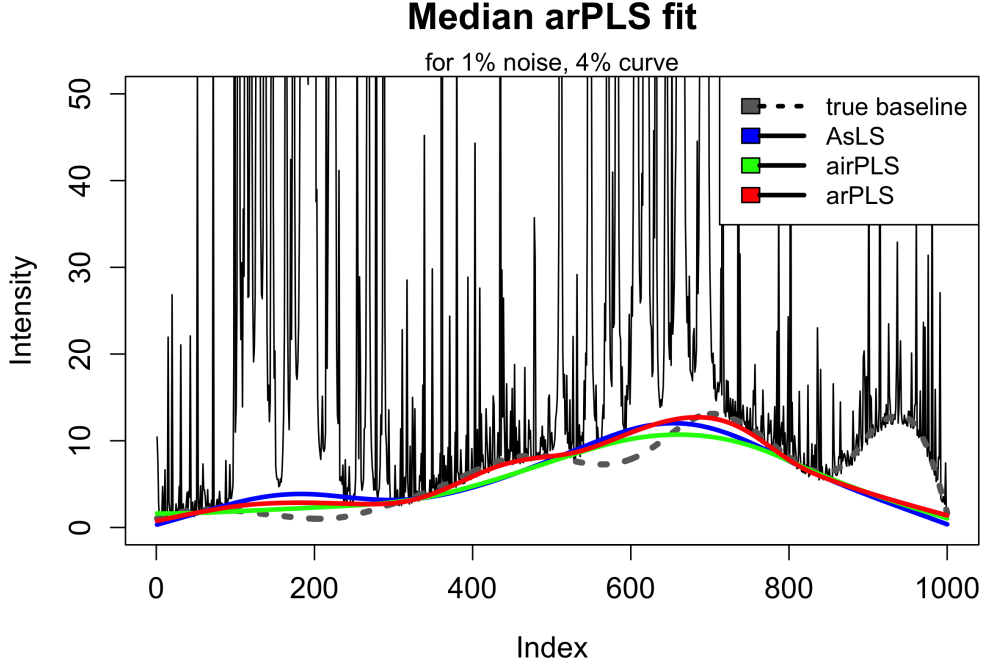


Figure 52: Median arPLS fit compared to AsLS and airPLS ($\alpha_N = 0.01$, $\alpha_C = 0.04$)

In certain areas of the spectra shown here, none of the methods tested accurately estimate the baseline. This is the case for the left and right signal regions in Figure 50 and the left signal region in Figure 51. In addition, AsLS and airPLS often interpret noise regions with baseline distortion as signal, underestimating the baseline curve (see Figure 50 between index 700 and 1000). All three methods underestimate the baseline in noise regions for curve levels of 2% and 4%, especially when baseline distortion is high (see the noise region on the leftmost end of Figure 51 and the noise region on the rightmost end of Figure 52).

These results seem to confirm Baek et al. (2015)'s claim that AsLS and airPLS tend to underestimate the baseline in noise regions. The smaller RMSE observed for arPLS seems to be due to the fact that arPLS does not underestimate the baseline in noise regions as much as the other two methods. The estimation of the baseline in signal regions seems to be about the same for all methods tested but is not necessarily satisfactory. Indeed, none of the methods are able to perfectly match the true baseline in signal regions.

5.3.2 Mean error

The distributions of normalized mean error values ($\widetilde{Mean\ Error}$) are plotted in Figure 53. The mean error is negative if a method overestimates the baseline on average: overestimating the baseline leads to a corrected spectrum that is generally lower than the true spectrum (see Section 5.2.2 for details concerning the calculation of the mean error). It is clear that all methods overestimate the baseline for curve levels of 1 and 2%, regardless of the noise level.

For 4% curve and all noise levels, AsLS and airPLS underestimate the baseline on average. AsLS and airPLS seem to transition from overestimation to underestimation as the curve level increases, whereas arPLS overestimates the baseline for all noise and curve conditions (except when noise and curve are maximal). Thus, arPLS offers baseline correction that is more consistent and less influenced by the particular noise and curve levels of a given spectrum.

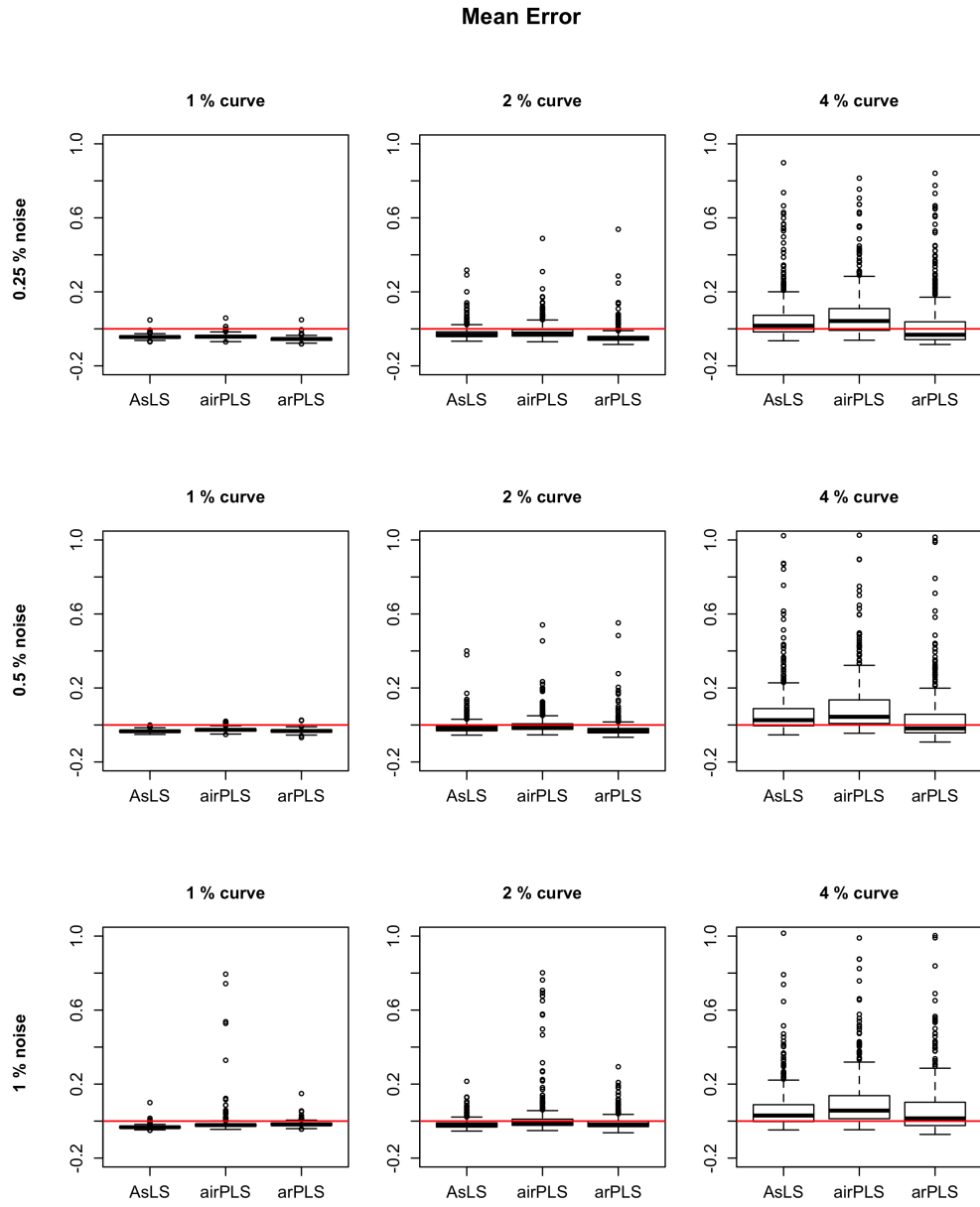


Figure 53: Normalized Mean Error for different parameter settings

5.3.3 Other measures

The boxplots of the other two performance measures, mean absolute error and maximum error, showed the same trends as the RMSE boxplots. These boxplots can be found in the Annex (see Section 7.3).

5.4 Conclusion

Compared to AsLS and airPLS, arPLS seems to be the best method for baseline correction when noise is high or medium-high, regardless of the degree of baseline distortion in the spectrum in question. The strength of the arPLS algorithm is that it is better able to estimate the baseline in noise regions. However, for spectra with a low level of noise, arPLS performs no better than AsLS and airPLS.

As expected, increasing baseline distortion resulted in increased error for all methods tested. It also increased the variability of the error of these methods, although this variability is likely explained by the particular design choices during simulation.

The boxplots in Figure 53 suggest that AsLS and airPLS transition from over-estimation to underestimation as the degree of curvature increases. Thus, the bias of these two methods depends on the particular character of the spectra in question. arPLS underestimates the baseline relatively consistently, meaning that it is less sensitive to differences in curve levels and thus more widely applicable.

While the iterative penalized least squares methods can estimate complex (i.e. non-linear) baseline distortion in signal regions better than noise and signal classification methods (see Section 4.2), their estimation of the baseline is far from perfect. The estimation of the baseline in signal regions remains a complex problem requiring further research and development.

6 Conclusion

In this thesis, two NMR pre-processing steps – normalization, scaling and transformation (NST) and baseline correction – were studied. After providing a review of the various methods described in the literature, a selection of the most promising methods were applied to real and simulated data sets. For baseline correction, these methods include AsLS, airPLS and arPLS. For NST, CSN, Linear Baseline Normalization, PQN, VSN, QN, Pareto scaling and Autoscaling were implemented. These methods were then evaluated according to different criteria.

In this final chapter, the performance of methods reported in the literature and in this thesis will be compared and discussed (see “Conclusions”) and directions for further research will be proposed (see “Further directions”).

6.1 Normalization, scaling and/or transformation (NST) methods

6.1.1 Conclusions

The performance measures used to evaluate the performance of the NST methods tested for this thesis measure the ability of an NST method to reduce overall differences between spectra obtained from similar samples. Thus, the results presented in this thesis cannot be compared to studies in the literature evaluating NST methods according to other criteria (i.e. the reduction of intensity-dependent bias, variance and differences between spectra, the preservation and/or increase of the contrast between noise and signal, the removal of dilution effects and the improvement of classification performance).

Urine data

Wang et al. (2013) found that CSN and PQN were among the best methods for reducing overall differences between spectra, and Kohl et al. (2012) found that PQN and QN were among the best (see Figure 18 in Chapter 2). These findings confirm the results for the urine data set used for this thesis, where CSN, PQN and QN were the top performing methods.

It is not altogether surprising that CSN was able to perform well in this study, despite its bad reputation in the literature. Indeed, the NMR spectra used here to evaluate the performance of NST methods were acquired from

urine samples from healthy subjects. Because there were no sick subjects in the data set, the variation between samples from different subjects was of an unspecific nature (similar to the samples used by Dieterle et al. (2006) and Wang et al. (2013), as explained in the Annex, Section 7.1.1). This type of spectra is by far the easiest to normalize, scale and/or transform because the spectra are already very similar. Thus, CSN’s ability to perform well in this study in terms of spectral comparability does not suggest that it would be an appropriate choice in terms of any other evaluation criteria. The literature would suggest that CSN is unable to achieve most other NST objectives (see Figure 18).

Surprisingly, Linear Baseline Normalization, which has not been shown to be the best method for increasing comparability of similar spectra, was one of the best methods for the urine data in this thesis. This could be because it has never been tested on a data set like ours, where there is little variation between spectra. The results outlined in Kohl et al. (2012) seem to suggest that Linear Baseline Normalization would not perform as well on more complex – and perhaps more realistic – data sets: when testing Linear Baseline Normalization on their data sets, Kohl et al. (2012) did not find that Linear Baseline Normalization greatly increased comparability.

Serum data

PQN and QN, both shown to have high performance in the literature in terms of reducing overall differences between spectra (Kohl et al. (2012), Wang et al. (2013)), also performed well when applied to the serum data set in this thesis. Unlike for the urine data, however, the strong performance of CSN was not observed for the serum data. The serum samples used were possibly more subject to imbalanced metabolite regulation than the urine samples. This seems to be confirmed by the fact that the top methods for the serum data were those that were performed in combination with Hochrein et al. (2015)’s pre-normalization step (a step that accounts for imbalanced metabolite regulation).

PQN and Linear Baseline Normalization, both with Hochrein et al. (2015)’s pre-normalization step, were among the best methods for the serum data set. The reduction of overall spectral differences has not yet been evaluated for these methods in the literature, so these results remain to be confirmed by a future study.

Overall

Based on the review of the literature (see Chapter 2) and the results presented here (see Chapter 3), the best-performing NST methods overall (i.e. for

different types of data sets and evaluation criteria) seem to be PQN and QN. For isolated NST objectives, the best NST method depends on the NST objective in question (see Figure 18). In terms of the reduction of overall differences between spectra, serum data sets may benefit from Hochrein et al. (2015)’s pre-normalization step more than urine data sets.

6.1.2 Further directions

PQN was found to be one of the best methods overall for the normalization of spectra. Torgrip et al. (2008) has demonstrated that Histogram Matching (HM) outperforms PQN in terms of the reduction of dilution effects. HM, or an improved version of HM, may be better than PQN for the other NST objectives as well, but no other authors have studied it yet. It would be interesting to build on the ideas of HM, developing a statistical method that searches for a factor minimizing the differences between the empirical distributions of each spectrum and a reference spectrum (rather than between the histograms of each spectrum and a reference spectrum).

The pre-normalization step proposed by Hochrein et al. (2015) has received little attention from other authors in the literature. Methods such as CSN, PQN, VSN and Linear Baseline Normalization combined with the pre-normalization step should be evaluated according to the criteria used in the literature (see the list in Section 2.1.1). This would make it possible to assess whether this pre-normalization step is useful for achieving NST objectives other than spectral comparability and technical reproducibility.

Lastly, it is unclear why the pre-normalization step was useful for the serum data set but not the urine data set. A deeper analysis of the effect of metabolite regulation on NMR spectra, as well as an investigation of the spectral differences between urine and serum spectra, could help explain this finding. Other feature pre-selection methods could also be developed.

6.2 Baseline correction methods

6.2.1 Conclusions

The performance of the three baseline correction algorithms tested – AsLS, airPLS and arPLS – has been compared in Baek et al. (2015), Zhang et al. (2010), Liu et al. (2014a) and Liu et al. (2014b). The performance observed

here for these baseline correction algorithms seems to confirm findings reported in the literature.

For example, airPLS was found to perform better than AsLS in Zhang et al. (2010). This was also observed in this thesis, with the precision that airPLS is only better than AsLS for curve levels of 1% and 2%, regardless of noise level.

The finding that arPLS performs better than AsLS and airPLS in noise regions (Baek et al., 2015; Liu et al., 2014b,a) was also confirmed by the results observed here. arPLS was better than AsLS and airPLS when noise was high or medium high (1% or 0.5%), regardless of curve level. The difference between methods was most striking in noise regions, where the baselines estimated by arPLS were closer to the true baseline than the baselines estimated by AsLS and airPLS. In signal regions, baseline estimation was comparable for all three methods.

The fact that arPLS more accurately estimates the baseline in noise regions makes it less likely to introduce artificial peaks or signal regions to spectra. The other methods tend to underestimate the baseline in noise regions, which often results in the creation of artificial mounds in baseline-corrected spectra. This is problematic because these artificial mounds could be mistaken for signal during multivariate analysis. Thus, arPLS should be generally favored over the other two methods.

One new contribution of this thesis was the evaluation of the average bias of the different baseline correction methods. On average, arPLS consistently overestimates the baseline, whereas AsLS and airPLS transition from overestimation to underestimation as curve level increases (regardless of noise level). arPLS has thus the advantage of having a bias that is less sensitive to differences in curve levels.

None of the baseline correction methods in the literature seem to correctly estimate the baseline in signal regions, especially in the presence of overlapping peaks. The signal and noise classification methods are particularly weak because they require the input of several user-defined parameters, rely on poor classification methods and estimate the baseline in signal regions using linear interpolation. The iterative penalized least squares methods seem to be better than noise and signal classification methods because they require fewer user-defined parameters and afford a better fit of the baseline in signal regions. However, this estimation of the baseline in signal regions remains far from perfect.

6.2.2 Further directions

Differences appear to exist between the different baseline correction methods tested here, but it is not clear if these differences have an actual impact on the quality of the data for multivariate analysis. Thus, it would be interesting to generate spectra belonging to two different classes and then to compare classification performance after application of each baseline correction method. This would make it possible to assess the practical impact of different methods.

The method of simulating spectra used here allowed an initial evaluation of baseline correction methods but suffered from several weaknesses. For one thing, because the curve level was adjusted by changing the standard deviation of the curve distribution, the baseline curve of spectra became more and more variable as the standard deviation of the curve distribution was increased. As a result, baselines in spectra with a low level of curve were fairly similar but were very different in spectra with a high level of curve. This design choice was made to ensure that the expected curve height would be equal to some proportion of the expected peak height. Other methods could be explored for increasing the level of curvature in the baseline while ensuring that baselines in the same category of curvature remain similar.

It should also be noted that no particular attention was paid to the fidelity of the simulated spectra to real spectra generated from biological samples. The heights of peaks in real spectra could be analyzed to identify suitable ranges of intensity for simulated peaks, and peak locations and regions observed in real spectra could be used to fix specific peak locations or regions. The parameters of the curve and noise distributions could also be determined by fitting the parameters to real data sets. Even the choice of distribution (i.e. log normal, uniform, etc.) could be re-evaluated by studying real spectra.

Finally, the results presented here describe the performance of the different baseline correction algorithms for different levels of noise and curvature. These findings would be even more useful if there were a method for automatically estimating the percent of curvature and noise present in a real spectrum. The choice of baseline correction method could thus be made automatically in function of the observed curve and noise levels.

6.3 Concluding remarks

Based on a review of the literature and the evaluation of certain pre-processing methods, it seems that arPLS is the most generally applicable and best performing baseline correction method and that PQN and QN are the best methods for normalization, scaling and transformation. Estimating NST method parameters on a reduced data set including only “unregulated” features also seems to improve the performance of NST methods for certain data sets.

7 Annex

7.1 NST evaluation methods in the literature

Different methods for evaluating the performance of normalization, scaling and transformation (NST) methods have been proposed in the literature (Dieterle et al. (2006), Parsons et al. (2007), Torgrip et al. (2008), Kohl et al. (2012), Wang et al. (2013), Hochrein et al. (2015)). In this section, the criteria used to evaluate methods, as well as the types of spectra on which methods are tested, will be described. Note that, due to time constraints, no deep critical analysis of the methods presented here will be provided.

7.1.1 Data used

In this section, several NMR data sets that have been used to evaluate NST methods will be described. These particular data sets will be referred to in Section 7.1.2.

Spike-in samples

Spike-in data sets are particularly useful for evaluating the bias and variance of intensity estimates because the concentrations of metabolites are known by experimental design. “Spike-ins” are metabolites that are artificially introduced to samples before NMR spectra are generated. When samples are designed such that a single spiked-in compound has the same concentration across all samples, the reproducibility of a given method can also be evaluated. Lastly, spike-in data sets can be useful for evaluating the preservation of the contrast between noise and signal, as the location and intensity of peaks representing signal are known by design.

Kohl et al. (2012) created spike-in data sets by adding differing concentrations of 8 metabolites (with concentrations varying from 0.0488 mmol/L to 6.25 mmol/L) to samples drawn from a homogenized pool of human urine samples. The total concentration of the added metabolites was the same for each different sample created.

Hochrein et al. (2015) also created spike-in samples, adding 8 metabolites (with concentrations varying from 0.05 mmol/L to 6.25 mmol/L) to samples drawn from a single urine sample (rather than a pool of urine samples). The total concentration (12.45 mmol/L) of these 8 metabolites was kept constant from sample to sample. However, an additional metabolite, glucose, was then

added so that the total concentration of added metabolites actually differed between samples (the glucose was added in 8 different concentrations ranging from 1.563 mmol/L to 200 mmol/L). This glucose metabolite was meant to serve as a “confounder”, violating the self-averaging principle (see Section 2.1.2 for a definition of this principle). The overall volume of each sample was the same.

Wang et al. (2013) created synthetic urine samples with unique metabolic profiles but little overall variation. Between nine and 17 compounds were added to each synthetic urine sample, with compound concentrations varying from 0.063 mmol/L to 1.1 mmol/L. Lastly, trimethylsilyl propionate (TSP) was added to each sample (in exactly the same concentration) in order to evaluate the reproducibility of each method.

Samples modified *in silico*

Rather than modifying the concentrations of metabolites in samples before generating spectra, modifications can be made afterwards, *in silico*. By adding an artificial peak of known intensity and then modifying the entire spectrum by some dilution factor, it is possible to evaluate an NST method’s ability to correct for dilution effects (the corrected intensity of the artificial peak can be compared to the known intensity).

Torgrip et al. (2008) applied *in silico* modifications to synthetic urine samples with relatively similar metabolic profiles. First, samples were created by adding 24 metabolites and 3 non-biological standards to synthetic urine samples. Nineteen of the 27 total compounds were added in constant concentrations across samples and only 7 metabolites were added in varying concentrations (between 5.5 mmol/L and 29.2 mmol/L). While the individual profiles of the resulting samples were unique, they were very similar overall.

After producing NMR spectra for each sample, three high-intensity synthetic peaks were added to each spectrum (the three peaks were of identical intensity intra-sample and variable intensity between samples). Then, each spectrum was multiplied by a different dilution factor, ranging from 0.4 to 2.6, resulting in a set of artificially diluted spectra.

Dilution effects were also simulated *in silico* by Dieterle et al. (2006). A “golden spectrum” was first created by calculating the median spectrum of 4023 NMR spectra of urine from control rats. An artificial peak of known intensity was added to the golden spectrum and then different spectra were generated based on arithmetic modifications of the golden spectrum. These modifications included multiplying the entire spectrum by different dilution

factors (1.1, 1.2, ..., 2), multiplying a single feature by different dilution factors, multiplying the entire spectrum and a single feature by different dilution factors and increasing a given total number of features (10, 20, ..., 200) by one percent of the total intensity of the golden spectrum.

Samples with unspecific variation

Samples with variation of an unspecific nature are generally the least demanding in terms of normalization, scaling and transformation. The small amounts of variation between samples are due to normal biological variation, not some kind of pathology, meaning that the spectra measured for these types of samples should already be very similar. These type of samples are particularly useful for evaluating the reproducibility of different methods.

Dieterle et al. (2006) collected samples of this type using urine from control animals (healthy animals not receiving any particular medication). Samples with unspecific variation can also be created using spike-ins with concentrations of very little variation (such as concentrations between 0.063 mmol/L and 1.1 mmol/L, as performed by Wang et al. (2013)).

Samples with dramatic variation

In contrast to samples from control animals, samples from patients with a pathology that dramatically influences metabolite regulation are arguably the most difficult samples to normalize, scale or transform. Hochrein et al. (2015) collected plasma samples from patients 24 hours after they received cardiac surgery. A little less than half of these patients developed acute kidney injury (AKI), which results in highly imbalanced metabolite regulation.

Classification samples

The ability of an NST method to improve classification performance is often tested on data sets including samples from two or more known classes (for example, sick vs healthy subjects). The classification data set used in Kohl et al. (2012) was composed of urine samples from patients with autosomal polycystic kidney disease (APKD) and healthy subjects. Parsons et al. (2007) also used a classification data set, which was composed of urine samples from two different breeds of dogs.

7.1.2 Evaluation criteria

This section will summarize the various evaluation criteria proposed in the literature to measure the performance of NST methods on NMR spectra. Each category of criterion will include a description of the methods (“Methods”) followed by a short discussion of their validity and applicability (“Discussion”). As mentioned at the beginning of this chapter, no deep critical analysis of the performance measures will be provided due to time constraints.

In this section, the notation x_{ij} represents the intensity of feature j in spectrum i after application of a given NST method.

Note that certain evaluation criteria can only be applied to specific types of data sets (see Section 7.1.1 for more information on the specific data mentioned in this section).

Reduction of overall difference between samples

Methods

Local reproducibility (one metabolite)

Wang et al. (2013) assessed reproducibility by applying various normalization methods to synthetic urine samples containing the same known concentration of a spiked-in compound (TSP, see 7.1.1 for more details). In this case, the best normalization method would be the method that yields exactly the same intensity estimate for the TSP peak in all spectra.

Wang et al. (2013) measured the reproducibility of a given method using the coefficient of variation of the estimated TSP peak intensity for all samples. Let t_i^{TSP} be the total intensity of the TSP peak in spectrum i after applying a given NST method. The peak intensities of each spectrum i are stored in a vector $\mathbf{t}^{TSP}$, from which the standard deviation S and mean $\bar{x}$ are calculated. The coefficient of variation is thus

$$cv = \frac{S}{\bar{x}}$$

A low coefficient of variation is indicative of good reproducibility.

Global reproducibility (PCA-based)

Another way of evaluating reproducibility is by studying the effect of different methods on the variance structure of control samples. Dieterle et al.

(2006) used NMR spectra of urine from control rats to evaluate normalization methods (see 7.1.1 for more details). Because these were control samples, the variation between samples was assumed to be minimal and of an unspecific nature (all variation being within the range of normal biological fluctuations). The percent of variance explained by the first three components of a Principal Components Analysis (PCA) was used as a criterion of reproducibility, where the percent of variance explained should be greater after normalization than before. Dieterle et al. (2006) do not provide any justification for this evaluation criterion.

Note that Torgrip et al. (2008) have questioned the use of PCA for comparing methods. While PCA did not allow them to distinguish the different methods they tested, their root mean square error (RMSE) method (see Section 7.1.2) highlighted significant differences between methods. According to the authors, this “indicates that the eigenspace of 1D NMR data can undergo considerable changes without affecting overall variance” (Torgrip et al., 2008).

Comparability

Another goal of normalization is to reduce overall differences between samples so that comparisons can be made more easily. Kohl et al. (2012) propose a measure of reduction of inter-sample variation that they applied to both a spike-in data set (see 7.1.1) and a classification data set (see 7.1.1).

Let x_{ij} be the corrected intensity of feature j for a given spectrum i and $x_{i'j}$ be the corrected intensity of feature j for a different spectrum i' . The difference between x_{ij} and $x_{i'j}$ is calculated, noted by δ_{jk} , where k indicates the particular combination of spectra in question ($k : 1, \dots, K = \frac{n(n-1)}{2}$). The difference between all possible pairs of spectra is calculated for feature j , yielding a vector of differences δ_j of length K .

Next, the median difference in this vector is identified:

$$\delta_{j_{med}} = \text{median}_{1 \leq k \leq K}(\delta_{jk}).$$

This process is repeated for all features $j : 1, \dots, p$, resulting in a vector δ_{med} of length p including the median differences for each feature. In order to evaluate the global comparability of the spectra after normalization, scaling and transformation, the absolute value of these median differences is averaged:

$$\bar{\delta} = \frac{1}{p} \sum_{j=1}^p |\delta_{j_{med}}|$$

Because $\bar{\delta}$ is an average of absolute value differences, a value close to zero indicates very good comparability between spectra (almost null absolute value differences).

The magnitude of $\bar{\delta}$ depends on the spread of the median differences. To facilitate comparison between different data sets, $\bar{\delta}$ is normalized by the interquartile range (IQR) of the median differences. The overall similarity between samples for a given method, noted by γ , is thus calculated:

$$\gamma = \frac{\bar{\delta}}{IQR(\delta_{med})}$$

Values of γ close to zero suggest good inter-sample comparability.

Discussion

If a spike-in data set is available, the reproducibility criterion proposed by Wang et al. (2013) would be an interesting performance measure. Dieterle et al. (2006) do not provide a theoretical justification for their reproducibility criterion, so it is unclear whether their method is valuable in practice.

For Kohl et al. (2012)'s comparability criterion, the motivation behind the use of the IQR as scaling factor (rather than the standard deviation or some other factor) is not provided. Clearly, a scaling factor must be used because $\bar{\delta}$ depends on the scale of the data set in question. However, the IQR is not directly related to the scale of the data and thus might not be an optimal scaling factor.

Reduction of intensity-dependent differences between spectra

Methods

Kohl et al. (2012) propose a method for evaluating an NST method's ability to reduce intensity-dependent differences between spectra that they applied to both a spike-in data set (see 7.1.1) and a classification data set (see 7.1.1).

Intensity-dependent differences for a single pair of spectra

Let x_{ij} be the normalized, scaled or transformed intensity of feature j for a randomly selected spectrum i and $x_{i'j}$ be the normalized, scaled or transformed intensity of feature j for a second randomly selected spectrum i' . The ratio between each feature j in the two spectra is calculated and stored in a vector of ratios, α . Each ratio $\hat{\alpha}_j$ represents the estimated fold change in

feature j when passing from spectrum i' to spectrum i .

$$\hat{\alpha}_j = \frac{x_{ij}}{x_{i'j}}$$

$$\hat{\boldsymbol{\alpha}} = (\hat{\alpha}_1, \dots, \hat{\alpha}_p)$$

Next, the mean of each feature j in both spectra is calculated.

$$\bar{x}_j = \frac{1}{2} (x_{ij} + x_{i'j})$$

$$\bar{\mathbf{x}} = (\bar{x}_1, \dots, \bar{x}_p)$$

In the absence of intensity-dependent differences between spectrum, the logged fold changes should be evenly scattered around a constant value for the entire dynamic range of intensities. To evaluate this, the logged vector of fold changes, $\log_2(\hat{\boldsymbol{\alpha}})$, is plotted against the vector of ranks for each mean, $r(\bar{\mathbf{x}})$. If the plotted points are evenly spread around a horizontal line, in this case $\log_2(\alpha) = 0$, no intensity-dependent differences between spectra are observed. Note that if $\log_2(\alpha) = 0$, the ratio between spectra is 1 (no fold change). In Kohl et al. (2012), $\log_2(\alpha)$ is noted M and $r(\bar{x})$ is noted $rank(A)$. The resulting plots are called “M-Rank(A) plots.”

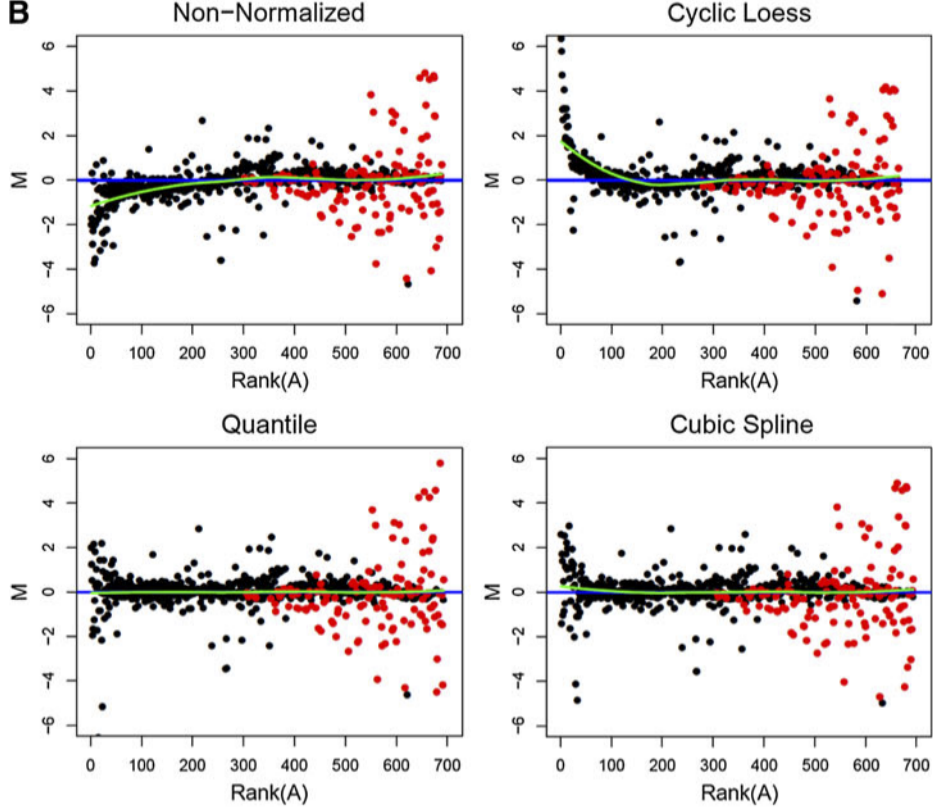


Figure 54: M-Rank(A) plots (Kohl et al., 2012)

As seen in Figure 54, a fitted curve (Loess line) can be superimposed onto the plot to evaluate how well the plotted points match up to the horizontal line $M = \log_2(\alpha) = 0$. Clearly, Quantile Normalization achieves a constant $\log_2(\alpha)$ for all intensities, suggesting that the difference between features in the pair of spectra in question remains about the same for all intensities. On the other hand, the differences between spectra for Cyclic Loess normalized and non-normalized spectra depend on intensity.

Overall intensity-dependent differences

The ratio vector $\hat{\alpha}$ and the rank vector $r(\bar{\mathbf{x}})$ are calculated using two randomly selected spectra, making it difficult to draw conclusions about overall performance. In order to evaluate the overall performance of a given NST method, all pairs of spectra must be considered. For each possible pair k of spectra ($k : 1, \dots, K = \frac{n(n-1)}{2}$), a Loess line $\mathbf{Loess}_k$ can be estimated as described above. Each element j of this vector represents the ratio $\hat{\alpha}$ associated with the average intensity of rank j . The average absolute Loess line,

$\overline{\text{Loess}}$, is then calculated as follows:

$$\overline{Loess_j} = \frac{1}{K} \sum_{k=1}^K |Loess_{jk}|$$

$$\overline{\text{Loess}} = (\overline{Loess_1}, \dots, \overline{Loess_p})$$

Plotting these average absolute Loess lines (see Figure 55) makes it possible to evaluate the performance of NST methods across all spectra in a given data set. If, for a given NST method, the average absolute Loess line remains approximately constant across all intensities, that method successfully reduces intensity-dependent differences between spectra. As seen in Figure 55, Li-Wong normalization yields average differences between spectra that are high in the lower dynamic range, with the average difference decreasing as intensity increases. After reaching a minimum, the average difference between spectra begins to increase with increasing intensity. In contrast, the average difference between spectra for Quantile Normalization remains relatively low and constant across the entire dynamic range.

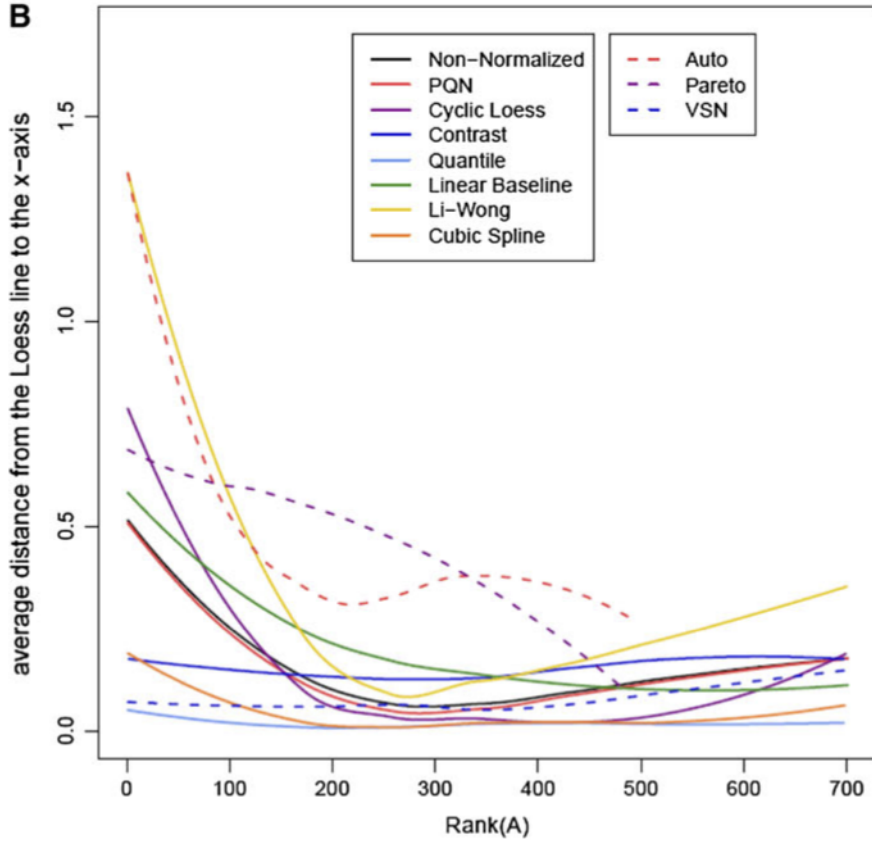


Figure 55: Average absolute Loess lines (Kohl et al., 2012)

Reduction of intensity-dependent bias

Ideally, the difference between known and estimated intensities in spectra should not depend on intensity. In practice, technical imperfections and biological biases, such as the presence of imbalanced metabolite regulation (see Section 2.1.2), can lead to inaccurate estimation of spectral intensities.

Methods

If fold changes between spectra are known by experimental design (as is the case for spike-in data sets, see 7.1.1), Hochrein et al. (2015) propose a way of measuring the reduction of intensity-dependent bias. Let t_{im} be the total intensity of metabolite m in spectrum i after application of a given NST method. The estimated fold change of metabolite m in a pair k of samples (spectra i and i') is given by

$$\hat{\alpha}_{mk} = \frac{t_{i'm}}{t_{im}},$$

and the known fold change for metabolite m is noted α_{mk} . The difference between these fold changes is given by

$$\Delta_{mk} = \hat{\alpha}_{mk} - \alpha_{mk}$$

and the vector containing the differences for all metabolites $m : 1, \dots, M$ for a given pair k of spectra is denoted

$$\Delta_{\mathbf{k}} = (\Delta_{1k}, \dots, \Delta_{Mk})$$

Now suppose that all n spectra can be divided into G total groups of spectra, where all spectra $\mathbf{x}_1, \dots, \mathbf{x}_{n_g}$ in a given group g are under the same experimental conditions. In Hochrein et al. (2015), for example, spectra are divided into $G = 8$ groups based their concentration in glucose. Each individual spectrum belonging to a given group g is paired with each individual spectrum in a different group g' , yielding a total of $n_g * n_{g'}$ pairs. The differences Δ for all of these pairs are pooled into a single vector, $\Delta_{gg'}$, of length $M * n_g * n_{g'}$.

Side-by-side boxplots of $\Delta_{gg'}$ for different pairs gg' can be used to visualize intensity-dependent bias (see Figure 56). Because the fold change for glucose is known for each pair of groups g and g' , the boxplots can be ordered from highest to lowest known fold change (see Figure 56, where groups 8 and 1 have the largest known fold change and group 4 and 5 have the smallest known fold change).

After NST, there is no intensity-dependent bias if the median of each vector $\Delta_{gg'}$ is zero, regardless of the fold change in question. A given NST method is also interesting if the IQR of each vector $\Delta_{gg'}$ is smaller than it would be in the absence of any NST method (which suggests that the bias is less variable).

In Figure 56, non-normalized spectra seem to exhibit no intensity-dependent bias (all medians are equal to zero), but the large IQRs for each boxplot suggest that the bias of non-normalized spectra is more variable than it is for Linear Baseline Mean-normalized spectra⁵. Linear Baseline Mean Normalization leads to an underestimation of fold changes, especially when the difference between glucose levels in the two groups g and g' is high. This suggests that Linear Baseline Normalization induces an intensity-dependent bias.

⁵Note that Linear Baseline Mean Normalization is the same as Linear Baseline Normalization (see Section 2.2.3) except that the reference spectrum used is the mean spectrum, rather than median spectrum

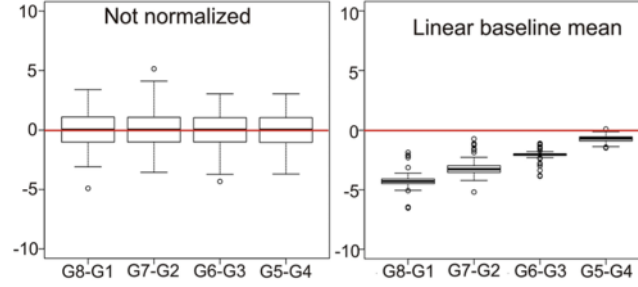


Figure 56: Boxplots of $\Delta_{gg'}$ for different glucose level pairs gg' (Hochrein et al., 2015)

Discussion

Once again, a spike-in data set must be available in order to apply Hochrein et al. (2015)'s criterion for evaluating intensity-dependent bias, which limits the applicability of this method.

Reduction of intensity-dependent variance

The standard deviation of NMR spectral intensities generally increases with an increase in intensity. One of the goals of NST is thus to ensure constant variance across all intensities.

Methods

To evaluate the reduction of intensity-dependent variance, Kohl et al. (2012) applied NST methods to a spike-in data set (see 7.1.1) and a classification data set (see 7.1.1). Then they plotted standard deviation against the rank of mean intensities, similar to the M-Rank(A) plots described in the previous section. To prepare these plots, three steps must be completed: the identification of the intensity intervals, the calculation of median mean intensities per interval and the calculation of the median standard deviations per interval.

Identification of intensity intervals

First, the features are ordered from smallest to largest average intensity. The average intensity of each feature across all transformed spectra $\mathbf{x}_i$ ($i : 1, \dots, n$) is given by

$$\bar{x}_j = \frac{1}{n} \sum_{i=1}^n x_{ij}.$$

The vector of ordered averages is given by

$$\bar{\mathbf{x}}_{()} = (\bar{x}_{(1)}, \dots, \bar{x}_{(p)}).$$

Next, the vector of ordered averages $\bar{\mathbf{x}}_{()}$ is split into B equal length intervals with p_b features each.

Calculation of median mean intensities per interval

Let I_b be the indexes of the features j in a given interval b . The mean of all features in a given interval b is calculated for each spectrum i .

$$\bar{x}_{ib} = \frac{1}{p_b} \sum_{j \in I_b} x_{ij}.$$

$$\bar{\mathbf{x}}_i = (\bar{x}_{i1}, \dots, \bar{x}_{iB})$$

Each interval b is thus associated with a vector of means $\bar{\mathbf{x}}_b = (\bar{x}_{1b}, \dots, \bar{x}_{nb})$. The median of each vector $\bar{\mathbf{x}}_b$ is used to represent the central intensity of interval b :

$$\bar{x}_{b,med} = \text{median}_{1 \leq i \leq n}(\bar{x}_{ib}).$$

$$\bar{\mathbf{x}}_{med} = (\bar{x}_{1,med}, \dots, \bar{x}_{B,med})$$

Calculation of median standard deviations per interval

The standard deviation of the features in a given intensity interval b is calculated for each spectrum i .

$$S_{ib} = \sqrt{\frac{1}{p_b - 1} \sum_{j \in I_b} (x_{ij} - \bar{x}_{ib})^2}$$

$$\mathbf{S}_i = (S_{i1}, \dots, S_{iB})$$

Calculating standard deviations “horizontally” (i.e. within a given interval in the same spectrum i) seems to be rather non-intuitive, but Kohl et al. (2012) do not justify this choice.

Each interval b is associated with a vector of standard deviations $\mathbf{S}_b = (S_{1b}, \dots, S_{nb})$. The median of each vector $\mathbf{S}_b$ is used to represent the standard deviation of interval b :

$$S_{b,med} = \text{median}_{1 \leq i \leq n}(S_{ib}).$$

$$\mathbf{S}_{med} = (S_{1,med}, \dots, S_{B,med})$$

Plotting standard deviation against the rank of mean intensities

The logged median standard deviations, $\log_2(\mathbf{S}_{med})$, are plotted against the ranks of the median averages, $r(\bar{\mathbf{x}}_{med})$.

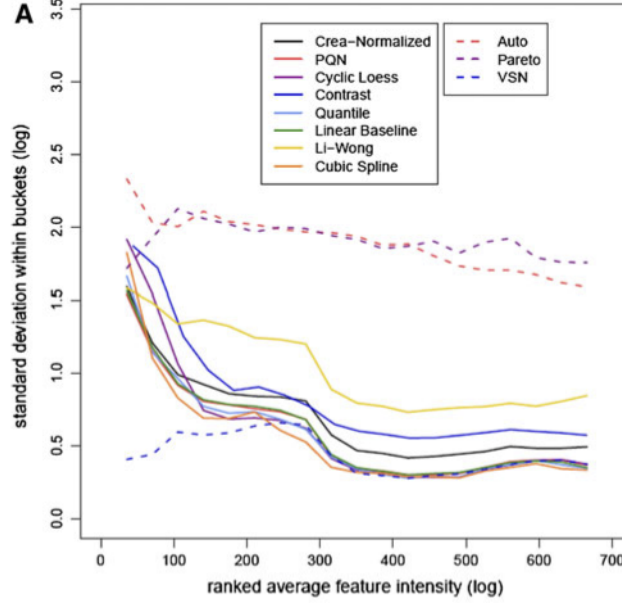


Figure 57: Logged median standard deviations ($\log_2(\mathbf{S}_{med})$) vs ranks of median average intensities, ($r(\bar{\mathbf{x}}_{med})$) (Kohl et al., 2012)

Ideally, $\log_2(S_{med})$ should be relatively low and constant across all intensities. Clearly, VSN is the only method with logged standard deviations that remain low and constant across all intensity intervals. While the logged standard deviations of Autoscaling and Pareto Scaling remain relatively constant, they are much larger than the other methods.

Discussion

While plotting standard deviations against the rank of spectral intensities is certainly an interesting way of evaluating intensity-dependent variance, the implementation proposed by Kohl et al. (2012) is problematic and seems unnecessarily complex. Rather than calculating the standard deviation of a single feature across multiple spectra, the standard deviation of several features from the same spectra is calculated. It is not clear how this calculation is representative of the relationship between feature variance and feature intensity. This method is also problematic because the scale of the standard deviations calculated depends on the NST methods used (different NST methods yield spectra with very different scales). Thus, it might be

inappropriate to compare methods using a graph such as the one in Figure 57.

Preserving and/or increasing the contrast between noise and signal

Methods

Preserving the contrast

A good normalization method’s performance should not come at the cost of reducing the biological signal. Kohl et al. (2012) added differing amounts of different “spike-ins” to a set of uniform urine samples (all urine samples came from a pool of combined urine samples, see 7.1.1). “Spike-ins” are artificially introduced metabolites of known concentrations. Because the non-spike-in features were of equal concentration across samples, the only variability between spectra for these features should be measurement error. On the other hand, spike-in features should be highly variable between samples.

Let $\mathbf{x}_i^{spike}$ be a vector of intensities for spectrum i including only those features j that are spike-ins. The vector of intensities for the remaining features j is defined as $\mathbf{x}_i^{non-spike}$. A ratio α_i for a given spectrum i is given by

$$\alpha_i = \frac{IQR(\mathbf{x}_i^{spike})}{IQR(\mathbf{x}_i^{non-spike})}.$$

For a given NST method, the ratios $\alpha_1, \dots, \alpha_n$ can be averaged and then compared with the average ratio in the absence of NST. If the average ratio, $\hat{\alpha}$, is smaller with a given method than it would be in the absence of NST, then that method reduces the signal to noise ratio and thus decreases the quality of the spectrum for later multivariate analysis.

Increasing the contrast

When spike-ins are present, their fold changes should be easily distinguished from fold changes associated with random variation. However, in raw data, small spike-in peaks can have lower intensities than noisy peaks. As a result, small spike-ins (or in a more realistic setting, potential biomarkers) can be overlooked, and noisy peaks can be identified as biomarkers. One goal of NST is thus to increase the distinction between spike-ins and noisy peaks, giving spike-ins relatively higher intensity than noisy peaks.

Kohl et al. (2012) identified the location of the spike-in peaks in their spectra before applying several NST methods to the them. Then they classified

intensities as “noise” or “spike-in” using a Support Vector Machine classifier and compared these classifications with the true labels, calculating the receiver operator characteristic (ROC) curves and the area under the curve (AUC) for these classifications.

Parsons et al. (2007) proposed another way of assessing the detection of biomarkers. Let the $n \times p$ matrix of spectra be given by $\mathbf{X}$. Each spectrum i in $\mathbf{X}$ belonged to one of $K = 2$ classes, $k : 1, 2$ (see 7.1.1). The matrix of spectra $\mathbf{X}$ can thus be subsetting, producing a different matrix for each class k , $\mathbf{X}_k$, with $k : 1, 2$. After performing a Principal Components Analysis (PCA) of the normalized spectra, they classified these spectra into the two different classes.

For each of the five largest features j in the PCA loadings plot (first and second principal components only), Parsons et al. (2007) compared the mean intensities of each class using a one-way ANOVA. Let x_{ijk} indicate the intensity of feature j in the i^{th} spectrum from the matrix of class k , $\mathbf{X}_k$. Let n_k be the number of spectra in $\mathbf{X}_k$. The mean intensity of feature j in class k is calculated as follows:

$$\bar{x}_{jk} = \frac{1}{n_k} \sum_{i \in X_k}^{n_k} x_{ijk}.$$

The overall mean intensity of feature j is calculated

$$\bar{\bar{x}}_j = \frac{1}{2} \sum_{k=1}^2 \bar{x}_{jk}.$$

The F-statistic for feature j is the Mean Square between classes divided by the Mean Square within classes, given by

$$MS_{within,j} = \sum_{k=1}^2 \frac{\sum_{i \in X_k} (x_{ijk} - \bar{x}_{jk})^2}{n_1 + n_2 - 2}$$

$$MS_{between,j} = \sum_{k=1}^2 n_k (\bar{x}_{jk} - \bar{\bar{x}}_j)^2$$

$$\frac{MS_{between,j}}{MS_{within,j}} \sim F_{j, df_1=1, df_2=\sum_k (n_k-1)}$$

NST methods that allow the detection of biomarkers should result in significant differences between classes for the largest features j in the PCA loadings plot.

Discussion

As with other methods previously presented, Kohl et al. (2012)’s criteria for evaluating the preservation and increase of the signal to noise ratio require the availability of spike-in data sets. The use of the IQR as a scaling factor is also non-intuitive but left unjustified by the authors.

Parsons et al. (2007)’s F-statistic method seems to be an interesting approach to evaluating the increase in contrast between noise and signal.

Removal of dilution effects (urine spectra)

Methods

In silico modification of spectra can be a useful way of evaluating the accurate estimation of a dilution factor in urine spectra. Starting with a “golden spectra” taken from their set of spectra, Dieterle et al. (2006) created different simulated spectral data sets (see 7.1.1), normalized each spectrum to a constant sum and then artificially set one of the features to a constant value (an intensity of 0.5). After “diluting” each data set *in silico*, various NST methods were applied. The quality of each NST method was evaluated by dividing the intensity of the artificial feature after NST, $x_{i,artificial}$, by the actual intensity of the feature, $\mu_{artificial}$ (0.5). The most optimal normalization procedure would yield a quotient of 1.

Torgrip et al. (2008) also evaluated the performance of normalization methods using simulations (see 7.1.1). For each original spectrum i in their data set, Torgrip et al. applied a different dilution factor, resulting in 50 simulated spectra for each original spectrum. A given original spectrum i with dilution factor k , $k : 1, \dots, K = 50$, is noted as $\mathbf{x}_{ik}$. Let $\hat{\alpha}_{ik}$ be the estimated dilution factor of spectrum $\mathbf{x}_{ik}$. For each NST method, Torgrip et al. (2008) regressed the vector of estimated dilution factors, $\hat{\alpha}_k = (\hat{\alpha}_{1k}, \dots, \hat{\alpha}_{nk})$, onto the vector of known dilution factors, α_k , and calculated the RMSE of this regression, $RMSE_k$. The mean RMSE across all dilution factors k was then calculated. The most optimal NST method was the one with the smallest mean RMSE, $\overline{RMSE}$.

$$\overline{RMSE} = \frac{1}{K} \sum_{k=1}^K RMSE_k$$

To determine whether different NST methods were significantly different from each other, Torgrip et al. also calculated an F-statistic. Let $\hat{\alpha}_{il}$ be the vector of dilution factors $\hat{\alpha}_{i1}, \dots, \hat{\alpha}_{iK}$ estimated after application of a given NST method l and α_i be a vector of known dilution factors. The difference

between these two vectors is given by

$$\Delta_{il} = \hat{\alpha}_{il} - \alpha_i$$

$$\Delta_{il} = (\Delta_{i1l}, \dots, \Delta_{iKl})$$

Let $\bar{\Delta}_{il} = \frac{1}{K} \sum_{k=1}^K \Delta_{ikl}$ be the mean of the vector of differences Δ_{il} after application of NST method l and S_{il}^2 be the estimated variance of Δ_{il} , calculated as follows:

$$S_{il}^2 = \frac{1}{K-1} \sum_{k=1}^K (\Delta_{ikl} - \bar{\Delta}_{il})^2$$

For a pair of different methods l and l' , the F-statistic is given by

$$F_i = \frac{S_{il}^2}{S_{il'}^2}$$

where $F_i \sim F_{df1=K-1, df2=K-1}$. Method l is considered significantly better than method l' if its estimated variance is significantly smaller.

Discussion

Torgrip et al. (2008)'s criterion for evaluating the estimation of a dilution factor yields individual F-statistics for each spectrum i . Thus, it would be interesting to develop some kind of global statistic summarizing the performance across all spectra.

Classification performance Many authors in the literature evaluate the quality of NST methods based on subsequent classification performance.

Methods

For each NST method tested on their classification samples (see 7.1.1), Parsons et al. (2007) performed Principal Component Analysis (PCA) on the spectra post-NST. Then they classified samples by applying Linear Discriminant Analysis (LDA) to the PCA scores for principal components one and two. Parsons et al. (2007) compared the performance of the different methods by cross-validation (percent of correct classifications).

Kohl et al. (2012) tested classification performance with differing numbers of total spectra, $n = 20, 40, 60, 80, 100$, from a classification data set (see 7.1.1). After applying NST methods to n randomly selected spectra from their data set, they performed classification using Support Vector Machine nested cross validation and calculated a ROC curve and the corresponding AUC. High AUC values indicate good classification performance.

Discussion

Both Parsons et al. (2007)'s and Kohl et al. (2012)'s classification performance measures seem useful for evaluating quality of classification. Parsons et al. (2007)'s measure is limited by the fact that it requires the selection of a subset of the principal components before classification is performed (in this case, 2 principal components).

7.2 Implementation and evaluation of certain NST methods

7.2.1 Application of the pre-normalization step to Linear Baseline Normalization, PQN and VSN

A given spectrum composed of a reduced set of features is given by

$$\mathbf{x}_{i*} = (x_{i1}, \dots, x_{ip*})$$

where $j^* : 1, \dots, p^*$ is the index corresponding to one of the features remaining in the data set.

Linear Baseline Normalization

For Linear Baseline Normalization, each spectrum $\mathbf{x}_i$ is normalized by dividing it by $\frac{x_{i,med}}{\text{median}_{1 \leq j \leq p}(z_j)}$, where $\mathbf{z}$ is a reference spectrum. When applying the pre-normalization step, this ratio is calculated using the reduced data set. In other words, the median of spectrum i , $x_{i,med}$, becomes

$$x_{i,med*} = \text{median}_{1 \leq j^* \leq p^*}(x_{ij*})$$

and the reference spectrum, $\mathbf{z}$, is recalculated using the reduced data set

$$\mathbf{z}_* = (\text{median}_{1 \leq i \leq n}(x_{i1}), \dots, \text{median}_{1 \leq i \leq n}(x_{ip*}))$$

$$\mathbf{z}_* = (z_1, \dots, z_{p*}),$$

resulting in a different median value

$$\text{median}_{1 \leq j^* \leq p^*}(z_{j*}).$$

PQN

For PQN, all features in $\mathbf{x}_i$ are divided by Q_i^{prob} (see Section 2.2.4). When applying the pre-normalization step, Q_i^{prob} is calculated using the reduced data set. First, a reference spectrum $\mathbf{z}_*$ is calculated

$$\mathbf{z}_* = (\text{median}_{1 \leq i \leq n}(x_{i1*}), \dots, \text{median}_{1 \leq i \leq n}(x_{ip*}))$$

Each feature in the vector $\mathbf{x}_{i*}$ is divided by the corresponding feature in the reference spectrum $\mathbf{z}_*$, yielding a vector of quotients

$$Q_{ij*} = \frac{x_{ij*}}{z_{ij*}}$$

$$\mathbf{Q}_{i*} = (Q_{i1*}, \dots, Q_{ip*}).$$

Q_{i*}^{prob} is defined as the median quotient

$$Q_{i*}^{prob} = \text{median}_{1 \leq j^* \leq p^*}(Q_{ij*}).$$

VSN

VSN depends on the estimation of a parameter λ (see Section 2.2.6). When applying pre-normalization, the parameter λ is optimized using the reduced data set, meaning that the function to minimize, $S(h, \lambda)$, becomes

$$S(h_*, \lambda) = \sum_{i=1}^n SSE_{i*} = \sum_{i=1}^n \sum_{j^*=1}^{p^*} \left(h_{ij*} - \frac{1}{n} \sum_{i=1}^n h_{ij*} \right)^2$$

7.2.2 Performance measures

Principal Component Analysis (PCA)

A total of p^* eigenvectors are calculated based on the data matrix $\mathbf{X}$, where a given eigenvector is noted $\mathbf{u}_{j^*}$, $j^* : 1, \dots, p^*$. Each row $\mathbf{v}_i^*$ of the reduced data matrix is composed of linear combinations of spectrum i with each eigenvector $\mathbf{u}_{j^*}$. A given dimension j^* of row i is the linear combination of the original spectrum i with eigenvector j^* :

$$\mathbf{v}_{j^*} = \mathbf{x}_i^\top \mathbf{u}_{j^*}$$

$$\mathbf{v}_i = (\mathbf{v}_{i1}, \dots, \mathbf{v}_{ip^*})$$

Inertia between and within groups

The total inertia of a matrix is a sum of the Between-group Inertia (B) and Within-group Inertia (W). The best NST method is the one with the largest B and the smallest W. Let x_{ijk} be the intensity of feature j for the i^{th} spectrum associated with subject k , with $j : 1, \dots, p$, $i : 1, \dots, n$ and $k : 1, \dots, K$. The average intensity of feature j for subject k is given by $\bar{x}_{jk} = \frac{1}{n} \sum_{i=1}^n x_{ijk}$. The overall average intensity for feature j is given by $\bar{\bar{x}}_j = \frac{1}{K} \sum_{k=1}^K \bar{x}_{jk}$.

The total inertia (T) is calculated as follows:

$$T = \frac{1}{K} \sum_{k=1}^K \frac{1}{n} \sum_{i=1}^n \frac{1}{p} \sum_{j=1}^p (x_{ijk} - \bar{\bar{x}}_j)^2$$

The total inertia can be expressed as the sum of the between-group inertia (B) and the within-group inertia (W).

$$T = B + W$$

where

$$B = \sum_{k=1}^K n \sum_{j=1}^p (\bar{x}_{jk} - \bar{\bar{x}}_j)^2$$

and

$$W = \sum_{k=1}^K \sum_{i=1}^n \sum_{j=1}^p (x_{ijk} - \bar{x}_{jk})^2$$

Dunn Index (DI)

For all pairs of clusters C_l and C_m ($C_l \neq C_m$), the distance between each spectrum i in C_l and each spectrum i' in C_m is calculated. The numerator of the Dunn Index represents the smallest distance between all pairs of spectra in all pairs of clusters C_l and C_m . This value is normalized by the maximum intra-cluster distance, Δ_k .

The Dunn Index for a given partition $\mathbf{C}$ is given by

$$DI_C = \frac{\min_{C_l, C_m, C_l \neq C_m} \{ \min_{i \in C_l, i' \in C_m} \text{dist}(\mathbf{x}_{il}, \mathbf{x}_{i'm}) \}}{\max_k \Delta_k} \quad (17)$$

Davies-Bouldin Index (DBI)

Let $\mathbf{A}_k$ be the medoid of cluster C_k . The variation of cluster C_k is represented by Γ_k :

$$\Gamma_k = \sqrt{\frac{1}{|C_k|} \sum_{i=1}^{|C_k|} \|\mathbf{x}_{ik} - \mathbf{A}_k\|_2}$$

The variation between two different clusters C_l and C_m is represented by $\gamma(C_l, C_m)$ and is calculated based on the medoid of each cluster:

$$\gamma(C_l, C_m) = \|\mathbf{A}_l - \mathbf{A}_m\|_2$$

For all pairs of clusters, the ratio between the sum of intra-cluster variation for both clusters (numerator of Equation 18) and the inter-cluster variation (denominator of Equation 18) is calculated. For each cluster C_k , the cluster C_m that maximizes this ratio is selected. These ratios are then averaged.

$$DBI_C = \frac{1}{K} \sum_{k=1}^K \max_{m, m \neq k} \left\{ \frac{\Gamma_k + \Gamma_m}{\gamma(C_k, C_m)} \right\} \quad (18)$$

The ratio in brackets in Equation 18 is maximized when the denominator is minimal and the numerator is maximal. The denominator is minimal when the medoids from both clusters are close and the numerator is maximal when the sum of intra-cluster variation is large. The index is minimal when, on average, the intra-cluster variation is small for two minimally distant clusters.

Rand Index (RI)

Let m_a be the number of pairs of spectra that are clustered together in both $\mathbf{C}^1$ and $\mathbf{C}^2$. Let m_b be the number of pairs of spectra that are clustered together in $\mathbf{C}^1$ but not in $\mathbf{C}^2$. Let m_c be the number of pairs of spectra that are clustered together in $\mathbf{C}^2$ but not in $\mathbf{C}^1$. Let m_d be the number of pairs of spectra that are in different clusters in both $\mathbf{C}^1$ and $\mathbf{C}^2$.

Consequently, $m_a + m_d$ represents the number of agreements between $\mathbf{C}^1$ and $\mathbf{C}^2$, $m_b + m_c$ represents the number of disagreements between $\mathbf{C}^1$ and $\mathbf{C}^2$ and $m_a + m_b + m_c + m_d$ represents the total number of agreements and disagreements.

The Rand Index is given by

$$RI_{C^1, C^2} = \frac{m_a + m_d}{m_a + m_b + m_c + m_d} \quad (19)$$

PLS-DA

Partial Least Squares (PLS) is a type of modeling that characterizes the relationship between a matrix of explanatory variables and a matrix of response variables. In this case, the matrix of explanatory variables used was an $n \times p$ matrix $\mathbf{X}$ (n spectra, p variables) containing all normalized, scaled or transformed spectra and the matrix of dummy response variables was an $n \times K$ matrix $\mathbf{Y}$ (n spectra, K dummy response variables). For a given spectrum i , $y_{ik} = 1$ if spectrum i was associated with subject k and $y_{ik} = 0$ otherwise.

Multiple coefficient of determination

Let $\bar{y}_k$ be the mean of $\mathbf{y}_k$ (column k in the matrix $\mathbf{Y}$), where $\bar{y}_k = \frac{1}{n} \sum_{i=1}^n y_{ik}$. The sum of squares for class k , SS_k , is calculated $SS_k = \sum_{i=1}^n (y_{ik} - \bar{y}_k)^2$. Let $\hat{y}_{ik}$ be the predicted value of dummy variable k for spectrum i . The fitted residual sum of squares, RSS , is calculated $RSS_k = \sum_{i=1}^n (y_{ik} - \hat{y}_{ik})^2$. The multiple coefficient of determination for a given subject k is thus:

$$R_k^2 = \frac{1 - RSS_k}{SS_k}$$

Cross-validated R^2

The predictive residual sum of squares, $PRESS$, is calculated using cross-validation. The spectra are divided into M partitions ($m : 1, \dots, M$) of equal size. For each iteration m , a PLS model is estimated using all spectra except those in partition m . The predicted values $\hat{y}_{ik}$ are then calculated for all spectra i belonging to partition m . The predictive residual sum of squares for response variable k , $PRESS_k$, is calculated as follows:

$$PRESS_k = \sum_{i \in m} \sum_{k=1}^K (\hat{y}_{ik} - \bar{y}_{ik})^2$$

The cross-validated R^2 (Q^2) is thus given by

$$Q^2_k = \frac{1 - PRESS_k}{SS_k}$$

For more information concerning PLS, please refer to Wold et al. (2001).

7.2.3 PCA plots

Urine data

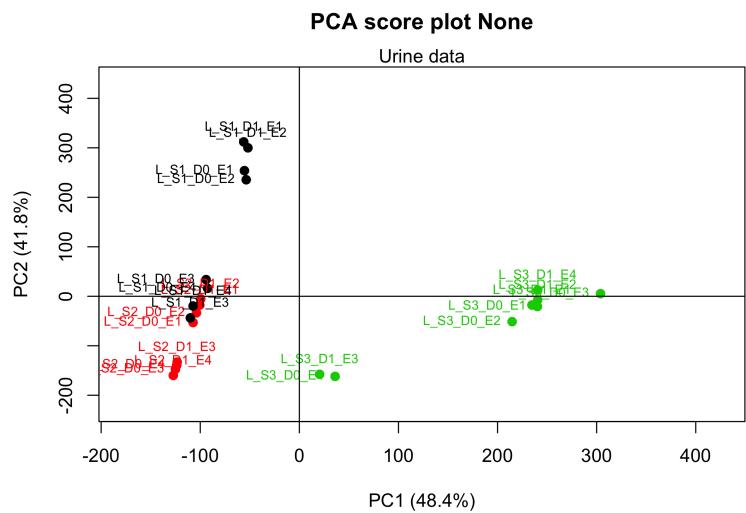


Figure 58: No NST method (PC1 and PC2)

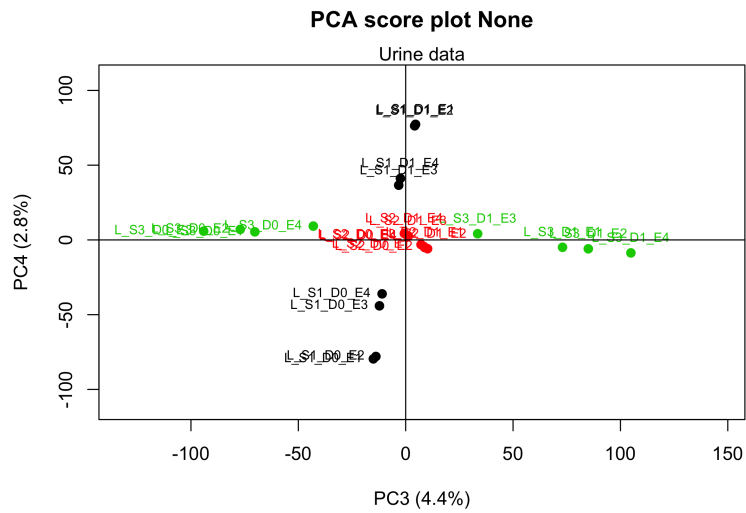


Figure 59: No NST method (PC3 and PC4)

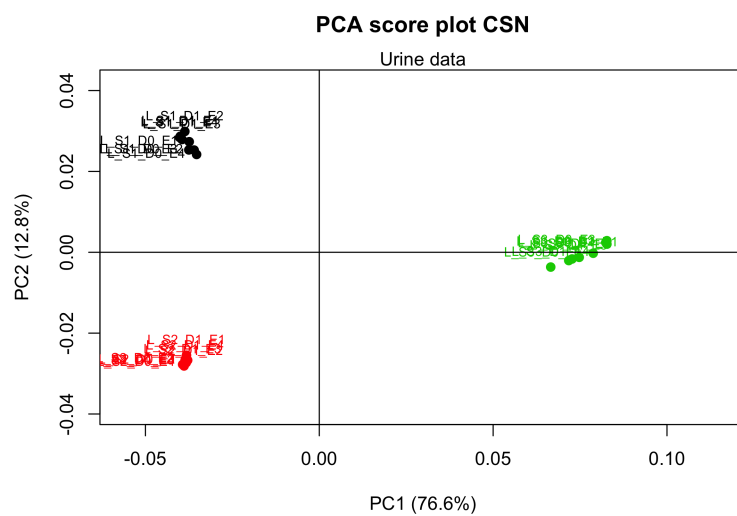


Figure 60: CSN (PC1 and PC2)

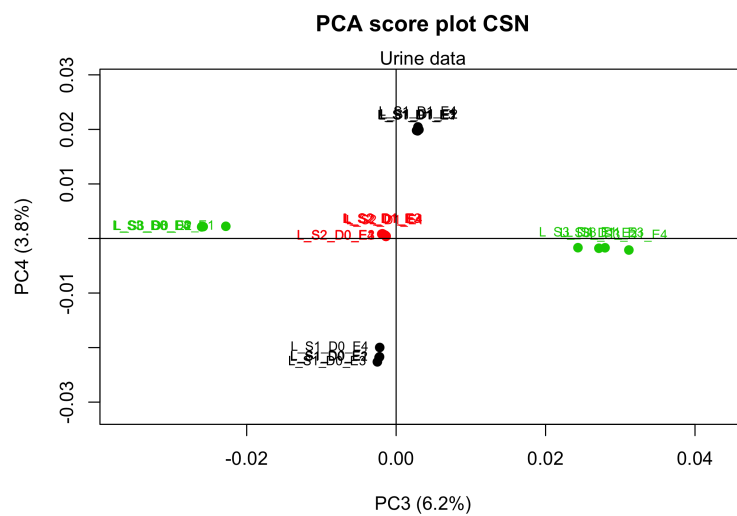


Figure 61: CSN (PC3 and PC4)

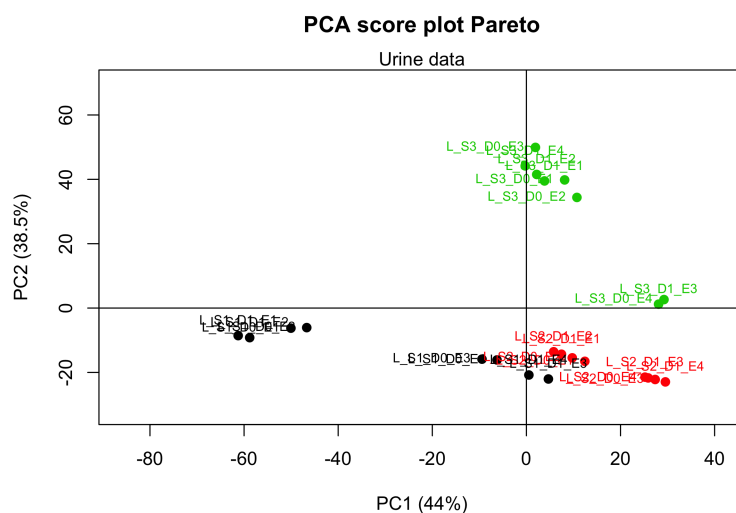


Figure 62: Pareto Scaling (PC1 and PC2)

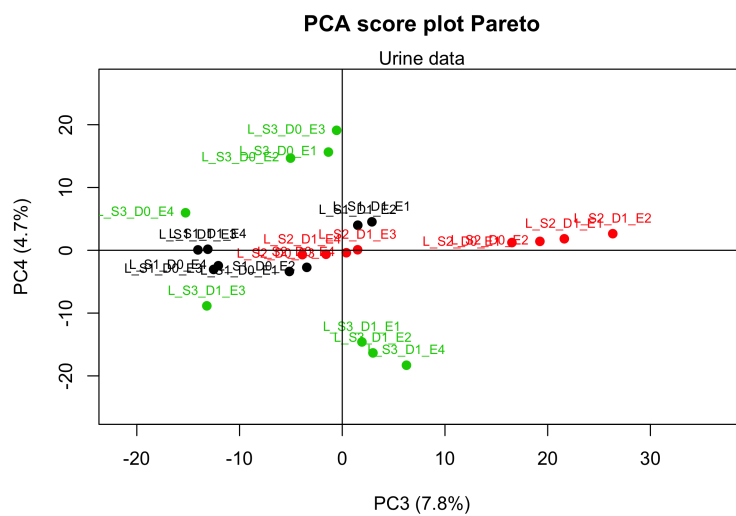


Figure 63: Pareto Scaling (PC3 and PC4)

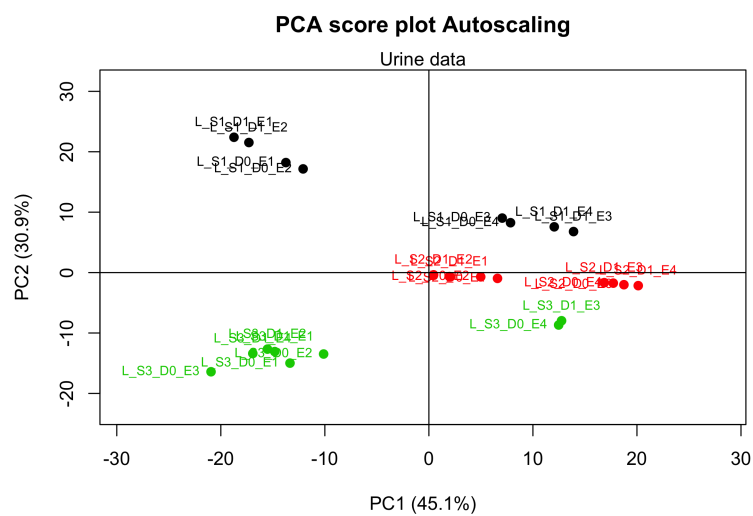


Figure 64: Autoscaling (PC1 and PC2)

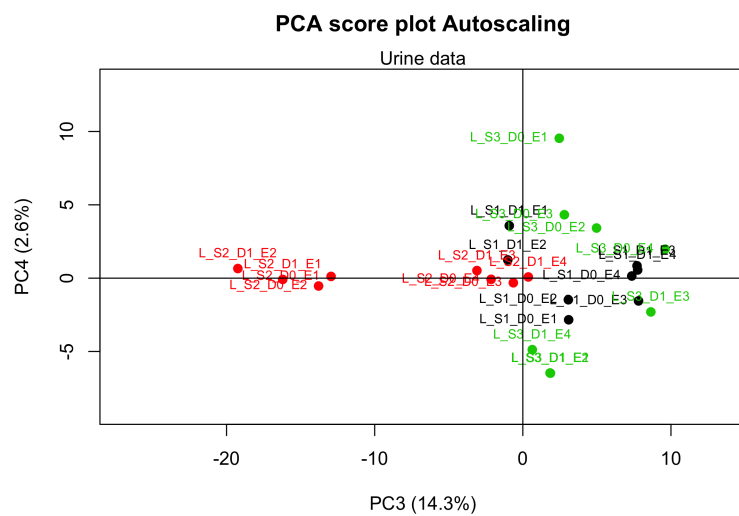


Figure 65: Autoscaling (PC3 and PC4)

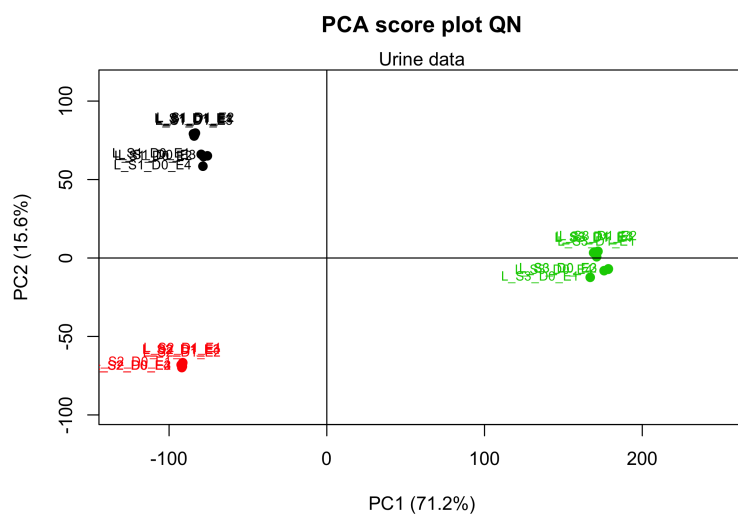


Figure 66: QN (PC1 and PC2)

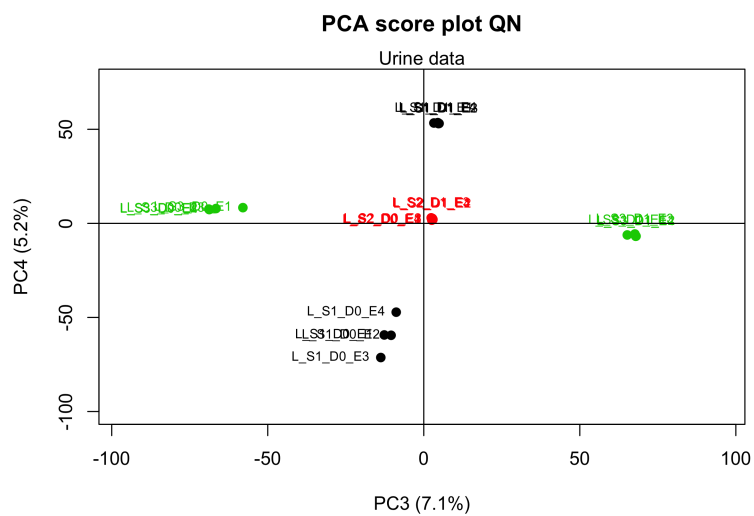


Figure 67: QN (PC3 and PC4)

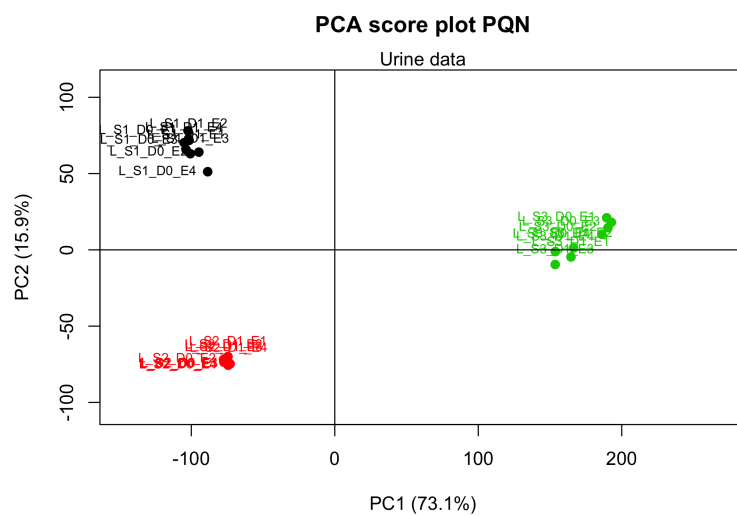


Figure 68: PQN (PC1 and PC2)

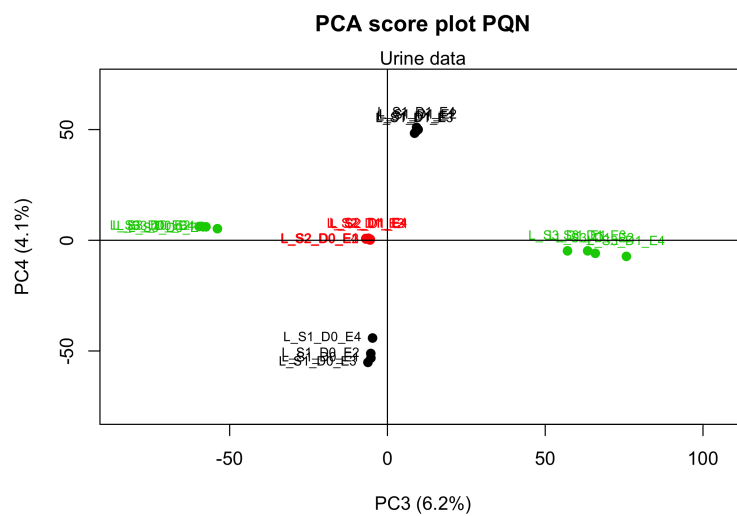


Figure 69: PQN (PC3 and PC4)

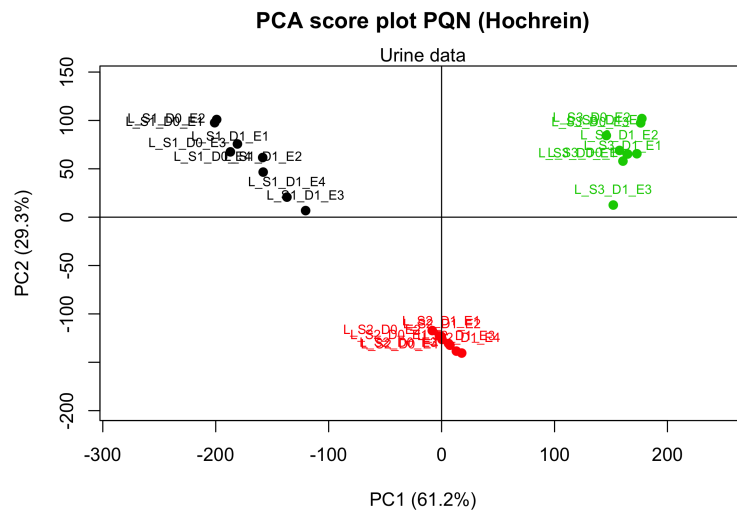


Figure 70: PQN with Hochrein et al. (2015)'s pre-normalization step (PC1 and PC2)

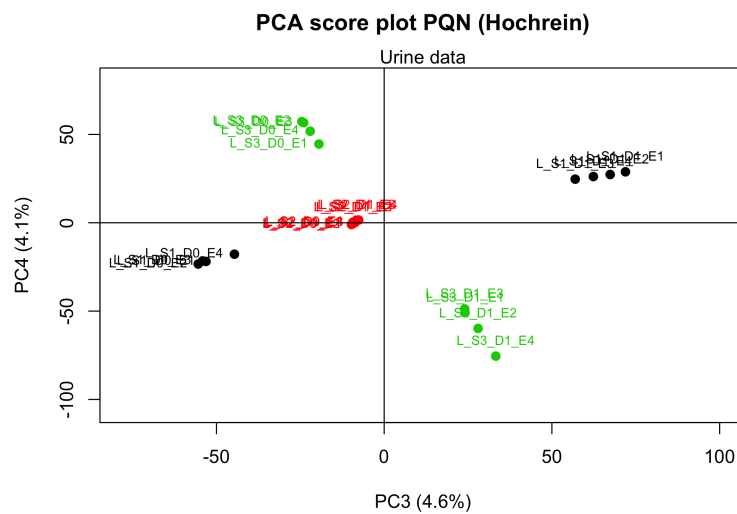


Figure 71: PQN with Hochrein et al. (2015)'s pre-normalization step (PC3 and PC4)

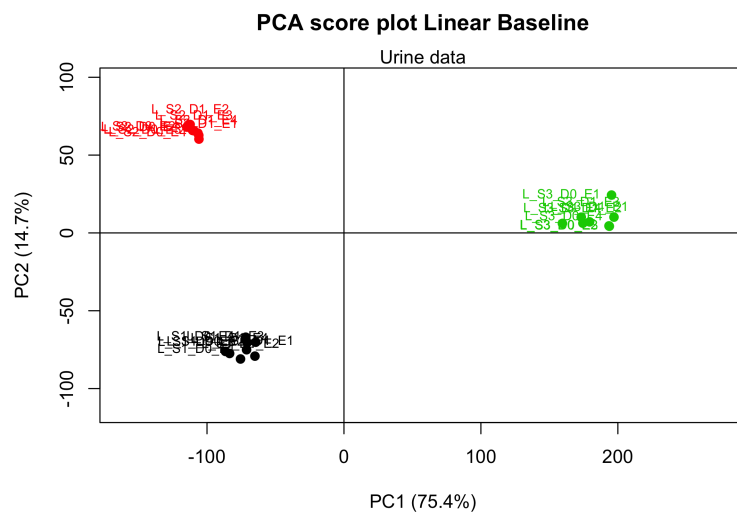


Figure 72: Linear Baseline (PC1 and PC2)

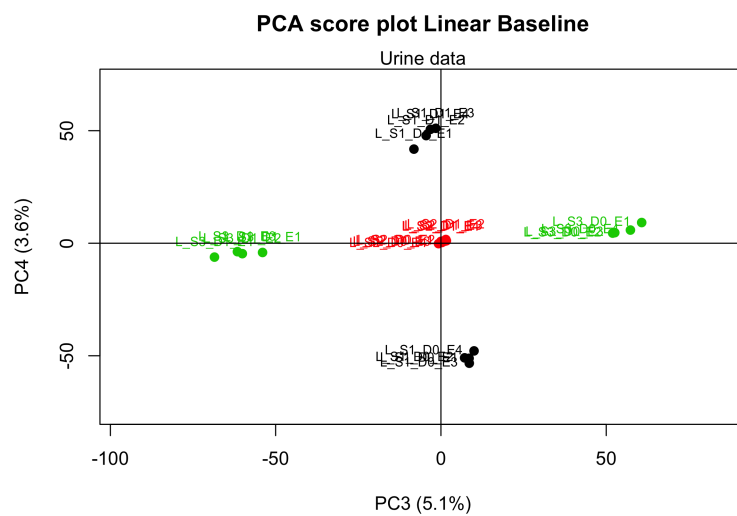


Figure 73: Linear Baseline (PC3 and PC4)

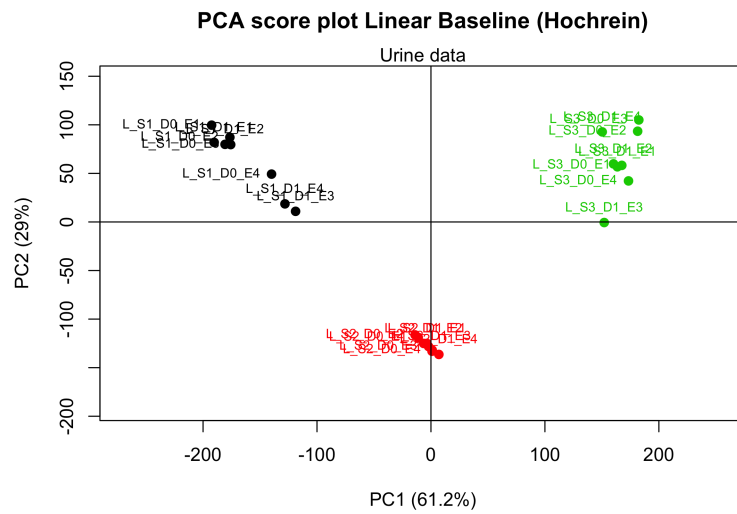


Figure 74: Linear Baseline with Hochrein et al. (2015)'s pre-normalization step (PC1 and PC2)

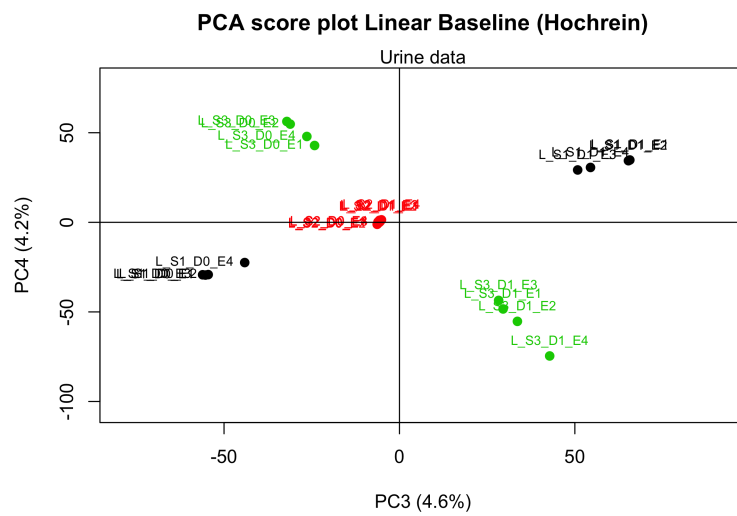


Figure 75: Linear Baseline with Hochrein et al. (2015)'s pre-normalization step (PC3 and PC4)

Serum data

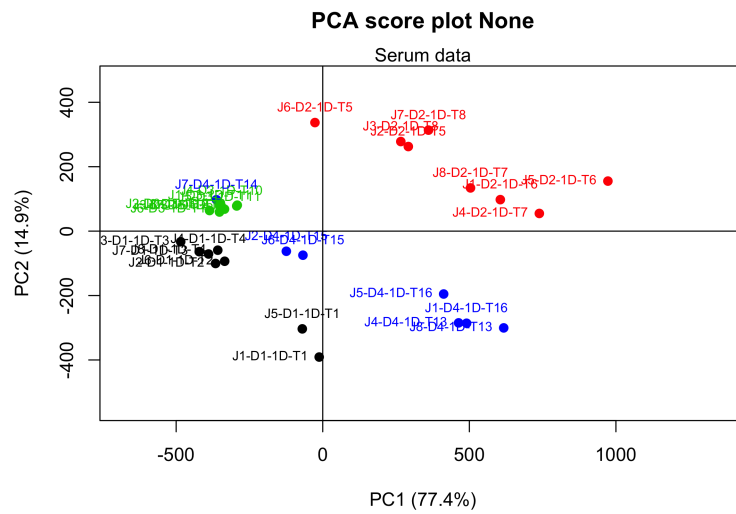


Figure 76: No NST method (PC1 and PC2)

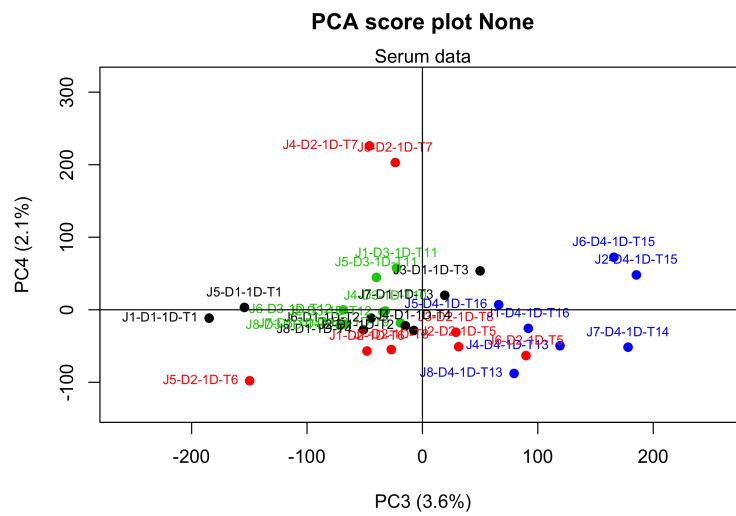


Figure 77: No NST method (PC3 and PC4)

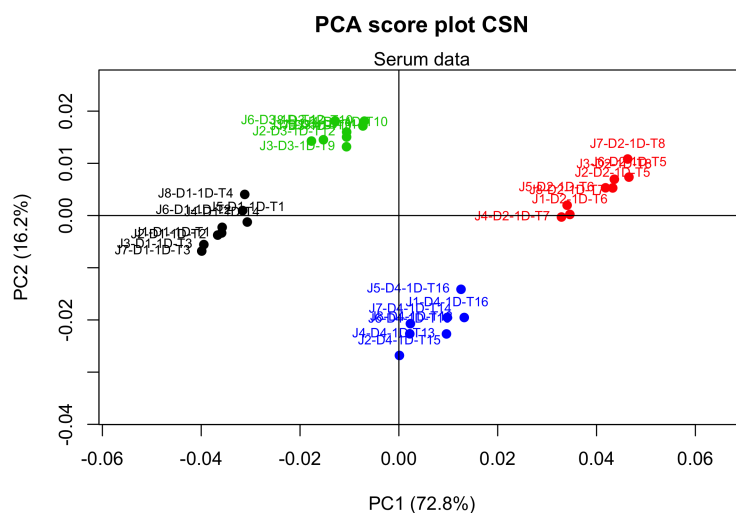


Figure 78: CSN (PC1 and PC2)

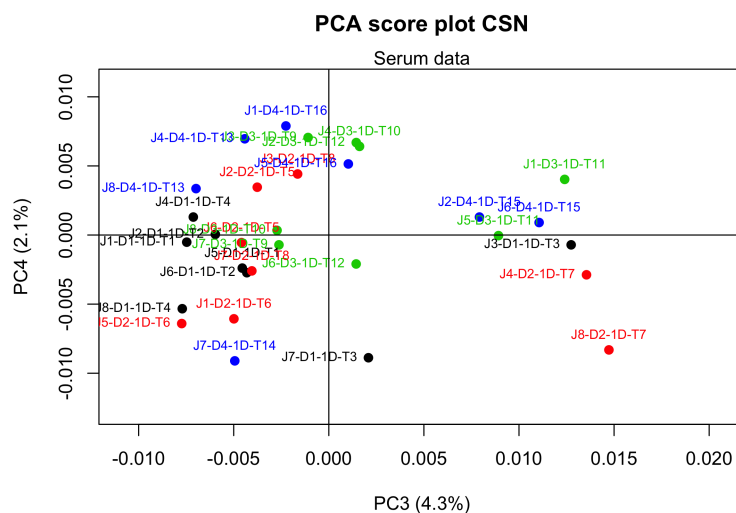


Figure 79: CSN (PC3 and PC4)

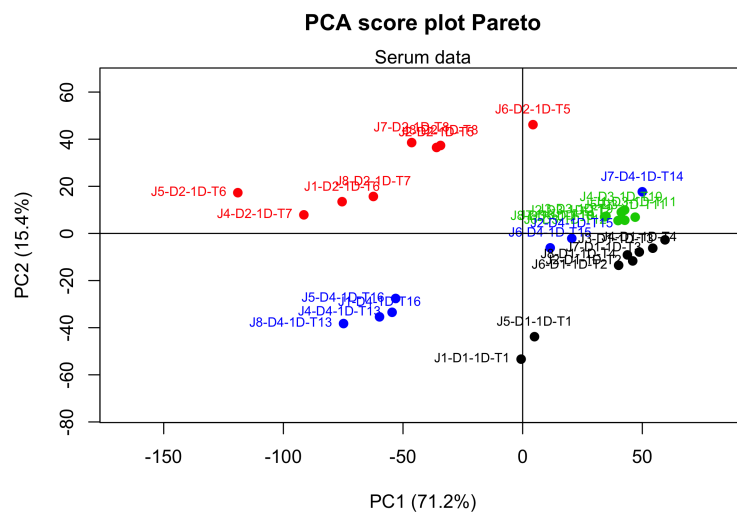


Figure 80: Pareto Scaling (PC1 and PC2)

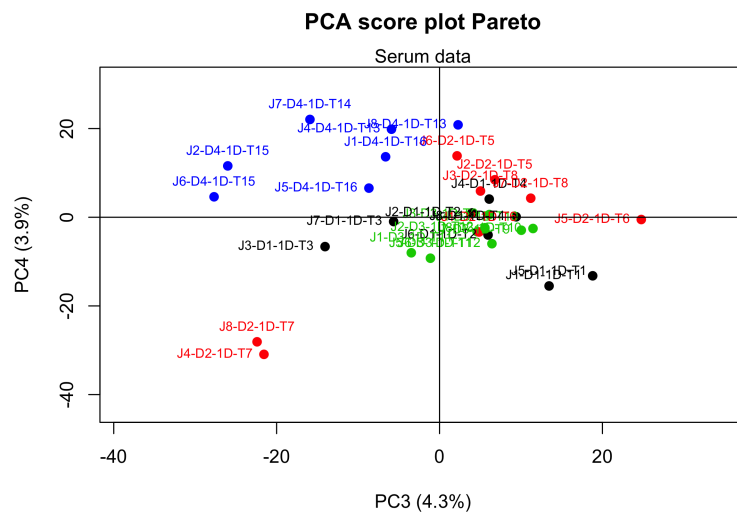


Figure 81: Pareto Scaling (PC3 and PC4)

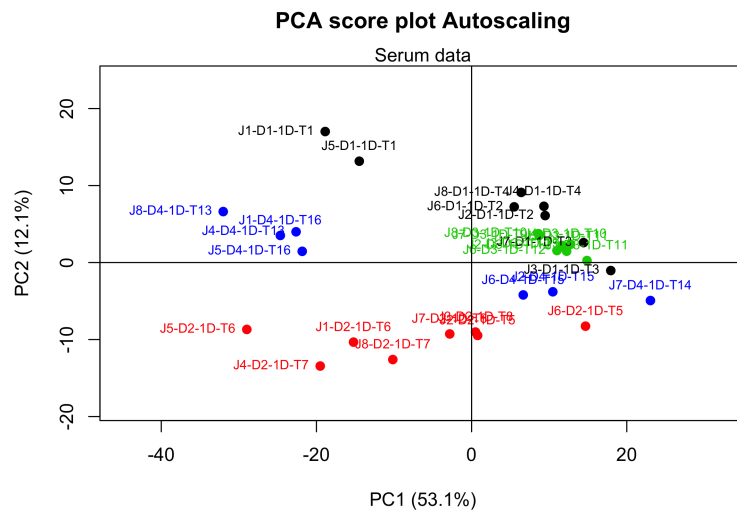


Figure 82: Autoscaling (PC1 and PC2)

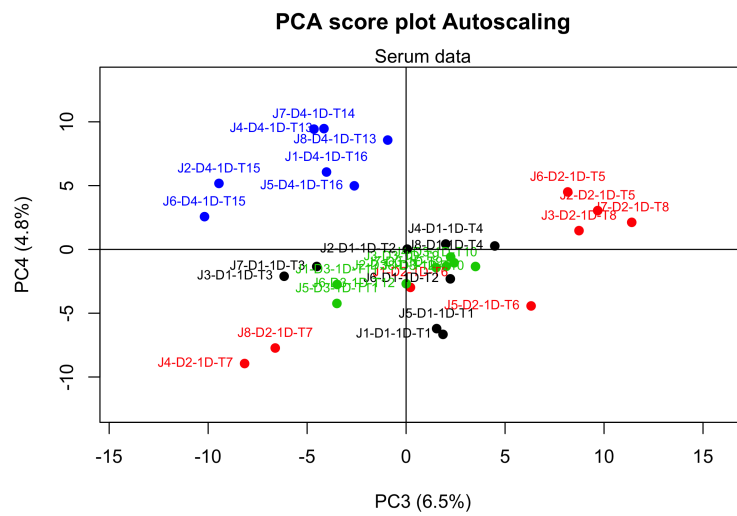


Figure 83: Autoscaling (PC3 and PC4)

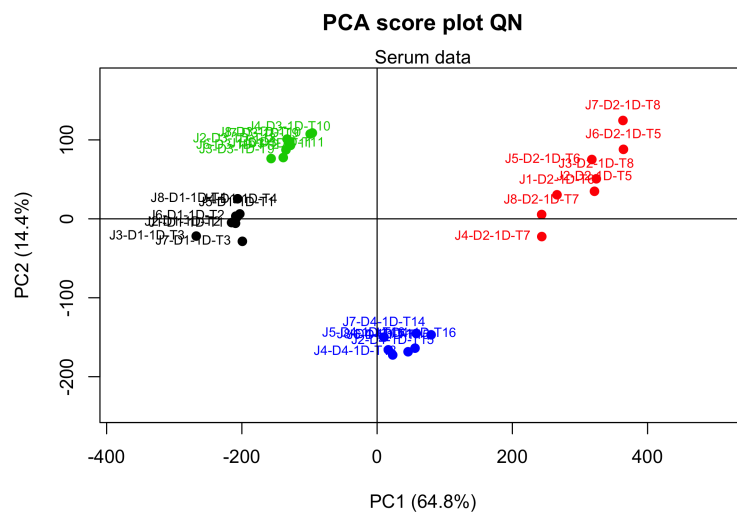


Figure 84: QN (PC1 and PC2)

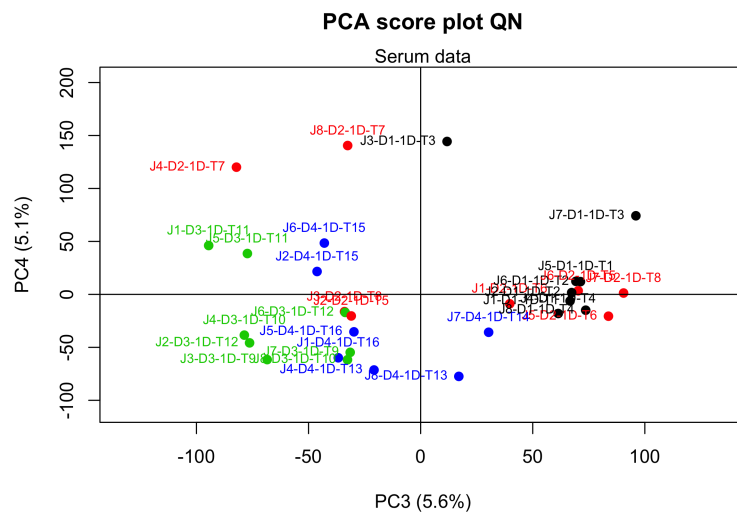


Figure 85: QN (PC3 and PC4)

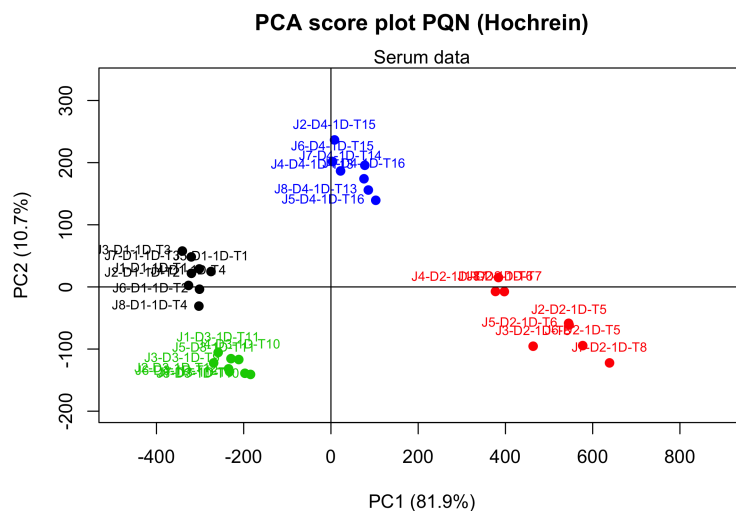


Figure 88: PQN with Hochrein et al. (2015)'s pre-normalization step (PC1 and PC2)

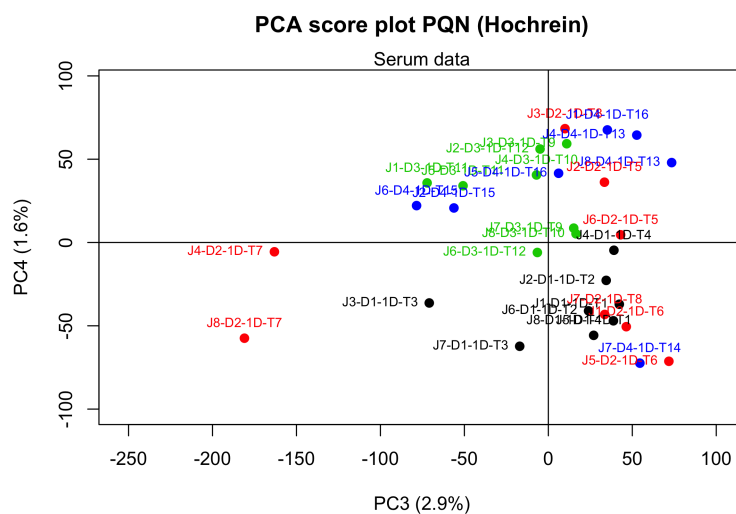


Figure 89: PQN with Hochrein et al. (2015)'s pre-normalization step (PC3 and PC4)

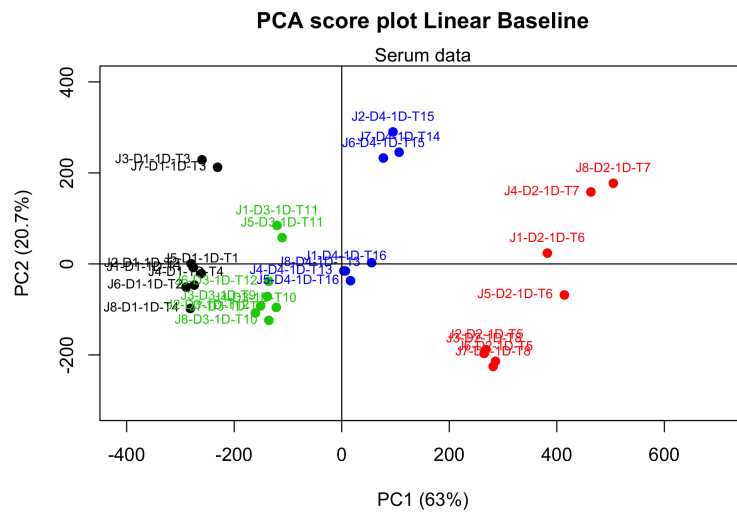


Figure 90: Linear Baseline (PC1 and PC2)

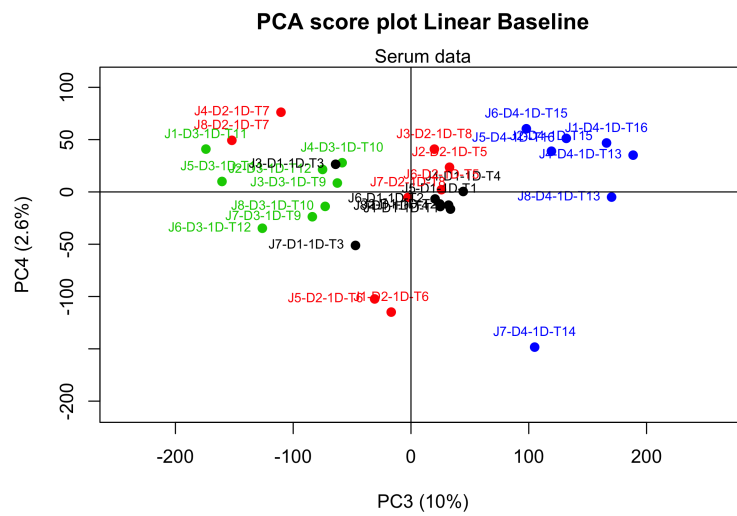


Figure 91: Linear Baseline (PC3 and PC4)

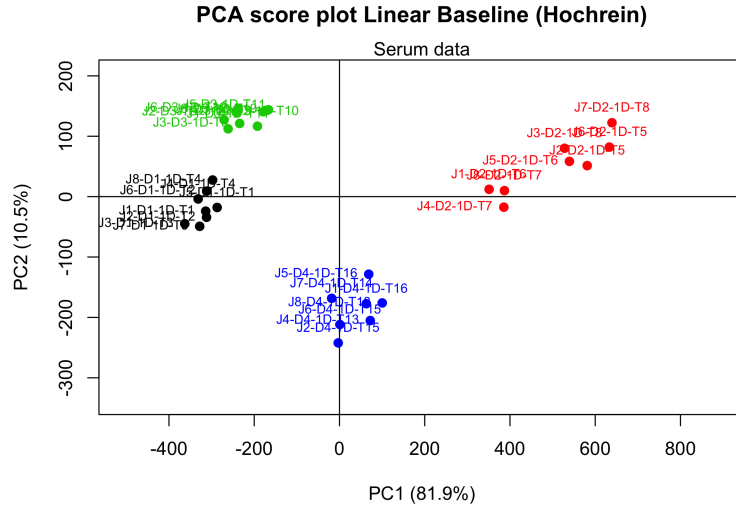


Figure 92: Linear Baseline with Hochrein et al. (2015)'s pre-normalization step (PC1 and PC2)

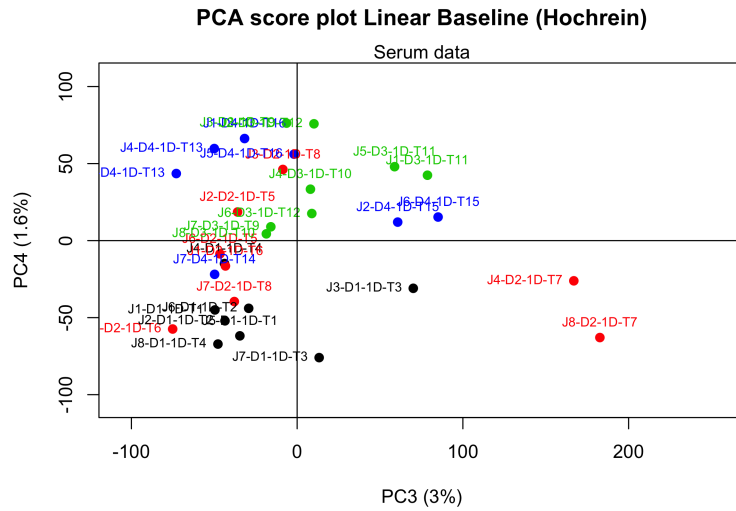


Figure 93: Linear Baseline with Hochrein et al. (2015)'s pre-normalization step (PC3 and PC4)

7.3 Implementation and evaluation of certain baseline correction methods

7.3.1 Additional boxplots

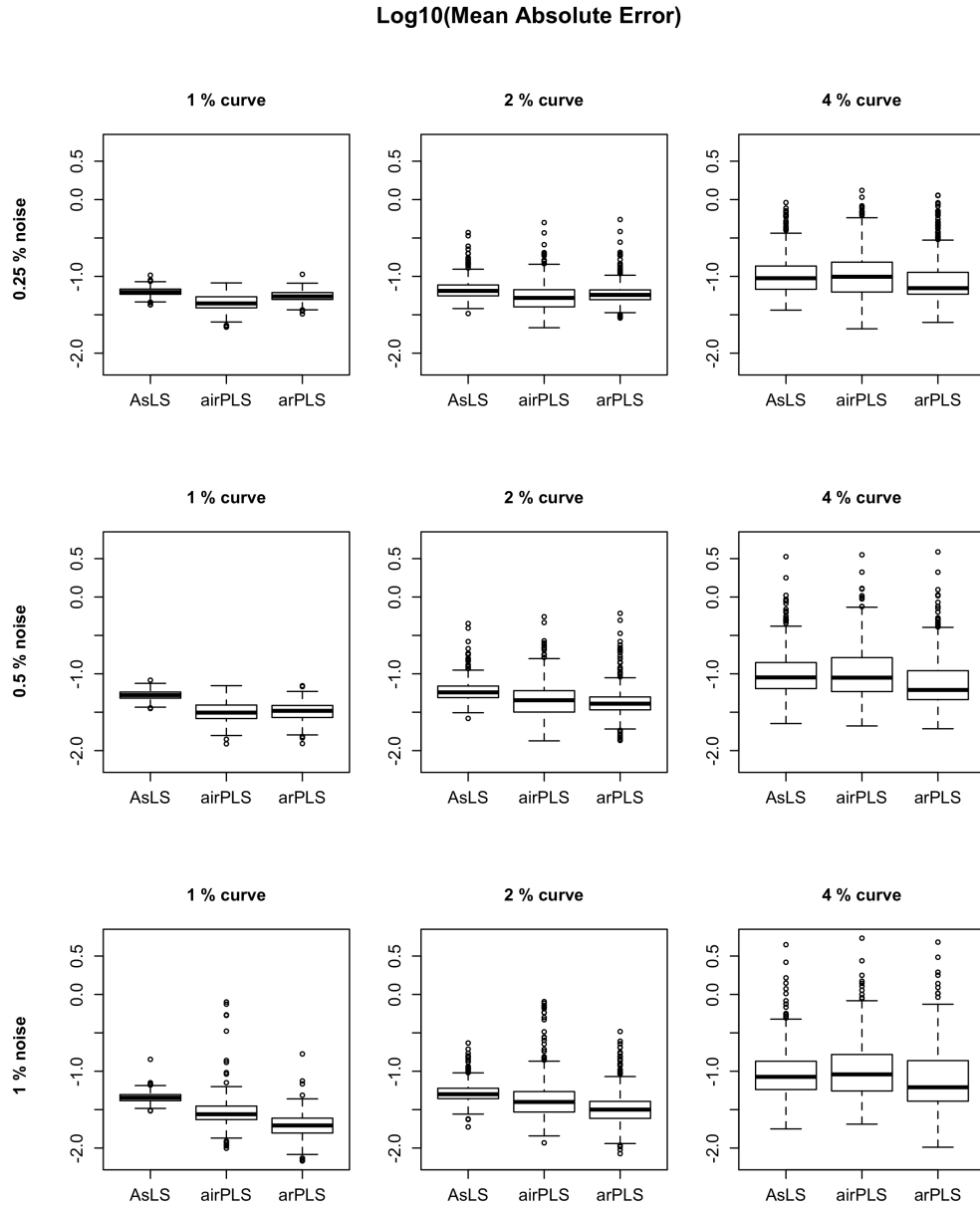


Figure 94: Normalized, log-transformed MAE for different parameter settings

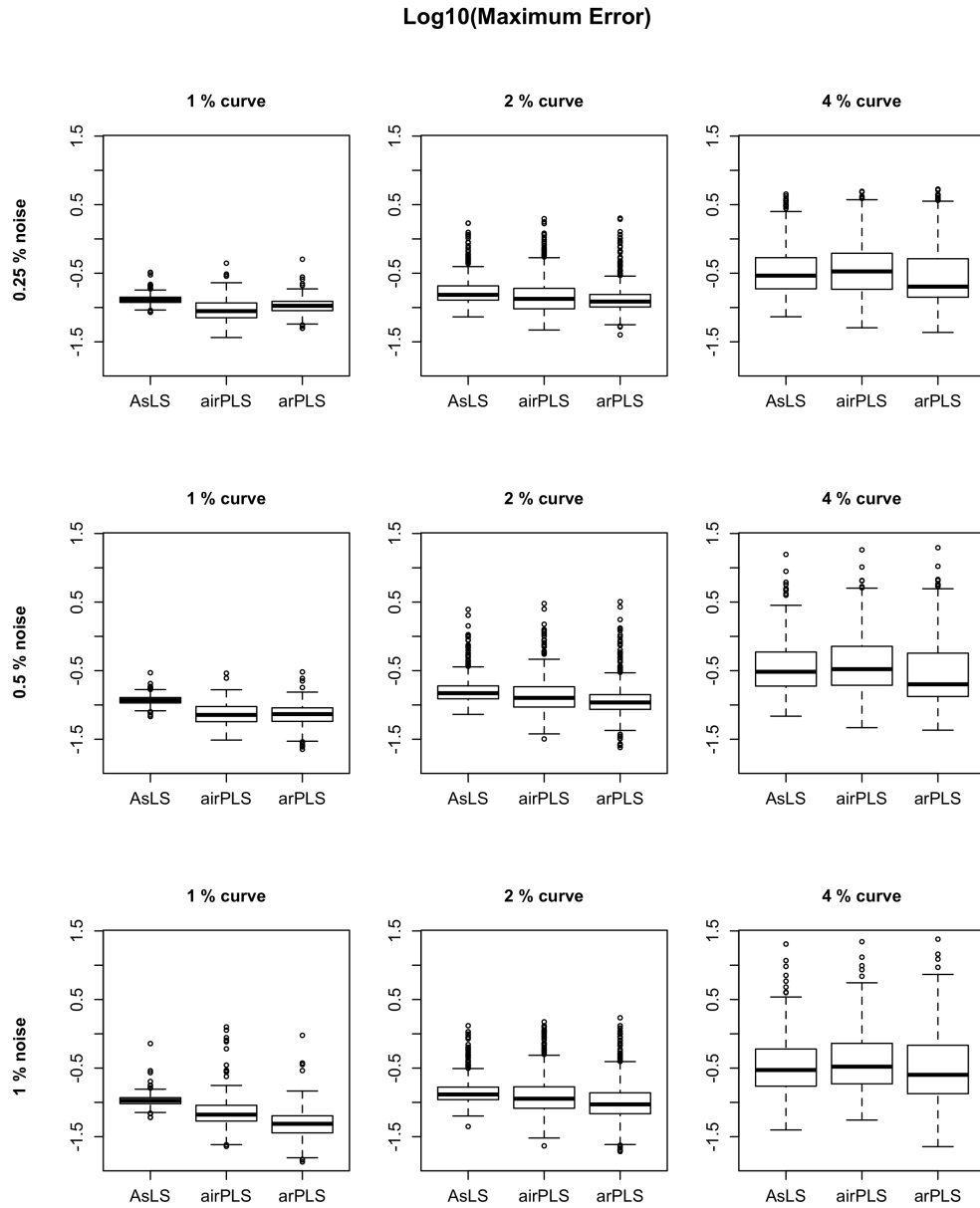


Figure 95: Normalized, log-transformed Max Error for different parameter settings

References

- Sung-June Baek, Aaron Park, Young-Jin Ahn, and Jaebum Choo. Baseline correction using asymmetrically reweighted penalized least squares smoothing. *Analyst*, 140(1):250–257, 2015.
- Benjamin M Bolstad, Rafael A Irizarry, Magnus Åstrand, and Terence P. Speed. A comparison of normalization methods for high density oligonucleotide array data based on variance and bias. *Bioinformatics*, 19(2):185–193, 2003.
- David Chang, Cory D Banack, and Sirish L Shah. Robust baseline correction algorithm for signal dense nmr spectra. *Journal of Magnetic Resonance*, 187(2):288–292, 2007.
- J Carlos Cobas, Michael A Bernstein, Manuel Martín-Pastor, and Pablo García Tahoces. A new general-purpose fully automatic baseline-correction procedure for 1d and 2d nmr data. *Journal of Magnetic Resonance*, 183(1):145–151, 2006.
- Frank Dieterle, Alfred Ross, Götz Schlotterbeck, and Hans Senn. Probabilistic quotient normalization as robust method to account for dilution of complex biological mixtures. application in 1h nmr metabonomics. *Analytical chemistry*, 78(13):4281–4290, 2006.
- Wolfgang Dietrich, Christian H Rüdel, and Markus Neumann. Fast and precise automatic baseline correction of one-and two-dimensional nmr spectra. *Journal of Magnetic Resonance (1969)*, 91(1):1–11, 1991.
- Blythe Durbin and David M Rocke. Estimation of transformation parameters for microarray data. *Bioinformatics*, 19(11):1360–1367, 2003.
- Blythe P Durbin, Johanna S Hardin, Douglas M Hawkins, and David M Rocke. A variance-stabilizing transformation for gene-expression microarray data. *Bioinformatics*, 18(suppl 1):S105–S110, 2002.
- Paul HC Eilers and Hans FM Boelens. Baseline correction with asymmetric least squares smoothing. *Leiden University Medical Centre Report*, 2005.
- Leslie R Euceda, Guro F Giskeødegård, and Tone F Bathen. Preprocessing of nmr metabolomics data. *Scandinavian journal of clinical and laboratory investigation*, 75(3):193–203, 2015.
- Baptiste Féraud, Bernadette Govaerts, Michel Verleysen, and Pascal De Tullio. Statistical treatment of 2d nmr cosy spectra in metabolomics: data preparation, clustering-based evaluation of the metabolomic informative

- content and comparison with 1h-nmr. *Metabolomics*, 11(6):1756–1768, 2015.
- Sergey Golotvin and Antony Williams. Improved baseline recognition and modeling of ft nmr spectra. *Journal of Magnetic Resonance*, 146(1):122–125, 2000.
- Jochen Hochrein, Helena U Zacharias, Franziska Taruttis, Claudia Samol, Julia C Engelmann, Rainer Spang, Peter J Oefner, and Wolfram Gronwald. Data normalization of 1h nmr metabolite fingerprinting data sets in the presence of unbalanced metabolite regulation. *Journal of proteome research*, 14(8):3217–3228, 2015.
- Daniel Holmes. Basic practical nmr concepts: A guide for the modern library, August 2013. URL <http://www2.chemistry.msu.edu/facilities/nmr/handouts.html>.
- Stefanie M Kohl, Matthias S Klein, Jochen Hochrein, Peter J Oefner, Rainer Spang, and Wolfram Gronwald. State-of-the art data normalization methods improve nmr-based metabolomic analysis. *Metabolomics*, 8(1):146–160, 2012.
- Ning Li, Yi peng Song, Huiru Tang, and Yulan Wang. Recent developments in sample preparation and data pre-treatment in metabonomics research. *Archives of biochemistry and biophysics*, 2015.
- Simon M Lin, Pan Du, Wolfgang Huber, and Warren A Kibbe. Model-based variance-stabilizing transformation for illumina microarray data. *Nucleic acids research*, 36(2):e11–e11, 2008.
- Xinbo Liu, Zhimin Zhang, Yizeng Liang, Pedro FM Sousa, Yonghuan Yun, and Ling Yu. Baseline correction of high resolution spectral profile data based on exponential smoothing. *Chemometrics and Intelligent Laboratory Systems*, 139:97–108, 2014a.
- Xinbo Liu, Zhimin Zhang, Pedro FM Sousa, Chen Chen, Meilan Ouyang, Yangchao Wei, Yizeng Liang, Yong Chen, and Chaoping Zhang. Selective iteratively reweighted quantile regression for baseline correction. *Analytical and bioanalytical chemistry*, 406(7):1985–1998, 2014b.
- Helen M Parsons, Christian Ludwig, Ulrich L Günther, and Mark R Viant. Improved classification accuracy in 1-and 2-dimensional nmr metabolomics data using the variance stabilising generalised logarithm transformation. *BMC bioinformatics*, 8(1):234, 2007.

- Parul V Purohit, David M Rocke, Mark R Viant, and David L Woodruff. Discrimination models using variance-stabilizing transformation of metabolomic nmr data. *Omics: a journal of integrative biology*, 8(2):118–130, 2004.
- Rejane Rousseau. *Statistical Contribution to the Analysis of Metabonomic Data in H-NMR Spectroscopy*. PhD thesis, Universite catholique de Louvain, 2011.
- Agnieszka Smolinska, Lionel Blanchet, Lutgarde MC Buydens, and Sybren S Wijmenga. Nmr and pattern recognition methods in metabolomics: from data acquisition to biomarker discovery: a review. *Analytica chimica acta*, 750:82–97, 2012.
- Marko Sysi-Aho, Mikko Katajamaa, Laxman Yetukuri, and Matej Orešič. Normalization method for metabolomics data using optimal selection of multiple internal standards. *BMC bioinformatics*, 8(1):1, 2007.
- CG Tang. An analysis of baseline distortion and offset in nmr spectra. *Journal of Magnetic Resonance, Series A*, 109(2):232–240, 1994.
- RJO Torgrip, KM Åberg, E Alm, I Schuppe-Koistinen, and J Lindberg. A note on normalization of biofluid 1d 1h-nmr data. *Metabolomics*, 4(2):114–121, 2008.
- Bo Wang, Aaron M Goodpaster, and Michael A Kennedy. Coefficient of variation, signal-to-noise ratio, and effects of normalization in validation of biomarkers from nmr-based metabonomics studies. *Chemometrics and Intelligent Laboratory Systems*, 128:9–16, 2013.
- Kuo-Ching Wang, San-Yuan Wang, Ching-hua Kuo, and Yufeng J Tseng. Distribution-based classification method for baseline correction of metabolomic 1d proton nuclear magnetic resonance spectra. *Analytical chemistry*, 85(2):1231–1239, 2012.
- Svante Wold, Michael Sjöström, and Lennart Eriksson. Pls-regression: a basic tool of chemometrics. *Chemometrics and intelligent laboratory systems*, 58(2):109–130, 2001.
- Christopher Workman, Lars Juhl Jensen, Hanne Jarmer, Randy Berka, Laurent Gautier, Henrik Bjørn Nielser, Hans-Henrik Saxild, Claus Nielsen, Søren Brunak, and Steen Knudsen. A new non-linear normalization method for reducing variability in dna microarray experiments. *Genome biol*, 3(9):1–16, 2002.

- Yuanxin Xi and David M Rocke. Baseline correction for nmr spectroscopic metabolomics data analysis. *BMC bioinformatics*, 9(1):324, 2008.
- Zhi-Min Zhang, Shan Chen, and Yi-Zeng Liang. Baseline correction using adaptive iteratively reweighted penalized least squares. *Analyst*, 135(5): 1138–1146, 2010.