Investigations into Background Correction and Retention Time Alignment to Enhance Quantitative Chemometric Analysis of Comprehensive Two-Dimensional Liquid Chromatography-Diode Array Detection Data

Robert Allen
Virginia Commonwealth University

Follow this and additional works at: https://scholarscompass.vcu.edu/etd

Part of the Chemistry Commons

© The Author

Downloaded from https://scholarscompass.vcu.edu/etd/2912

This Dissertation is brought to you for free and open access by the Graduate School at VCU Scholars Compass. It has been accepted for inclusion in Theses and Dissertations by an authorized administrator of VCU Scholars Compass. For more information, please contact libcompass@vcu.edu.
Investigations into Background Correction and Retention Time Alignment to Enhance Quantitative Chemometric Analysis of Comprehensive Two-Dimensional Liquid Chromatography-Diode Array Detection Data

A Dissertation submitted in partial fulfillment of the requirements for the degree of Doctorate in Philosophy in Chemistry at Virginia Commonwealth University.

By

ROBERT CRAIG ALLEN
Masters of Science in Chemistry, George Mason University, 2004
Bachelor of Science in Chemistry, George Mason University, 2001

Director: Sarah C. Rutan
Professor, Department of Chemistry

Virginia Commonwealth University
Richmond, Virginia
December 2012
Acknowledgements

The work described in this dissertation was not generated in a vacuum and credit needs to be given where it is deserved. First and foremost, I would like to acknowledge the contributions of Dr. Sarah C. Rutan. She has provided me with a considerable amount of support in regards to learning the Matlab programming environment and continuous debugging services these last four years. She has also been very forthcoming about my many flaws and has constantly strived to improve me as a researcher. Secondly, I would like to acknowledge the support and good-natured harassment I received from Dr. Hope P. Bailey over the same time period. She has consistently pointed out problems and solutions to the projects I have worked in. Thirdly, I would like to acknowledge the support I received in Minnesota while I was visiting the Carr research group at the University of Minnesota. I would like to thank Dr. Peter W. Carr and Dr. Marcelo R. Filgueira for their hospitality and forbearance while I attempted to learn a difficult set of concepts in a very short period of time. In addition to the level of academic support I received, I feel it is necessary to also acknowledge the support I received outside of VCU that has allowed me to focus on this degree. I am indebted to Park Previs, Kristopher Scott, Paul Felts, Thomas Fields, William Altizer, and many others for providing me with ample opportunity to blow off steam and enjoy their company over many board game sessions. Finally, I would also like to acknowledge my immediate family; John B. Allen, Catherine C. Allen, John B. Allen, Jr., and Katherine M. Allen, for standing by me and allowing me to focus on what I needed to do to complete this work.
Table of Contents

Acknowledgements .................................................................................................................. ii

List of Tables ............................................................................................................................ ix

List of Figures .......................................................................................................................... xi

List of Abbreviations and Symbols .......................................................................................... xx

List of Variables .................................................................................................................... xxiv

Abstract .................................................................................................................................. xxix

Chapter

1. Overview and Objectives ................................................................................................. 1

2. Overview of Comprehensive Two-Dimensional Liquid Chromatography ............... 4

   2.1 Basics of Chromatography ......................................................................................... 4

   2.2 Necessity of Two-Dimensional Chromatography ..................................................... 9

   2.3 Promise of Two-Dimensional Chromatography ......................................................... 10

   2.4 Optimizing $\beta$ ........................................................................................................ 13

   2.5 Optimizing $f_{c_0v}$ .................................................................................................... 18

   2.6 Sampling Device Setup Strategies ........................................................................... 26

3. Overview of Two-Dimensional Chromatography Quantification ............................. 30

   3.1 Transitioning from One-Dimensional to Two-Dimensional Quantitation ........... 30

   3.2 Trilinear and Quadrilinear Methods ....................................................................... 37

   3.3 Alignment Strategies ............................................................................................... 40
3.4 Alternatives to PARAFAC .................................................................45
3.5 Determination of the Number of Components .................................48
3.6 Initialization ....................................................................................52
3.7 Constraints ....................................................................................54

4. Impact of Background Correction on Peak Detection, Fractional Coverage, and Integration ........................................................................................................56

4.1 Introduction ....................................................................................57
4.2 Materials and Methods .....................................................................61
  4.2.1 Chemicals ..................................................................................61
  4.2.2 Samples and LC×LC-DAD instrumentation ...............................61
  4.2.3 Background correction techniques .............................................62
  4.2.4 Peak detection methods .............................................................64
  4.2.5 Adjusting peak counts .................................................................69
  4.2.6 Fraction coverage calculations ....................................................69
  4.2.7 Quantitation methods .................................................................70
4.3 Results and Discussion .....................................................................71
  4.3.1 Standard replicate peak counts ....................................................71
    4.3.1.1 Analysis of total peak counts ..................................................72
    4.3.1.2 Analysis of region A peaks .....................................................73
    4.3.1.3 Analysis of region B peaks .....................................................75
    4.3.1.4 Analysis of region C peaks .....................................................76
    4.3.1.5 Analysis of region D peaks .....................................................77
    4.3.1.6 OBGC using a median filter ..................................................78
4.3.2 Maize samples ................................................................. 80
4.3.3 Fractional coverage ...................................................... 87
4.3.4 Quantitation ................................................................. 90

4.4 Conclusions .................................................................. 92

5. Reconstruction of the First Dimension Peak Shape ......................... 95

5.1 Introduction ................................................................. 95

5.2 Sampling of First Dimension ........................................... 97

5.3 Experimental .............................................................. 99

5.3.1 Data sets ................................................................. 99

5.3.2 Alignment approach .................................................. 101

5.3.3 Interpolation implementation ...................................... 103

5.3.4 PARAFAC analysis .................................................... 106

5.4 Results and Discussion ................................................ 107

5.4.1 Retention time prediction .......................................... 107

5.4.2 Calibration curves ..................................................... 109

5.4.3 Experimental data ..................................................... 112

5.5 Conclusions .................................................................. 115

6. Development of a Two Dimensional Alignment Approach ............. 116

6.1. Introduction .............................................................. 116

6.2 Method Overview ....................................................... 117

6.2.1 Selection of an appropriate region of the two-dimensional chromatogram for analysis ........................................ 118

6.2.2 Selection of the number of components .......................... 118
6.2.3 Initial estimate generation and identification of components .......119
6.2.4 Three-way PARAFAC analysis on each sample .......................121
6.2.5 Component matching between samples .................................125
6.2.6 Alignment of the second dimension and the first
dimension .................................................................................. 129
6.2.7 Four-way PARAFAC analysis on the aligned data set .............130

6.3 Experimental ............................................................................. 131
   6.3.1 Data sets ............................................................................. 131
   6.3.2 Data analysis scheme ......................................................... 135

6.4 Results and discussion ............................................................. 136
   6.4.1 Simulated data sets ............................................................. 136
   6.4.2 Urine data set .................................................................... 137
   6.4.3 Phenytoin data set ............................................................. 140

6.5 Conclusion ................................................................................ 143

7. Comparison of Three PARAFAC Based Methods .....................145
   7.1 Introduction ............................................................................. 145
   7.2 Experimental ......................................................................... 148
      7.2.1 PARAFAC based methods .............................................. 148
      7.2.2 Background correction .................................................. 149
      7.2.3 Data sets .......................................................................... 149
         7.2.3.1 Phenytoin ................................................................. 149
         7.2.3.2 Simulations .............................................................. 150
         7.2.3.3 Quantitation study ................................................... 153
7.2.3.3.1 Chemicals ......................................................153
7.2.3.3.2 Sample preparation .......................................153
7.2.3.3.3 Instrument configuration ...............................155
7.2.3.4 Maize recovery ................................................155
  7.2.3.4.1 Chemicals ....................................................156
  7.2.3.4.2 Sample preparation .......................................156
  7.2.3.4.3 Instrument configuration ...............................157
7.3 Results and Discussion ............................................158
  7.3.1 Phenytoin ........................................................158
  7.3.2 Simulations .......................................................161
    7.3.2.1 Inter-sample shifting only .............................161
    7.3.2.2 Inter- and intra-sample shifting .....................163
  7.3.3 Quantitation study .............................................164
    7.3.3.1 Region 1 ....................................................165
    7.3.3.2 Region 2 ....................................................172
  7.3.4 Maize recovery ...............................................177
7.4 Conclusions ......................................................179
8. Conclusions and Future Work .....................................181
  8.1 Reflections on Chapter 4 .......................................181
  8.2 Reflections on Chapter 5 .......................................183
  8.3 Reflections on Chapter 6 .......................................183
  8.4 Reflections on Chapter 7 .......................................185
  8.5 Final Thoughts ..................................................187
List of References .................................................................................................................. 189
Appendix A ............................................................................................................................. 203
Appendix B ............................................................................................................................. 207
Appendix C ............................................................................................................................. 215
Appendix D ............................................................................................................................. 221
Appendix E ............................................................................................................................. 300
Vita ........................................................................................................................................ 302
List of Tables

Table 1: Thresholds (mAU) used for peak detection of the maize replicates ........................................66

Table 2: Thresholds (mAU) used for peak detection of the maize replicates after OBGC background correction ........................................................................................................66

Table 3: Average “peak” manual counts for the maize replicates .........................................................80

Table 4: Calculated fractional coverage values from eleven maize replicates before and after background correction ........................................................................................................88

Table 5: Percent relative standard deviations of six fully resolved peaks present in the standard samples ........................................................................................................................................91

Table 6: Peak areas used to generate the sampled $^1$D peak for the simulated data .....................101

Table 7: Calculated slope and intercept (with associated errors) for the simulated data set from the relative areas obtained from PARAFAC ........................................................................111

Table 8: Average percent standard error of prediction and percent relative standard deviation for the validation samples within the simulated data ........................................................................112

Table 9: Percent relative standard deviations of the relative peak areas for the five experimentally generated peaks ........................................................................................................112
Table 10: Percent recoveries for the four simulated data sets after analysis by MCR-ALS and PARAFAC ..........................................................137

Table 11: Percent relative standard deviations for Peaks 11 and 12 from the urine data set .....138

Table 12: Direct and standard calibration curves for phenytoin ........................................141

Table 13: Calculated concentrations for the phenytoin data set by SAAM, PARAFAC2, and SCP ........................................................................................................160

Table 14: The calculated simulated phenytoin $^1$D peak area and corresponding standard deviation from calibration curves with inter-sample shifting after analysis by SAAM, PARAFAC2, or SCP ......................................................................................163

Table 15: The calculated $^1$D peak area and corresponding error from calibration curves with inter- and intra-sample shifting after analysis by SAAM, PARAFAC2, or SCP ..............164

Table 16: Percent recoveries and associated standard deviations for 5-hydroxy-tryptamine, 3-chloropyridine, and anthranilic acid without background correction prior to chemometric analysis ..................................................................................................................169

Table 17: Percent recoveries and associated standard deviations for 5-hydroxy-tryptamine, 3-chloropyridine, and anthranilic acid with background correction prior to chemometric analysis ...........................................................................................................170

Table 18: Percent recoveries and associated standard deviations for 4-indolyl acetate and indole-3-acetonitrile before and after background correction .................................................................175

Table 19: Percent recoveries and associated standard deviations for anthranillic acid spiked into diluted maize extracts after SAAM, PARAFAC2, SCP, and MCR-ALS analysis ...............178
List of Figures

Fig. 1: Illustration of a van Deemter curve from Eq. 7 using $A=1$, $B=3$, and $C=0.05$. The blue line represents the $A$ term as a function of $u_e$, the red line represents the $B$ term as a function of $u_e$, the green line represents the $C$ term as a function of $u_e$, and the purple line represents the resulting plate height ($H$) as a function of $u_e$. The minimum of the purple line is used to establish the $H_{\text{min}}$ and $u_{\text{opt}}$ .......................................................... 7

Fig. 2: The modification made by Stoll et al. to an Hewlett Packard 1090 LC instrument in order to reduce the dwell volume of the second dimension for a LC×LC system. Reproduced from reference [22] with permission from Elsevier ............................................................... 17

Fig. 3: Conceptual diagrams of advancing gradients in the $^2$D of LC×LC. From top to bottom: “full in fraction” (FIF), “segment in fraction” (SIF), “continuously shifting” (CS). The dashed line is the corresponding $^1$D gradient. Reproduced from reference [43] with permission from Elsevier ........................................................................ 25

Fig. 4: Typical setup of the sampling device for LC×LC. The red and green colors denote the changes in the flow path depending on the position of the valve within the sampling device ..... 26

Fig. 5: An example of a trapping column based sampling device for LC×LC. Reproduced from reference [44] with permission from Wiley .................................................................................. 27

Fig. 6: An example of the use of an active splitter immediately after the sampling device to control the amount of $^1$D effluent transferred to the $^2$D column. Reproduced from reference [46] with permission from American Chemical Society ........................................................................ 28

Fig. 7: An example of the modified sampling device used in sLC×LC. Panel A shows the first loop being filled with $^1$D effluent. Panel B shows the second loop being filled with $^1$D effluent. Panel C shows the injection of the contents of the first loop onto the $^2$D column. Reproduced from reference [47] with permission from Elsevier ........................................................................ 29
Fig. 8: Illustration of the use of derivatives for a 2D peak to determine the peak maximum, the peak start, and the peak end. The blue line corresponds to the non-derivative peak. The green line corresponds to the first derivative of the peak. The red line corresponds to the second derivative of the peak. Adapted from reference [50] with permission from Elsevier ..........................31

Fig. 9: Five scenarios common in LC×LC when determining the degree of overlap. If the calculated overlap is greater than or equal to the overlap criterion, peak A and B are considered to be part of the same 2D peak. Reproduced from reference [50] with permission from Elsevier .................................................................................................................................32

Fig. 10: A representative three dimensional peak shape where A signifies the point with the highest intensity pixel and B signifies the region where the pixels with the lowest intensity are located ....................................................................................................................................................................................33

Fig. 11: A representation of the selected border of a GC×GC 2D peak by the drain algorithm. The letters a-f denote non-continuous portions of a 2D chromatogram merged into the overall 2D peak. Reproduced from reference [56] with permission from Elsevier ..........................................................................................................................................................................................34

Fig. 12: A representation of the triangle summation method devised by Mondello et al. for a single 2D peak. Reproduced from reference [57] with permission from American Chemical Society ..........................................................................................................................35

Fig. 13: A schematic representation of the four-way structure of multiple LC×LC-DAD samples where I denotes the number of points in second dimension, J denotes the number of point in the first dimension, L denotes the number of points in the UV-VIS spectrum, and K denotes the total number of samples. Adapted from reference [2] with permission from Wiley .................................38

Fig. 14: A schematic representation of the selected warping path by DTW between a reference chromatogram (Signal R) and a target chromatogram (Signal T). Reproduced from reference [69] with permission from American Chemical Society .................................................................................................................................41

Fig. 15: A scree plot (from the reshaped four-way phenytoin data set discussed in Chapters 6 and 7) is generated by plotting the log of the singular value by the component number. The bend, indicated by A, is the normal criteria used to determine the number of components. The gap, indicated by B, is the previous method used by Porter et al. and Zhu et al. for determining the number of components. The gap, indicated by C, is the method used by Bailey et al. and is used by the semi-automated alignment method (SAAM). Reproduced from reference [94] with permission from Elsevier .................................................................................................................................................49
Fig. 16. LC×LC-DAD chromatogram of a blank sample at 220 nm ........................................59

Fig. 17. Schematic diagram of the system of peak classification used in this chapter. (A) Peaks detected in the region of the injection ridge. (B) Peaks detected in the region of the re-equilibration ridge. (C) Peaks generated by the background correction process. (D) Region where “real” peaks are presumed to exist ...............................................................69

Fig. 18. The average number of peaks determined in each region shown in Fig. 17 for the standard samples. Various peak detection methods are denoted as: Derivative detection—high threshold (a); drain algorithm detection without baseline correction—high threshold (b); drain algorithm based detection with baseline correction—high threshold (c). (d-f) are analogous to (a-c), except using the low threshold. Panel (A) no background correction, Panels (B-D) after background correction using the Direct (B), SVD-BC (C), and AWLS (D) techniques respectively. The color patterns correspond to: red (peaks located in region A of Fig. 17), green (peaks located in region B of Fig. 17), purple (peaks located in region C of Fig. 17), cyan (peaks located in region D of Fig. 17). The horizontal lines in (A-D) are the true peak totals, 23 and 21, upon which the detection thresholds were based, for the high and low detection threshold respectively ...............................................................71

Fig. 19. Impact of the various background correction techniques on the background artifacts (data shown for the third standard replicate at 220 nm): (A) the injection ridge (box 1) and artifact peaks (box 2) from 1.75 to 3.5s in the second dimension and at 25.6 min in the first dimension. Curve a is the raw signal (before background correction) and curves b-d show the corrected signals upon use of the Direct, SVD-BC, and AWLS techniques respectively; (B) the injection ridge from 3 to 28 min in the first dimension at 1.9375 s in the second dimension. The letters a-d have the same meaning as in A; (C) the re-equilibration ridge from 9.5 to 10.9 s in the second dimension at 25.6 min in the first dimension. The letters a-d have the same meaning as in A; (D) the solvent peaks from 1.75 to 4. s at 3.8 min (immediately after dead marker) in the first dimension. The letters a-d have the same meaning as in A ...............................................................74

Fig. 20. Impact of the LC Image baseline correction options of the second dimension chromatogram from the third standard replicate at 220 nm and 25.6 min. The blue (solid line) chromatogram is before baseline correction and the red (dashed line) chromatogram is after baseline correction applied within LCImage. (A) No background correction (B-D); background removal by the Direct, SVD-BC, and AWLS techniques respectively ..............................................75

Fig. 21: Peak counts obtained from the five standards replicates using A) high peak detection thresholds and B) low detection thresholds as listed in Table 2. The colors correspond to the colored regions in Fig. 17 with red indicating peaks detected from the injection ridge and cyan indicating peaks detected due to “real” peaks. The horizontal black bars indicate the expected peak counts (21 for the high detection threshold and 23 for the low detection threshold) ........79
Fig. 22: A plot of the number of merged 2D peaks as a function of the degree of spectral correlation and retention time drift tolerance allowed between successive 2D chromatograms at the low detection threshold without performing background correction. The blue diamonds are peaks merged using a 0.0625 s tolerance; the red squares are peaks merged using a 0.1250 s tolerance; and the green triangles are the peaks merged using a 0.1875 s tolerance ...

Fig. 23. The impact of the AWLS and OBGC techniques on the signal intensity of a selected peak. A) The AWLS technique was found to retain on average 75±2 % of the net peak height compared to the SVD-BC method. A second dimension chromatogram of a peak from the first standard replicate at 220 nm and 15.4 min is shown. The blue line was not background corrected. The red line was background corrected by the Direct method. The green line was background corrected by the SVD-BC method. The purple line was background corrected by the AWLS method; B) The spectra of the peak shown in A with the purple line representing after AWLS background correction and the red line after Direct background correction; C) The peak heights of six selected peaks from the standard replicates as a response to changing the weighting factor (p-value) for the AWLS background correction technique. The heights after AWLS were divided by the heights after Direct to give the relative % remaining after background correction; D) The peak heights of the same six peaks chosen in C after background correction using OBGC by median filter for different filter window widths ...

Fig. 24. The average number of peaks determined for the maize sample in each region shown in Fig. 17 after background correction for (A) no background correction, (B-D) background correction using the Direct, SVD-BC, and AWLS techniques respectively. The colors and letters (a-f) have the same meaning as previously described in Fig. 18. The shaded boxes in (A-D) represent the range of estimated true peak totals obtained by manual peak detection and using a 0.125 s retention time drift tolerance ...

Fig. 25. Plots of the sixth maize replicate at 220 nm. Red circles or outlines indicate where a merged peak was detected at the higher threshold in Table 1. (A) Derivative approach using Stevenson’s method. (B) Drain algorithm with baseline correction using LC Image. (C) Manual peak detection using Matlab with a 0.0625 s retention time drift allowed between each second dimension chromatogram ...

Fig. 26. Retention times of peaks detected using the derivative-based method at a high threshold for the eleventh maize replicate at 220 nm. The blue diamonds (◇) are the retention times of peaks detected without using background correction. The red squares (□) are the retention times of peaks detected using the following background correction techniques A) AWLS using a weighting factor of 0.001; B) OBGC using a filter window width of 5; C) OBGC using a filter window width of 25; D) OBGC using a filter window width of 45; E) OBGC using a filter window width of 45 after linear-correlation alignment in the second dimension; F) OBGC using a filter window width of 45 after optimized-COW on the second dimension ......
Fig. 27. Retention times of peaks detected using the derivative-based method at a low threshold for the eleventh maize replicate at 220 nm. The symbols and A-F plots are the same as described in Fig. 26.

Fig. 28: (A) An illustration of an LC×LC chromatogram. The dashed lines indicate the time points where the sampling device injects the 1D eluent into the 2D column. (B) The 1D peak pattern from Fig. 29A generated by summing all values along the dashed lines. The alternating ▼ and ● markers represent either the use of two loops or two 2D columns in LC×LC. (C) Illustration of the parameters that determine the sampling phase, φ.

Fig. 29: An illustration of the impact of a changing $t_R$ on the shape of the sampled 1D peak. Adapted from reference [19] with permission from Elsevier.

Fig. 30: (A) A contour plot of a representative LC×LC chromatogram. The dashed line indicates the 2D time point chosen to generate the representative 1D chromatogram for interpolation purposes. (B) The representative 1D chromatogram generated in 30A. The solid and dashed lines represent two different samples of the same peak at two different retention times. (C) 1D chromatograms after the cubic spline interpolation technique has been applied to the sampled 1D chromatograms from 30B. The vertical dashed line indicates the point to which the dashed peak is being aligned. (D) The resulting resampled 1D peak after alignment and resampling of the interpolated 1D chromatogram.

Fig. 31: A plot of the retention times after reconstruction of the first dimension for the simulated data set versus the actual retention times used for the generation of the simulated data. The solid line is the actual retention times plotted against the actual retention times. The four interpolation techniques, PCHIP (□), spline polynomial (△), Fourier zero-filling (∗), and Gaussian fitting (○), are compared against this line.

Fig. 32: Plots of the standard deviations for the normalized relative peak areas obtained from PARAFAC for (A) Peak 1, (B) Peak 2, and (C) Peak 3. The solid black line is the average standard deviation for all techniques for each peak.

Fig. 33: A) Contour plot of the MCR-ALS component from the twelfth replicate sample of the urine sample showing the two peaks of interest within the urine data set. The dashed white line indicates the 1D point at which the component will be split into two components. B) The resulting MCR-ALS component after splitting the original MCR-ALS component into two components. C) The newly generated MCR-ALS component created when the component was split into two components.
**Fig. 34:** A) A contour plot of the MCR-ALS component containing the phenytoin peak (50 ppb) from the sixth sample of the phenytoin data set. B) A plot of the $^2$D chromatograms from the second to fifth $^1$D data points in 34A. The numbers (in s) indicate the position of the peak maximum for the corresponding color $^2$D chromatogram. C) The $^2$D chromatograms shown in 34B after shifting all of the $^2$D chromatograms to the earliest maximum (12.60 s). D) The resulting contour plot of the MCR-ALS component after aligning the $^2$D chromatograms.

**Fig. 35:** Schematic of the spectral component matching system. The reference components have been labeled in the following order: 1) background, 2) analyte, 3) background, 4) analyte. The sample components have been labeled in the following order: 1) analyte, 2) background, 3) background, 4) analyte. A) Matrix generated from all possible Pearson correlation coefficient pairs. B) Removal of the raw background components (1 and 3) from consideration. C) Removal of the sample background components (2 and 3) from consideration. D) The resulting analyte component matches between the raw reference spectra and the three-way PARAFAC spectra for this sample.

**Fig. 36:** Schematic of how the direction of the alignment is determined. The white and black triangles represent the earliest and latest eluting peaks. The arrow indicates the direction of alignment. A) The distance between the black triangle and the median is greater than the distance between the white triangle and median resulting in the peak being aligned to the right. B) The distance between the white and black triangles and the median is equal resulting in the peak being aligned to the middle of the data window. C) The distance between the white triangle and the median is greater than the distance between the black triangle and the median resulting in the peak being aligned to the left.

**Fig. 37:** A) A contour plot of the seventh sample from the urine data set. Shown in the middle of the contour plot are the two selected peaks with the same spectra. The absorbance bar on the right of the contour plot shows the intensities of the peaks present. B) The resolved spectra of the two peaks with the calculated Pearson correlation coefficient shown. C) Contour plot of the reconstructed peak shapes from the resolved four-way PARAFAC components for the two peaks of interest. The positions of the two peaks are different from Fig. 38A due having been aligned prior to reconstruction.

**Fig. 38:** A) A contour plot of the thirteenth sample from the phenytoin data set (25 ppb spiked waste water treatment). The intensity on the right shows the relative absorbance (in mAU) of each of the peaks present. B) The resolved spectra of the phenytoin and interferent after four-way PARAFAC analysis. The solid line is the phenytoin spectra and the dashed line is the interferent spectra. The calculated Pearson correlation coefficient is also shown. C) The resolved phenytoin peak after reconstruction from the components obtained from four-way PARAFAC analysis. D) The resolved interferent peak after reconstruction from the components obtained from four-way PARAFAC analysis. The distortion of the upper shape of the peak is caused by the use of selectivity in the second dimension.
Fig. 39: Plots of the excitation (left column) and emission (right column) spectra from the four-way data set used by Hong and Harshman to compare SFA to PARAFAC2. The symbols F1-4 indicate the component numbers. Reproduced from reference [89] with permission from Wiley. 

Fig. 40: Schematic setup of the simulations performed in order to compare the ability of SAAM, SCP, and PARAFAC2 to handle retention time shifting.

Fig. 41: A) A contour plot of the mean Euclidian distance from 4.29 s in the first dimension and 14.35 s in the second dimension. The mean Euclidian distance was calculated from the first and second dimension retention times for eighteen samples for each simulation. B) A contour plot of the mean correlation coefficient calculated as described in the text.

Fig. 42: A contour plot of the chromatogram of the first replicate of the lowest calibration standard at 220 nm of the quantitation data set. The red numbers indicate the peaks that were quantified. The numbers correspond to the following compounds: 1) 5-hydroxy-tryptamine; 2) 3-chloropyridine; 3) anthranilic acid; 4) 2-bromopyridine; 5) indole-3-acetyl-L-lysine; 6) indole-3-acetyl-L-glycine; 7) 4-indolyl acetate; and 8) indole-3-acetonitrile.

Fig. 43: Chromatograms of the maize spiked standards used in the maize recovery study. A) Low level anthranilic acid calibration point from the first replicate; B) Diluted maize sample from the first replicate; C) Resulting low level anthranilic acid spiked maize sample from the first replicate.

Fig. 44: Contour plots of the resolved 2D chromatograms for A) the first replicate of the unspiked waste water treatment sample using a seven component PARAFAC2 model (phenytoin component); B) the first replicate of the unspiked waste water treatment sample using a four component PARAFAC2 model (phenytoin component); C) the phenytoin component from a seven component SCP model; and D) the interferent component from a seven component SCP model.

Fig. 45: Illustration of the two regions of the quantitation study analyzed by SAAM, PARAFAC2, and SCP. Region 1 contained the peaks corresponding to 5-hydroxy-tryptamine, 3-chloropyridine, and anthranilic acid. Region 2 contained the peaks corresponding to 2-bromopyridine, indole-3-acetyl-L-lysine, and indole-3-acetyl-L-glycine.

Fig. 46: A) Plot of the resolved sequential 2D chromatograms obtained from MCR-ALS analysis of the first replicate from the low calibration point. The resolution between the two peaks was estimated as 0.13. B) Plot of the true spectra for 5-hydroxy-tryptamine and 3-chloropyridine.
with the calculated correlation coefficient for the two spectra. C) Plot of the 3-way PARAFAC analysis resolved spectra for 5-hydroxy-tryptamine and 3-chloropyridine with the calculated correlation coefficient from the first replicate of the low calibration point. The blue lines in all three panels correspond to 3-chloropyridine and the green lines in all three panels correspond to 5-hydroxy-tryptamine .................................................................166

Fig. 47: A plot of the calculated 1D peak areas after 4-way PARAFAC analysis from SAAM using simulated data. The simulation was conducted using a 1D peak area of 42 .......................171

Fig. 48: Mesh plots of the resolved chromatograms from MCR-ALS analysis for 5-hydroxy-tryptamine, 3-chloropyridine, and anthranilic acid. The left column is the result obtained if background correction was not performed prior to MCR-ALS analysis. The right column is the result obtained if background correction was performed prior to MCR-ALS analysis ..........171

Fig. 49 Plots of the resolved spectra if A) background correction was not performed prior to MCR-ALS analysis; B) background correction was performed prior to MCR-ALS analysis; C) An overlay of the resolved 5-hydroxy-tryptamine spectra if background correction was not performed prior to MCR-ALS analysis (solid line) and if background correction was performed prior to MCR-ALS analysis (dashed line). The colors in A, B, and C correspond to blue (3-chloropyridine), red (anthranilic acid), and purple (5-hydroxy-tryptamine). The arrows in (B) indicate points of difference in the resolved spectrum compared to (A) ...................................172

Fig. 50: Plots of the resolved 2-bromopyridine component (blue) and indole-3-acetyl-L-lysine and indole-3-acetyl-L-glycine component (red) for A) PARAFAC2 and B) SCP .................176

Fig. 51: Plots of the optimized resolved 2D profiles from SAAM analysis. A) Profiles obtained if component splitting is not used. B) Profiles obtained if component splitting is used on all samples. C) Profiles obtained if extra components are added to the samples impacted by background while component splitting was being performed. D) Profiles obtained after OBGC was performed prior to SAAM. The blue peak corresponds to 3-chloropyridine. The red peak corresponds to indole-3-acetyl-L-lysine. The green peak corresponds to indole-3-acetyl-L-glycine. The cyan peak in B corresponds to indole-3-acetyl-L-glycine that was placed into the background ........................................................................................................176

Fig. 52: Plots of the log (base 10) of the singular values from SVD of the maize recovery data set if A) background correction has not been performed; or B) if background correction has been performed ........................................................................................................................................178
Fig. 53: Plots of the normalized intensity versus the concentration for A) Background correction performed prior to MCR-ALS analysis; B) SAAM analysis; C) PARAFAC2 analysis; D) SCP analysis. The blue diamonds correspond to anthranilic acid samples consisting of water and MeOH. The red squares correspond to the different levels of anthranilic acid spiked into the dilute maize samples. The red square calibration standard zero corresponds to the diluted maize sample without anthranilic acid.
List of Abbreviations and Symbols

1D: first chromatographic dimension

1D: one-dimensional

2D: second chromatographic dimension

2D: two-dimensional

ALS: alternating least squares

ACN: acetonitrile

AWLS: asymmetric weighted least squares

°C: degree Celsius

CS: continuously shifting

COW: correlation optimized warping

CVD: chemical vapor deposition

DAD: diode array detector

DTW: dynamic time warping
EMG: exponentially modified Gaussians

FA: factor analysis

FIF: full in fraction

GC: gas chromatography

GC×GC: comprehensive two-dimensional gas chromatography

GRAM: generalized rank annihilation method

IE: imbedded error

IKSFA: iterative key set factor analysis

IND: factor indicator test

IOPA: iterative orthogonal projection approach

KSFA: key set factor analysis

LC: liquid chromatography

LC×LC: comprehensive two-dimensional liquid chromatography

LC-LC: heart cutting chromatography

LOD: limit of detection

MCR: multivariate curve resolution

MeOH: methanol
M-S-F: Murphy-Schure-Foley criterion

OBGC: orthogonal background correction

OPA: orthogonal projection approach

PARAFAC: parallel factor analysis

PARAFAC2: parallel factor analysis version 2

% RSD: percent relative standard deviation

RESO: Ratio of Eigenvalues calculated by Smooth principal component analysis (PCA) and those calculated by Ordinary PCA

ROD-IND: ratio of derivatives of error indicator function

RP: reversed phase

SAAM: semi-automated alignment method

SCP: shifted candecomp/PARAFAC

sLC×LC: selective LC×LC

SEP: standard error of prediction

SFA: shifted factor analysis

SIF: segment in fraction

SOT: statistical overlap theory

SVD: singular value decomposition
SVD-BC: singular value decomposition-based background correction

THF: tetrahydrofuran

TLC: thin layer chromatography

µL: microliter

UV-VIS: ultraviolet-visible
List of Variables

$1_n$: 1D peak capacity

$2_t_g$: 2D gradient time

$1_t_R$: 1D retention time

$2_t_R$: 2D retention time

$2_n$: 2D peak capacity

$1_\sigma$: 1D peak width

$<1_\sigma_{eff}>$: average effective 1D peak width

A: total area of the Gaussian peak

A: eddy dispersion term in a van Deemter curve

A, B, C: two-way arrays generated by a three-way PARAFAC analysis

A, B, C, D: two-way arrays generated by a four-way PARAFAC analysis

$a_{in}, b_{jn}, c_{in}$: elements of the A, B, and C two-way arrays generated by a three-way PARAFAC analysis

$e_{ij}$: elements of the E two-way array generated by a three-way PARAFAC analysis

$a_{in}, b_{jn}, c_{kn}, d_{ln}$: elements of the A, B, C, and D two-way arrays generated by a four-way PARAFAC analysis

$e_{ijkl}$: elements of the E two-way array generated by a four-way PARAFAC analysis
\( a, b, c, d, e \): Gaussian fitting initialization parameters

\( \alpha \): saturation factor

\( a, b \): empirically fitted parameters for estimating \( n_c \)

\( B \): longitudinal dispersion term in a van Deemter curve

\( <\beta> \): average undersampling term

\( C \): resistance to mass transfer term in a van Deemter curve

\( D_m \): mobile phase diffusion coefficient

\( d_c \): inner diameter of a capillary

\( \mathbf{E} \): an error array with dimensions corresponding to the dimensions of \( \mathbf{X} \)

\( \mathbf{F} \): abstract row vector two-way array from IKSFA and IOPA

\( f_{\text{cov}} \): fractional coverage

\( F \): flow rate

\( F_i \): column selectivity

\( H \): height of an individual theoretical plate

\( H_{\text{min}} \): minimum possible value of \( H \)

\( I, J, K, L \): Number of data points in the second dimension, first dimension, sample dimension, and spectral dimension for LC\( \times \)LC-DAD data

\( k' \): retention factor
\( k'_{\text{last}} \): retention factor for the last eluting peak

\( \kappa \): fitting parameter for undersampling equation

\( L \): length of a column

\( \mathbf{M} \): a regression matrix of weighting coefficients for the contributions of the abstract spectra to the data set

\( m \): number of true peaks in SOT

\( M_R \): modulation ratio

\( N \): number of theoretical plates

\( N \): number of components

\( n_{c,2D} \): ideal 2D peak capacity

\( n'_{c,2D} \): corrected 2D peak capacity

\( n^*_{c,2D} \): effective 2D peak capacity

\( n_{c,\text{grad}} \): gradient peak capacity

\( n_{c,\text{iso}} \): isocratic peak capacity

\( n_m \): number of moles of the analyte in the mobile phase

\( n_s \): number of moles of the analyte in the stationary phase

\( n_{\text{true}} \): number of true peaks in SOT

\( n_{\text{type}} \): number of a given type of peak in SOT
ΔP: pressure drop

\textbf{R}: resolved chromatograms from MCR-ALS

R_s: resolution

\textbf{S}: resolved spectra from MCR-ALS

σ: standard deviation

\textbf{T}: center of the first dimension sampling window

t_0: dwell time of a column

t_R: retention time of a peak

t_{R,1}: retention time of the earlier eluting peak

t_{R,2}: retention time of the later eluting peak

t_{re-eq}ii: time required to re-equilibrate a column

t_{R,first}: retention time of the first peak

t_{R,last}: retention time of the last peak

t_s: 2D sampling interval

\phi: first dimension sampling phase

\textbf{U}: abstract chromatograms from SVD

u_e: interstitial velocity
$u_{\text{opt}}$: optimal interstitial velocity

$\mathbf{V}$: abstract spectra from SVD

$\tilde{\mathbf{V}}^T$: the singly truncated (to contain the first $N$ columns) $\mathbf{V}$ matrix from SVD

$\tilde{\mathbf{V}}$: doubly truncated $\mathbf{V}$ two-way array from SVD

$\eta$: viscosity

$\mathbf{W}$: two-way array of singular values from SVD

$w_{\text{avg}}$: average base width

$\mathbf{X}_{\text{sample}}$: truncated sample matrix

$\mathbf{X}_{\text{sample, bc}}$: the background corrected 2D chromatogram

$\mathbf{X}$: three-way or higher array

$\mathbf{X}_{\text{fit}}$: three-way or higher fitted array

$\mathbf{X}_{IJKL}$: four-way array of size $IJKL$
Abstract

INVESTIGATIONS INTO BACKGROUND CORRECTION AND RETENTION TIME ALIGNMENT TO ENHANCE QUANTITATIVE CHEMOMETRIC ANALYSIS OF COMPREHENSIVE TWO-DIMENSIONAL LIQUID CHROMATOGRAPHY-DIODE ARRAY DETECTOR DATA

By: Robert Craig Allen

A dissertation submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy at Virginia Commonwealth University.

Virginia Commonwealth University, 2012

Major Director: Sarah C. Rutan, Professor of Chemistry

The focus of the projects presented here was to develop possible solutions to three issues commonly encountered during chemometric analysis of comprehensive two-dimensional liquid chromatography diode array detector (LC×LC-DAD) data. The focus of the first project was to determine a means of performing background correction that removed two background ridges. The methods of simply subtracting out a mean blank sample, singular value decomposition based background correction (SVD-BC) and asymmetrically weighted least squares (AWLS) were compared. AWLS was found to be the only background correction technique to fully remove the ridges. However, AWLS was also found to attenuate the peak intensity by approximately 25% due to over fitting of the background at the lower wavelengths.

The focus of the second project was the investigation of five common interpolation strategies for the reconstruction of the sampled first dimension peak. The interpolation strategy that best reproduced the original first dimension retention time was Gaussian fitting. This was
expected given that the simulated data set was generated using a Gaussian model for the peak shape.

An algorithm, semi-automated alignment method (SAAM), was then developed that allowed for each peak to be aligned independently of the other peaks in the data set. SAAM was validated using both simulated and experimental data. The simulated results indicated that SAAM produced percent recoveries close to 100%. SAAM was also compared to iterative key set factor analysis-alternating least squares (IKSFA-ALS) for the analysis of phenytoin in a waste water treatment plant effluent. SAAM produced a concentration of 26±3 ppb compared to 39±9 ppb from IKSFA-ALS. While these results are very different, the result produced by SAAM is still within the experimental error of the reference 2D-LC/MS/MS method, 42±19.

Finally, SAAM was compared to two existing literature methods. A mixture of simulated and experimental data sets was used to measure the accuracy and precision of the results. SAAM was found to be impacted less by intra- and inter-sample retention time shifting then PARAFAC2. SAAM and shifted candecomp/PARAFAC were found to produce very similar results. However, SAAM was found to experience some difficulty producing accurate and precise results with some of the experimental data sets.
CHAPTER 1: Overview and Objectives

The work presented in this dissertation seeks to apply existing chemometric analysis methods and to develop new methods to address common data analysis problems in comprehensive two-dimensional liquid chromatography–diode array detector (LC×LC-DAD) data. During the development of the iterative key set factor analysis–alternating least squares method (IKSFA-ALS), Bailey et al. identified five major inherent problems found in analyzing LC×LC-DAD chromatograms: 1) data size and complexity, 2) spectral overlap, 3) chromatographic overlap, 4) retention time shifts, and 5) presence of a background signal. The data size and complexity issue refers to the difficulty in selecting the correct number of components to solve for with chemometric methods. The spectral overlap issue refers to the presence of two or more spectrally similar or identical peaks within the same chemometrically resolved chromatogram. The chromatographic overlap issue refers to two or more peaks having poor chromatographic resolution but possessing dissimilar enough spectra that chemometric resolution is possible. The retention time shift issue refers to the merging of peaks as shifting occurs or the difficulty in tracking a single peak as shifting occurs. The presence of a background signal refers to the response of the DAD to refractive index changes. With the exception of the first problem, the research contained in Chapters 4-7 attempts to provide solutions (or at the very least workarounds) to each of these problems [1].
In order to provide a foundation for the research chapters, Chapters 2 and 3 present background information relating to the basic operation of LC×LC-DAD systems and some of the current chemometric methods available for analyzing the resulting data. These chapters have been written with an eye towards presenting the information as a rationale for the choices made in the development of the chromatography and chemometric methods.

The spectral overlap problem is addressed in the method described in Chapter 6. The method utilizes the same strategy as Tistaert et al. [2] to split spectrally rank deficient components into two or more components. The split is accomplished by applying a chromatographic selectivity constraint to a region of either the first or second chromatographic dimension. In addition, the spectrum of the component being split is copied and applied to the new component.

The chromatographic overlap problem is addressed through the use of parallel factor analysis to resolve each peak into a different component assuming that the spectra are dissimilar enough, Chapter 6. However, the ability to correctly resolve different components is directly impacted by the retention time shifting. In order for parallel factor analysis to produce an accurate depiction of the peak, any retention time shifting between samples or within samples needs to be addressed.

The retention time shifting problem is addressed in the method described in Chapter 6. Unlike previous alignment strategies, the alignment method described in Chapter 6 does not align every peak to the same reference sample. Instead, each peak is separately aligned based on the calculated distance from the middle of the window. Each peak is then shifted to the retention time furthest from the middle of the window. This alignment strategy is utilized to prevent the shape distortion of peaks located on the edge of the window.
One problem not mentioned by Bailey et al. [1] is the need to provide a sufficient number of data points for alignment in the first chromatographic dimension. Chapter 5 investigates common interpolation techniques to allow for an accurate recreation of an unsampled first dimension peak. The techniques were compared on their ability to correctly estimate the unsampled first dimension retention time. After interpolation was conducted, the resulting chromatograms were aligned using a simple linear shift and then resampled to prevent Matlab from running out of memory. In addition, the precisions of component areas obtained from parallel factor analysis after alignment were calculated for each technique.

The background signal problem is addressed in Chapter 4. Chapter 4 investigates three spectroscopic-based techniques and one chromatographic-based technique for performing background correction. The results are presented in terms of each technique’s ability to remove two prominent background artifacts. In addition, two literature peak detection methods are investigated to determine which one returns a more reliable peak count.

Finally, a direct comparison is performed between the technique described in Chapter 6 and three literature-based methods [3-5]. The comparison evaluates the ability of the four methods to handle shifting in both the first and second dimensions using both simulated and experimental data sets. These data sets were chosen in order to evaluate both the accuracy and precision of each of these methods. In addition, the impact of background correction on chemometric analysis is also investigated.
CHAPTER 2: Overview of Comprehensive Two-Dimensional Liquid Chromatography

2.1 Basics of Chromatography

Chromatography is the separation of analytes in either space, as in thin layer chromatography (TLC), or in time, as in gas chromatography (GC) or liquid chromatography (LC). Separation is achieved by the repeated transfer of the analytes between two phases. In the three example methods given, the two phases consist of a stationary phase and a mobile phase. The relative affinity of an analyte for the stationary phase is known as the retention factor, $k'$, and is defined at any point in time and position along the bed according to Eqn. 1 [6],

$$k' = \frac{n_s}{n_m}$$ (1)

where $n_s$ is the number of moles of the analyte in the stationary phase and $n_m$ is the number of moles of the analyte in the mobile phase. By convention, the stationary phase is placed in the numerator and the mobile phase is placed in the denominator. Obviously, if two analytes have the same $k'$, then separation is not achieved.

Two different modes of bonded phase LC are available that make use of the polarity of the stationary phase to influence $k'$. If the stationary phase is polar, then the separation is considered to occur under normal phase conditions. If the stationary phase is nonpolar, then the separation is considered to occur under reversed phase (RP) conditions. Since RP conditions are the dominant form of chromatography used in LC, the remainder of the discussion is written from a reversed phase perspective. In RP-LC, the stationary phase typically consists of polar porous SiO$_2$ particles that have been chemically modified with nonpolar ligands. The particles
are packed into the LC column using high pressure, with a frit preventing the particles from exiting from the other side of the column. After the packing is complete, the entrance of the column is capped off using another frit. In order to preserve the integrity of the packing of the particles, mobile phase is pumped through the column in the same direction that the column was packed. The mobile phase is typically water with some fraction of organic modifier added to decrease the polarity of the mobile phase. Tetrahydrofuran (THF), acetonitrile (ACN), and methanol (MeOH) are the most commonly used organic modifiers.

The degree of separation between two analytes depends on the magnitude of the difference between their respective retention times. The degree of separation between two analytes is defined by Eq. 2 [6]

\[ R_s = \frac{t_{R,2} - t_{R,1}}{w_{avg}} \]  

(2)

where \( t_{R,1} \) and \( t_{R,2} \) are the retention times of the earlier and later eluting peaks, respectively, and \( w_{avg} \) is the average base width of the two peaks. Typically, the shape of a chromatographic peak is approximated by a Gaussian shape, Eq. 3 [6]

\[ C(t) = \frac{A}{\sigma \sqrt{2\pi}} e^{-\frac{(t-t_R)^2}{2\sigma^2}} \]  

(3)

where \( A \) is the total area of the Gaussian peak, \( \sigma \) is the standard deviation, and \( t_R \) is the retention time of the peak. This Gaussian model assumes that the peak width, \( w \), is equal to \( 4\sigma \).

RP-LC can be performed under isocratic or gradient conditions. Isocratic conditions correspond to experiments where the fraction of organic modifier present in the mobile phase remains the same throughout the chromatographic run. Gradient experiments are based on changing the fraction of organic modifier during the chromatographic run. Gradients are usually performed in a linear fashion with the initial mobile phase composition consisting primarily of
aqueous solvent. As the chromatographic run progresses, increasing amounts of organic modifier are added to the mobile phase.

Under isocratic conditions, $\sigma$ is related to $t_R$ and the number of theoretical plates ($N$) as shown in Eq. 4. [6]

$$\sigma = \frac{t_R}{\sqrt{N}}$$

(4)

The retention time of a peak is directly related to $k'$ by Eq. 5 [6],

$$t_R = t_o(1+k')$$

(5)

where $t_o$ is the dwell time of the column (the amount of time required for an unretained compound to reach the end of the column). The number of theoretical plates ($N$) is directly related to the length of the column ($L$) and the height of an individual theoretical plate ($H$) as shown in Eq. 6 [6].

$$N = \frac{L}{H}$$

(6)

$N$ can either be increased by increasing $L$ or decreasing $H$. Since increasing $L$ results in longer analysis times, minimizing $H$ is the preferred choice for increasing $N$. $H$ can be minimized by optimizing the van Deemter equation, Eq. 7 [6]

$$H = A + \frac{B}{u_e} + Cu_e$$

(7)

where $A$ is the “eddy” dispersion term, $B$ is the longitudinal dispersion term, $C$ is the resistance to mass transfer term and $u_e$ is the interstitial velocity. An illustration of the optimization of the van Deemter equation is shown in Fig. 1.
When optimizing the van Deemter equation, the goal is to produce the smallest $H_{\min}$ possible. This is achieved by selecting the optimum interstitial velocity ($u_{opt}$). $H_{\min}$ can be calculated directly from $A$, $B$, and $C$ (Eq. 8) and $u_{opt}$ can be calculated from $B$ and $C$ (Eq. 9)

$$H_{\min} = A + 2\sqrt{BC}$$

$$u_{opt} = \frac{B}{\sqrt{C}}$$ 

As seen in Eq. 7, the $A$ term is not affected by the velocity. This is because the $A$ term is a measure of how random the packing for a given column is and, by extension, how efficient the column performs axial transport. Since it is the random nature of the packing that allows for axial transport, axial transport increases as the particle packing becomes increasingly random. As a result of greater axial transport, the more uniform the overall interstitial velocity of a given analyte and conversely the smaller the $A$ term. Therefore, one way to improve $A$ is to ensure
that the column being used has been packed correctly. Currently, all commercial columns meet this criterion, but caution should be used if a column is packed in-house. The second way to improve A is to use smaller particles, which will decrease the time required to move from one flow stream to another.

As mentioned previously, B is a measure of the amount of longitudinal dispersion. Longitudinal dispersion is the natural broadening of the peak due to diffusion of the analyte through the mobile phase. As the mobile phase velocity is increased, less time is allowed for the analyte to disperse along the length of the column. This reduced dispersion results in a decreased H at higher velocities, as shown in Fig. 1. However, the opposite effect occurs for the C term if the mobile phase velocity increases. The C term represents the amount of time required for the analyte to travel to and from the stationary phase within the particle pores to the bulk mobile phase between the particles. As the mobile phase velocity increases, some of the analyte molecules are not able to partition out of the stationary phase within the pores before the bulk of the analyte molecules elute down the column.

The ultimate goal of maximizing N is to reach the highest possible peak capacity for a given analysis time. Peak capacity is defined as the maximum number of peaks possible existing within a set amount of time for a given R_s. Under isocratic conditions and assuming R_s is 1, peak capacity is calculated by Eq. 10,

\[
n_{c,iso} = 1 + \frac{\sqrt{N}}{4} \ln(1 + k'_{last})
\]  

(10)

where \(k'_{last}\) is the retention factor of the last eluting peak. Under gradient conditions and assuming R_s is 1, peak capacity is calculated by Eq. 11

\[
n_{c,grad} = 1 + \frac{(t_R, last - t_R, first)}{w_{avg}}
\]  

(11)
where $t_{R,first}$ and $t_{R,last}$ are the retention times of the first and last eluting peaks, respectively. In gradient elution, peak widths are assumed to be approximately equal due to a focusing effect at the head of the column. Since the initial mobile phase composition is primarily aqueous and therefore very polar, the analytes prefer to occupy the stationary phase. Due to this focusing, peak widths from gradient conditions do not dramatically change depending on $t_R$. It is due to this decoupling of peak width from $t_R$ that gradient conditions produce higher peak capacities than isocratic conditions. As a result, gradient conditions are the chromatographic conditions used to generate the data in Chapters 4-7 and the remainder of this discussion will focus on the use of gradient conditions.

2.2 Necessity of Two-Dimensional Chromatography

In the mid-1980s, Davis and Giddings developed the statistical overlap theory (SOT) to examine complex chromatograms for the likelihood that a peak present within a chromatogram is a singlet, \textit{i.e.} the $R_s$ is at least 1.0 for both adjacent peaks [7-9]. Davis and Giddings defined complex chromatograms as chromatograms containing a substantial number of overlapped peaks. SOT was developed using a combination of simulated Gaussian peak and tested using experimentally convoluted exponentially modified Gaussians (EMG). The key concept of SOT is that peaks present within the chromatogram are randomly positioned. This randomness ensures that some of the observed peaks are singlets while others are a combination of several peaks. The key parameter of SOT is the saturation factor ($\alpha$) which is the ratio of the number of true peaks present in the chromatogram divided by the peak capacity. Using SOT, the number of singlets, doublets, triplets, \textit{etc.} ($n_{type}$) can be calculated using Eq. 12,
\[ n_{\text{type}} = n_{\text{true}} (e^{-2\alpha})(1 - e^{-2\alpha})^{(m-1)} \]  

(12)

where \( n_{\text{true}} \) is the true number of peaks and \( m \) is the desired type (i.e., for singlets, \( m=1 \)). In addition, the total number of observed peaks (\( n_{\text{total}} \)) present in the chromatogram can be calculated by Eq. 13,

\[ n_{\text{total}} = n_{\text{true}} e^{-\alpha} \]  

(13)

From their simulations, Davis and Giddings concluded that it is possible to use the interval between observed peaks to estimate \( n_{\text{true}} \). While analyzing the experimental chromatograms, Davis and Giddings determined that there were 50 to 100% more peaks present within a chromatogram then were visually observable. In addition, Davis and Giddings concluded from analyzing select regions of the experimental chromatograms that there was only a 4 to 19% probability that a given peak was a singlet. Based on results obtained when developing SOT, Giddings [10] postulated that a second separation dimension was needed to fully resolve all compounds within a highly complex mixture.

### 2.3 Promise of Two-Dimensional Chromatography

Two-dimensional (2D) chromatography encompasses three distinct approaches. The first approach is what is known as heart cutting chromatography. In heart cutting chromatography (LC-LC), a selected portion of the effluent from the first column is transferred to a second dimension of separation. Typically, heart cutting is used for targeted analysis of a few compounds from complex samples. The second approach is what is known as offline LC×LC. In offline LC×LC, the entire effluent from the first dimension is collected into distinct fractions and each fraction is then injected onto the second dimension (2D) column. The advantage of this approach over the first approach is that it is possible to separate more compounds than using the
heart cutting approach. However, this approach relies on the ability to collect a large number of µL size fractions and then to reinject each of these fractions onto the second dimension. The third approach is what is known as online LC×LC. Like offline LC×LC, LC×LC collects fractions of the first dimension (1D) effluent. However, unlike offline LC×LC, in online LC×LC the different effluent fractions from the first dimension are collected by a sampling device and then immediately injected onto the 2D column during the same analysis time. The amount of time required to perform this technique is less than the typical analysis time of an offline LC×LC analysis. It is the third approach that will be the focus of the rest of the discussion. Both the offline and online approaches are considered as comprehensive LC×LC. Comprehensive implies that all of the fractions collected from the 1D column are injected into the 2D column. Giddings proposed that the peak capacity of a comprehensive 2D separation, $n_{c,2D}$, was equal to the product of the individual peak capacities, Eq. 14,

$$n_{c,2D} = n_{c,1D} n_{c,2D}$$

(14)

where $n_{c,1D}$ is the peak capacity of the first dimension and $n_{c,2D}$ is the peak capacity of the second dimension [10].

Before applying Giddings product rule to online LC×LC, Eq. 14 needs to be modified to take into account the impact of the sampling device on the overall peak capacity of the 2D separation space. To account for the necessary sampling of the first dimension in order to make use of an additional separation dimension, Davis et al. [11] developed Eq. 15,

$$n'_{c,2D} = n_{c,1D} n_{c,2D} \frac{1}{<\beta>}$$

(15)

where $n'_{c,2D}$ is the corrected 2D peak capacity and $<\beta>$ is the undersampling correction factor. The average undersampling correction factor is determined by Eq. 16,
\[ <\beta> = \sqrt{1 + \kappa \left( \frac{t_s}{\sigma} \right)^2} \]  

(16)

where \( \kappa \) is a empirical fitting parameter (determined to be 0.214±0.010 from simulated data) \( t_s \) is the first dimension sampling time and \( \sigma \) is the peak width of the 1D peak [11]. Davis et al. derived this equation by simulating 2D chromatograms and then using 2D SOT [12] to determine the peak widths of the first dimension. When conducting these simulations, Davis et al. made several assumptions regarding the nature of the 1D peaks. First, the peaks were assumed to be completely Gaussian in nature. Second, the peaks were generated under gradient elution conditions, \textit{i.e.}, all of the peaks had approximately the same base width. Third, the entire volume of the first dimension was transferred to the second dimension by the sampling device. Fourth, the ability of the second dimension to perform the separation was independent of 2D sampling time, \( t_s \). They calculated that the average effective 1D peak width, \( \langle \sigma_{\text{eff}} \rangle \) could be calculated using \( \sigma \) and \( t_s \) by Eq. 17,

\[ \langle \sigma_{\text{eff}} \rangle = \sqrt{\sigma^2 + 0.21t_s^2} \]  

(17)

They then were able to relate Eq. 17 to Eq. 16 by a relationship previously established by Liu and Davis [12], Eq. 18, to directly relate \( <\beta> \) to \( t_s \).

\[ \langle \sigma_{\text{eff}} \rangle = \sigma <\beta> \]  

(18)

Eq. 15 therefore represents the peak capacity limit for a defined set of experimental conditions for a given 2D separation.

However, it is entirely possible that the sample being analyzed does not fully cover the 2D separation space. Therefore, an additional term needs to be added to Eq. 15 to account for the degree of orthogonality of the 2D separation space. The degree of orthogonality of the 2D
separation is a result of how dissimilar the retention mechanisms of the first dimension and second dimension are compared to one another with respect to the sample being analyzed. The degree of orthogonality is calculated by determining the fractional coverage, $f_{\text{cov}}$, of the sample within a 2D chromatogram [13, 14]. With the introduction of $f_{\text{cov}}$, Eq. 15 is modified into Eq. 19 by multiplying Eq. 15 by $f_{\text{cov}}$ [15]. Eq. 19 therefore gives the effective peak capacity, $n^*_{c,2D}$, of a particular set of 2D experimental conditions for a given sample.

\[
n^*_{c,2D} = n_c^* n_c \frac{1}{f_{\text{cov}}} \tag{19}
\]

2.4 Optimizing $\beta$

In order to ensure that Eq. 19 produces a large theoretical peak capacity, $\beta$ needs to be minimized. As shown by Eq. 16, $\beta$ can be adjusted by changing the ratio between $t_s$ and $\sigma$. The primary means of adjusting this ratio is to optimize $t_c$. Potts et al. studied the impact of changing $t_s$ on $n'_{c,2D}$ [16]. The results were based on a homologue series of phenones and a small group of peptides. The second dimension column used for the homologous series was a Halo C18 RP column and the second dimension gradient was varied from 20% ACN to a variable final mobile phase composition depending on the second dimension gradient time ($t_g$) used. In addition, the results obtained in this study were generated using a $\kappa$ value of 3.35 which was determined using Eq. 20 from Davis et al. [11]

\[
<\beta> = \sqrt{1 + \frac{13.7}{N^2}}
\tag{20}
\]
where $N$ was determined by isocratic measurement of heptanophenone. In this study, the $t_s$ used in Eq. 16 was determined by adding a 3 s system re-equilibration time to the $^2t_g$. Based on results obtained from varying $^2t_g$, Potts et al. derived Eq. 21,

$$^2n_c = a(1 - e^{(-b^2t_s)})$$

where $a$ and $b$ are empirically derived fitting parameters, to estimate $^2n_c$. While $a$ and $b$ differed in magnitude depending on whether the phenone homologous series or a peptide mix was used, Potts et al. determined that the maximum value of $^2n_c$ was located between a $t_s$ of 12 and 21 s with an estimated maximum occurring at 14 s. If the value of $t_s$ was lower than 12 s, the calculated $^2n_c$ sharply decreased. However, if the value of $t_s$ was greater than 21 s, the calculated $^2n_c$ decreased more gradually in a somewhat linear fashion until at 50 s approximately 30% of the maximum $^2n_c$ had been lost. In addition to determining an optimal $t_s$ for maximizing Eq. 16, Potts et al. made one other important conclusion. They concluded that a limit exists for the payoff of maximizing $^1n_c$. After this limit is reached, they concluded that the only improvement to Eq. 16 occurs from maximizing $^2n_c$. Huang et al. confirmed the optimal range of 12 to 21 s for $t_s$ when analyzing complex maize samples [17].

Care must be taken, however, when optimizing Eq. 16. Murphy, Schure, and Foley performed a study to determine the necessary number of times a $^1D$ peak needed to be sampled in order to maintain a 2D $R_s$ of 1 [18]. The criterion established by this paper is commonly referred to as the Murphy-Schure-Foley (M-S-F) criterion. The M-S-F criterion states that a $^1D$ peak needs to be sampled approximately three (when the phase of the $^1D$ peak is 0) to four (when the phase of the $^1D$ peak is $\pi$) times. The phase of the $^1D$ peak is defined in terms of the $^1D$ $t_R$, $^1t_R$, in comparison to the center of the first dimension sampling window ($T$). Thekkudan et al. [19] developed Eq. 22,
\[ \phi = \frac{2\pi(T - t_R)}{t_s} \]  

(22)

to calculate the phase angle, \( \phi \), for a given \(^1\)D peak. The M-S-F criterion is based on the assumption that the base width of a \(^1\)D peak was 8\( \sigma \). Later, Seeley [20] studied the impact of duty cycles on the M-S-F criterion. A duty cycle was defined as the efficiency of transfer between the first and second dimensions by the sampling device. A duty cycle of 1 was considered to be completely comprehensive as is the case for LC\( \times \)LC. Seeley, unlike Murphy, Schure, and Foley, determined that the number of times a \(^1\)D peak needed to be sampled was independent of the phase if a Gaussian distribution was used to fit the sampled \(^1\)D peak. Seeley concluded that a \(^1\)D peak with a duty cycle of 1 only needed to be sampled three times to maintain a 2D \( R_s \) of 1.

However, given that the base width of a peak is typically defined as 4\( \sigma \), the M-S-F criterion uses an unusual definition for peak width. Khummeung et al. [21] developed modulation ratio (M\(_R\)) based on the standard peak width definition of 4\( \sigma \), Eq. 23,

\[ M_R = \frac{4\sigma}{t_s} \]  

(23)

for comprehensive two-dimensional gas chromatography (GC\( \times \)GC), although this principle also applies to LC\( \times \)LC. The primary conclusion of this study was that for quantitative results, a M\(_R\) of 3 is required. This result was consistent with the M-S-F criterion. Thekkudan et al. performed a similar study later for LC\( \times \)LC and concluded that for “well resolved” (i.e. 2D \( R_s \) equal to or greater than 1.5) a M\(_R\) of 2 to 5 was sufficient for simulated Gaussian peaks [19].

In order to achieve the optimal \( t_s \) as determined by Potts et al. [16] and Huang et al., the \(^2\)D gradient needs to be fast. By necessity, this requirement imposes two conditions. First, the dwell time of the system (the amount of time required for the beginning of the \(^2\)D gradient to
reach the beginning of the column) needs to be very low. Second, the amount of time required for the column to re-equilibrate (hereafter defined as $t_{\text{re-eq}}$) after the end the gradient needs to allow for a minimum of three column volumes of mobile phase to flush the column [16]. Both Potts et al. [16] and Huang et al. [17] used a re-equilibration time of 3 s for the $^2$D gradient. In order to achieve such a small $t_{\text{re-eq}}$, the dwell volume of the system needs to be very low and the flow rate of the second dimension needs to be very fast.

Stoll et al. accomplished the task of greatly reducing the dwell volume of a LC×LC system by making use of two second dimension pumps, as shown in Fig. 2 [22]. The dwell volume was reduced by allowing one of the $^2$D pumps to re-equilibrate while the other $^2$D pump performed the gradient elution. By using this setup, Stoll et al. was able to reduce the dwell volume of the system from 350 µL to 20 µL. Recently, the mixing chambers in new pumping systems (such as Agilent’s 1290 series pumps [23]) possess low enough dwell volumes that such extreme system modifications are no longer necessary. Schellinger et al. [24] used the modified system of Stoll et al. [22] to test the reproducibility of the system in terms of $t_R$. Schellinger et al. determined that for non-ionizable analytes the run-to-run retention time precision was smaller than 0.002 min with $t_{\text{re-eq}}$ corresponding to only two column volume flushes. In order to achieve full equilibrium of the column, the column would have needed to have been flushed with ten to fifteen column volumes.
While the solution to achieving a very low dwell volume is system dependent, the very fast flow rate requirement can be achieved by adjusting the experimental conditions. Both Potts et al. and Huang et al. used a flow rate of 3 mL/min in the second dimension. At normal operating temperatures of 25 °C, a 3 mL/min flow rate would result in an enormous amount of backpressure on the 2D pump, well outside the typical operating limit of 400 bar for even short columns. In both studies, the 3 mL/min flow rates were achieved by operating the second dimension at high temperature (specifically 100 °C).

Teutenberg defines high temperature as corresponding to temperatures of the mobile phase between 60 and 374 °C [25]. Teutenberg then suggests an operational range of 60 to 200 °C as being acceptable, given the current level of column technology. The primary benefit of operating the 2D column within this range is the reduced viscosity of the mobile phase. The reduced viscosity increases the diffusion coefficient ($D_m$) of the solutes in the mobile phase. This increase in $D_m$ directly impacts the B and C terms in Eq. 7, given that $B$ is directly proportional to $D_m$ and $C$ is inversely proportional to $D_m$ [25]. This increase in $D_m$ does not impact $H_{min}$ since according to Eq. 8 the change in $D_m$ would cancel out. However, $u_{opt}$ is impacted according to Eq. 9 since $u_{opt}$ would increase in direct proportion to $D_m$. The increased
\( u_{\text{opt}} \) is accommodated by a corresponding decrease in pressure. Although the columns currently used in RP-LC are packed columns, typically the Hagen-Pouiseuille equation is used to show the impact of reduced viscosity (\( \eta \)) on the pump pressure (\( \Delta P \)) for capillary columns [25], Eq. 24,

\[
\Delta P = \frac{128 \eta LF}{d_c^4 \pi}
\]

where \( F \) is the flow rate of the mobile phase and \( d_c \) is the inner diameter of the capillary. As can be seen from Eq. 24, the increase in \( F \) is offset by decreasing the viscosity of the mobile phase.

Abbott et al. determined that proper pre-heating of the mobile phase prior to the column was necessary to ensure that the high temperature effectively enhanced the performance of the separation [26]. Typically, a small loop of metal tubing, with a very small volume, is used as a column pre-heater. Metal is used in the place of poly ethyl ethyl ketone (PEEK) tubing due to the greater thermal conductivity of metal versus PEEK. In addition, Thompson et al. determined that a 2.1 mm wide column was better suited to high temperatures than a 4.6 mm wide column due to a lessening of thermal mismatch between the mobile phase outside of the column and inside the column [27]. As a result, Thompson et al. concluded that with the use of narrower columns, due to the reduced interstitial velocities commonly used with narrower columns, the mobile spends more time in the eluent pre-heater coil before reaching the column. As such, the mobile phase has more time to reach the desired temperature prior to entering the column. In addition, the necessary volume of the eluent pre-heater coil was less if a narrower column was used.

2.5 Optimizing \( f_{\text{cov}} \)

The second term of Eq. 19 that needs to be optimized is the degree of fractional coverage.
Fractional coverage is measured on a scale of 0 to 1 where 0 is a complete lack of orthogonality (i.e. a straight line) and 1 is completely orthogonal (i.e. the entire 2D separation space is used). Several methods have been examined for the calculation of $f_{cov}$. Gilar et al. proposed the use of a grid of boxes across the 2D separation space. The number of boxes is equal to the number of compounds in the sample or chosen by the user if the number of compounds is not known [13]. If a peak is located within a given box, then the box is considered to be occupied. Obviously, if the size of the box is very large then it is more probable that a given box is occupied. Conversely, the smaller the boxes are, the less likely the chance that a box is filled. Rutan et al. compared the capability of the Gilar box method to ecological home range methods for the calculation of $f_{cov}$ using a known set of $t_R$ values [14]. Rutan et al. determined that the minimum convex hull method (which works by placing a “rubber band” around the edge $t_R$ point values within the 2D space) provided results as precise as those from the Gilar box method. The primary advantage of the minimum convex hull method, compared to the Gilar box method, is that the minimum convex hull method does not require an arbitrarily defined parameter and as such is user independent. Based on the recommendation of Rutan et al., the minimum convex hull will be used to calculate $f_{cov}$ in Chapter 4.

In order to maximize $f_{cov}$, the appropriate selection of $^1$D and $^2$D columns needs to be addressed. Before a column for the first dimension can be considered, an appropriate column for the second dimension needs to be selected to allow for the conditions stated in Section 2.4 to be met. Teutenberg focuses on four main types of phases: 1) silica-based bonded phases; 2) titanium dioxide-based phases; 3) polymeric based phases (such as polystyrene); and 4) zirconium dioxide-based stationary phases [25]. Silica based stationary phases using ethylene bridges between the silica allow for the silica to be used at pHs between 3 and 6 and at high
temperature (100 °C) [25]. However, the majority of silica-based bonded stationary phases available are reversed phase alkyl chains. Therefore, an alternative type of column is needed to ensure that the degree of orthogonality is low if a reversed phase alkyl chain-based stationary phase is used for the 1D column.

In order to make zirconium and titanium dioxide stationary phases capable of reversed phase retention mechanisms, the surface of the particles is modified with a hydrophobic coating (either a polymer coating for titanium dioxide and zirconium dioxide or carbon for zirconium dioxide). The majority of the titanium and zirconium dioxide-based columns are produced by Zirchrom [28]. The titanium dioxide-based columns (trade name Sachtopore) were studied in 2000 by Winkler and Marme for their retention characteristics [29] and are a very recent addition to their product catalog. In comparison, carbon-coated zirconium dioxide-based particles have been studied as early as 1990 for RP-LC [30]. Teutenberg et al. examined the ability of bare silica (at 120 °C) and metal oxide (at 150 °C) particles to withstand high temperature and acidity [31]. The columns were heated for five hours and then allowed to cool to room temperature. Teutenberg et al. concluded that the performance of the bare silica based particles was found to rapidly degenerate. In comparison, the metal oxide particles were found to be rugged and stable over the length of the analysis. However, this analysis was conducted using an aqueous mobile phase without any organic modifier. Later, Teutenberg noted that in the case of polymer-coated titanium dioxide stationary phases that the polymer coating can be removed with organic solvent at high temperature [25].

Weber et al. prepared carbon clad zirconium dioxide particles by using chemical vapor deposition (CVD) to deposit the carbon onto the surface of the zirconia dioxide [30]. CVD was carried out at elevated temperatures with toluene providing the carbon source. The particles
were then washed with THF or heptane to remove byproducts of the CVD process. Weber et al. concluded that the unique nature of the carbon coated zirconium dioxide particles allows for a unique retention compared to traditional RP bonded silica particles. In addition, Weber and Carr tested the ability of the carbon-clad zirconium dioxide particles to separate isomers. They found that in comparison to traditional RP bonded phases that the retentivity of the carbon-clad zirconium dioxide particles was greater, and these materials were better able to separate isomers [32]. Later, Weber et al. determined that the use of saturated hydrocarbons for the deposition of carbon onto zirconium dioxide particles resulted in a more efficient column than if unsaturated hydrocarbons were used [33]. Weber et al. also determined that the carbon coated zirconium dioxide particles possessed electronic (π-π) interactions in addition to hydrophobicity. Columns made using this stationary phase were used for data analyzed in Chapters 4-6.

Paek et al. attempted to improve the carbon clad stationary phases by using alumina in place of zirconium [34]. While comparing the carbon clad alumina stationary phase to the carbon clad zirconia stationary phase, Paek et al. determined that the carbon clad alumina stationary phase maintained the selectivity and retentivity of the carbon clad zirconia phase. In addition, Paek et al. determined that the carbon clad alumina stationary phase required fewer monolayers of carbon than the carbon clad zirconia stationary phase. However, while the number of monolayers was less for alumina, the percent by weight of carbon was 24 % compared to zirconia’s 8 % weight percent of carbon, resulting in increased retention. The increased retention resulted in reduced peak distortion from increasing amounts of sample solvent, making this column extremely suitable for the second dimension of a LC×LC system. However, an alumina-based stationary phase is problematic to produce due to the difficulty in producing LC grade alumina. To address this problem, Paek et al. deposited carbon onto an
alumina modified silica particle [35]. By using a silica particle, Paek et al. avoided the necessity of using LC grade alumina particles. The alumina was applied to the surface of the silica particle by titration, forming approximately a monolayer. The carbon was then deposited using CVD. Once again, the carbon phase was approximately four to five layers thick (25 and 32 % weight by carbon), and this phase provided a chromatographic performance very similar to the carbon clad alumina particles. Finally, Paek et al. extended their approach to core shell silica particles [36]. The primary advantage of using core shell particles compared to fully porous particles is reduced mass transfer (C term from Eq. 7) and a lower backpressure on the 2D pump. Paek et al. determined that with the carbon clad core shell silica particles, the strong solvent from the first dimension had less of an impact on the peak shape and width of the 2D peaks. Columns made using this stationary phase were used for data analyzed in Chapter 7.

Given that the carbon clad columns described above are the current columns of choice for ultra-fast high temperature LC×LC, an appropriate 1D column needs to be determined that is highly orthogonal to the separation mechanism of the 2D column. The parameters needed to estimate the degree of orthogonality for silica based columns can be approximated by means of the linear solvation energy relationship equation, Eq. 25 [37],

$$\log k' = \log k'_{ref} + \eta' H + \sigma' S^* + \beta' A + \alpha B + \kappa' C$$

(25)

where $\eta'$, $\sigma'$, $\beta'$, $\alpha$, and $\kappa'$ are solute specific characteristics and $H$, $S^*$, $A$, $B$, and $C$ are the matching column characteristics. Each of the matched pairs, i.e., $\sigma'S^*$, represents a particular interaction between the solute and column. $\eta'H$ interactions represent the degree of hydrophobic interaction between the solute and column. $\sigma'S^*$ interactions represent the degree of steric resistance of the bound ligands to accommodate the solute when the solute partitions into the stationary phase., $\beta'A$ interactions represent the degree of hydrogen bond interactions between a
basic hydrogen bond solute and column (due to non-ionized surface silanols). αB interactions represent the degree of hydrogen bond interactions between hydrogen bond acidic solute and column (due to sorbed water). κ'C interactions represent the degree of cation exchange between positively charged solutes and column (due to ionized surface silanols). $k_{ref}'$ is the $k'$ of ethylbenzene. Ethylbenzene was used as a reference since its retention is predominately due to its hydrophobicity. Once the parameters in Eq. 25 have been determined, the selectivities of columns are compared using by the column selectivity function, [38], Eq. 26.

$$F_s = \sqrt{\left[12.5(H_2 - H_1)\right]^2 + \left[100(S^*_2 - S^*_1)\right]^2 + \left[30(A_2 - A_1)\right]^2 + \left[143(B_2 - B_1)\right]^2 + \left[83(C_2 - C_1)\right]^2}$$

(26)

where the subscripts refer to the two columns being considered.

Zhang and Carr made use of $F_s$ to devise a visual approach for the comparison of column selectivities [39]. They devised four different normalized plots (a $S^*$-B-C triangle, a $S^*$-A-C triangle, a $S^*$-A-B triangle, and an A-B-C triangle) while comparing 366 commercial RP columns. Normalization was carried out by dividing each term ($S^*_1$, $S^*_2$, $A_1$, $A_2$, $B_1$, $B_2$, $C_1$, and $C_2$) by the hydrophobicity ($H_1$ and $H_2$) [39]. Gu et al. used the $S^*$-B-C triangle plot to determine the optimal $1^D$ column for LC×LC assuming that the $2^D$ column was a carbon clad zirconium oxide column [40]. Gu et al. compared six $1^D$ columns (with a carbon clad metal oxide column as the $2^D$ column) for vegetable and fruit extract samples. The six $1^D$ columns were determined to be the most representative of the commercially available RP-LC columns. Gu et al. used the Gilar box counting method to determine the degree of orthogonality between the six $1^D$ columns. The overall conclusion of this study was that as long as the $2^D$ column was a carbon clad metal oxide phase it did not matter what the $1^D$ column was. The differences in calculated $f_{cov}$ values were determined to be not significantly different for the six $1^D$ columns investigated [40]. However, the primary requirement for a $1^D$ column still holds. The retention of the sample on
The 1D column should be low enough that small amounts of organic modifier are necessary to elute the sample. This allows for the sample to be strongly retained at the head of the 2D column.

After the 1D and 2D columns have been selected, \( f_{\text{cov}} \) can be further optimized by allowing for a changing gradient in the second dimension. Jandera et al. describes three different types of 2D gradients: 1) “full in fraction” (FIF) gradients where the organic modifier is ramped from 0 to 100% organic modifier every \( t_s \); 2) “segment in fraction” (SIF) gradients where the starting point of the 2D gradient is changed during the course of the run; and 3) “continuously shifting” (CS) gradients where a 2D gradient is carried out over the entire course of the 1D elution [41]. Since an optimized parallel 2D gradient possesses the same duration as the 1D gradient, an optimized parallel 2D gradient does not need to be re-equilibrated every \( t_s \). An illustration of the three types of 2D gradient profiles is presented in Fig. 3. A comparison of the three approaches was made using typical RP-LC alkyl bonded silica columns. Specifically, the 1D column was a Discovery HS PEG 2.1x150 mm with 5 \( \mu \)m particles and the 2D columns were C18 2.7 \( \mu \)m core shell 3x30 mm columns (either Acentis Express or Kinetex). The analytes used in this study were phenolic acids and flavones.

The FIF gradients were found to have the benefit of compressing the peak widths the most but at the cost of 2D separation space usage. The 2D peaks were found to occur mainly only within a diamond stretching from the lower left corner of the 2D separation space to the upper right corner of the 2D separation space. The SIF gradients were found to exhibit a mix of properties of the CS and FIF gradients with some compression of the peaks occurring while allowing for a smaller re-equilibration time due to the reduced difference between the initial and final organic modifier concentrations. The CS gradients were found to increase the widths of the
2D peaks due to the relatively isocratic nature of the 2D gradient; however, the 2D gradient allowed for a greater portion of the 2D separation space to be utilized due to the lack of a need for re-equilibration of the 2D column. They did note that the end of the 2D gradient still needed to be adjusted to allow for all analytes to elute prior to the sampling device switching. In addition, due to the changing 2D gradient and the fact that 1D peaks are sampled multiple times, the retention times of successive 2D peaks from the same 1D peak were different. As a result, Cesla et al. developed an algorithm to correct for gradient induced shifting of the peaks between consecutive 2D chromatograms. Finally, while this approach was successfully applied to sparsely populated chromatograms, further work is being done to test the approach on more complicated real world samples [42].

![Fig. 3](image-url): Conceptual diagrams of advancing gradients in the 2D of LC×LC. From top to bottom: “full in fraction” (FIF), “segment in fraction” (SIF), “continuously shifting” (CS). The dashed line is the corresponding 1D gradient. Reproduced from reference [43] with permission from Elsevier.

2.6 Sampling Device Setup Strategies
At the heart of the LC×LC system is the sampling device. Many different types of sampling device configurations have been designed, with each type having both advantages and disadvantages. The standard sampling device setup is illustrated in Fig. 4, where the red and green colors illustrate the changing 1D flow paths. At first, the effluent from the 1D column is collected in the first loop while the 2D pump is pushing mobile phase from the second loop onto the 2D column, Fig. 4A. Later, after the valve within the sampling device switches, the flow paths change as illustrated in Fig. 4B. Now, the effluent from the 1D column is collected in the second loop while the previously collected 1D effluent is injected onto the 2D column by the 2D pump. This process repeats every t_s time interval until the end of the analysis. The data analyzed in Chapter 5 were produced using this type of sampling device setup.

**Fig. 4:** Typical setup of the sampling device for LC×LC. The red and green colors denote the changes in the flow path depending on the position of the valve within the sampling device.

In addition to the loops shown in Fig. 4, guard columns containing the same material as the 2D column have been used to trap analyte present in the 1D effluent. An example of such a setup is shown in Fig. 5. In Fig. 5, the RP trap 1 and 2 boxes indicate the guard columns. Pepaj *et al.* used the trapping columns to capture human saliva proteins eluting from a pH controlled first dimension separation. Since the trapping columns contain a polymer-coated silica particle stationary phase, the human saliva proteins were captured, because the 1D mobile phase effluent was completely aqueous. Pepaj *et al.* avoided having to elute the proteins through the entire
trapping column by employing a backflushing method [44]. A danger inherent with this strategy is the amount of time required for the \(^2\)D gradient to reach the front of the trapping column, although the same could be said if the proteins were allowed to elute through the trapping columns. This amount of time extends the duration of \(t_s\), which in turn increases \(\beta\).

**Fig. 5:** An example of a trapping column based sampling device for LC×LC. Reproduced from reference [44] with permission from Wiley.

Sweeney and Shalliker have devised a heart-cutting approach using a trapping column (a C18 column) for low molecular weight polystyrene oligomers, while the \(^1\)D column contained C4 modified silica particles, and the \(^2\)D column contained carbon clad zirconia particles [45]. The temperature of the trapping column was adjusted from 0 to 50 °C. Sweeney and Shalliker were able to successfully transfer effluent from the \(^1\)D column to the trapping column thirty two times before observing a substantial decrease in the \(R_s\) for the \(^3\)D separation. Like Pepaj et al., Sweeney and Shalliker employed backflushing to elute the oligomers from the trapping column. However, given that the molecular weights of the polystyrene oligomers were 770 Da, this approach may not be suitable for lower molecular weight compounds typically analyzed by LC×LC.
Filgueira et al. modified the apparatus illustrated in Fig. 4 by adding a splitter pump [46]. Instead of tubing going directly to waste, a length of tubing connected the splitter pump to the sampling device, Fig. 6. The splitter pump controls how much of the \( ^1 \)D effluent is transferred to the sampling device. The primary advantage of this sampling device setup is the expansion of available space within the first dimension. Typically, the first dimension is performed at low flow rates to prevent too much of the \( ^1 \)D effluent from being injected onto the \( ^2 \)D column at a time. However, with the sampling device setup shown in Fig. 6, the flow rate of the first dimension can be performed at closer to normal levels, allowing for less time to re-equilibrate the column between samples, as well as reducing the dwell time of the system. The primary disadvantage is that the sample needs to be concentrated because only a portion of the sample reaches the \( ^2 \)D column while the rest of the sample proceeds to waste. This sampling device setup was used to produce some of the data analyzed in Chapters 4 and 7.

![Diagram](attachment:image.png)

**Fig. 6:** An example of the use of an active splitter immediately after the sampling device to control the amount of \( ^1 \)D effluent transferred to the \( ^2 \)D column. Reproduced from reference [46] with permission from American Chemical Society.

Finally, Groskreutz et al. developed selective \( \text{LC} \times \text{LC} \) (s\( \text{LC} \times \text{LC} \)) by creating a sampling device design that bridges the gap between LC-LC and LC\( \times \)LC, illustrated in Fig. 7 [47, 48].
This sampling device allows for multiple selected portions of the $^1$D effluent to be stored in loops (labeled as L1 to L6 in Fig. 7) and then subsequently injected onto the $^2$D column. The primary advantage of this sampling device design is the decoupling of the $^2$D analysis time from $t_s$. For example, some of the data analyzed in Chapters 6 and 7 are produced using a $^2$D analysis time of 20 s with a $t_s$ of 2 s. If the data had been produced using the sampling device design in Fig. 6, then the $^2$D analysis time and $t_s$ would have to of been equal, i.e., 20 s or 2 s. The primary disadvantage of this sampling device design is the complexity of the setup and the requirement that the sampling devices switch simultaneously. If the sampling devices do not switch simultaneously, then some of the $^1$D effluent is lost.

Fig 7: An example of the modified sampling device used in sLC×LC. Panel A shows the first loop being filled with $^1$D effluent. Panel B shows the second loop being filled with $^1$D effluent. Panel C shows the injection of the contents of the first loop onto the $^2$D column. Reproduced from reference [47] with permission from Elsevier.
CHAPTER 3: Overview of Two-Dimensional Chromatography Quantification

3.1 Transitioning From One-Dimensional to Two-Dimensional Quantitation

Quantification is accomplished in one dimensional (1D) chromatography by summation of the area between a peak and a determined baseline. However in LC×LC, as mentioned in section 2.4, a single 1D peak is sampled into multiple 2D peaks. This sampling introduces two problems not present in 1D quantification: 1) the need to merge multiple 2D peaks into a single 2D peak and 2) a decreased signal to noise ratio for each 2D peak due to dilution. Three different approaches to peak merging and quantification have been proposed for 2D data.

The first approach was proposed by Peters et al. [49]. This approach is an extension of normal 1D peak detection techniques to 2D peaks. As with 1D peak detection techniques, this approach is typically performed on a single wavelength after the raw data has been reshaped into a 2D chromatogram. The approach relies on the use of the derivatives to determine where a 2D peak starts and stops and the retention time of the peak based upon user defined thresholds, illustrated in Fig. 8. The retention time of the peak is first determined by determining the time at which the first derivative crosses zero. Alternatively, the second dimension can be used to determine the retention time by selecting the minimum time point of the second derivative curve as being the retention time. Proceeding outward from the estimated retention time, the start and stop of the 2D peak are determined when the second derivative passes through the user defined threshold with a negative slope. In order to account for the presence of noise in the raw chromatogram, the Savitzky-Golay derivative smoother is used to smooth the chromatogram.
while simultaneously calculating the derivative [50]. The width of the Savitzky-Golay smoothing window is determined by the Durbin-Watson test. The Durbin-Watson test determines the optimal level of smoothing by calculating the residual of each data point after smoothing and then calculating a correlation value. The Durbin-Watson test converges (indicating that optimal smoothing has been performed) if the calculated statistic is 2 [50].

![Diagram](image)

**Fig. 8:** Illustration of the use of derivatives for a 2D peak to determine the peak maximum, the peak start, and the peak end. The blue line corresponds to the non-derivative peak. The green line corresponds to the first derivative of the peak. The red line corresponds to the second derivative of the peak. Adapted from reference [50] with permission from Elsevier.

After each 2D chromatogram has been analyzed, the approach then merges the prospective 2D peaks based on two criteria; unimodality and overlap. The overlap criterion assures that the second dimension peaks being merged possess a minimum degree of overlap in their start and stop times. An illustration of how the overlap criterion is performed is shown in Fig. 9. In Fig. 9, five different scenarios are illustrated with the x-axis representing the 1D time scale and the y-axis representing the 2D time scale. In each example, the fraction of overlap between peak A and B is calculated with A in the numerator and B in the denominator. If the calculated fraction exceeds the established overlap criterion then the two prospective 2D peaks are considered to be part of the same 2D peak. The unimodality criterion assures that each 2D
peak will only possess a single maximum. The peak heights at the respective retention time are compared and the 2D peak is split if a peak height increases after a minimum has been reached. This approach was later modified when Stevenson et al. squared the normalized signal of each 2D chromatogram [51]. The idea was that by squaring each 2D peak to increase its size, the impact of noise on the chromatogram was reduced. Consequently, this modified approach was shown to possess the ability to detect peaks at lower intensities than is otherwise possible, assuming an appropriate threshold was selected. This modified approach is examined in Chapter 4.

![Diagram](image)

**Fig. 9:** Five scenarios common in LC×LC when determining the degree of overlap. If the calculated overlap is greater than or equal to the overlap criterion, peak A and B are considered to be part of the same 2D peak. Reproduced from reference [50] with permission from Elsevier.

The second approach was developed by Reichenbach et al. [52, 53]. Unlike the first approach which analyzes the data in a sequential fashion, Reichenbach et al. developed a procedure that utilizes both the 1D and 2D peak shapes when determining which 2D peaks should be merged to form a single 2D peak. This approach treats a 2D chromatogram at a single wavelength as a pixelated picture. Peak detection is then carried out using the drain algorithm [54], a modified version of the inverted watershed algorithm [55], illustrated in Fig. 10. The drain algorithm begins by identifying the pixel with the highest intensity in the chromatogram, labeled as A in Fig. 10. The algorithm then proceeds down the peak assigning pixels to the 2D
peak until a minimum is reached, labeled as B in Fig. 10. The borders of the 2D peak are determined using the unimodality criterion. The base of the peak is defined by determining the minimum unassigned pixels adjacent to the peak. In the example illustrated by Fig. 10, the base of the peak would occupy that dark trench at the bottom of the peak and the ridge rising on all sides of the peak would be the beginnings of other 2D peaks. However, since the drain algorithm treats all minimum pixels as part of the same peak, the use of the drain algorithm potentially results in 2D peaks that are not chemically possible, as shown in Fig. 11.

![Diagram](image)

**Fig. 10:** A representative three dimensional peak shape where A signifies the point with the highest intensity pixel and B signifies the region where the pixels with the lowest intensity are located.

As demonstrated by Vivo-Truyols *et al.*, the drain algorithm results in non-continuous 2D peak profiles along the edge of a 2D peak, letters a-f in Fig. 11 [56]. As will be shown in Chapter 4 though, if the baseline correction option is used then the non-continuity does not significantly impact the quantification. Two additional criteria are used in to make certain that the selected region of the 2D is stored as a peak. A collection of pixels is determined to be a 2D peak if the sum of intensities for a sufficient number of pixels around the maximum pixel (the minimum number being defined by the user) exceed a user defined minimum value. The effectiveness of the drain algorithm for LC×LC data is examined in Chapter 4.
Fig. 11: A representation of the selected border of a GC×GC 2D peak by the drain algorithm. The letters a-f denote non-continuous portions of a 2D chromatogram merged into the overall 2D peak. Reproduced from reference [56] with permission from Elsevier.

The third approach was developed by Mondello et al. while investigating the change in limit of detection (LOD) for LC×LC relative to 1D-LC [57]. In order to merge multiple 2D peaks and then quantify the resulting 2D peak, they developed a new software package, Chrom\textsuperscript{square}. Peak merging is accomplished by identifying peaks that elute within consecutive 2D data point windows. The prospective peaks are then checked to ensure that only one maximum is present. Finally, the program allows for retention time drifting in consecutive 2D chromatograms by enabling a drift tolerance. Unlike the Peters et al. and Stevenson et al. methods, Mondello et al. does not reshape the raw chromatogram into a 2D format. Instead, once the retention time of a peak has been determined, the peak merging takes into account the modulation frequency and detector sampling to predict the next occurrence of the 2D peak. If a peak is present at the predicted retention time and obeys the unimodality rule then the prospective peak is matched to the previous peak. This process continues until all peaks have been matched. Quantification is accomplished through the use of “data point triangles”, illustrated in Fig. 12.
Fig. 12: A representation of the triangle summation method devised by Mondello et al. for a single 2D peak. Reproduced from reference [57] with permission from American Chemical Society.

The “data point triangles” work by reflecting an integration line between the curve of a 2D peak and summing the resulting triangles to achieve the peak area. Mondello et al. asserted that this method was superior to the method used by Reichenbach et al. However, in a rebuttal, Reichenbach [58] showed that this method of peak integration was identical to simply summing the area under a curve and therefore was identical to the approach used by Reichenbach et al. Using the “data point triangles” method, Mondello et al. determined that the LOD for LC×LC doubled in comparison to 1D-LC. Mondello et al. attributed this increase in the LOD to the dilution of the sample by the sampling process. The results obtained by Mondello et al., however, are not the same as previously determined by Schure [59].

In 1999, Schure studied the impact of dilution on the LOD by theoretical means, assuming that isocratic conditions were used [59]. Schure determined that the dilution factor for LC×LC was 2500 if a split factor of 17.8 was used between the first and second dimensions. The split factor indicates that this dilution factor only holds true if 17.8% of the 1D effluent is transferred to the second dimension. However, Horvath et al. later concluded that this dilution was too pessimistic and that the dilution factor was only 200-300, an order of magnitude less [60]. Unlike Schure, Horvath et al. based their results on theoretical calculations, as well as experimental results under gradient conditions. The decrease in dilution factor was attributed to
an increase in the $^{2}$D peak height compared to the first dimension due to focusing of the $^{1}$D effluent at the head of the $^{2}$D column. Horvath et al. noted that the amount of organic modifier present in the $^{1}$D effluent played a significant role in the width and height of the $^{2}$D peaks. Larger and narrower $^{2}$D peaks were found to occur when lower organic modifier was present in the $^{1}$D effluent.

Later, Harynuk et al. calculated the error introduced by the sampling device for GC×GC via a series of simulated data sets [61]. Harynuk et al. compared the impact of simply summing the areas of the $^{2}$D peaks to the use of a generalized rank annihilation method (GRAM), a multi-way method, in terms of percent residual standard deviation (% RSD). Unlike the sampling device in LC×LC, the sampling device in GC×GC is not currently composed of two loops. Instead, the effluent from the first dimension is cooled down using liquid N$_{2}$ and then vaporized into the second dimension. As such, the impact of % RSD was analyzed solely through the error in correctly summing multiple $^{2}$D peaks and the impact of the sampling phase on the shapes of the $^{2}$D peaks. Harynuk et al. determined that in order to achieve a % RSD of 1, a $M_{R}$ of 3, calculated according to Eq. 23, is necessary. A $M_{R}$ of 3 corresponds to four or five $^{2}$D peaks for each $^{1}$D peak. Harynuk et al. also determined that if a minimum of three $^{2}$D peaks were summed then the % RSD was considered to be minimized. However, Harynuk et al. determined if only two $^{2}$D peaks were present, corresponding to a sampling phase of approximately 0.5, calculated according to Eq. 22, then the % RSD was 6%. Finally, these authors determined that as the $M_{R}$ decreased from 3.0 to 1.5, % RSDs of 1.8% and 0.6% % RSDs, corresponding to $M_{R}$ values of 1.5 and 3.0, were obtained by GRAM. Harynuk et al. concluded that if a multi-way method was being used to quantify the peaks then a minimum $M_{R}$ of 3.0 should be used to ensure that a sufficient number of first dimension data points are present.
Thekkudan et al. extended the work of Harynuk et al. to LC×LC using a combination of simulated and experimental data [62]. Thekkudan et al. compared the summing of the 2D peak areas to fitting a Gaussian curve across the 2D peak areas and calculating the area under the curve. Thekkudan et al. determined that the Gaussian fitting technique gave % RSDs less than 1% assuming that three 2D peaks were present. This matches what Harynuk et al. previously determined [61]. However, if there were only two 2D peaks above the baseline the accuracy and precision of the Gaussian fitting method become quite poor [19]. Based on simulations, Thekkudan et al. determined that the % RSD was approximately 2% if the M_R was between two and five. Thekkudan et al. also concluded that the Gaussian fitting technique performed the same regardless of the sampling phase ratio if the M_R was between two and five.

The majority of these approaches suffer from the same disadvantages. Since each of these approaches relies on some form of integration to determine the volume of a 2D peak the approaches are only accurate for peaks with a R_s of at least 1.5. However as was indicated in section 2.2, SOT predicts that only a small number of visible peaks will be composed of a single analyte. Also, each of these approaches assumes that consecutive 2D peaks are from the same 1D peak. This disadvantage may be addressed by comparing the spectra of the prospective 2D peaks. However, this solution assumes that the two prospective peaks are pure enough to ensure a correlation value sufficient to meet the required standard. A possible solution to both the disadvantages would be to use multi-way methods.

3.2 Trilinear and Quadrilinear Methods

The notation that will be used for the rest of this discussion is an underscored bold capital, \( \mathbf{X} \), for arrays with dimensions greater than two, bold capital letter, \( X \), for two-way arrays, a bold
lower case letter, \( v \), for vectors and an italicized (lower or upper case) letter, \( a \) or \( A \), for scalars as suggested by Kiers [63].

Multi-way methods, such as GRAM and parallel factor (PARAFAC) analysis, decompose the raw data, \( \mathbf{X} \), into two or more matrices. The number of matrices generated is determined by the dimensionality of the data. In the case of LC×LC-DAD data, a single sample is three dimensional in nature with dimensions \( I \times J \times L \) where \( I \) is the number of data points in the second dimension, \( J \) is the number of data points in the first dimension and \( L \) is the number of channels recorded by the multi-channel detector, as illustrated in Fig. 13. In the case of a DAD, the addition of the \( L \) dimension allows for compounds to be identified by three different characteristics, the \( 1 \)D retention time, the \( 2 \)D retention time, and the ultraviolet-visible (UV-VIS) spectrum of the compound.

![Figure 13](image)

**Fig. 13:** A schematic representation of the four-way structure of multiple LC×LC-DAD samples where \( I \) denotes the number of points in second dimension, \( J \) denotes the number of points in the first dimension, \( L \) denotes the number of points in the UV-VIS spectrum, and \( K \) denotes the total number of samples. Adapted from reference [2] with permission from Wiley.

The GRAM model was originally developed for two-way data to calculate the concentration of an analyte in an unknown sample [64, 65]. GRAM is a non-iterative eigenvalue/vector based analysis method. As such, GRAM would return the component profiles of the two dimensions, i.e., the emission and excitation spectra in the case of fluorescence data. The component profiles are saved as eigenvectors and the concentrations are saved as eigenvalues. The third dimension must consist only of two samples [66]. In the case of quantification, the two samples consist of the unknown sample and a standard sample where the
concentration of the analyte of interest in the standard is known. The ability to quantify the relative concentration of only two samples is the primary disadvantage of using GRAM for LC×LC-DAD data.

An alternative is to use the PARAFAC model. A three dimensional PARAFAC model is defined by Eq. 27 [66, 67],

\[ x_{i,j,l} = \sum_{n=1}^{N} a_{i,n} b_{j,n} c_{l,n} + e_{i,j,l} \]  

(27)

where \( x_{i,j,l} \) is a specific data point within the three dimensional space, \( a_{i,n} \), \( b_{j,n} \), and \( c_{l,n} \) are elements of \( A \), \( B \), and \( C \). \( A \), \( B \), and \( C \) are two-way arrays with a variable number of rows (the number depends on which dimension of the data they are modeling) and \( N \) columns. \( e_{i,j,l} \) is the residual element from the error two-way array and \( N \) is the total number of components. \( A \), \( B \), and \( C \) are not tied to a particular chemical nature and can be changed depending on how the initialization is accomplished (further discussed in section 3.6). This model can be re-written as Eq. 28 to solve for a particular dimension, \( A \) in this example,

\[ X_{I,J,L} = A(C \odot B) + E \]  

(28)

where \( \odot \) is the Khatri-Rao product. Eq. 28 can be easily extended into four dimensions by Eq. 29,

\[ X_{I,J,K,L} = A(D \odot C \odot B) + E \]  

(29)

where \( D \) is a two-way array containing the resolved component profiles for the fourth dimension. Unlike GRAM, an algorithm for PARAFAC has been developed using alternating least squares (ALS) where the algorithm is terminated based on one of three conditions being met. The algorithm is typically terminated when a maximum number of iterations has been reached, when the difference in the sum squared residuals between \( X \) and \( X_{fit} \) (where \( X_{fit} \) is the \( X_{I,J,L} \) or \( X_{I,J,K,L} \)
produced by Eqs. 28 and 29 without the addition of E) reaches a selected convergence criterion, or when the algorithm fails to improve the quality of the fit for a predetermined number of times successively. The primary advantage of the PARAFAC model over the GRAM model is the ability to apply constraints to the components to improve the resolved matrices. The PARAFAC algorithm has been shown to produce a unique solution, implying that only one correct solution exists, provided the data meets certain criteria [68]. Uniqueness is achieved, in the case of four dimensional data, if the components in three matrices of the four matrices are linearly independent and that no two components in the fourth matrix are linearly dependent [67]. In addition to the non-linear dependence requirement, the PARAFAC model requires that the number of components must be correctly chosen and that \( X \) has a multilinear structure for uniqueness to occur. If the number of components is not correct, the final \( A, B, C, \) and \( D \) matrices will result in solutions that may be completely different. Multilinearity requires that the retention time of each peak be completely reproducible from sample to sample. Given that compounds are expected to produce the same UV-VIS spectrum under the same conditions for all samples, often \( X \) undergoes chromatographic alignment in the first and second dimension or both prior to PARAFAC analysis.

### 3.3 Alignment Strategies

Retention shifting can occur in three different ways in LC×LC data: 1) shifting in the first dimension between samples (inter-sample shifting); 2) shifting in the second dimension between samples (inter-sample shifting); and 3) shifting between second dimension chromatograms within samples (intra-sample shifting). Three main strategies have been developed to correct
retention shifting in one dimension: 1) dynamic time warping (DTW), 2) correlation optimized warping (COW), and 3) an iteratively based linear shift.

DTW corrects retention time shifting by calculating the minimum distance between two orthogonal vectors for each point, as illustrated in Fig. 17 [69, 70]. Wang and Isenhour used DTW to align chromatograms from GC-FTIR and GC-MS instruments [69]. Wang and Isenhour noted that the two chromatograms needed to fulfill certain conditions to allow for accurate alignment by DTW: 1) the order of the peaks needs to be the same in both chromatograms (i.e., no peak swapping), 2) the magnitude of the retention time shifts need to be small, and 3) the peak heights and areas for each peak should be approximately the same in both samples. These conditions indicate that the two chromatograms need to be very similar in order to minimize the non-diagonal portions of the path, illustrated by the vertical line above c(k) in Fig. 14. In an effort to decrease the number of non-diagonal elements, Clifford et al. [71] modified DTW to use a variable penalty to minimize the number of non-diagonal elements; however, some non-diagonal alignment was found to still occur. The impact of the non-diagonal elements is that different samples result in different length vectors. The different length vectors results in the ability to only analyze two samples at a time.

Fig. 14: A schematic representation of the selected warping path by DTW between a reference chromatogram (Signal R) and a target chromatogram (Signal T). Reproduced from reference [69] with permission from American Chemical Society.
As an alternative to DTW, Nielsen et al. [72] developed the COW algorithm. COW uses piecewise linear alignment, along with linear interpolation warping, to align shifted peaks between samples. Like DTW, COW requires the use of a reference signal to act as a template to which the rest of the target chromatogram is to be aligned. The data is divided into a number of sections, also known as segments, and each segment is cross correlated to the corresponding reference signal segment. The sample segment is either stretched or shrunk, using linear interpolation, to maximize the correlation coefficient between the reference and sample signals. Nielsen et al. verified the use of the COW algorithm by successfully aligning peaks in both simulated and experimental data sets. While using COW to align two different data sets, van Nederkassel et al. noted that determining the optimal segment length and slack (how much warping was allowed) was a time consuming process [73]. To reduce the time required, Skov et al. developed an automated optimization algorithm for determining the optimal segment length and slack based on user selected limits [74]. Skov et al. noted that the appearance of tailing and/or fronting in chromatograms negatively impacts the ability of COW to successfully align the peaks. Skov et al. also noted that if the peak was represented by a low number of data points (7 to 10 data points) then the COW algorithm could result in significant changes to the peak area. This finding has implications for the application of COW to the sampled first dimension given the suggested M_R of 2 to 5.

An alternative means of aligning chromatograms that does not depend on warping, the use of iterative linear shifts based on the selection of a maximum correlation coefficient have been explored for aligning a single dimension. Krebs et al. [75] used landmark selection, the selection of marker compounds in the reference chromatogram, to align GC–MS data and accounted for inconsistent peak shifts with a cubic spline function. The cubic spline function
was used to interpolate the peaks to allow for peak alignment that would not be possible using the original time points. Johnson et al. [76] also used peak matching as means of aligning peaks in a series of GC chromatograms. The alignment was conducted using a linear shift in the time axis followed by a calculation of the correlation coefficient between the sample signal and the reference signal. The peaks were considered to be aligned when the correlation coefficient reached its maximum value. This technique assumes that the closest peak to the peak of interest is the correct peak, which may not be the case in a complex sample, and relies upon a window established around the peak of interest so that all peaks within that window are aligned to the reference peak.

To account for the possibility of peaks having the same retention time but different spectra, Bortolato et al. [77] used PARAFAC to extract the pure components from a three-way data set (time, spectral, and sample) in order to align the chromatographic dimension in the presence of an interferent. They utilized Andersson and Bro’s N-way toolbox [78] to conduct the PARAFAC analysis on a reference signal and a sample signal. The chromatograms were aligned by iteratively shifting the chromatograms, analyzing the new matrix using PARAFAC, and accepting the results when the error term of the PARAFAC equation reaches a minimum. The missing values generated by the shift were replaced by a column of zeros in both the time and spectral dimensions. The procedure was successfully tested on both simulated and experimental data and was found to correctly align both sets of data. Tistaert and van der Heyden used COW in conjunction with multivariate curve resolution by ALS to align resolved chromatograms while accounting for spectral differences in peaks [79]. This approach is similar to the approach taken in Chapter 6.
To account for the two chromatographic dimensions present in GC×GC data, several of the one dimensional methods have been modified in order to align the resulting two dimensional chromatograms. Zhang et al. [80] modified the COW algorithm, developed by Nielsen et al. [72], to allow a second chromatographic dimension to be aligned simultaneously with the first chromatographic dimension. This 2D COW program was applied to GC×GC/TOF-MS data using the total ion counts (TIC) chromatograms. The 2D COW algorithm is applied to small sections of the 2D chromatogram in order to reduce the analysis time and improve the alignment.

Fraga et al. [81] aligned a series of GC×GC chromatograms using a stepwise linear alignment along both chromatographic dimensions. This alignment strategy is a modification of a one dimensional procedure developed by Prazen et al. [82]. Briefly, the number of components is determined in the sample by using singular value decomposition (SVD). Then the singular values of the remaining components, i.e. noise, are summed up and divided by the sum of all of the singular values. This procedure is repeated as the chromatogram is incrementally aligned until a minimum value is obtained for the variance of the noise.

Pierce et al. [83] modified a one dimensional alignment technique they had previously developed [84] in order to align GC×GC chromatograms of gasoline and diesel samples. The method relies upon dividing the two dimensional chromatogram into sections and each section is shifted between samples followed by a check on the correlation coefficient. The data section is first linearly interpolated along the first chromatographic dimension to provide the necessary amount of data points required for the algorithm to work. The shift that produces the highest correlation coefficient is kept as the alignment for that particular section of the chromatogram.
### 3.4 Alternatives to PARAFAC

As an alternative to directly aligning chromatograms prior to PARAFAC analysis, Kiers et al. [4] developed the PARAFAC2 model to deal with retention shifting, by relaxing the constraints in the time dimension. In order to understand the PARAFAC2 model, it will be necessary to present another version of the three-way PARAFAC model for a single LC-DAD sample (A: time, B: UV-Vis spectrum, C: sample), Eq. 30,

\[ X_k = AD_k B^T + E_k \]  

(30)

where \( X_k \) is a \( ILxK \) two-way array of the \( k^{th} \) sample of \( X \), \( A \) is a \( IxN \) two-way array, \( B \) is a \( JxN \) two-way array, \( D_k \) is a \( N xN \) two-way array which contains the weights for the \( k^{th} \) sample of \( X \) along the diagonal, and \( E_k \) is a \( IxJ \) error two-way array for the \( k^{th} \) sample. Eq. 30 is modified into the PARAFAC2 model by assuming that \( A_{k_1}^T A_{k_2} = A_{k_2}^T A_{k_2} \) (where \( k_1, k_2 = 1 \) to \( K \)) for all pairs of samples resulting in Eq. 31,

\[ X_k = A_k D_k B^T + E_k = P_k H D_k B^T + E_k \]  

(31)

where \( P_k \) is an \( IxN \) two-way array and \( H \) is an \( N xN \) two-way array. The addition of the \( A_{k_1}^T A_{k_2} = A_{k_2}^T A_{k_2} \) constraint imposes on the model the requirement that \( H \) remains the same for all samples. This allows PARAFAC2 to function in the same manner as PARAFAC for three-way data sets without the need for pre-alignment, as long as shifting occurs in only one dimension.

Kiers et al. extended the concept of the PARAFAC2 model to a fourth dimension with Eq. 32.

\[ X_k = P_k H D_k (B \odot A)^T + E_k \]  

(32)

It should be noted that Eq. 32 is very similar to the four-way PARAFAC model shown in Eq. 29 and simply includes the updated shifting dimension constraint. While PARAFAC2 can be extended to four-way data, PARAFAC2 was designed to handle shifts in only one dimension and an algorithm currently does not exist for handling shifts in two dimensions.
In order to test the ability of PARAFAC and PARAFAC2 to handle retention time shifts, Bro, Andersson, and Kiers [85] compared the ability of PARAFAC and PARAFAC2 to analyze 1D-LC fluorescence data (with both excitation and emission spectra) and simulated data without aligning the chromatograms first. PARAFAC2 was found to produce results closer to the actual values than PARAFAC. However, due to the inability to place restrictions on the shifting dimension in PARAFAC2, PARAFAC2 produced profiles that were not as smooth as those produced by PARAFAC. In addition, they noted that the PARAFAC2 algorithm requires additional dimensions, beyond the third dimension, to be unconstrained (although this may no longer be true given the newest PLS toolbox from Eigenvector Research). Later, Skov et al. [86] used PARAFAC2 and shift correction followed by PARAFAC to analyze GCxGC-TOFMS chromatograms of a single peak of bromobenzene. The shift correction relied on iteratively moving the chromatogram and then calculating the correlation coefficient. The maximum correlation coefficient obtained by the shift was used as the alignment. Skov et al. determined that the shift correction followed by PARAFAC allowed for peak detection at a lower signal-to-noise ratios and lower concentrations compared to PARAFAC2.

A second alternative to pre-alignment, shifted factor analysis (SFA), was developed by Harshman et al. [87-89]. SFA was originally developed based on a bilinear model, X with dimensions of IxJ for a single LCxLC-DAD sample at a particular wavelength, with a shifting parameter applied to one of the dimensions as defined by Eq. 33,

\[ x_j = \delta_{t_j}(A)b_j \]  

where \( x_j \) is the \( j^{th} \) column of \( X \), \( \delta \) is the shifting operator, \( t_j \) is the \( j^{th} \) row of \( T \) (a JxN two-way array that holds the degree of shifting), \( A \) is an IxN two-way array being shifted, and \( b_j \) contains the \( j^{th} \) row of \( B \) (a JxN two-way array) in its diagonal. In this example, the shifting function is
being performed to correct for intra-sample shifting. The SFA model can be extended to four-way data by Eq. 33,

\[ \mathbf{X}_{jl} = \delta_j(\mathbf{A}) \mathbf{b}_j \mathbf{d}_j \mathbf{C}^* + \mathbf{E}_{jl} \]  

(34)

where <> indicates that the \( j^{th} \) or \( l^{th} \) row of \( \mathbf{B} \) or \( \mathbf{D} \) occupies the diagonal of a \( N \times N \) two-way array. SFA tracks the degree of shifting through an iterative process. The <> nomenclature is identical to the \( \mathbf{D}_k \) nomenclature used in PARAFAC2 in Eq. 31. Due to the shifting, SFA required the development of a quasi-ALS algorithm [88]. The unshifted dimension (\( \mathbf{B} \) in Eq. 33) is determined by iteratively shifting each component within \( \mathbf{A} \) until an optimal shift is reached. \( \mathbf{A} \) is then determined component by component after the previously determined shifts are removed. This process continues until a termination condition is reached. In the case of four-way data, two additional steps would be present regarding the \( \mathbf{C} \) and \( \mathbf{D} \) matrices. The shifting for the \( \mathbf{A} \) matrix is determined component by component by an exhaustive line search that constantly updates the degree of shifting based decreasing calculated residuals. Obviously, as the amount of data becomes larger, the amount of time required for each calculation increases greatly adding to the time required for each iteration [89].

In order to reduce the amount of time required for each iteration, Mørup et al. introduced the concept of calculating the degree of shifting using the Fourier domain [5]. While Mørup et al. developed the shifted CANDECOMP/PARAFAC (SCP) for neuroimaging data, Mørup et al. suggested that SCP also could be applied to chromatographic data. Mørup et al. defined the SCP model, where \( \mathbf{B} \) is the two-way array allowed to shift, by Eq. 35,

\[ x_{i,j,l} = \sum_{n=1}^{N} a_{i,n} b_{j-l+n} c_{i,n} + e_{i,j,l} \]  

(36)

where \( \tau_{k,n} \) is the degree of shifting for each component between each sample. Mørup et al. determined a closed form equation for \( \tau_{k,n} \) and included it in Eq. 35 to give Eq. 36,
where \( \tilde{b}_{j,n} \) is the element obtained by Fourier transformation of the \( B \) matrix. Mörup et al. tested SCP against PARAFAC for electroencephalography (EEG) data and determined that PARAFAC resulted in degenerate (multiple components with the same profile) profiles. In addition, although an equation was not been expressly derived during their paper, Mörup et al. developed a MATLAB function capable of handling four-way data. However, like PARAFAC2, SFA (and by extension SCP) was designed to only handle shifts in one dimension. However, unlike PARAFAC2, Harshman et al. theorized a possible model for SFA that accounted for shifts in two dimensions but did not devise an actual algorithm \(^{87}\).

### 3.5 Determination of the Number of Components

As was mentioned in section 3.2, PARAFAC requires the correct number of components to be selected to ensure that the correct result is obtained. Typically, singular value decomposition (SVD) is used as an aid to determine the number of components. For a single LC×LC-DAD sample \( X_{ijL} \), SVD \(^{90}\) is defined according to Eq. 37,

\[
X = UWV^T
\]  

where \( U \) is a \( IJ \times W \) (where \( W \) is the number of rows or columns, whichever is smaller, of \( X \)) two-way array containing the abstract \( 1 \)D and \( 2 \)D chromatograms, \( W \) is a \( W \times W \) two-way array containing the singular values along the diagonal, and \( V \) is a \( L \times W \) two-way array containing the abstract spectra. The majority of the manual and automated methods for determining the correct number of components rely on the use of the singular values present along the diagonal of \( W \). The manual methods involve looking at a plot of the singular values, illustrated in Fig. 15, and
arbitrarily (possibly including some experimental knowledge such as how many peaks are present) deciding how many components are required to obtain the desired result.

![Scree Plot](image)

**Fig. 15:** A scree plot (from the reshaped four-way phenytoin data set discussed in Chapters 6 and 7) is generated by plotting the log of the singular value by the component number. The bend, indicated by A, is the normal criteria used to determine the number of components [91]. The gap, indicated by B, is the previous method used by Porter *et al.* [92] and Zhu *et al.* [93] for determining the number of components. The gap, indicated by C, is the method used by Bailey *et al.* [3] and is used by the semi-automated alignment method (SAAM). Reproduced from reference [94] with permission from Elsevier.

Unlike manual selection, automated techniques are based mostly on four approaches: 1) singular value ratios (or something very similar) [95-108]; 2) frequency analysis of the abstract chromatograms [109]; 3) cross validation [110-112]; and 4) morphological approaches [113, 114]. For the purposes of this discussion, only those studies which utilized automated methods will be discussed through several comparison studies.

Meloun *et al.* compared thirteen different singular value ratio based methods using simulated and experimental spectra data to determine the accuracy of the methods with respect to the selection of the correct number of components [102]. Meloun *et al.* concluded that for experimental data the RESO (Ratio of Eigenvalues calculated by Smooth principal component
analysis (PCA) and those calculated by Ordinary PCA test [100] and the factor indicator (IND) test [95, 115] should be used.

Chen et al. developed RESO, a technique very similar to SVD [100]. Chen et al. specifically note that while RESO works better than other index methods based on their results, RESO may not be the best choice for all data. This statement in their conclusion implies that a universal method has not yet been determined. Also, Chen et al, state that since RESO was developed for smooth spectra (as in the case of UV-UVis spectra), RESO should not be applied to non-smooth spectra (such as MS spectra), limiting its effectiveness.

Malinowski developed the IND test, based on the eigenvalues obtained from SVD, to automatically calculate the required number of components for factor analysis (FA) [95]. The IND test is designed to select the correct number of components when the result from the IND calculation reaches a minimum as a function of the number of components. Malinowski determined the following observations when using IND: 1) the error in the data needs to be random and fairly homogeneous across the entire data set for a minimum to be reached; 2) depending on the amount of excessive error in the data, additional components may be necessary. Elbergali et al. also compared Malinowski’s IND, Malinowski’s imbedded error (IE) [95, 115] test in addition to Malinowski’s F-test, among others using experimental data [101]. Elbergali et al. concluded that the IE and the IND tests gave the best result with the F-test close behind (although the F-test required the user to pick a significance level). Wasim and Brereton also compared several methods in their effectiveness to determine the correct number of components for LC-NMR data [104]. Wasim and Brereton were interested in which methods could be fully automated (an important requirement to minimize the necessity of user interaction) while ensuring an accurate component determination. Wasim and Brereton decided
that Malinowski’s F-test [97] and the ratio of derivatives of error indicator function (ROD-IND) test were able to be fully automated. However, Wasim and Brereton determined that these tests were highly dependent on the number of artifacts present in the data. As the number of artifacts was increased, the likelihood that the correct number of components is selected decreases. However, it should be noted that although the IND and IE tests devised by Malinowski were shown to correctly predict the number of components, Malinowski has continued to design new tests [103, 107, 108]. Since Malinowski continued to design new tests, it can be surmised that current tests may not be sufficient for all data sets. However, until an in depth study is conducted, we do not know how these tests will perform with LC×LC-DAD data.

An alternative technique is to use cross-validation. In cross-validation, a portion of the data is left out and a new model is generated from the remaining data. The portion of the data left out is then predicted from the new model. In the case of component determination, the purpose of cross validation is to determine the number of components without including noise components (this definition holds true for all methods not just cross validation). Bro et al. compared different types of cross-validation with an eye towards accuracy of component prediction as well as computational effort [112]. Bro et al. concluded that an eigenvector based cross validation performed the best when a small number of variables and samples being present in the data set. However, the typical LC×LC-DAD sample analyzed by PARAFAC consists of four hundred thousand data points with typically thirty to fifty samples. As such, attempts to use cross-validation techniques on multiple samples have resulted in memory problems in MATLAB. Within an individual sample, cross-validation has been found to over determine the correct number of components [94].
3.6 Initialization

Smilde, Bro, and Gealdi state in their book, “Multi-way Analysis: Applications in the Chemical Sciences”, that the proper initialization of multi-way methods results in a faster analysis and avoids the prospect of converging to a local minima [66]. In order to initialize a four-way PARAFAC analysis, three of the four matrices (for example B, C, and D) need to be determined prior to PARAFAC analysis. The one matrix not pre-determined (for example A) is calculated using the initialized matrices and then used to update the other three matrices. In his PARAFAC tutorial paper, Bro mentions several initialization strategies such as the use of randomized matrices as initial guesses or basing initial guesses from U and V obtained from SVD [67]. However, Bro noted that the more trilinear X is, the less likely a local minima is obtained as the result. Fraga and Corley used GRAM to initialize PARAFAC analysis of LC×LC-UV data and found that the use of GRAM improved the accuracy of the quantitation [116]. However, as was previously noted in Section 3.2, the use of GRAM requires that only two samples are being analyzed. Porter et al. [92] used a flexible version of multivariate curve resolution (MCR) by ALS developed by Bezemer and Rutan [117] to initialize PARAFAC. Porter et al. found that the use of MCR-ALS allowed for the selective application of constraints, discussed in section 3.7, and the imposition of quadrilinearity by PARAFAC. This same approach was used to initialize PARAFAC in Chapters 5-7. MCR-ALS is based on a bilinear model [118] defined by Eq. 38,

\[ X = RS^T + E \]  

(38)

where \( C \) is a \( I \times N \) matrix containing the resolved chromatograms and \( S \) is a \( J \times N \) matrix containing the resolved spectra. The ALS algorithm of MCR-ALS consists of the iterative implementation of Eqs. 39 and 40,
\[ S = R^\dagger X \quad (39) \]
\[ R = X(S^T)^\dagger \quad (40) \]

where \( \dagger \) is the Moore-Penrose-matrix, \textit{i.e.}, the pseudo inverse matrix, \((X^TX)^{-1}X^T\).

As with PARAFAC, MCR-ALS requires that either \( R \) or \( S \) be used as an initial guess. Porter \textit{et al.} used resolved spectra from a simple unconstrained PARAFAC analysis to initialize the MCR-ALS step [92]. Bailey and Rutan later refined this approach by using a faster method that focuses solely on the spectra of \( X \) [3]. Bailey and Rutan used a modified form of key set factor analysis (KSFA), iterative key set factor analysis (IKSFA), [115, 119] to select a number of spectra that were orthogonal to the rest of the spectra in \( X \).

The KSFA algorithm was modified to account for the background signature present in the raw data. IKSFA is designed to search through the two way raw data set and retrieve the rows that are most orthogonal to (different from) each other up to the number of components specified. IKSFA makes use of SVD to decompose the raw data into the corresponding \( U \), \( W \), and \( V \) matrices. \( U \) and \( W \) are multiplied together to form an abstract row-factor matrix, \( F \). In order to eliminate the impact of the changing magnitude of \( F \) from row to row on the selection procedure, the rows of \( F \) are normalized according to Eq.41,

\[ \tilde{F} = \frac{f_m}{\left( \sum_{m=1}^{M} f_{mm}^2 \right)^{1/2}} \quad (41) \]

where \( f_m \) is the \( m^{th} \) row of \( F \), \( M \) is the total number of rows and \( \tilde{F} \) is the resulting row normalized \( F \) matrix. The normalized row with the largest absolute value is selected as the first key row. The other rows in the normalized matrix are then compared to the first key row. A matrix (consisting of two rows by two columns), Eq. (42)
is generated and the row, $\tilde{r}_{key1,2}$, that provides the greatest determinant is selected as the next key row. A third row is then inserted into the matrix (forming a 3 row by 3 column matrix) and the maximum determinant is determined. This process continues until $N$ key rows have been identified. However, the key factors obtained by this process may not be the most ideal, therefore an iterative procedure is carried out. Using the key rows determined by IKSFA, each key row is replaced iteratively by each normalized row for all $M$ rows until the absolute value of the determinant obtained is greater than the previously determined determinant. This process continues until the determinant does not change between iterations. The resulting key rows can be selected from the raw data as the most representative portions of the raw data set corresponding to the number of components chosen.

3.7 Constraints

Constraints are commonly used to help multi-way methods reach convergence faster and to ensure that the profiles produced are chemically relevant. The constraints used in Chapters 5-7 were non-negativity, unimodality, and selectivity [120-122]. The non-negativity constraint ensures that the profiles obtained during multi-way analysis contain values that are only greater than zero for the constrained components. Non-negativity is implemented in Chapters 5-7 by taking the average of each data point and the corresponding absolute value. The value of $1\times10^{-30}$ is then added to the result. Unimodality, which is typically used only in the chromatographic dimension, ensures that only one maximum is present in the constrained resolved profiles. Unimodality is implemented in Chapters 5-7 by halving the smaller peak during each iteration.
In essence, the smaller peak is smoothed away. Selectivity restricts either the chromatographic or spectral dimensions by ensuring that the component cannot be located within certain regions of the resolved components. This is accomplished by setting the intensity of the component to zero within those regions. Instead of applying the constraints in the statistically correct manner [121], the constraints implemented in Chapters 5-7 were applied after each ALS iteration to reduce the analysis time as described by Bezemer and Rutan [117].
CHAPTER 4: Impact of Background Correction on Peak Detection, Fractional Coverage, and Integration

The goal of this chapter is to compare the effectiveness of several background correction strategies, specifically with respect to the accuracy and precision of peak detection, calculation of fraction coverage, and peak quantification. A new technique for bilinear background correction for the three-dimensional data sets produced by a LC×LC-DAD system, singular value decomposition-based background correction (SVD-BC), is also proposed. This technique is similar to the asymmetric weighted least squares (AWLS) approach developed by Boelens et al. [123]. The orthogonal background correction (OBC) technique developed by Filgueira et al. [46] will also be investigated to allow for a comparison between spectral and chromatographic background correction techniques. In addition, a simple background subtraction technique based on a blank measurement is also investigated. Upon removal of the background signal, two automated peak detection methods, previously described in Chapter 3, section 3.1, (Stevenson et al. [51] and Reichenbach et al. [52, 53]) were used to determine whether or not peak detectability was compromised or improved after application of the background removal algorithm. The peaks were also detected manually to serve as a reference method. The background correction techniques were evaluated based on their ability to remove the injection and re-equilibration ridges while preserving the manual peak count. In addition, the impact of background correction on the calculated fractional coverage using the minimum convex hull [14]
and quantification of selected standard replicate peaks was examined. Portions of this chapter are reproduced from reference [124] with permission from Elsevier.

4.1. Introduction

With recent advances in comprehensive two-dimensional separations, as well as the inherent complexity (i.e., multidimensionality) of the data, comes the need for software and algorithms with the ability to characterize the separation system and generate reproducible quantitative results. System performance is typically measured by evaluating the effective utilization of the separation space [13, 125], i.e., fractional coverage. A comparison of peak counts for different separation conditions (i.e., different stationary phase and mobile phase combinations) also yields performance information about the different systems. For complex biological samples, this comparison can be a daunting task, due to the high degree of peak overlap in both chromatographic dimensions. Furthermore, as was explained in Chapter 2, section 2.4, to preserve the first dimension’s chromatographic resolution, each peak appearing in first dimension must be sampled several times thereby resulting in the same species appearing in several successive second dimension chromatograms [18]. Therefore, as was explained in Chapter 3, section 3.1, to accurately count the number of peaks present in the final 2D chromatogram, the peaks corresponding to the same species need to be properly identified and merged to yield a single 2D peak.

Subsequent to peak detection, peak quantification is carried out. One of the most common methods for quantifying peaks is the simple summation of the areas of the consecutive 2D peaks belonging to a single analyte [19, 126, 127]. This is the approach taken by the derivative-based detection method. The derivative-based detection method relies on the use of
derivatives to determine the start and stop of each 2D peak. After the 2D peaks are merged into 2D peaks, the areas of each 2D peak can be summed to generate the 2D peak area. While not explicitly implemented in the work of Stevenson et al. [51], this is obviously possible once the peak boundaries are determined, provided the background contributions to the signal are accounted for appropriately. As long as the baseline is linear, this is straightforward.

The alternative approach being investigated is the use of the drain algorithm to quantify the 2D peak. The drain algorithm assigns pixels adjacent to a maximum to the same 2D peak. In the drain algorithm [52, 53], the intensities of the pixels assigned to each 2D peak are summed to quantify the peak in question. However, an issue arises with the drain algorithm that can affect quantification. As Vivó-Truyols and Janssen [56] and Bailey et al. [1] point out, the watershed algorithm allows for non-continuity across a 2D chromatogram, illustrated by Fig. 11 in Chapter 3, from reference [56]. The non-continuity is a result of the watershed algorithm selecting non-adjacent parts of a 2D chromatogram on the edge of a merged 2D peak. While this non-continuity does not affect peak detection, non-continuity may impact the quantification of the merged 2D peaks.

The performance of both peak detection and peak quantification algorithms can be severely compromised by the presence of irregular background contributions. Thus, a critical step in developing reliable peak detection and quantification algorithms is often the effective removal of background signals. Background contributions are a more serious problem with ultra-fast LC×LC-DAD than with GC×GC-TOF/MS due to large baseline signals generated in optical detectors by the very fast and large changes in the refractive index of the effluent during gradient elution LC×LC [128]. There are two main features of the background that must be addressed when considering ultra-fast LC×LC chromatography. First, there is a gradually
increasing ridge that appears at the beginning of the 2D chromatograms over the course of the 1D gradient; this is caused by the difference in the composition of the 1D effluent and the initial composition of the 2D gradient. We call this the injection ridge (see Fig. 16).

This ridge increases in size as the 1D effluent composition becomes increasingly different from the initial composition of the 2D gradient. Second, there is another ridge of constant height that appears at the end of the 2D gradient; this is due to the rapid change in the 2D eluent composition back to its initial value. We call this the re-equilibration ridge (see Fig. 16). These features appear at all wavelengths and are due to changes in the refractive index in the detector cell which form a dynamic lens in the detector cell [129, 130], leading to an increase in the apparent absorbance. The magnitude of these features depends on the gradient time in the second dimension [131]; at the higher flow rates used in ultra-fast LC×LC-DAD, these features are greatly increased.

Fig. 16. LC×LC-DAD chromatogram of a blank sample at 220 nm.
Several approaches for background removal for comprehensive two-dimensional chromatograms have been described in the literature. Reichenbach et al. [132] performed a background removal on digital images of GC×GC-FID data. This was done by estimating the background level for each second dimension chromatogram across the chromatographic image based on structural and statistical properties of the GC×GC-FID data. Subsequent subtraction of the background level left a chromatogram in which the peaks were above the non-zero mean background [132]. Zhang et al. [133] suggested a chemometric technique for subtracting the background drift from a trilinear data set. However, this technique assumes that the data are completely trilinear, which is not necessarily the case for LC×LC-DAD data, due to shifting between consecutive second dimension chromatograms. In addition, due to the typically large number of compounds found in complex ultra-fast LC×LC-DAD samples compared to the experimental data used by Zhang et al., the determination of the correct number of components required for trilinear decomposition of the sample is difficult. Porter et al. [92] used a similar chemometric background correction approach based on a combination of MCR-ALS and PARAFAC, but this approach could only be used on small sections of the chromatogram. Therefore, alternative global background correction techniques (based on either analysis in the spectral or chromatographic dimension) will be investigated and compared based on their ability to remove the artifact ridges and their ability to not compromise peak detection and quantification.
4.2. Experimental

All calculations and data analysis were carried out using Matlab version 7.12.0.635 (R2011a, Mathworks, Inc., Natick, MA) on a Lenovo Win 7 PC laptop with an Intel Core i5-2410 M @ 2.30 GHz and 6.00 GB of RAM.

4.2.1 Chemicals

Chromatographic grade water was from Sigma-Aldrich (St. Louis, MO), and acetonitrile was obtained from J.T Baker (Phillipsburg, NJ, USA). Reagent grade perchloric acid was purchased from Mallinkrodt Baker (Paris, Kentucky, USA). All materials were used as received. All mobile phases were prepared gravimetrically (± 0.01 g) and used without any further filtration.

4.2.2 Samples and LC×LC-DAD instrumentation

The data consisted of two sets of samples: five replicates of a standard mixture containing 22 analytes (thiourea, 5-hydroxy-L-tryptophan, indole-3-acetyl-L-aspartic acid, indole-3-acetyl-L-glutamic acid, tryptophan, anthranilic acid, indole-3-acetyl-L-glycine, 5-hydroxy-tryptamine, indole-3-acetyl-ε-L-lycine, indole-3-acetyl-β-D-glucose, indole-3-acetamide, indole-3-carboxylic acid, indole-3-acetyl-L-isoleucine, indole-3-propionic acid, indole-3-ethanol, tryptamine, indole-3-butyric acid, indole-3-acetonitrile, indole-5-carbonitrile, 4-indolyl acetate, nitroethane, and nitropropane) and eleven replicates of an extraction of maize seeds [17]. Four replicates of a blank sample were also provided for use in generating the background correction models. The mobile phases used for both the first and second dimensions were a 10 mM perchloric acid solution in water for mobile phase A and neat acetonitrile for mobile phase B. A 4.6 mm × 100
mm Zorbax SB-C3 column packed with 3.5 µm particles was used in the first dimension. The first dimension gradient was 0 % B at 0 min, 56 % B at 24.5 min, 0 % B at 24.51 min, with a total analysis time of 32 min. The first dimension flow rate was 0.5 mL/min and the column was maintained at a temperature of 40 °C. A 2.1 mm × 33 mm column packed in-house with ZirChrom CARB 3.0 µm particles was used in the second dimension. The second dimension gradient was 0 % B at 0 min, 100 % B at 0.15 min, 0 % B at 0.16 min, with a total analysis time of 0.2 min. The second dimension flow rate was 3 mL/min and the column was maintained at a temperature of 110 °C. The instrument configuration was the same as previously reported by Filgueira et al. in the split mode [46].

4.2.3 Background correction techniques

The standard mixture and maize replicates were examined using the four background correction techniques and without background correction (treated as a control group). The first technique (Direct) subtracted the average blank from each of the “unknown” samples (standard mixture and maize). The average blank was generated by averaging the chromatograms of the four blank samples at each wavelength collected.

The second technique was the SVD-BC method based on a doubly truncated \( V \) matrix (\( \tilde{V} \)) from SVD, Eq. 37. The double truncation is accomplished by using only selected rows from the \( V \) matrix (440 nm to the highest wavelength collected) and only the first \( N \) columns. Before the procedure used to determine \( N \) will be described, the remainder of the SVD-BC technique will be explained. To estimate the contribution of the background to the sample chromatogram, a truncated \( \tilde{X}_{sample} \) matrix (consisting of \( IJK \) rows and the columns
corresponding from 440 nm to the highest wavelength collected) was fit to the doubly truncated \( \tilde{V} \) matrix, as follows:

\[
M = \tilde{X}_{\text{sample}} (\tilde{V}^T)^\dagger
\]

Eq. (43)

where \( M \) is a regression matrix of weighting coefficients for the contributions of the abstract spectra to the data set, and the superscript \( \dagger \) indicates the pseudoinverse of the \( \tilde{V}^T \) matrix. To generate the background corrected sample matrix, the following calculation is carried out:

\[
X_{\text{sample, bc}} = X_{\text{sample}} - MV^T
\]

Eq. (44)

where \( X_{\text{sample, bc}} \) indicates the background corrected chromatogram and \( V^T \) is the singly truncated (to contain the first \( N \) columns) \( V \) matrix from Eq. 37.

The number of components, \( N \) was determined by carrying out the background correction method described above for a blank chromatogram, with the number of components of interest set at one, and incremented until the standard deviation of the resulting background corrected chromatogram reached its minimum standard deviation. This procedure was carried out using the four blank samples. The rationalization for this approach was that a reduction in the background fluctuations would result in a decrease in the standard deviation of the background signal. \( N \) was determined to be three for three of the blanks; the third blank only required two components to reach its minimum standard deviation. An augmented blank matrix was then created by stacking the four blanks (so that the dimensions were \( 4I \times L \)), and SVD was then carried out on this blank. The resulting \( V^T \) matrix was then used as the basis for the background correction of the standard mixture and maize replicates.

The third technique, AWLS, was developed by Boelens et al. [123]. This technique uses the same \( V^T \) matrix as the proposed background correction method. However, instead of
truncating the $\tilde{V}^T$ matrix to contain only the longer wavelengths, the AWLS fitting method generates a chromatographic model of the background based on the entire wavelength range. Using an asymmetric system of weights (positive residuals were given a weight of 0.001 for this study), fitting is performed on the sample until only positive intensities remain.

The fourth technique, OBGC, was developed by Filgueira et al. [46]. Unlike the previous two techniques, this background correction technique relies on determining the baseline of each 1D chromatogram (the number of baselines determined is equal to the number of data points in the second dimension). Once a 1D baseline has been determined, the estimated 1D baseline is subtracted from its associated 1D chromatogram. Since this method is applied at a single wavelength, OBGC is performed on all wavelengths from 200 to 640 nm separately. The function used to estimate the 1D baseline was the median filter function used by Filgueira et al. [46]. The effect of the width of the median filter function was investigated by adjusting the width from 5 to 155 in increments of 10.

4.2.4 Peak detection methods

Prior to analysis, the standard mixture and maize chromatograms were cropped from 3 min to 28 min in the first dimension and 1.75 s to 10.9375 s in the second dimension. This cropping scheme was based on the retention time of the dead volume marker, the end of the second dimension gradient, and the end of the first dimension gradient. Two automated peak detection methods were used in this study. The first was the derivative based detection algorithm of Stevenson et al. (distributed by Hearn Scientific software, Melbourne, VIC, Australia) and implemented in Mathematica 8 (Wolfram Research, Champaign, IL) [51]. The cropped data sets were imported from Matlab into Mathematica by reshaping the two-dimensional chromatogram
at a single wavelength (220 nm) into a three-column matrix. The first column contained the first dimension time scale (in increments of 0.2 min), the second column contained the second dimension time scale (in increments of 0.0125 s), and the third column contained the absorbance intensities (in mAU).

Once the selected files were imported into Mathematica, the parameters for the analysis were adjusted for each individual chromatogram. Parameter names (Capitalized) are as specified in the Stevenson Mathematica code. The cut time (CutTime) was set to -1 in order for the entire dataset to be analyzed; the dead time (VoidTime) was set to 0.01; the threshold criteria was set to 2, indicating that the threshold value is equivalent to the value of the detection threshold (thrh2), which was manually determined for each replicate. Due to a shifting baseline as each replicate was injected throughout the series, the threshold was set separately for each individual standard replicate. Two detection thresholds were established for each standard replicate. The higher threshold was based on detecting 21 visually confirmed peaks present in all of the standard replicates. The lower threshold was based on detecting 23 visually confirmed peaks present in four of the five standard replicates. Two low intensity peaks were not visually detected in the first standard replicate. The thresholds used for the maize replicates without background correction and after background correction by the Direct, SVD-BC, and AWLS techniques were the mean high and low thresholds from the standard replicates, as shown in Table 1. The thresholds used for the maize replicates after background correction by the OBGC technique were the mean high and low thresholds from the standard replicates, as shown in Table 2.
Table 1: Thresholds (mAU) used for peak detection of the maize replicates

<table>
<thead>
<tr>
<th>Method</th>
<th>Derivative based detection</th>
<th>Drain algorithm detection</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>High</td>
<td>Low</td>
</tr>
<tr>
<td>No background</td>
<td>6.700</td>
<td>5.925</td>
</tr>
<tr>
<td>correction</td>
<td>3.540</td>
<td>2.800</td>
</tr>
<tr>
<td>Direct</td>
<td>5.180</td>
<td>2.550</td>
</tr>
<tr>
<td>SVD-BC</td>
<td>2.120</td>
<td>1.400</td>
</tr>
<tr>
<td>AWLS</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 2: Thresholds (mAU) used for peak detection of the maize replicates after OBGC background correction

| Filter window width | 5  | 15 | 25 | 35 | 45 | 55 | 65 | 75 | 85 | 95 | 105 | 115 | 125 | 135 | 145 | 155 |
|---------------------|----|----|----|----|----|----|----|----|----|----|-----|-----|-----|-----|-----|-----|-----|
|                     | 3.84| 4.58| 4.58| 4.58| 4.6| 4.6| 4.6| 4.6| 4.6| 4.6| 4.6| 4.6| 4.6| 4.6| 4.6| 4.6|
|                     | 2.15| 2.7 | 2.7 | 2.7 | 2.7 | 2.7 | 2.725| 2.725| 2.725| 2.725| 2.725| 2.725| 2.725| 2.725| 2.725| 2.725|

The peak maximum was determined from the first derivative x-intercept (PeakMethod was set to 1); the first derivative threshold (Thrfd) was set to 0.01, and the second derivative
threshold (Thrsd) was set to 0.01 also. The peak region overlap threshold (ThrOV) was set to 0.5; this deviates from the suggested setting (0.2) according to Peters et al. [49]. This change in overlap criteria was necessary to ensure that some of the lower intensity peaks in the standard replicates were not combined into a 2D peak. The peak maximum profile (PeakMaxProfile) was set to 1, which analyzes the peak maximum profile (plot of first dimension retention time as a function of detector response for all first dimension peaks) and separates peaks if there are valleys in the profile. The (PeakMaxInterpolation) was set to 1 to extrapolate the data points between the points in the (PeakMaxProfile) from the two dimensional data set and again looked for valleys between peaks in the first dimension.

The second automated peak detection algorithm used in this paper was developed by Reichenbach et al. [52] and is implemented in the LCImage software. The version of LCImage (LC Image, LLC, Lincoln, NE) used in this study was vR2.2. The cropping of the chromatograms and the selection of appropriate thresholds followed the same procedure as previously described for the derivative method. In addition to selecting appropriate thresholds the following parameters were established for peak detection; the sigma of the Gaussian smoothing used by LCImage was set to 0.1 in the first dimension and 1.0 in the second dimension; the minimum peak area was set to 15 (i.e., the number of pixels required for the detected “blob” to be recorded); the minimum peak volume was 0 (i.e., the sum of the detected pixels); and the minimum peak reference was set to ‘absolute’ (detection was based on set mAU thresholds). The same parameters were applied to both the standards and maize replicates.

In addition to counting the number of detected peaks after background correction, LCImage’s built-in baseline correction option was also utilized. The baseline correction option within the LCImage software works as follows [132]: First, each second dimension
chromatogram is divided into halves, called strides. Second, the baseline correction option determines the five pixels with the lowest intensity in each stride. Third, the mean and median of the two pixels adjacent to each of the chosen five pixels are calculated to generate the local baseline intensity. Fourth, the pixels that fall within the selected baseline intensity range are identified in each stride. Fifth, interpolation is performed between each pixel using a piecewise cubic spline and the resulting interpolated image is subtracted from the chromatogram.

As a comparison to the automated peak detection methods, manual peak counting was carried out on three of the maize sample replicates (the first, sixth, and eleventh respectively) without background correction being performed. The first and second dimension retention times of each second dimension peak was determined by visually inspecting each second dimension chromatogram. The visual inspection was carried out on 1.25 s increments of the 12 s second dimension chromatogram. This increment was chosen to reduce the influence of strongly absorbing peaks on the visual plot of the data. Unlike the automated peak detection methods, the starts and stops of each second dimension peak were not determined during manual peak counting. Instead, the prospective second dimension peaks were merged based on the unimodality criterion and a maximum allowable degree of retention time shift between consecutive second dimension maxima. Since the degree of peak overlap is not easily determined when using manual peak counting, a new constraint was used to merge prospective second dimension peaks. This constraint was based on limiting the maximum allowed retention time drift between prospective second dimension peaks to a user defined tolerance. To gauge the impact of the constraint on the merging process, three tolerance values (0.0625 s, 0.125 s, and 0.1875 s) were used.
4.2.5 Adjusting peak counts

After the total number of peaks was determined, the chromatogram was divided into regions, as shown in Fig. 17. The peaks that were close to the injection (Fig. 17, regions A and C) and re-equilibration (Fig. 17, region B) ridges were removed from the peak total. Removal of the peak counts in these regions resulted in a peak count that could reasonably be expected to represent true chromatographic peaks (Fig. 17, region D).

![Fig. 17: Schematic diagram of the system of peak classification used in this chapter. (A) Peaks detected in the region of the injection ridge. (B) Peaks detected in the region of the re-equilibration ridge. (C) Peaks generated by the background correction process. (D) Region where “real” peaks are presumed to exist.](image)

4.2.6 Fractional coverage calculations

The retention times determined by the derivative based detection method were used to calculate the fractional coverage of the maize samples before background correction and after background correction by the AWLS and OBGC techniques. The AWLS technique was performed using a weighting factor of 0.001 and the OBGC technique was performed using median filter windows of 5, 25, and 45. In addition, the fractional coverage was calculated for the OBGC technique with a median filter window of 45 after chromatographic alignment using the linear cross correlation and optimized COW methods. The linear cross correlation method
was performed using a maximum possible shift of 0.0625 s between successive 2D chromatograms and the optimized COW segment and slack parameters were determined using the algorithm developed by Skov et al. [74] for each 2D chromatogram. Due to time constraints, optimized COW was only performed on the first, sixth, and eleventh maize replicate with each alignment taking approximately 9 hours. The fractional coverage was calculated using a minimum convex hull with the area of each polygon normalized by dividing by 229.6875 which corresponds to the total area of the region selected for peak counting.

4.2.7 Quantitation methods

Six well resolved peaks were selected for quantification in the standard samples. A window surrounding each peak was chosen so that the peak of interest was fully present in the window (including any tailing); only one peak was present in each window. Quantification was accomplished by both manual integration assuming a linear baseline and by locating the respective “blob” in the “blob” detection table of LCImage. Stevenson et al. used a power function in their peak detection procedure [51]; therefore quantification was not performed using their technique. Manual integration was accomplished by plotting the sequence of second dimension chromatograms for the peak of interest, manually drawing the baseline of the peaks present in the sequential chromatograms, and summing up the area of the peaks after removal of the baseline.
4.3. Results and Discussion

4.3.1 Standard Replicate Peak Counts

The peak counts given by the two detection methods before background correction and after background correction are illustrated in Fig. 18. The impact of the background correction techniques on each peak count is described below. The different classes of peaks mentioned in Fig. 18 are defined in Fig. 17, where region A includes peaks detected in the injection ridge region, B includes peaks detected in the re-equilibration ridge region, C includes peaks generated during background correction (artifact peaks), and D includes the estimated number of true peaks in the sample (real and artifact peaks).

Fig. 18: The average number of peaks determined in each region shown in Fig. 17 for the standard samples. Various peak detection methods are denoted as: Derivative detection—high threshold (a); drain algorithm detection without baseline correction—high threshold (b); drain algorithm based detection with baseline correction—high threshold (c). (d-f) are analogous to (a-c), except using the low threshold. Panel (A) no background correction, Panels (B-D) after background correction using the Direct (B), SVD-BC (C), and AWLS (D) techniques respectively. The color patterns correspond to: red (peaks located in region A of Fig. 17), green (peaks located in region B of Fig. 17), purple (peaks located in region C of Fig. 17), cyan (peaks located in region D of Fig. 17). The horizontal lines in (A-D) are the true peak totals, 23 and 21, upon which the detection thresholds were based, for the high and low detection threshold respectively.
4.3.1.1 Analysis of Total Peak Counts

The average total number of peaks found for the five standard samples is shown in Fig. 3 as the cumulative height of all bars. With no background or baseline correction, the drain algorithm detected more peaks (Fig. 18A, bars c-d) than did the derivative approach (Fig. 18A, bars a-b). This is due to how the drain algorithm determines the presence of a peak. As shown in Fig. 9 of reference [54], this algorithm determines that a given pixel is a maximum when it is surrounded by enough smaller pixels to satisfy the minimum area requirement. After a pixel has been so identified, the volume of the pixels is determined by summing their intensities. If the calculated volume exceeds the minimum peak volume requirement, the maximum and the surrounding pixels are labeled as a peak. Clearly more peaks will be found without background correction than with background correction. However when the LCImage baseline correction is used, the volume of the detected peaks is reduced thereby preventing the drain algorithm from considering some smaller detected maxima as peaks. This results in a reduced total number of detected peaks (see Fig. 18A, bars e-f). As expected the number of peaks detected increased upon decreasing the detection threshold (Fig. 18A, bars b, d and f relative to bars a, c, and e), although the increase varied with detection method. Except for the case of the drain algorithm without baseline correction (Fig. 18B, bars c-d), the Direct technique produced peak totals very similar to the peak totals obtained when background correction was not performed. The primary difference between columns c-d in Figs. 18A and 18B is due to a large reduction in the B region peaks, which will be explained in more detail in section 3.1.3. At the higher detection threshold level the SVD-BC technique (see Fig. 18C) also found fewer peaks when baseline correction was used than when not used (see Fig. 18A). However, when the lower detection threshold was used (Fig. 18C, bars b and d), the number of peaks found did not differ much from the number found
without baseline correction. Use of baseline correction in LCImage reduced the peak totals primarily by decreasing the number of artifact peaks; a further explanation will be given in section 3.1.4. Finally, the AWLS technique gave peak totals approximately equal to the presumed true peak totals.

4.3.1.2 Analysis of Region A Peaks

The injection ridge (shown in Fig. 16 and Fig. 17, region A) caused by the solvent mismatch between the first and second dimensions was still largely present after background correction, (see Fig. 19A (box 1) and Fig. 19B), by the Direct and to a lesser extent the SVD-BC technique. Consistent peak counts were obtained before background correction and after background correction by the Direct and AWLS techniques (see red bars in Figs. 18A, 18B and 18D) regardless of detection threshold or method. The number of injection ridge peaks detected with the SVD-BC technique changed when the threshold was lowered, (see red bars in Fig. 18C). The SVD-BC technique was able to reduce the injection ridge so that the portion of the ridge occurring early in the first dimension was not detected at the higher threshold, Fig. 19B. However, because the SVD-BC technique was not able to completely remove the injection ridge, the lower threshold gave more peaks.
Fig. 19: Impact of the various background correction techniques on the background artifacts (data shown for the third standard replicate at 220 nm): (A) the injection ridge (box 1) and artifact peaks (box 2) from 1.75 to 3.5s in the second dimension and at 25.6 min in the first dimension. Curve a is the raw signal (before background correction) and curves b-d show the corrected signals upon use of the Direct, SVD-BC, and AWLS techniques respectively; (B) the injection ridge from 3 to 28 min in the first dimension at 1.9375 s in the second dimension. The letters a-d have the same meaning as in A; (C) the re-equilibration ridge from 9.5 to 10.9 s in the second dimension at 25.6 min in the first dimension. The letters a-d have the same meaning as in A; (D) the solvent peaks from 1.75 to 4. s at 3.8 min (immediately after dead marker) in the first dimension. The letters a-d have the same meaning as in A.

The use of the baseline correction option in LCImage did reduce the number of detected peaks by the drain algorithm. The effect of the baseline correction option on a selected second dimension blank chromatogram is shown in Fig. 20. For each of the techniques examined, the baseline correction option reduced the overall signal intensity requiring the high and low detection thresholds to be lowered, as shown in Table 1.
Fig. 20: Impact of the LC Image baseline correction options of the second dimension chromatogram from the third standard replicate at 220 nm and 25.6 min. The blue (solid line) chromatogram is before baseline correction and the red (dashed line) chromatogram is after baseline correction applied within LCImage. (A) No background correction (B-D); background removal by the Direct, SVD-BC, and AWLS techniques respectively.

4.3.1.3 Analysis of region B peaks

Fig. 19C compares the ability of the three background correction techniques to remove the re-equilibration ridge. Detection of peaks in this region differed depending on the detection technique used. No peaks were found in this region by the derivative approach. In contrast, the drain algorithm detected peaks in this region both before and after background correction. As explained in section 4.2.4, the drain algorithm determines that a peak is present when a sufficient number of pixels are located adjacent to a given maximum and the sum of the intensities of the pixels are greater than a given volume. Two factors then contribute to peak detection in this region. First, the minimum volume was set to zero for this study. While this ensured that all small peaks are detected, it also increases the chance that false peaks are counted. Second, since
the background correction techniques did not completely remove the re-equilibration ridge, an adequate number of pixels with intensities greater than the threshold remained to satisfy the area requirement. In addition, the use of the baseline correction option found in LCImage increased the number of peaks detected in this region. The increased detection of peaks after baseline correction in LCImage was due to an apparent “wraparound” effect (a 2D peak is located on both the top and bottom of the chromatogram simultaneously) in the detection software. It is therefore possible, that the detected peaks are due to the injection ridge and not the re-equilibration ridge. It can be seen that the SVD-BC and AWLS methods reduced the background in this region the most, but a significant amount of noise was introduced with the SVD-BC method.

4.3.1.4 Analysis of region C peaks

Fig. 19A (box 2) shows that artifact peaks can be introduced by the background correction procedure. As shown by the purple bars in Fig. 18, the largest number of artifact peaks was introduced by the SVD-BC technique, Fig. 18C; this is true for both detection methods. A possible explanation is that the chromatographic model generated during the SVD-BC technique did not adequately represent the background at the shorter wavelengths. This result indicates that that background models for the Direct and AWLS techniques are better than the SVD-BC techniques at shorter wavelengths. Curiously, the drain algorithm detected a smaller number of artifact peaks for the SVD-BC technique. The number of pixels surrounding the maxima is probably too small to satisfy the minimum area requirement. The drain algorithm did detect artifact peaks after application of the Direct technique but the use of the baseline correction option almost completely eliminated of the artifact peaks. The impact of the baseline
correction option on the chromatograms is shown in Fig. 20. The baseline correction option was able to generate an almost zero-baseline immediately after the injection ridge in region C. Also, the presence of the purple bar in Fig. 18A for columns d and e was due to a single standard replicate. The low detection threshold used for this replicate was the lowest. As such, additional peaks would have been detected in the other replicates had the low detection thresholds been lowered further. These peaks are not real however. Instead they are a result of the poor baseline correction immediately after the injection ridge, illustrated by the blue line in Fig. 19A box 2.

4.3.1.5 Analysis of region D peaks

In addition to real solute peaks, two types of artifact peaks were observed in the D region of Fig. 17, depending on the background correction and detection method used. Although these peaks are clearly artifacts, we have decided to include them in the estimated true peak totals since this region of the chromatogram is occupied by real peaks in the maize samples. The first type is due to solvent peaks, Fig. 19D. Since the solvent peaks are not completely reproducible between second dimension injections, the Direct and SVD-BC background correction techniques overcompensate in this region of the 2D chromatogram and generate the artifactual peaks. Likewise, the LCImage baseline correction option is unable to fully model the background in this region and this also leads to artifact peaks being generated. The second type of artifact peaks is due to the inability of the background correction techniques, primarily for SVD-BC but also AWLS to a lesser degree, to account for peak tailing. The most probable reason is the lack of incorporation of lower wavelengths into the background model.

The black horizontal lines in Fig. 18 represent the true estimated peak counts for the standards replicates. By looking at the cyan bars, we can compare the ability of the background
correction techniques to detect the actual peaks, once the artifact peaks are accounted for. Prior to background correction, the derivative based approach and LCImage before baseline correction detect approximately the correct number of peaks at the high threshold. However, when the detection threshold is lowered, the detection methods pick up the solvent artifact peaks mentioned previously. After baseline correction is applied, LCImage further inflates the number of detected artifact peaks in this region due to overcompensation. Of the three background correction techniques, only AWLS produces accurate peak counts at both detection thresholds. However, the LCImage baseline correction option once again introduces artifact peaks into the estimated true peak totals.

4.3.1.6 OBGC using a median filter

As indicated in Chapter 4, section 4.2.3, OBGC was performed using a median filter function available within MATLAB. The impact on varying the width of the filter window is shown in Fig. 21 (using the derivative based detection method) for the standards data set at the high detection thresholds (Fig. 21A) and low detection thresholds (Fig. 21B) listed in Table 2. Unlike the other background correction techniques, the samples after using the OBGC technique were only analyzed using the derivative-based method. This was due to a previous determination that the derivative based method detected a greater number of peaks for complex samples. At the high detection thresholds, the number of region D peaks found were equal to the true number of peaks regardless of the width of the filter window. With the exception of using a 5 point filter window width, the number of region A peaks was approximately the same regardless of the width of the filter window. A trend was observed (for window widths greater than 35) where the number of region A peaks decreased slightly as the filter
window width increased. A reasonable explanation for this downward trend is not that more of the injection ridge was being corrected. Rather, more of the injection ridge remained allowing for the derivative method to merge more of the injection ridge peaks together into single 2D peaks.

Fig. 21: Peak counts obtained from the five standards replicates using A) high peak detection thresholds and B) low detection thresholds as listed in Table 2 for the OBGC background correction technique. The colors correspond to the colored regions in Fig. 17 with red indicating peaks detected from the injection ridge and cyan indicating peaks detected due to “real” peaks. The horizontal black bars indicate the expected peak counts (21 for the high detection threshold and 23 for the low detection threshold).

A different story is found looking at the lower detection thresholds results. Unlike at the higher detection thresholds, the number of region D peaks changed as the width of the filter window increased. Starting at a width of 55, a small number of additional peaks were detected at 2D t_Rs slightly greater than the t_R of the dead volume peak. As was stated in section 4.3.1.5, while these additional peaks are obviously not real peaks, in a complex sample these peaks would automatically be included as “real” peaks due to their location in the chromatogram. A visual examination of the 2D chromatograms in this area does indicate some noise being present but the intensity of the noise does not reach the detection threshold. The squaring of the signal by Stevenson et al. may be the reason why some of the noise in this region is detected as peaks by the derivative method. As with the higher detection thresholds, the number of region A peaks
is also approximately the same if a filter window width other than 5 is used. The number of region A peaks at the low detection threshold did increase for the same reason as was stated for the SVD-BC technique compared to the high detection threshold. Based on the results shown in Fig. 21B, a filter window width of 45 will be considered the optimum filter window width if the median filter is used. This consideration is due to the lack of extra region D peaks and, with the exception of the 5 point filter window width, this filter window width produces the smallest number of region A peaks.

4.3.2 Maize samples

The true number of peaks present in the maize sample is not known. Therefore, visual inspection and peak counting was performed to establish a reference number of 2D peaks. First, a total of 441 ± 14 individual 2D peaks were observed (measured for the 1st, 6th and 11th maize replicates). Next, peaks in consecutive second dimension chromatograms were merged to form 2D peaks. The total number of peaks detected depended on the retention time shift tolerance, as shown in Table 3 and Fig. 22.

Table 3: Average “peak” manual counts for the maize replicates

<table>
<thead>
<tr>
<th>Method</th>
<th>0.0625 s retention time drift tolerance</th>
<th>0.1250 s retention time drift tolerance</th>
<th>0.1875 s retention time drift tolerance</th>
</tr>
</thead>
<tbody>
<tr>
<td>High threshold</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>No background correction</td>
<td>214±11</td>
<td>175±10</td>
<td>164±9</td>
</tr>
<tr>
<td>Direct</td>
<td>191±8</td>
<td>160±9</td>
<td>153±7</td>
</tr>
<tr>
<td>SVD-BC</td>
<td>102±9</td>
<td>92±6</td>
<td>89±7</td>
</tr>
<tr>
<td>AWLS</td>
<td>85±3</td>
<td>76±2</td>
<td>75±1</td>
</tr>
<tr>
<td>Low threshold</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>No background correction</td>
<td>256±11</td>
<td>206±11</td>
<td>193±8</td>
</tr>
<tr>
<td>Direct</td>
<td>226±10</td>
<td>184±10</td>
<td>172±7</td>
</tr>
<tr>
<td>SVD-BC</td>
<td>227±11</td>
<td>187±7</td>
<td>167±18</td>
</tr>
<tr>
<td>AWLS</td>
<td>116±6</td>
<td>103±4</td>
<td>102±4</td>
</tr>
</tbody>
</table>
Fig. 2: A plot of the number of merged 2D peaks as a function of the degree of spectral correlation and retention time drift tolerance allowed between successive 2D chromatograms at the low detection threshold without performing background correction. The blue diamonds are peaks merged using a 0.0625 s tolerance; the red squares are peaks merged using a 0.1250 s tolerance; and the green triangles are the peaks merged using a 0.1875 s tolerance.

None of the manual peak counts, Table 3, were the same as previously reported by Filgueira et al. [46], who report 128 peaks in this sample. The closest peak counts (prior to performing background correction) were from the largest allowed retention time drift. However, since this degree of retention time shifting was not observed in the standards replicates, the largest retention time drift tolerance will not be considered for comparison to the automated peak detection methods. Instead, the second column comprising the average retention time shift of the standards peaks with the addition of the upper bound of the standard deviation (0.125 s) will be considered as the true peak count for the maize sample. Also, as shown in Fig. 22, the use of spectral correlation to verify whether two 2D peaks should be merged does not have a significant effect unless correlation values greater than 0.9 are used. This indicates that the spectra for the majority of the compounds within the maize sample are very similar (either as a result of poor resolution or similar structures). However, given that the derivative based and drain algorithm
methods do not use the spectral correlation in merging 2D peaks, the spectral correlation requirement will not be used to merge peaks to generate the basis for the “true” number of 2D peaks in the maize sample.

Instead of manually counting peaks after each background correction technique is implemented, the retention times before background correction were used to estimate the number of peaks above a given threshold. The Direct technique was found to retain approximately the same number of peaks as if background correction was not performed. While the AWLS technique resulted in the lowest peak count, this can be explained by the signal loss at the wavelength used for detection, illustrated in Fig. 23A. Use of the AWLS technique resulted in a mean 25±2 % loss in peak intensity compared to the SVD-BC technique. The loss in intensity is due to the AWLS degrading the spectra of the peaks at lower wavelengths (200 to 300) nm although curiously AWLS does not remove the background from wavelengths higher than 300 nm, as shown in Fig. 23B. This intensity loss was found to be systematic across a given peak. The impact of the weighting factor on peak height is examined in Fig. 23C. As the weighting factor was decreased, the use of AWLS resulted in more of the peak height being retained after background correction. However, some loss in peak height was still observed even for the smallest weighting factors. In contrast, as the weighting factor was increased, the six peaks rapidly lost more of their height as AWLS overcompensated in removing the background at the lower wavelength. In comparison, as shown in Fig. 23D, the OBGC technique produced an equivalent peak height for all six peaks. As such the OBGC technique will be used for background correction prior to multivariate analysis in Chapter 7.
Fig. 23: The impact of the AWLS and OBGC techniques on the signal intensity of a selected peak. A) The AWLS technique was found to retain on average 75±2 % of the net peak height compared to the SVD-BC method. A second dimension chromatogram of a peak from the first standard replicate at 220 nm and 15.4 min is shown. The blue line was not background corrected. The red line was background corrected by the Direct method. The green line was background corrected by the SVD-BC method. The purple line was background corrected by the AWLS method; B) The spectra of the peak shown in A with the purple line representing after AWLS background correction and the red line after Direct background correction; C) The peak heights of six selected peaks from the standard replicates as a response to changing the weighting factor (p-value) for the AWLS background correction technique. The heights after AWLS were divided by the heights after Direct background correction to give the relative % remaining after background correction; D) The peak heights of the same six peaks chosen in C after background correction using OBGC by median filter for different filter window widths.

The reason why the SVD-BC technique resulted in a lower peak count is not immediately clear. A possible reason is that some of the peaks determined by manual detection were not real peaks but instead were due to fluctuations in the background. The peak counts at the lower threshold, however, were approximately the same as the Direct technique. The same trends were
also seen when the automated peak detection methods were used, Fig. 24. The patterns of the counts in the different regions are similar to those observed for the standard replicates, Fig. 18.

**Fig. 24:** The average number of peaks determined for the maize sample in each region shown in Fig. 19 after background correction for (A) no background correction, (B-D) background correction using the Direct, SVD-BC, and AWLS techniques respectively. The colors and letters (a-f) have the same meaning as previously described in Fig. 18. The shaded boxes in (A-D) represent the range of estimated true peak totals obtained by manual peak detection and using a 0.125 s retention time drift tolerance.

The cyan bars from Fig. 24 can be directly compared to the manual estimated peak totals. The Direct and AWLS techniques produced the closest estimated true peak totals to the peak counts found using a retention time drift tolerance of 0.125 s. However, as has been previously shown, AWLS produces a lower peak total compared to Direct. Also, after using the Direct technique, the derivative based detection method produced higher estimated true peak totals. The drain algorithm produced much lower estimated true peak totals, especially after baseline correction.
Fig. 25 illustrates the positions of the 2D peaks for the sixth maize replicate. The retention times of the 2D peaks are shown as circles in Figs. 25A and 25C. The boxes present in Fig. 25B are the regions designated to each 2D peak. Fig. 25A, generated using the code from Stevenson et al., is visually comparable to the manually detected peak positions, Fig. 25C. Some of the 2D peaks detected by the manual and derivative approaches were not detected by LCImage, Fig. 25B, after baseline correction. These missing peaks are possibly due to two limitations of the method. Since LCImage treats the chromatogram like an image, the drain algorithm is unable to detect peaks that are co-eluting which do not possess an apparent maximum on the shoulder of a larger peak when looking at the entire chromatogram. In order to detect co-eluting peaks, the user is required to manually select suspected overlapped peaks and apply a built-in deconvolution algorithm. This option was not tested during this study. Also, unlike the manual drift constraint and the derivative overlap criterion, a setting does not exist in LCImage to determine the degree of merging between consecutive second dimension peaks.
Fig. 25: Plots of the sixth maize replicate at 220 nm. Red circles or outlines indicate where a merged peak was detected at the higher threshold in Table 1. (A) Derivative approach using Stevenson’s method [51]. (B) Drain algorithm with baseline correction using LC Image. (C) Manual peak detection using Matlab with a 0.0625 s retention time drift allowed between each second dimension chromatogram.
4.3.3 Fractional coverage

The fractional coverage calculated for the maize samples as a function of which background correction was performed is shown in Table 4. The calculated fractional coverages are based on the region of the 2D separation space available to peaks, i.e., the region defined by the dead volume marker and the end of the 1D and 2D gradients. The fractional coverages after background correction were normalized to allow for a direct estimation of the impact of a particular background correction technique on the calculated fractional coverage.

Even though the AWLS technique was found to reduce peak intensity by approximately 25%, the AWLS technique was found to retain more of the non-background correction fractional coverage. This is shown in Figs. 26A and 27A. The peaks lost due to using the AWLS background correction technique were predominately in the center of the mass of peaks. The loss of the calculated fractional coverage was primarily to four peaks along the upper edge of the peak mass for the higher detection threshold (Fig. 26) and the same peaks were not detected in the lower threshold (Fig. 27). In addition, a peak located on the left edge of the peak mass was not detected at the lower detection threshold reducing the calculated fractional coverage even further. This was further compensated by the detection of additional peaks by AWLS, indicated by the arrow in Fig. 27A, that appears to have not been detected prior to background correction. However, this does not indicate that the same results would be obtained for other complex samples. It may have simply been happenstance that more of the edge 2D peaks were retained after background correction which most probably will not be the case for all samples.
Table 4: Calculate fractional coverage values from eleven maize replicates before and after background correction.

<table>
<thead>
<tr>
<th>Method</th>
<th>Weighting factor or filter window width</th>
<th>Mean</th>
<th>Relative mean compared to no background correction</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>High peak detection threshold</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>None</td>
<td>N/A</td>
<td>0.40±0.01</td>
<td>1.000</td>
</tr>
<tr>
<td>AWLS</td>
<td>0.001</td>
<td>0.344±0.009</td>
<td>0.858</td>
</tr>
<tr>
<td>OBGC</td>
<td>5</td>
<td>0.302±0.004</td>
<td>0.753</td>
</tr>
<tr>
<td>OBGC</td>
<td>25</td>
<td>0.301±0.005</td>
<td>0.750</td>
</tr>
<tr>
<td>OBGC(a)</td>
<td>45</td>
<td>0.299±0.005</td>
<td>0.744</td>
</tr>
<tr>
<td>OBGC(b)</td>
<td>45</td>
<td>0.297±0.005</td>
<td>0.739</td>
</tr>
<tr>
<td>OBGC(c)</td>
<td>45</td>
<td>0.297±0.004</td>
<td>0.753</td>
</tr>
<tr>
<td>Low peak detection threshold</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>None</td>
<td>N/A</td>
<td>0.43±0.03</td>
<td>1.000</td>
</tr>
<tr>
<td>AWLS</td>
<td>0.001</td>
<td>0.39±0.01</td>
<td>0.904</td>
</tr>
<tr>
<td>OBGC</td>
<td>5</td>
<td>0.383±0.004</td>
<td>0.894</td>
</tr>
<tr>
<td>OBGC</td>
<td>25</td>
<td>0.35±0.03</td>
<td>0.812</td>
</tr>
<tr>
<td>OBGC(a)</td>
<td>45</td>
<td>0.382±0.006</td>
<td>0.891</td>
</tr>
<tr>
<td>OBGC(b)</td>
<td>45</td>
<td>0.381±0.006</td>
<td>0.889</td>
</tr>
<tr>
<td>OBGC(c)</td>
<td>45</td>
<td>0.41±0.01</td>
<td>0.947</td>
</tr>
</tbody>
</table>

\(a\) without alignment
\(b\) after linear correlation alignment
\(c\) after optimized COW using only the first, sixth, and eleventh maize replicates.

In comparison, the use of OBGC with the median filtering function resulted in fractional coverages that were approximately equivalent regardless of the filter window width or whether the 2D chromatogram had been aligned prior to background correction. The one exception to this trend was the use of COW to align the 2D chromatogram followed by detection at the lower threshold. This exception is due to a single peak (indicated by the arrow in Fig. 27F). This peak is not present in the other samples because it was merged into adjacent 2D peaks. COW shifted the peak far enough that the overlap criterion (0.5 in this case) was not sufficient to indicate that this peak belonged in the same 2D peak it had previously included in. As with AWLS, OBGC by median filtering resulted in smaller peaks located in the center of the peak mass being lost while larger ones on the periphery were retained. However, unlike with AWLS, OBGC by
median filtering should retain more of the peaks on the edge of the peak mass than the interior. The rationale behind this statement is that the median filter will perform less overcorrection of the background on the edges of a large peak mass versus the center of the peak mass. This has been observed visually although not quantified. As such, if a more suitable means of performing OBGC can be found, it is entirely possible that the fractional coverage ratio of OBGC to non-background correction may equal one regardless of the detection threshold used.

**Fig. 26**: Retention times of peaks detected using the derivative-based method at a high threshold for the eleventh maize replicate at 220 nm. The blue diamonds (◊) are the retention times of peaks detected without using background correction. The red squares (□) are the retention times of peaks detected using the following background correction techniques A) AWLS using a weighting factor of 0.001; B) OBGC using a filter window width of 5; C) OBGC using a filter window width of 25; D) OBGC using a filter window width of 45; E) OBGC using a filter window width of 45 after linear-correlation alignment in the second dimension; F) OBGC using a filter window width of 45 after optimized-COW on the second dimension.
Fig. 27: Retention times of peaks detected using the derivative-based method at a low threshold for the eleventh maize replicate at 220 nm. The symbols and A-F plots are the same as described in Fig. 26. The arrow in A indicates the detection of false positives from the AWLS technique. The arrow in F indicates the incomplete merging of two previously merged 2D peaks due to warping by COW.

4.3.4 Quantitation

The % RSDs of six selected peaks from the standard replicates are shown in Table 5. A series of Levene’s tests to compare the variances [134] were carried out to determine if there is a statistical difference whether or not background correction was performed. No statistical differences between before and after background correction were found if manual integration was used as the quantification method. Likewise, statistical differences before and after background correction were not found if LCImage was used, regardless of whether the baseline
correction option was employed. However, statistical differences were found when the variances from LCImage before baseline correction were compared to the variances from manual integration and LCImage after baseline correction. The likely reason for this statistical difference is once again the lack of a non-zero baseline after background correction. Since LCImage treats the chromatogram as a series of pixels, the magnitude of background noise from each pixel is included with that of the peak upon conducting quantification. An additional comparison between the variances from manual integration and LCImage after baseline correction did not show a statistical difference. So, while the scale of the % RSD values are lower for the manual integration, a statistical improvement cannot be found if LCImage after baseline correction is used instead.

Table 5: Percent relative standard deviations of six fully resolved peaks present in the standard samples

<table>
<thead>
<tr>
<th>Method</th>
<th>Manual Integration</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Peak 1</td>
<td>Peak 2</td>
<td>Peak 3</td>
<td>Peak 4</td>
<td>Peak 5</td>
<td>Peak 6</td>
</tr>
<tr>
<td>No background correction</td>
<td>1.08</td>
<td>1.96</td>
<td>1.46</td>
<td>1.90</td>
<td>1.62</td>
<td>2.09</td>
</tr>
<tr>
<td>Direct</td>
<td>0.66</td>
<td>2.17</td>
<td>1.60</td>
<td>2.20</td>
<td>1.67</td>
<td>1.37</td>
</tr>
<tr>
<td>SVD-BC</td>
<td>0.92</td>
<td>2.37</td>
<td>1.55</td>
<td>2.96</td>
<td>2.94</td>
<td>2.85</td>
</tr>
<tr>
<td>AWLS</td>
<td>2.09</td>
<td>4.18</td>
<td>1.71</td>
<td>2.92</td>
<td>0.39</td>
<td>2.24</td>
</tr>
<tr>
<td>Drain algorithm without baseline correction</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>No background correction</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Direct</td>
<td>14.34</td>
<td>7.74</td>
<td>6.16</td>
<td>4.18</td>
<td>3.00</td>
<td>3.32</td>
</tr>
<tr>
<td>SVD-BC</td>
<td>14.89</td>
<td>22.46</td>
<td>10.70</td>
<td>11.59</td>
<td>4.80</td>
<td>6.71</td>
</tr>
<tr>
<td>AWLS</td>
<td>25.98</td>
<td>14.23</td>
<td>7.85</td>
<td>8.86</td>
<td>2.32</td>
<td>6.93</td>
</tr>
<tr>
<td>Drain algorithm with baseline correction</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>No background correction</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Direct</td>
<td>1.45</td>
<td>2.28</td>
<td>1.59</td>
<td>2.37</td>
<td>1.31</td>
<td>1.76</td>
</tr>
<tr>
<td>SVD-BC</td>
<td>1.38</td>
<td>2.12</td>
<td>1.33</td>
<td>1.62</td>
<td>1.43</td>
<td>1.20</td>
</tr>
<tr>
<td>AWLS</td>
<td>2.70</td>
<td>2.93</td>
<td>1.34</td>
<td>2.79</td>
<td>1.67</td>
<td>1.67</td>
</tr>
<tr>
<td></td>
<td>2.18</td>
<td>3.58</td>
<td>4.38</td>
<td>3.95</td>
<td>1.54</td>
<td>1.42</td>
</tr>
</tbody>
</table>
4.4. Conclusions

After background correction, none of the background correction techniques were able to provide a completely zero baseline. While the baseline was not zero, the AWLS technique did achieve a complete elimination of the injection and re-equilibration ridges. However, due to the loss of signal intensity, the user is warned that the use of standards to determine the detection threshold may not be possible with the AWLS technique. As a result of the loss of signal intensity, a large reduction in the estimated true peak totals was observed for the AWLS. This reduction may prove problematic when attempting to assess the quality of the separation as smaller outlying peaks may not be detected. In terms of quantification, the AWLS technique was found to produce the most consistent % RSD values regardless of the quantification approach used.

In comparison to AWLS, the Direct technique did not remove the injection and re-equilibration ridges. Unlike the SVD-BC technique, the Direct technique introduced a smaller number of artifact peaks which are easily accounted for based on their location in the chromatogram. The Direct technique also resulted in a larger peak count compared to the AWLS technique due to a lack of signal loss. This preservation of signal intensity allows for the detection of lower intensity peaks that may not be possible using the AWLS technique. The Direct technique can be improved if the baseline correction option found in LC Image is incorporated into the derivative based-detection method. This would allow for a further decrease in the threshold and possibly in the number of artifact peaks detected. However, the Direct technique is greatly dependent on the ability of the system to reproduce the baseline between samples.
The proposed SVD-BC technique is not recommended for background correction. While, SVD-BC did reduce or eliminate the background ridges, an unacceptable number of noise peaks were introduced with sufficient height to be detected at the lower detection threshold. OBGC was found to eliminate the re-equilibration ridge but not completely eliminate the injection ridge. The use of a simple linear shift followed by cross correlation to account for the oscillation in the 2D chromatograms caused by the shunt did reduce the number of region A peaks detected. However, optimized COW on each successive pair of 2D chromatograms resulted in a greater number of region A peaks detected versus the simple linear shift approach.

In terms of which detection method is preferable, the derivative-based method is recommended to allow for the detection of co-eluting compounds that are not detected by the LCImage software. A modification to the method is recommended to allow for the user to define the maximum degree of retention time drift between consecutive second dimension peaks. However, the LCImage software with the baseline correction option implemented for quantification due to the ease of use for well resolved peaks. If the peaks of interest have a resolution less than one than it is recommended to use one of the quantification methods examined in Chapter 7 instead.

The implementation of background correction techniques resulted in fractional coverages lower than was determined without performing background correction. The AWLS technique resulted in higher calculated fractional coverages compared to the OBGC technique for both the high and low detection threshold. However, this may have been purely coincidental. Due to the inability to predict the amount of signal loss for a given peak, AWLS may not present the same trend if a different sample is used. The OBGC technique, although it resulted in a lower calculated fractional coverage, is expected to possess a more consistent result regardless of the
sample. However, the median filter may not be the best function to calculate the $^1$D baseline for complex samples. Further research should be conducted to determine a more optimum function that does not result in the peak losses shown in this chapter.
CHAPTER 5: Reconstructing the First Dimension Peak Shape

The focus of this chapter is on utilizing common interpolation techniques to recreate the unsampled 1D peak shapes during chromatographic alignment of LC×LC-DAD data. Interpolation was used to generate a sufficient number of data points in the sampled first chromatographic dimension to allow for alignment of retention times from different samples. Five different interpolation methods, linear interpolation followed by cross correlation, piecewise cubic Hermite interpolating polynomial, cubic spline, Fourier zero-filling, and Gaussian fitting, were investigated. The fully aligned chromatograms, in both the first and second chromatographic dimensions, were analyzed by PARAFAC analysis to determine the relative area for each peak in each sample. Portions of this chapter are reproduced from reference [135] with permission from Elsevier.

5.1. Introduction

The quantification performed in Chapter 4 was for peaks that were well resolved, producing % RSD values typically between 1-2%. However, integration techniques rapidly lose accuracy and precision as peaks become poorly resolved, i.e., for each peak to be at least 95% pure, the R_s needs to be greater than or equal to 0.8 for peaks of equal height [136]. An alternative to simply integrating the peaks is to utilize multivariate techniques. Bailey and Rutan [3] used MCR-ALS to analyze five isolated peaks from six replicate samples of LC×LC-DAD standard chromatograms. The resulting resolved chromatograms were manually integrated to
determine the relative area of each of the peaks. While this approach resulted in relative areas with percent relative standard deviations (% RSD) ranging from 1.4 to 4.7%, the manual integration approach can become very labor intensive if a large number of peaks are present or if a large number of samples are analyzed. For chromatograms containing a larger number of peaks or larger sample sizes, a more automated quantification approach is desirable. Both GRAM and PARAFAC have been automated using the MATLAB programming environment to analyze GC×GC and LC×LC chromatograms [116, 137-140]. However, as was stated in Chapter 3, section 3.2, in order to correctly implement either GRAM or PARAFAC, the chromatographic dimensions need to be aligned so that the peaks occur at the same retention time for each sample.

While the second chromatographic dimension is easily aligned by a simple linear shift due to the high data collection frequency [81], the alignment of the first dimension is complicated by the sampling process that occurs between the two chromatographic columns. As noted by Fraga et al. [81], the 1D peaks typically possess a low data density (three to four data points per peak for LC×LC chromatograms), thereby requiring interpolation to insure that precise shifting of peaks between samples is possible. The predominant interpolation strategy found within the literature for aligning the first dimension of two dimensional chromatographic data consists of performing a linear interpolation on the existing data points in order to ensure that a sufficient number of points exist for alignment [83, 116]. Typically a localized region of the two dimensional chromatogram containing the peaks of interest is selected for alignment and then interpolated. A reference chromatogram (typically a standard) is chosen as the basis to which the other chromatograms will be aligned. The second and first dimensions are then iteratively shifted until the criteria for selecting the optimal shift has been met. However, the use of linear interpolation in the first dimension may prove to be insufficient to correctly align the
peaks for two reasons. First, the peaks within a LC×LC chromatogram may experience non-linear shifts, resulting in different sampled 1D peak shapes, due to changes in pump performance or ambient temperature, column degradation, or evaporation of volatile compounds from the mobile phase [141]. Second, the \( M_R \) typically used for LC×LC experiments is much less than the \( M_R \) in GC×GC experiments [83, 142, 143]. An alternative approach that will be explored in this chapter is to use common interpolation techniques to recreate the original 1D peak shape and align the reconstructed 1D peak to a common retention time.

5.2 Sampling of first dimension

The position of the 1D peak within the 2D sampling window determines the pattern of the 1D sampled peak shape. An example of the sample pattern resulting from the sampling device is illustrated in Fig. 28. The dashed lines in Fig. 28A represent the individual 2D chromatograms rearranged into a two dimensional structure. The length of time between each valve switch determines the length of the 2D chromatograms and the number of 2D chromatograms corresponds to the number of times the 1D effluent is sampled. Once the area of each of the sequential 2D chromatograms is summed, the pattern illustrated in Fig. 28B is obtained. Each marker represents the total amount of signal (area) present in each 2D chromatogram. The alternating ▼ and ● markers represent the alternating 1D column effluent samples from the two sampling loops onto the 2D column. The shape of the peak pattern is determined by the sampling phase according to Eqn. 22 in Chapter 2, section 2.4. The meaning of these parameters is depicted graphically in Fig. 28C. If the retention time of the peak is in the center of the window then the peak has a sampling phase of \( \phi = 0 \) and is considered to be completely “in-phase”, Fig. 29A. If the retention time of the peak exists at the edge of the sampling window then the peak
has a sampling phase of $\phi = \pm \pi$ and is considered to be completely “out-of-phase”, Fig. 29D. If the retention time of the peak exists somewhere in between then the peak has a sampling phase between 0 and $\pi$, illustrated by Fig. 29B and C.

Fig. 28: (A) An illustration of an LC×LC chromatogram. The dashed lines indicate the time points where the sampling device injects the 1D eluent into the 2D column. (B) The 1D peak pattern from Fig. 28A generated by summing all values along the dashed lines. The alternating ▼ and ● markers represent either the use of two loops or two 2D columns in LC×LC. (C) Illustration of the parameters that determine the sampling phase, $\phi$. 

98
5.3. Experimental

All work was conducted on a Dell® Optiplex 755, Intel(R) Core™2 Duo CPU, E6550 @ 2.33 GHz, 3.23 GB of RAM within the confines of MATLAB® software 2009a (Mathworks, Inc. Natick, MA) version 7.8.0.347.

5.3.1 Data sets

Two different types of data, a simulated set and an experimental set, were examined to test the five different interpolation techniques. In order to ensure that the simulated data closely modeled an experimentally obtained data set, the simulated data set was created using experimentally comparable peak widths, retention time shifts, and background signals. The 1D chromatogram, with a time scale of 0.35 to 2.8 min sampled at 0.35 min intervals and a peak width defined by a $\sigma$ of 0.1335 min, was constructed using an in-house sampled Gaussian function created in MATLAB® to account for the 1D sampling, previously described by Thekkudan, Rutan, and Carr [19]. The 1D peak area column in Table 6 shows the areas used to
generate the sampled $^1$D peak for both the calibration and validation samples. A normal Gaussian distribution (with an area of one) was used for the $^2$D chromatogram, with a time scale of 0.025 to 3.375 s at 0.025 s intervals and a $\sigma$ of 0.2125 s. After the $^2$D and $^1$D pure component profiles were generated individually, the 2D pure component chromatograms were generated by taking the outer product of the $^1$D and $^2$D chromatographic vectors. These 2D chromatograms are then reshaped into a one dimensional structure (a vector), where the $^2$D chromatograms for each sample are appended sequentially, and the individual samples appended in turn. The spectral information is then incorporated by calculating the outer product of these chromatographic vectors with a pure component spectral vector (arbitrarily taken as the spectrum of 4-methylthioamphetamine [144]), measured at 4 nm intervals from 200 nm to 700 nm. The data structure is then reshaped into a four-way structure and then added onto a pre-existing, experimental background. This approach was conducted for thirty-five samples: fifteen calibration samples and twenty validation standard samples. The fifteen calibration samples consisted of a five point calibration curve with three replicates at each calibration level interspersed throughout the data set. Four validation samples were used, with five replicates per validation sample, to gauge the effectiveness of the five interpolation methods in aligning the data set.
Table 6: Peak areas used to generate the sampled $^1$D peak for the simulated data

<table>
<thead>
<tr>
<th>Calibration Level$^a$</th>
<th>$^1$D peak area</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>26.00</td>
</tr>
<tr>
<td>2</td>
<td>34.00</td>
</tr>
<tr>
<td>3</td>
<td>42.00</td>
</tr>
<tr>
<td>4</td>
<td>10.00</td>
</tr>
<tr>
<td>5</td>
<td>18.00</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Validation Level$^b$</th>
<th>$^1$D peak area</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>39.25</td>
</tr>
<tr>
<td>2</td>
<td>12.98</td>
</tr>
<tr>
<td>3</td>
<td>14.57</td>
</tr>
<tr>
<td>4</td>
<td>26.92</td>
</tr>
</tbody>
</table>

$^a$Each calibration level consisted of three replicates.

$^b$Each validation level consisted of five replicates.

The experimental data set was obtained from Dr. Carr’s research group at the University of Minnesota and has been previously analyzed by Bailey and Rutan [3]. This data set consisted of six replicate samples containing seven well resolved standard compounds. A localized region around five of the standard peaks was selected according to two conditions. First, the boundaries of each localized region were selected so that only a single standard peak was present. Second, each localized region was large enough to ensure that the standard peak of interest was fully present in all six replicate samples. This data set was interpolated and aligned in the first dimension and then aligned in the second dimension. PARAFAC was then used to determine the relative areas of five of the seven peaks. The two peaks not interpolated were affected by adjacent contaminant peaks and as the peak phase shifted, the contaminants coeluted with these peaks.

5.3.2 Alignment approach

The approach for the interpolation and alignment of each peak can be seen in Fig. 30. At first, a representative $^1$D chromatogram (denoted by the dashed line in Fig. 30A) is chosen at a
wavelength where the peak of interest strongly absorbs. The representative 1D chromatogram was chosen at 216 nm to ensure that a sufficient amount of signal for each peak was present in all samples, Fig. 30B. This representative 1D chromatogram is then interpolated to provide nine data points between each sampled point, and the maximum position of each peak present is determined. Once the interpolated position of each peak is determined, each 1D chromatogram across the second and spectral dimensions is interpolated and then shifted to the earliest retention time for that peak, Fig. 30C. The 1D chromatogram is then resampled, by taking every ninth data point starting from the first data point, in order to reduce the number of data points. This helps to ensure that the computer does not run out of memory while implementing the PARAFAC analysis. The 2D chromatograms are then aligned by determining the maximum position of each peak in the 2D chromatogram at a given 1D point, illustrated by the dashed line in Fig. 30D, and then shifting each 2D chromatogram to the earliest retention time observed for that peak.
Fig. 30: (A) A contour plot of a representative LC×LC chromatogram. The dashed line indicates the 2D time point chosen to generate the representative 1D chromatogram for interpolation purposes. (B) The representative 1D chromatogram generated in 30A. The solid and dashed lines represent two different samples of the same peak at two different retention times. (C) 1D chromatograms after the cubic spline interpolation technique has been applied to the sampled 1D chromatograms from 30B. The vertical dashed line indicates the point to which the dashed peak is being aligned. (D) The resulting resampled 1D peak after alignment and resampling of the interpolated 1D chromatogram due to computer memory issues. The vertical dashed line indicates the 1D time point chosen to determine the 2D time points for the 2D alignment.

5.3.3 Interpolation implementation

The varying requirements of each interpolation technique necessitated different approaches in applying the techniques to the raw data set. The Hermite polynomial and spline techniques were implemented using the pchip and spline functions available in MATLAB® [145], respectively. The inputs for these functions were the 1D chromatogram being interpolated and the new time scale (0.35 to 2.8 min at 0.035 min intervals for the simulated data) to which the 1D chromatograms were to be interpolated. Both the pchip and spline functions utilize a cubic polynomial to fit to the data. The only difference between how the two functions apply the
cubic polynomial is the manner in which the second derivative of the polynomial is used. In the spline function, the second derivative of the polynomial is continuous, thereby allowing the resulting interpolated peak to possess maxima and minima between the original sampled data points. In contrast, the second derivative of the polynomial used in the pchip function is not continuous. This lack of continuity in the second derivative ensures that the original shape of the peak is preserved with only a minimal degree of curvature existing between data points, resulting in an interpolated peak that retains the original maxima and minima of the sampled 1D peak.

The Fourier zero-filling and Gaussian fitting techniques were implemented using in-house MATLAB® functions. Prior to performing Fourier zero-filling, a time axis corresponding to the sampled 1D chromatogram and a time axis corresponding to the interpolated 1D chromatogram were generated depending on the size (number of data points) of the first dimension. In the case of the simulated data set, the size of the first dimension was even, necessitating truncation of the data in accordance to the requirement that the frequency domain contain an odd number of points [146]. This truncation resulted in the sampled time axis occurring from 0.35 to 2.45 min at 0.35 min intervals and the interpolated time axis occurring from 0.35 to 2.765 min at 0.035 min intervals. If the size of the first dimension was odd, then the sampled time axis was from 0.35 to 2.8 min at 0.35 min intervals and the interpolated time axis was from 0.35 to 3.115 min at 0.035 min intervals. The different interpolated time axis is necessary to account for MATLAB® treating the 1D data as cyclical when performing the inverse Fourier transform. After the creation of the sampled time axis and the interpolated time axis, the Fourier transform was applied to the 1D chromatogram to convert from the time domain into the frequency domain. A number of zeros equal to the difference between the original number of data points and the desired number of data points were inserted just after the median point in the
frequency domain. The inverse Fourier transform was then applied to the modified $^1$D frequency data to convert the $^1$D chromatogram back into the time domain, obtaining an interpolated $^1$D chromatogram. The original total intensity of the data is now distributed over a larger number of data points, resulting in the $^1$D chromatogram possessing reduced signal intensity compared to the original $^1$D chromatogram. To account for this change in signal intensity, the interpolated $^1$D chromatogram was multiplied by the ratio between the number of data points after interpolation and before interpolation. This resulted in the interpolated $^1$D chromatogram having the same maximum intensity as the original data.

Gaussian fitting was accomplished through the use of the nonlinear least-squares function, \textit{lsqnonlin}, available in the optimization toolbox for MATLAB®. In order to perform the nonlinear least-squares fit of the $^1$D peak, initial estimates for the Gaussian parameters (area, position, $\sigma$, and background level) of the peak were determined. The initial estimate for the peak area was determined by Eq. 45,

$$\text{Area} = a + b + c - 1.5(d + e)$$  \hspace{1cm} (45)

where $a$ and $c$ are the values of the data points to either side of the peak maximum, $b$ is the value of the peak maximum, and $d$ and $e$ are the value of background levels at either end of the $^1$D peak, as shown in Fig. 30B. The peak area was bounded to within $\pm 50\%$ of this initial guess. In order to determine the initial estimate for the peak position, a spline curve was applied to the sampled $^1$D peak and the resulting maximum was chosen as the peak position. A value of 0.21 min was used as the initial estimate for $\sigma$. Furthermore, an upper and lower bound of 0.28 min and 0.14 min was used to reduce the chances of overfitting the original $^1$D peak if an insufficient number of points (< 4) was available for fitting. The background level was calculated by averaging the first and last data points (points $d$ and $e$ in Fig. 30B) in the sampled $^1$D
chromatogram. Using the determined regression parameters for the peak area, retention time and 
σ, a Gaussian peak was generated using the sampled time axis (0.35 min to 2.8 min at 0.35 min 
intervals for the simulated data). In order to correctly scale the interpolated 1D peak for each 2D 
time point, linear least squares regression was used to fit the individual 1D chromatograms at 
each 2D time to obtain the correct amplitude. Using these amplitudes (areas), interpolated 
Gaussian peaks were generated using the interpolated time axis (0.35 min to 2.8 min at 0.035 
min intervals for the simulated data), and alignment was carried out as described previously.

5.3.4 PARAFAC analysis

After each interpolation technique was applied and alignment subsequently carried out, 
the data sets were analyzed using an in-house PARAFAC function. The PARAFAC function 
was designed using an ALS algorithm as described by Smilde et al. [66]; however, it was 
expanded to account for the fourth data dimension (sample). Non-negativity, unimodality, and 
selectivity were implemented independently for each dimension and component using the same 
approach previously implemented by Bezemer and Rutan [117]. The PARAFAC analysis was 
conducted with a maximum of 2000 iterations and a convergence criterion of 1 x 10⁻¹⁰. The peak 
components were constrained by applying spectral selectivity from 440 nm to 700 nm, 
unimodality in both the second and first dimensions, and non-negativity was applied to all 
dimensions. The background components were constrained by applying non-negativity to the 
sample dimension and the second and first dimensions.
5.4. Results and Discussion

5.4.1 Retention time prediction

The predicted retention times for the simulated data set were calculated for each of the four proposed interpolation techniques over the course of all of the samples. A plot of the predicted retention times for the four interpolation techniques of the simulated data versus the actual retention times used to generate the peaks can be seen in Fig. 3. The solid line found in Fig. 3 represents the actual retention times of the simulated peak.

The squares (□) show the predicted retention times obtained from the PCHIP interpolation procedure for the thirty-five samples. The PCHIP predicted retention times were found to be 1.4 minutes for the first sixteen samples and 1.75 minutes for the remaining nineteen samples. This abrupt shift from one retention time to another is due to the nature of the PCHIP interpolation. PCHIP interpolation is almost a linear interpolation between existing data points. A small amount of curvature is allowed to exist between data points as a result of the second derivative being non-continuous. Due to the shape preserving nature of the PCHIP technique, the peak maxima observed in the sampled $^1$D peaks remains the maxima after interpolation. The abrupt change in predicted retention time is due to a change in the $^1$D peak sampling phase. The sampling phase of the simulated $^1$D peak is 2.73 at sample fifteen, 3.07 at sample sixteen, and 2.77 at sample seventeen. Since the sampled $^1$D sampling phase passes through $\pi$ between samples sixteen and seventeen, the predicted retention time abruptly shifts when the signal in the $^1$D at 1.75 minutes becomes larger than the signal in the $^1$D at 1.4 minutes. The linear interpolation used in the literature also produces the same predicted retention times as the PCHIP interpolation technique.
Fig. 31: A plot of the retention times after reconstruction of the first dimension for the simulated data set versus the actual retention times used for the generation of the simulated data. The solid line is the actual retention times plotted against the actual retention times. The four interpolation techniques, PCHIP (◻), spline polynomial (▲), Fourier zero-filling (×), and Gaussian fitting (○), are compared against this line.

The triangles (▲) in Fig. 31 show the predicted retention times obtained from the spline interpolation procedure for the thirty-five samples. Unlike the PCHIP interpolated peak, the predicted retention time of the spline interpolated 1D peak changes over the course of the samples. This change is due to the difference in how the spline is implemented versus PCHIP. Both techniques use a cubic polynomial. However, where the PCHIP technique only requires the first derivative to be continuous, the spline interpolation requires that both the first and second derivatives are continuous. This results in the maximum of the interpolated 1D peak shifting from the existing data points to points in between. The spline interpolated peak begins to take on the more rounded shape of the simulated peak. However, while this is an improvement over the PCHIP interpolation, the predicted retention times do not closely correspond to the actual retention times of the simulated peak.
The $\times$’s in Fig. 31 show the predicted retention times obtained from the Fourier zero-filling interpolated $^1$D peak. The predicted retention times were closer to the actual retention times compared to the Hermite and spline interpolation techniques. However, the predicted retention times from the Fourier zero-filling oscillated between either being too high or too low.

The circles (o) in Fig. 31 show the predicted retention times obtained from the Gaussian fitting interpolated $^1$D peak. Unlike the previous three interpolation techniques, the Gaussian fitting interpolation technique produces predicted retention times that are nearly identical to the actual retention times used to generate the simulated peak. Since the original sampled $^1$D peak was created using a Gaussian curve, the fact that the Gaussian fitting interpolation technique produces the most reliable calculated retention times makes sense. However, the Gaussian fitting interpolation only produces an accurate depiction of the unsampled $^1$D peak if the fitting is properly conducted, i.e., all parameters are properly constrained. If the bounded range for the $\sigma$ value, from section 5.3.3, is not relatively close to the actual sigma, then the resulting Gaussian fitted interpolated $^1$D peak will be either too narrow or too wide. While the width of the peak does not have an impact on the predicted retention times, an erroneous $\sigma$ value will significantly affect the accuracy of the results (i.e., relative peak areas) obtained from the PARAFAC analysis.

5.4.2 Calibration curves

In order to further measure the effectiveness of each interpolation technique, the results from the PARAFAC analysis were used to generate a calibration curve for the simulated calibration points. The Excel linest function was used to calculate the slopes and intercepts with the corresponding errors, shown in Table 7, for each of the five interpolation techniques. The relative standard deviations of the slopes were used to provide an initial estimate of how well the
PARAFAC relative areas fit to the calculated trend line, shown in Table 7. The relative slope errors were calculated by dividing the error by the slope and multiplying by one hundred. The relative slope errors for each of the interpolation techniques and the unaligned calibration curve were calculated to be as follows: 17.9 % for the unaligned calibration curve, 11.5 % for the linear with cross correlation of the $^1$D chromatogram calibration curve, 4.7 % for the PCHIP calibration curve, 2.9 % for the spline calibration curve, 3.7 % for the Fourier calibration curve and 1.9 % for the Gaussian calibration curve. From the relative slope errors given in Table 7, the unaligned data set, as expected, possessed the most error in the calibration points. This is easily explained by the lack of multilinearity in the unaligned data set. The relative slope error for the linear interpolation followed by cross correlation of the $^1$D chromatogram was better than the unaligned relative slope error. However, in comparison to the relative slope errors of the other interpolation techniques, the relative slope error of the linear interpolation followed by cross correlation of the $^1$D chromatogram was much higher. This large difference in the standard errors may be due to the manner in which the cross correlation alignment approach works. Unlike the four proposed interpolation techniques, the cross correlation alignment approach may not necessarily align a peak to the same point. In addition, the shape of the interpolated peak may not be consistent across all of the samples. Both of these reasons could result in the aligned data set not being completely multilinear. The PCHIP and Fourier zero-filling interpolation techniques produced the next set of comparable relative slope errors. The PCHIP relative slope error is due to the shape preserving nature, as discussed previously, of the technique. Likewise, the Fourier zero-filling relative slope error may be explained by a similar shape preserving ability. Finally, the spline and Gaussian fitting interpolation techniques produced the lowest relative slope errors. The most probable reason for the small relative slope errors is their ability
to impose a uniform shape on the interpolated $^1$D peak. This imposition of a uniform shape aids in forcing multilinearity for the simulated data set.

**Table 7:** Calculated slope and intercept (with associated errors) for the simulated data set from the relative areas obtained from PARAFAC

<table>
<thead>
<tr>
<th>Interpolation Method</th>
<th>Slope</th>
<th>Intercept</th>
<th>Standard error of the regression ($s_y$)</th>
<th>Relative Standard Deviation of Slope (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unaligned</td>
<td>1.2(0.2)</td>
<td>-1(5)</td>
<td>9.2</td>
<td>17.9</td>
</tr>
<tr>
<td>Linear</td>
<td>1.0(0.1)</td>
<td>-3(3)</td>
<td>4.9</td>
<td>11.5</td>
</tr>
<tr>
<td>PCHIP</td>
<td>1.21(0.06)</td>
<td>0(2)</td>
<td>2.5</td>
<td>4.8</td>
</tr>
<tr>
<td>Spline</td>
<td>1.39(0.04)</td>
<td>0(1)</td>
<td>1.8</td>
<td>2.9</td>
</tr>
<tr>
<td>Fourier</td>
<td>1.70(0.06)</td>
<td>2(2)</td>
<td>2.8</td>
<td>3.5</td>
</tr>
<tr>
<td>Gaussian</td>
<td>1.73(0.03)</td>
<td>0(1)</td>
<td>1.4</td>
<td>1.7</td>
</tr>
</tbody>
</table>

In addition to calculating the relative slope error, the calibration curves were used to determine the calculated concentration of the validation samples. The calculated validation sample concentrations were used to determine the percent standard error of prediction (% SEP), a measure of the accuracy of the approach, and the average percent relative standard deviation (% RSD), a measure of precision. Table 8 shows the calculated % SEP and % RSD values for the unaligned data and for the different interpolation techniques. As expected when the data was not aligned prior to analysis by PARAFAC, the resulting relative peak areas were the most inaccurate and imprecise. Of the five interpolation techniques examined, the linear followed by cross correlation of the $^1$D chromatogram was the most inaccurate but the imprecision of the technique was on par with most of the other interpolation techniques, except for the Gaussian fitting. Of the remaining four interpolation techniques, the Gaussian fitting calculated concentrations were the most accurate and precise. This observation is not surprising, as Gaussian peak shapes were used to generate the simulated data.
### Table 8: Average percent standard error of prediction and percent relative standard deviation for the validation samples within the simulated data

<table>
<thead>
<tr>
<th>Interpolation Method</th>
<th>% SEP</th>
<th>Average % RSD</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unaligned</td>
<td>23.8</td>
<td>4.0</td>
</tr>
<tr>
<td>Linear</td>
<td>14.2</td>
<td>2.4</td>
</tr>
<tr>
<td>PCHIP</td>
<td>9.3</td>
<td>2.5</td>
</tr>
<tr>
<td>Spline</td>
<td>8.0</td>
<td>2.2</td>
</tr>
<tr>
<td>Fourier</td>
<td>8.7</td>
<td>2.5</td>
</tr>
<tr>
<td>Gaussian</td>
<td>4.1</td>
<td>1.1</td>
</tr>
</tbody>
</table>

### 5.4.3 Experimental data

From the results of the simulated data analysis, the Gaussian fitting interpolation technique was expected to be the most reliable technique for aligning the experimental sampled 1D peaks. Each of the five peaks in the experimental data set were interpolated, aligned, and analyzed using PARAFAC. Since a calibration curve was not available for the experimental data, only the precision measurement, % RSD, is shown in Table 9. Since this data set has been previously analyzed by Bailey and Rutan [3], a column titled MCR-ALS, where the results from this MCR-ALS analysis are reported, was included for comparison purposes. None of the four interpolation techniques were able to match the degree of reproducibility obtained by Bailey and Rutan [3]. This MCR-ALS method, which requires manual integration, while somewhat tedious and less automated than PARAFAC, does not require that the data have a multilinear structure.

### Table 9: Percent relative standard deviations of the relative peak areas for the five experimentally generated peaks

<table>
<thead>
<tr>
<th>Peak</th>
<th>Unaligned</th>
<th>MCR-ALS [3]</th>
<th>Linear</th>
<th>PCHIP</th>
<th>Spline</th>
<th>Fourier</th>
<th>Gaussian</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>13.0</td>
<td>1.6</td>
<td>4.29</td>
<td>6.27</td>
<td>5.24</td>
<td>5.05</td>
<td>4.39</td>
</tr>
<tr>
<td>2</td>
<td>16.8</td>
<td>2.2</td>
<td>6.76</td>
<td>4.33</td>
<td>3.77</td>
<td>4.60</td>
<td>5.84</td>
</tr>
<tr>
<td>3</td>
<td>7.45</td>
<td>4.7</td>
<td>13.9</td>
<td>19.6</td>
<td>8.41</td>
<td>17.0</td>
<td>16.8</td>
</tr>
<tr>
<td>4</td>
<td>11.4</td>
<td>3.5</td>
<td>12.8</td>
<td>12.9</td>
<td>12.1</td>
<td>11.2</td>
<td>11.8</td>
</tr>
<tr>
<td>5</td>
<td>17.0</td>
<td>1.4</td>
<td>15.9</td>
<td>11.8</td>
<td>13.6</td>
<td>10.1</td>
<td>10.4</td>
</tr>
<tr>
<td>Average</td>
<td>13.1</td>
<td>2.7</td>
<td>10.7</td>
<td>11.0</td>
<td>8.61</td>
<td>9.58</td>
<td>9.83</td>
</tr>
</tbody>
</table>
In addition to visually comparing the % RSDs for the five techniques vs. the unaligned % RSD, Levene’s test was used to determine if there was a statistical difference between the different normalized relative peak areas [134]. Levene’s test is a robust method that works under the assumption that the underlying data possesses a normal distribution. Since the number of data points for each technique is small, the Brown and Forsythe test could not be used to determine if the data deviates from a normal distribution [134]. The critical F-value for Levene’s test is 2.53 obtained from an α of 0.05 with 5 and 30 degrees of freedom respectively. Levene’s test revealed that the normalized relative peak areas for Peaks 1, 2, and 3 possessed statistically significant differences. The reason for the statistical significant difference for normalized relative peak areas for Peaks 1 and 2 is due to the large unaligned standard deviation, seen in Fig. 32A and 32B.

![Fig. 32: Plots of the standard deviations for the normalized relative peak areas obtained from PARAFAC for (A) Peak 1, (B) Peak 2, and (C) Peak 3. The solid black line is the average standard deviation for all techniques for each peak.](image-url)
The standard deviations for the unaligned normalized relative peak areas are a clear outlier in comparison to the standard deviations from the other techniques. The unaligned standard deviation was much larger due to the degree of peak shifting for Peaks 1 and 2 relative to the other peaks. In both cases, the normalized relative peak areas for samples three and four were higher due to the presence of the peak existing in the center of the data window for these samples. The reason for the statistical significant difference for the normalized relative peaks areas for Peak 3 is not as easily explained, Fig. 32C. In comparison to the retention time shifting of Peaks 1 and 2, Peak 3 experiences less retention time shifting over the course of the experimental run. An examination of the normalized relative peak areas reveals that for the PCHIP, Fourier zero-filling, and Gaussian fitting techniques, the normalized relative peak area for sample six of the experimental data set is less than the other techniques. An identifiable cause for this decreased value has not been determined.

Even though the Gaussian fitting was found to produce the most accurate and precise calculated concentrations for the simulated data set, the Gaussian fitting for the experimental peaks did not produce better results than the other techniques. One possible reason for this deviation from the simulated data is the inability of the non-linear least squares fitting to correctly account for the background signal within the sampled $^1$D peak. The background used in the simulated data set was not the same background present in the experimental data set. As a result, the resulting fitting parameters calculated for the experimental data set were not as consistent as the fitting parameters calculated for the simulated data set. This deviation in the fitting parameters results in the interpolated Gaussian peak showing a greater degree of inconsistency, in comparison to the simulated interpolated Gaussian peak, across each of the replicate samples. A second consideration is that the actual experimental data may not follow the
Gaussian model exactly, such that for the Gaussian method specifically may not give improved results for the experimental data.

5.5. Conclusions

The five interpolation techniques were successfully applied to LC×LC-DAD data during the alignment process. From the simulated data set, the Gaussian interpolation technique was found to produce the best % SEP and % RSD in comparison to the other interpolation techniques. This was expected due to the Gaussian nature of the simulated chromatographic peaks and was expected to carry over into the experimental data set. However, when the techniques were applied to the experimental data, the Gaussian fitting was found to not be statistically better or worse than the other interpolation techniques. The results from MCR-ALS reported previously [3] still provide the best precision for the experimental data set. This approach has the advantage that it is not affected by retention time shifts in the first and second dimension separations, but is more tedious and less automated than PARAFAC, especially with respect to the final quantification step which requires manual assignment of the chromatogram baseline. To reduce the potential impact of the background on the Gaussian fitting interpolation technique, an approach consisting of performing PARAFAC on each sample individually is being developed. This method will enable alignment of more complex chromatograms; because the peaks being aligned will be constrained to have the same spectral features. The new approach will isolate the sampled 1D peak signature from the background and hopefully reduce the variations in the Gaussian fitting parameters between samples. This approach is discussed in great detail and validated in Chapter 6.
CHAPTER 6: Development of a Two Dimensional Alignment Approach

The focus of this chapter is the generation and testing of an approach to correct retention time shifting in the first and second dimension prior to PARAFAC analysis for LC×LC-DAD data. The alignment strategy utilizes the spectra of the compounds to independently align the peaks without the need for a reference sample. Peak alignment is achieved by shifting the optimized chromatographic component profiles from a three-way PARAFAC analysis model applied to each sample. To ensure accurate shifting, components are matched up based on their spectral signature and the position of the peak in both chromatographic dimensions. The degree of shift, for each peak, is determined by calculating the distance between the median data point of the respective dimension (in either the second or first chromatographic dimension) and the maximum data point of the peak furthest from the median. All peaks that were matched to this peak are then aligned to this common retention data point. Portions of this chapter are reproduced from reference [94] with permission from Elsevier.

6.1. Introduction:

As was previously explained in Chapter 3, section 3.2, PARAFAC analysis requires retention time shifting to be corrected prior to using the PARAFAC algorithm. The common approach for aligning 2D chromatograms is to linearly shift the sample chromatogram relative to the reference chromatogram and use a defined metric to determine the point of maximum alignment [76, 77, 81, 83, 116]. Although these techniques successfully aligned the respective data sets, most of the data sets aligned were either single peaks or replicate samples of several
samples. Several problems potentially exist if these methods are applied to data sets with more heterogeneity between samples. First, each of these techniques requires that a suitable reference sample be selected prior to alignment. Typically the chromatogram with the most peaks is selected, although this does not guarantee that these same peaks are present in every sample. Second, all of these techniques do not take into account the chemical signatures of the peaks when alignment is performed. These techniques only align the data based on the raw shape of the chromatograms. There then exists the possibility that incorrect peak alignment could occur based on the appearance of a peak within a sample at approximately the same retention time as a peak in the reference chromatogram. Zhang et al. did propose that the use of the SIM chromatograms be used to provide an overall warping signal if this is a possibility when performing the 2D COW technique [80]. However, due to the large presence of background contributions, this approach is not applicable to LC×LC-DAD data. An alternative approach that will be discussed in this chapter is to handle shifts for each peak independent of other peaks in the samples.

6.2 Method Overview

The semi-automated alignment method is designed to align the peaks within the localized data region based on their PARAFAC resolved spectral components. The semi-automated alignment method consists of six general steps: i) selection of an appropriate region of the two-dimensional chromatogram for analysis, ii) selection of the number of components, iii) initial MCR-ALS analysis and identification of components, iv) three-way PARAFAC analysis on each sample, v) component matching between samples, vi) alignment of the second and first
dimensions, and vii) four-way PARAFAC analysis on the aligned data set. Each of these steps is discussed in further detail below.

6.2.1 Selection of an appropriate region of the two-dimensional chromatogram for analysis

While analyzing maize samples by LC×LC-DAD, Porter et al. discovered that PARAFAC was not capable of analyzing the entire sample at once [92]. Instead, Porter et al. localized the PARAFAC analysis around a peak of interest. The localized region was selected to ensure that the peak was completely within the region in both the second and first dimension. Also, sufficient space around the peak was included to allow for successful accounting of background components. Bailey et al. further refined this approach by choosing a localized region that allowed for the shifting of the peak in both the second and first dimension [3]. The approach used in this method to determine the range of the localized region for analysis is the same approach used by Bailey et al.

6.2.2 Selection of the number of components

As stated in the introduction, PARAFAC requires that the correct number of components be selected for analysis. We have tried several automated methods to determine the number of components, such as cross-validation [112] or an F-test [104], however, the resulting number of components was always overestimated. Instead, a process has been developed to provide the user with enough information to perform an educated guess for the number of components present in both the entire four-way data set and for each of the samples. To generate the necessary information, SVD is performed on both the entire four-way data set and on each three-way data set corresponding to each sample. Once SVD is performed on the overall data set, and
on each of the samples, the resulting log (base 10) of the singular values is plotted vs. the corresponding component number. Fig. 17, Chapter 3, section 3.5, is an example of the resulting scree plot. The user is prompted to choose the number of components for the both the entire raw data set and for each sample. The number of components in a scree plot is typically chosen by selecting the number of components at the point at which the slope of the plot changes suddenly, indicated by point A in Fig. 17 [91]. Alternative approaches for interpretation of the scree plot have included the identification of the largest gap in the plot, point B [92, 93]. Unfortunately, due to the variability in the singular values between the entire raw data set and each sample, neither of these approaches provided a fool-proof method for determining the number of components. The criterion used in developing this method is to identify a break within the singular values, denoted by C in Fig. 17, that accounts for the number of visible peaks present (by inspection of a contour plot) and the presence of background components [3].

6.2.3 Initial estimate generation and identification of components

The PARAFAC algorithm used in this method allows for the implementation of non-negativity, unimodality, and selectivity constraints for each component within each mode selectively. These constraints were implemented in a similar fashion to that described by Bezemer and Rutan for MCR-ALS [117]. The non-negativity constraint is used in all four modes: second dimension, first dimension, sample, and spectra. Unimodality is only applied in the second and first dimension. The unimodality constraint used in this method halves the intensity of the lesser peak over successive iterations, effectively removing the peak from the component [120]. The more common vertical and horizontal implementations of unimodality were not able to be used due to the possibility of noisy profiles in the second and first dimension.
The selectivity constraint is implemented in the second dimension, the first dimension, the sample dimension, and the spectral dimension. However, to correctly construct each constraint, the components within each mode need to have been previously identified as either an analyte component or a background component. The procedure used to generate an initial estimate of each component for each mode is an approach similar to the IKSFA-ALS method developed by Bailey and Rutan [3]. The user has the choice to use either IKSFA [115, 119] or orthogonal projection approach (OPA) [147] to generate the initial guesses. The OPA algorithm used in this method has been modified to allow for the same iterative approach used in IKSFA and will be called iterative orthogonal projection approach (IOPA) to distinguish from normal OPA. Based on the number of components selected for the raw data set and each sample, IKSFA (or IOPA) produces the most dissimilar set of spectra possible for the selected number of components. These spectra are used as initial estimates for MCR-ALS to generate the chromatographic profiles of each component in both the entire raw data set and for each sample. These resolved two-way chromatographic profiles are reshaped into four-way structures and plotted as contour plots. The user is directed to identify the analyte and background components. Once the components for the entire raw data set have been identified, the user is prompted, for each analyte component, to select the point at which selectivity will be implemented in the spectral dimension, i.e., the wavelength above which the analytes are not expected to absorb. MCR-ALS is then performed on the entire raw data set again and the resulting constrained spectra are then used as the reference spectra for the component matching step described in section 6.2.5.
6.2.4 Three-way PARAFAC analysis on each sample

To allow for each peak within the data set to be aligned separately, three-way PARAFAC is performed on each sample to obtain the component profiles in the second mode, first mode, and spectral mode. Two benefits accrue from applying three-way PARAFAC to each sample separately. First, the likelihood of overfitting the data due to non-existent components present in other samples is minimized. Second, applying three-way PARAFAC to each sample allows for each sample to be constrained by trilinearity. To ensure that each sample is trilinear prior to performing three-way PARAFAC analysis, three options are given to allow the user to better constrain and correct the analyte components.

The first option is whether the user wishes to split an analyte component into two or more components in the second dimension, the first dimension, or both. This option is included in the method due to the possibility that more than one peak may be present in an analyte component, because two compounds have highly similar spectra. If this is the case, the use of unimodality during the three-way PARAFAC analysis would result in the smaller of the two peaks not being included in the analysis. The approach used to split the analyte component is a modification of the approach used by Tistaert et al. [2]. An example of this process is shown in Fig. 33 where a single PARAFAC component containing two peaks is being split into two analyte components.

In Fig. 33A, the first dimension is selected as the dimension where the component is being split, and a point is selected where the split will occur, denoted by the dashed white line. Next, an identical duplicate of the analyte component in Fig. 33A is generated as a new component. Since the MCR-ALS components will be recombined prior to the three-way PARAFAC analysis, the sections of the analyte components outside of the specified region need to be set to zero. In the example in Fig. 33, the region to the right of the dashed line is set to
zero, Fig. 33B, and the intensities at the point at which the component was split, the dashed line, are then multiplied by 0.5. Likewise, the region of the newly generated component, Fig. 33C, to the left of the dashed line is then set to zero and halved at the dashed line. To ensure that the split components do not recombine during three-way PARAFAC analysis, a chromatographic selectivity constraint, in this case in the first dimension, is used so that the region where the peak is not present is set to zero for each component. Finally, the number of components for this sample is increased by the number of additional components created by splitting.

**Fig. 33**: A) Contour plot of the MCR-ALS component from the twelfth replicate sample of the urine sample showing the two peaks of interest within the urine data set. The dashed white line indicates the 1D point at which the component will be split into two components. B) The resulting MCR-ALS component after splitting the original MCR-ALS component into two components. C) The newly generated MCR-ALS component created when the component was split into two components.
To provide the requisite trilinearity for three-way PARAFAC analysis, the user next has the option to correct for within sample retention time shifts. If a peak does not possess the same retention time in each 2D chromatogram, then that analyte component does not possess trilinearity. The process to correct for within sample retention time shifting is shown in Fig. 34. In Fig. 34A, the analyte component has a skewed contour, indicating that the retention times between the 2D chromatograms are not identical. The user selects the 1D time points where the peak is present, 7.86 min, 7.89 min, 7.92 min and 8.00 min in Fig. 34A, and the 2D chromatograms corresponding to those 1D data points are plotted, Fig. 34B. The user is prompted to select the maximum positions of the 2D chromatograms, which give the numbers shown in Fig. 34B. From these maximum positions, each of the 2D chromatograms is shifted so that the maxima all occur at the earliest selected point, Fig. 34C. To accommodate change in the length of the 2D chromatogram, each 2D chromatogram is extended by using the intensity value at the latest point so that all of the 2D chromatograms remain the same size. When the aligned analyte component is plotted as a contour plot, the resulting shape of the peak is less skewed, as shown in Fig. 34D.
Fig. 34: A) A contour plot of the MCR-ALS component containing the phenytoin peak (50 ppb) from the sixth sample of the phenytoin data set. B) A plot of the $^2$D chromatograms from the second to fifth $^1$D data points in 34A. The numbers (in s) indicate the position of the peak maximum for the corresponding color $^2$D chromatogram. C) The $^2$D chromatograms shown in 34B after shifting all of the $^2$D chromatograms to the earliest maximum (12.60 s). D) The resulting contour plot of the MCR-ALS component after aligning the $^2$D chromatograms.

The third option is to constrain the second dimension or first dimension with selectivity. Four choices are available to the user. The user can decide to constrain each analyte component in either the second dimension or first dimension, in both the first and second dimensions, or not constrain the analyte component at all. However, if the analyte was previously split in either the second dimension or first dimension, the option to implement selectivity on that component in the split dimension is not available because the selectivity constraint has been implemented previously.

After all three options have been either implemented or rejected, the modified MCR-ALS components, both analyte and background, are reconstructed into a pseudo- data set as follows
where the matrix $\mathbf{R}$ contains the resolved chromatographic profiles and the matrix $\mathbf{S}$ contains the resolved spectra. The residuals are not added since, in theory, the residuals should just be random noise. At this point, three-way PARAFAC is performed on the reconstructed data for each individual sample.

**6.2.5 Component matching between samples**

After three-way PARAFAC is performed on each sample separately, the possibility exists that the component profiles may not be in the same order for each sample. A matching scheme was created to allow for analytes to be identified in comparison to the reference spectra generated in the MCR-ALS step as described in section 6.2.3. In addition, the general trend of the position of the analyte peaks in both the second dimension and first dimension is taken into consideration to prevent outliers with the same spectra from being aligned incorrectly.

The general procedure for matching the spectral component profiles between samples is illustrated in Fig. 35. A table, containing the Pearson correlation coefficients between all possible pairs, is constructed with the reference components occupying the rows and the sample components occupying the columns, as shown in Fig. 35A. At the start of the matching, the user is prompted to select a minimum threshold for confirming a spectral match; 0.9 has proven more than sufficient to ensure accurate matching for the data sets examined in this article. Since only analyte components are aligned, the background components in the reference data set, identified as components one and three in this example, are removed from consideration, the grayed out boxes in Fig. 35B. The removal of background components is then repeated for those components which had been identified as background in the sample of interest, Fig. 35C. Once
the background components have been removed from consideration, starting with the first available correlation coefficient, each coefficient is compared to the threshold. If the correlation coefficient is larger than the threshold and larger than any previous coefficients then that coefficient is selected as a match. This process is repeated until all components are matched or all remaining coefficients are less than the threshold, Fig. 35D. A slight modification exists if a matched component was previously split within a sample. In this case, the generated component is not directly compared to the reference spectra. Instead, only after a match has been determined between one of the reference spectra and the pre-split component, the split component is automatically matched to the same reference spectrum. Once the spectral matching is complete for all samples, a trail of component matches from the samples for each reference component has been generated and will be used as a template for matching the $^2$D components between each sample.
Fig. 35: Schematic of the spectral component matching system. The reference components have been labeled in the following order: 1) background, 2) analyte, 3) background, 4) analyte. The sample components have been labeled in the following order: 1) analyte, 2) background, 3) background, 4) analyte. A) Matrix generated from all possible Pearson correlation coefficient pairs. B) Removal of the raw background components (1 and 3) from consideration. C) Removal of the sample background components (2 and 3) from consideration. D) The resulting analyte component matches between the raw reference spectra and the three-way PARAFAC spectra for this sample.

After all of the components for each sample have been matched to the reference spectra, incorrect spectral matches are eliminated based on the peak positions in the second dimension. Unlike the spectral matching procedure, verifying component matches between samples in the second dimension does not use correlation coefficients. This change in strategy is due to the inconsistency in the position of the peak within the second dimension and the lack of a set reference 2-D chromatograms. Instead, the 2-D matching scheme uses the maximum positions of the analyte components from the three-way PARAFAC analysis. The 2-D chromatograms are smoothed using a Whitaker smoothing using a weighting parameter of 9 [148, 149] prior to
determining the maximum position to reduce the impact of noise on both the verification of components and the alignment of the second dimension in section 6.2.6. Using the spectral component trail for each reference data set component identified as an analyte, linear regression is used to calculate a slope and intercept from the maximum positions, $y$, and the corresponding sample number, $x$. The residuals are then calculated for every sample. Starting with the largest residual, the residual is then compared to a user defined threshold, the absolute maximum amount of deviation from the line allowed for each of the positions. If the residual is greater than the threshold, the corresponding sample and component is eliminated as a match, and linear regression is performed again. The comparison of the residuals to the threshold is continued until all residuals are less than or equal to the threshold. These samples are used as the starting variables for the matching within the first dimension.

Peak position matching for the first dimension follows the same procedure except that the $1D$ component profiles are interpolated prior to matching. Interpolation is performed to alleviate the sampling effects on the shapes of the $1D$ chromatograms and the apparent retention times. The interpolation used in this method is either Gaussian fitting or EMG fitting depending on the user’s choice. Gaussian fitting should be sufficient in most cases except where significant tailing occurs in the first dimension. Gaussian fitting was chosen as the interpolation technique based on results obtained from comparing five different interpolation strategies in a previous study [135], described in Chapter 5. Also, the flat baseline obtained when implementing Gaussian fitting allows for peak shifting while keeping the number of the data points in the resolved profile equal to the number of data points in the original profile, which is required for implementing the data reconstruction step described in section 6.2.7. The matching results from the first dimension are then used to initialize the alignment of the second and first dimensions.
6.2.6 Alignment of the second dimension and the first dimension

The alignment for each reference component is conducted separately from the other components, taking into consideration components that were previously split within certain samples. This approach allows for each peak to be aligned individually instead of relying on an overall alignment of the localized window. This approach was also designed to prevent peaks that exist on the edge of the localized window from being distorted during alignment. Within the second and first dimensions, the positions of the peak closest to the beginning and end of the localized window are determined based on the component trail, as shown in Fig. 36. The distance between each point and the median of the window is calculated. Three possible scenarios arise depending on the distance. The first scenario, Fig. 36A, occurs when the greater distance between the positions to the median is positive. This results in the peak within this component being shifted to a later retention time within the localized window. The second scenario, Fig. 36B, occurs when the distance between the two points to the center of the window is equal. The peaks within this component are then aligned so that the retention time of the peak is set to the center of the localized region. The third scenario, Fig. 36C, occurs when the greater distance between the positions to the center is negative. The peaks are then shifted to an earlier retention time within the localized region. In each case, after being shifted, the lengths of the 2D or 1D chromatograms are different between each sample. To compensate for this, the baseline of the Gaussian fitting is extended to ensure that the chromatogram lengths are equal for all samples. Likewise, the baseline of the 2D chromatogram is also extended to ensure that the number of points in the 2D chromatograms remains the same for each sample. Depending on the direction of the shift, the baseline is extended in the opposite direction.
**Fig. 36:** Schematic of how the direction of the alignment is determined. The white and black triangles represent the earliest and latest eluting peaks. The arrow indicates the direction of alignment. A) The distance between the black triangle and the median is greater than the distance between the white triangle and median resulting in the peak being aligned to the right. B) The distance between the white and black triangles and the median is equal resulting in the peak being aligned to the middle of the data window. C) The distance between the white triangle and the median is greater than the distance between the black triangle and the median resulting in the peak being aligned to the left.

### 6.2.7 Four-way PARAFAC analysis on the aligned data set

After the $^2$D and $^1$D chromatograms are aligned, the three different modes for each sample are recombined according to

$$x_{ijkl} = \sum_{n=1}^{N} a_{i,n} b_{j,n} c_{k,n} d_{l,n}$$ (47)

To maintain the size of the aligned data set compared to the raw data set, the first dimension is resampled to remove the interpolated data points. The reconstructed four-way data are then reshaped into a two-way structure. The IKSFA (or IOPA) results from section 6.2.3 are then used to reinitialize a constrained (using the same constraints as used previously) MCR-ALS of this reconstructed two-way data set. The resolved chromatographic and spectral profiles are then
used as initial estimates for four-way PARAFAC analysis. The resulting sample component profiles provide the relative concentrations of the analytes within each sample.

6.3 Experimental:

The development and validation of the method was conducted on a Dell® Optiplex 755, Intel(R) Core™2 Duo CPU, E6550 @ 2.33 GHz, 3.23 GB of RAM. The software used to implement the method was MATLAB® software 2009a (Mathworks, Inc. Natick, MA) version 7.8.0.347. The non-linear least squares Gaussian and EMG fitting required the use of the MATLAB optimization toolbox.

6.3.1 Data sets

The method was developed and validated using four simulated and two experimental data sets. The simulated data sets were constructed using the same methodology as previously reported by Allen and Rutan [135] and discussed in Chapter 5. The simulated data sets consisted of four separate peaks; A, B, C, and D. Peak A was the primary peak of interest in all four simulated data sets and consisted of a five-level calibration, with each level in triplicate, and four “unknown” samples, with each unknown in quintuplicate. Peak B was an interferent peak present in each of the “unknown” samples in the second simulation. Peak C was a secondary peak with the same style of calibration curve and “unknown” samples as Peak A in the third simulated data set. Peak D was an interferent present in some calibration samples and “unknown” samples for the fourth simulated data set. Peak D also possessed an identical spectra to peak A.
The first experimental data set was obtained from the Carr research group at the University of Minnesota and consisted of fourteen replicate samples (with a 2D run time of 21 s and a 1D run time of 30 min) of a urine sample [128]. A small section, 2.15 s in the 2D and 2.45 min in the 1D, was selected for analysis, shown in Fig. 37A. The resulting size of the urine data set was 87 data points in the second dimension, 8 data points in the first dimension, 14 data points in the sample dimension, and 126 data points in the spectral dimension. The spectral range of the data set was 200 to 700 nm with an interval between collected wavelengths of 4 nm. This section contained four peaks, the two primary peaks of interest (Peak 11 and Peak 12, respectively, as identified by Bailey and Rutan [1] (shown in Fig. 2 of reference [3]) located approximately in the center of the section and two coeluting peaks in the upper left corner of the section. The two primary peaks were chosen for analysis due to both peaks possessing the same spectra [1], Fig. 37B.
Fig. 37: A) A contour plot of the seventh sample from the urine data set. Shown in the middle of the contour plot are the two selected peaks with the same spectra. The absorbance bar on the right of the contour plot shows the intensities of the peaks present. B) The resolved spectra of the two peaks with the calculated Pearson correlation coefficient shown. C) Contour plot of the reconstructed peak shapes from the resolved four-way PARAFAC components for the two peaks of interest. The positions of the two peaks are different from Fig. 38A due having been aligned prior to reconstruction.

The second experimental data set was obtained from the Stoll research group at Gustavus Adolphus College, MN and consisted of two sets of calibration samples [47, 48]. The first set of calibration samples was a series of duplicate spikes (0, 25, 50, 75, 150 ppb) of phenytoin in distilled water. The second calibration curve was a standard addition (0, 25, 50, 75, 150 ppb) analysis of a 1000x concentrated waste water extract sample containing phenytoin. Like the distilled water calibration samples, each of the levels in the standard addition experiment was duplicated. In addition to the phenytoin peak, the phenytoin data set also contained an unknown interferent peak that was severely overlapped with the phenytoin peak. A localized region (8.75
to 20.0 s in the second dimension and 7.82 to 8.02 min in the first dimension) surrounding the phenytoin and interferent peak was selected to minimize extraneous influences (*i.e.*, other interferents or background fluctuations) on the resulting relative areas, as shown in Fig. 38A. The resulting size of the phenytoin data set was 225 data points in the second dimension, 6 data points in the first dimension, 20 data points in the sample dimension, and 101 data points in the spectral dimension. The spectral range of the data set was 200 to 600 nm with an interval between collected wavelengths of 4 nm.

**Fig. 38:** A) A contour plot of the thirteenth sample from the phenytoin data set (25 ppb spiked waste water treatment). The intensity on the right shows the relative absorbance (in mAU) of each of the peaks present. B) The resolved spectra of the phenytoin and interferent after four-way PARAFAC analysis. The solid line is the phenytoin spectra and the dashed line is the interferent spectra. The calculated Pearson correlation coefficient is also shown. C) The resolved phenytoin peak after reconstruction from the components obtained from four-way PARAFAC analysis. D) The resolved interferent peak after reconstruction from the components obtained from four-way PARAFAC analysis. The distortion of the upper shape of the peak is caused by the use of selectivity in the second dimension.
6.3.2 Data analysis scheme

The resulting relative areas from the four-way PARAFAC analysis and the methods developed by Tistaert et al. [2] and Bailey et al. [1] were normalized by dividing each relative area by the maximum relative area for each component. This normalization scheme was carried out for the four simulated data sets and the two experimental data sets. The normalization was conducted to allow for a better comparison between each of the methods.

The first, second, and third simulated data sets were analyzed using the method described in this chapter, applying four-way PARAFAC to the unaligned data set, and applying MCR-ALS to each data set. The fourth simulated data set was analyzed using the method described in this paper, applying four-way PARAFAC to the unaligned data set, and applying the MCR-ALS with unimodality method [2]. Where possible, the constraints for the analysis of the four data sets were identical within the limitations of the methods. The component containing the two peaks with identical spectra in simulation four was split at the same 1D data point for each appropriate sample for both the semi-automated alignment method and MCR-ALS with unimodality.

The resulting relative areas obtained by four-way PARAFAC analysis and summed chromatographic components from MCR-ALS were used to calculate the accuracy of the calculated values for the “unknown” samples for peaks A and C using the following formula:

\[
\% \text{ recovery} = 100 + \frac{(\text{actual-expected})}{\text{expected}} \times 100
\]

(48)

where actual is the value obtained from using the slope and intercept obtained from linear regression and expected is the value originally used as the area of the sampled 1D peak. The average % recovery between the four “unknown” samples was then calculated. The corresponding errors for the % recoveries were determined by calculating the % RSD from the
normalized four-way PARAFAC relative areas. Appropriate error propagation was used to generate the error for the average % recovery of the four “unknown” samples.

Both of the experimental data sets were analyzed using the method described in this chapter. The MCR-ALS with unimodality method [2] was used to analyze the urine data set and the % RSDs of the summed resulting peak chromatograms were compared to the relative areas % RSDs from the present semi-automated alignment method. The component containing peaks 11 and 12 was split at the same 1D point for both methods. All other constraints used were applied in the same fashion as previously described for the simulated data sets.

The IKSFA-ALS-ssel method [1] was used with the phenytoin data set to provide a comparison to results obtained from the semi-automated alignment method. The unknown phenytoin sample concentrations were calculated using both the direct and standard addition calibration curves. The reported errors for the phenytoin concentrations were determined using a 95 % confidence interval. Instead of manually integrating the peaks as was done by Tistaert et al. [2] and Bailey et al. [1], the resulting chromatographic components were summed to generate the relative peak areas. This approach was used to provide a more direct comparison to the relative areas produced by PARAFAC.

6.4 Results and Discussion

6.4.1 Simulated data sets

The % recoveries and % RSDs for the four simulated data sets are shown in Table 10. The semi-automated alignment method produced the % recoveries closest to 100 % for each of the four simulated data sets. The summed MCR-ALS-ssel components produced the next set of % recoveries closest to 100 %. The large % recovery for simulation 3C (348 %) was due to the
inability of MCR-ALS-ssel to adequately remove the background contributions from the peak C component. As a result of this lack of background removal, the calibration curve for the simulation 3C MCR-ALS-ssel component had a negative slope compared to the slopes of the semi-automated alignment method. Unlike, simulation 2, two peaks were present in both the calibration and unknown samples. The presence of both peaks led the algorithm to generate a computational form of cross contamination and shifted some of the relative areas for both peaks A and C into background components.

Table 10: Percent (%) recoveries for the four simulated data sets after analysis by MCR-ALS and PARAFAC

<table>
<thead>
<tr>
<th>Simulation</th>
<th>MCR-ALS-ssel</th>
<th>Aligned PARAFAC</th>
</tr>
</thead>
<tbody>
<tr>
<td>1A</td>
<td>111.9±5.5</td>
<td>98.0±1.9</td>
</tr>
<tr>
<td>2A</td>
<td>107.6±4.7</td>
<td>98.6±1.3</td>
</tr>
<tr>
<td>3A</td>
<td>109.5±7.8</td>
<td>98.3±2.4</td>
</tr>
<tr>
<td>3C</td>
<td>348±22</td>
<td>100.0±3.0</td>
</tr>
<tr>
<td>4A</td>
<td>111.7±6.1a</td>
<td>99.3±1.9</td>
</tr>
</tbody>
</table>

*a calculated from % recoveries shown in Table 11 from reference [2].

6.4.2 Urine data set

Peaks 11 and 12 have been previously analyzed by Bailey *et al.* [1] and Tistaert *et al.* [2] using IKSFA-ALS-ssel and IKSFA-ALS-ssel with unimodality, respectively. The resulting % RSDs for each technique, calculated by manual integration of the resolved component, were found to be 1.04 and 3.58 by Bailey *et al.* [1] and 2.33 and 3.68 by Tistaert *et al.* [2]. By comparison, the % RSDs obtained using the present semi-automated alignment method were
4.01 and 2.67, as shown in Table 11. A more direct comparison can be made between the proposed method and the IKSFA-ALS-ssel method of Tistaert by summing up the resulting chromatographic profiles obtained. Because the present semi-automated alignment method used a localized region of the urine data smaller than the region previously analyzed by Tistaert et al., this smaller localized region has been re-analyzed using the IKSFA-ALS-ssel with unimodality method and the calculated % RSDs are reported in Table 11. The resulting % RSDs from the summation method were 11.0 and 10.3 for peaks 11 and 12, respectively, vs. 4.01 and 2.67 for the present method (values mentioned previously).

**Table 11: Percent relative standard deviations for Peaks 11 and 12 from the urine data set**

<table>
<thead>
<tr>
<th></th>
<th>Semi-automated alignment</th>
<th>IKSFA-ALS-ssel&lt;sup&gt;a&lt;/sup&gt;</th>
<th>IKSFA-ALS-ssel with unimodality&lt;sup&gt;b&lt;/sup&gt;</th>
<th>IKSFA-ALS-ssel with unimodality&lt;sup&gt;c&lt;/sup&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>Peak 11</td>
<td>4.0</td>
<td>1.04</td>
<td>2.33</td>
<td>11.0</td>
</tr>
<tr>
<td>Peak 12</td>
<td>2.7</td>
<td>3.58</td>
<td>3.68</td>
<td>10.3</td>
</tr>
</tbody>
</table>

<sup>a</sup> from reference [1]  
<sup>b</sup> from reference [2]  
<sup>c</sup> component areas determined by the simple summation method

Even though a different window size was used in each of the previous papers, the resulting % RSDs were comparable taking into account the various limitations of each method. The IKSFA-ALS-ssel method used by Bailey *et al.* did not allow the separation of the two peaks into two components. Instead, Bailey *et al.* had to make a best guess of how to manually integrate the peaks leading to a favored integration of peak 11 in their analysis. The IKSFA-ALS-ssel with unimodality method used by Tistaert *et al.* did allow for the splitting of the two peaks into two different components. However, as noted by Tistaert *et al.*, the split was not optimal for every sample, resulting in some cross contamination between the two components.

While the semi-automated alignment method produced slightly better precision for peak 12 in comparison to the IKSFA-ALS-ssel and IKSFA-ALS-ssel with unimodality, the % RSD
for peak 11 was slightly higher. A possible reason for the slightly higher % RSD was that peak 11 was aligned so that the entire peak was not present in the data window. This is due to the location of peak 11 in the fourteenth sample occurring at the far left side of the window. While this allowed peak 11 and peak 12 to be further resolved during the alignment, the resulting four-way PARAFAC component for peak 11 did not return to zero in the first dimension, shown in Fig. 37C. The decision to limit the width of the data window to 2.45 min, instead of 2.8 min, in the first dimension was due to the previous analysis of Bailey et al. [1]. Bailey et al. identified approximately four additional peaks to the immediate left of the data window that would have been included in the analysis had the window been expanded. An attempt was made at analyzing this larger window but the resulting spectra obtained by IKSFA and IOPA did not allow MCR-ALS to generate reliable chromatographic initial guesses for the three-way PARAFAC step of the semi-automated alignment method.

As mentioned above, a more direct comparison between the % RSDs obtained by the semi-automated alignment method can be made by comparing the summed areas from the IKSFA-ALS-ssel with unimodality method. Both methods were constrained as closely as possible to one another (the chromatographic selectivity constraints were not possible due to the limitations of the implementation of the IKSFA-ALS-ssel with unimodality method) and the components were split at the same points in every sample. The % RSD from the simple summation method using the IKSFA-ALS-ssel method with unimodality was calculated to be 11.0 and 10.3 for Peaks 11 and 12 respectively, Table 11. The large difference between the simple summation method and PARAFAC can be attributed to two factors. First, due to the data reconstruction used in the semi-automated alignment method, much more noise is eliminated using this method. This can be seen in the fit errors of the two methods; 2.51% for the IKSFA-
ALS-ssel-unimod method and 11.9 % for the semi-automated alignment method. The most likely reason for this difference in the fit errors is due to the discarding of the error matrices from the MCR-ALS and three-way PARAFAC analysis for each individual sample. The second factor was the ability of the semi-automated alignment method to implement an additional constraint beyond what was possible in the IKSFA-ALS-ssel with unimodality method. The unimodality constraint, as implemented by the IKSFA-ALS-ssel method with unimodality, simply splits components into two or more components, thereby preserving each individual analyte. However, the unimodality constraint used by the semi-automated alignment method is the more traditional version where only one maximum is permitted in each component. In its current form the IKSFA-ALS-ssel method with unimodality is unable to implement this version of unimodality due to the augmented nature of the chromatographic dimension. This limitation of the IKSFA-ALS-ssel with unimodality method results in small contributions of background remaining within the resolved analyte component.

**6.4.3 Phenytoin data set**

The phenytoin data set analyzed in this paper has been previously analyzed by Groskreutz et al. [48] and discussed in more detail by Bailey et al. [150] with reported concentrations for the average phenytoin concentration in waste water samples of 42 ± 1 ppb from the standard addition curve, 36 ± 1 ppb from the direct calibration curve (both calculated by manually integrating MCR-ALS-ssel components) and 42 ± 19 ppb by LC-LC MS/MS (all error ranges are given at 95 % confidence intervals). These calculated concentrations were similar to those obtained by directly summing the MCR-ALS-ssel components after careful application of the chromatographic selectivity constraint. The summed MCR-ALS components resulted in
32±2 ppb and 37±7 ppb for the direct and standard addition calibration curves, as shown in Table 12 (errors listed are based on a 95% confidence interval). The results obtained by the semi-automated alignment method differed depending on the type of fitting used, Gaussian or EMG. If the first dimension was fitted using a Gaussian curve then the calculated concentrations of the unknown were 23±5 ppb and 26±3 ppb from the direct calibration and standard addition curves, respectively, Table 12. If the first dimension was fitted using an EMG curve then the calculated concentrations of the unknown were 28±5 ppb and 31±3 ppb from the direct calibration and standard addition curves respectively, as shown in Table 12.

Table 12: Direct and standard calibration curves for phenytoin

<table>
<thead>
<tr>
<th></th>
<th>Linear regression parameters</th>
<th>R²</th>
<th>Standard error (s_y)</th>
<th>Concentration (ppb)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Direct calibration</strong></td>
<td>Semi-automated alignment with Gaussian fitting</td>
<td>Slope = 0.0060(0.0003) Intercept = 0.00(0.03)</td>
<td>0.976</td>
<td>0.055</td>
</tr>
<tr>
<td></td>
<td>Semi-automated alignment with EMG fitting</td>
<td>Slope = 0.0060(0.0003) Intercept = -0.00(0.02)</td>
<td>0.979</td>
<td>0.051</td>
</tr>
<tr>
<td></td>
<td>IKSFA-ALS-ssel b</td>
<td>Slope = 0.0022(0.0003) Intercept = 0.07(0.03)</td>
<td>0.994</td>
<td>0.022</td>
</tr>
<tr>
<td><strong>Standard addition</strong></td>
<td>Semi-automated alignment with Gaussian fitting</td>
<td>Slope = 0.0056(0.0002) Intercept = 0.14(0.02)</td>
<td>0.989</td>
<td>0.030</td>
</tr>
<tr>
<td></td>
<td>Semi-automated alignment with EMG fitting</td>
<td>Slope = 0.0054(0.0002) Intercept = 0.17(0.02)</td>
<td>0.989</td>
<td>0.030</td>
</tr>
<tr>
<td></td>
<td>IKSFA-ALS-ssel b</td>
<td>Slope = 0.0027(0.0007) Intercept = 0.44(0.06)</td>
<td>0.940</td>
<td>0.070</td>
</tr>
</tbody>
</table>

a calculated concentrations were compared to the LC-LC MS/MS concentration of 42±19 ppb obtained by Groskreutz et al. [48]
b calculated using the simple summation method
A possible reason for the increase in calculated concentrations from the EMG fitting compared to the Gaussian fitting is an approximate average factor of 1.23 increase in the fitted 1D areas from the EMG fitting compared to the Gaussian fitting. While not an overly large increase in fitted areas, the 1.23 factor increase in fitted areas correlates to an approximate increase in the calculated concentrations for both the direct calibration and standard addition curves. Regardless of which 1D peak model is used, the resulting calculated concentrations of phenytoin are still lower than the reported literature concentrations. The lower concentrations may be due to several challenges that were encountered during the analysis of the phenytoin data set.

The first challenge was the irregular shifting of the phenytoin peak in both the second and first dimension. The irregular shifting in the second dimension was compounded by the presence of noise in the three-way PARAFAC 1D component for one of the unknown replicate samples. The irregular shifting in the first dimension resulted in different apparent peak shapes which were then fitted to either a Gaussian or EMG curve. Neither the Gaussian nor the EMG curves were able to completely reproduce the exact structure of the 1D peak. The introduction of small discrepancies between replicate samples, during the fitting step of the semi-automated alignment method, did not produce calibration curves as linear as those reported by Groskreutz et al. (R² = 0.988 for the standard addition and R² = 0.998 for the direct calibration) [48], as shown in Table 12. In addition to the irregular shifting of retention times between samples in the second and first dimensions, retention time shifting was observed within some of the samples. While an attempt was made to correct for these shifts, some information loss most likely occurred during the three-way PARAFAC step due to the enforcement of trilinearity. The third challenge was the presence of the interferent which coeluted very closely with the phenytoin peak in the spiked
waste water treatment samples, Fig. 38A. After the semi-automated alignment method analysis was concluded, the resolved spectral signature of the phenytoin component and interferent component were compared. The resulting correlation coefficient of 0.9687 indicated why the semi-automated alignment method had difficulty separating the two peaks into different components, Fig. 38B. Once again, partial compensation for this challenge was achieved by selective use of a chromatographic selectivity constraint in the second and first dimensions, Figs. 38C and 38D. Given the low intensity of the phenytoin peak in the unknown samples, as well as the noisy nature of the resolved component, it is entirely possible that some of the phenytoin component was included in the interferent component. While the calculated concentrations from using the semi-automated alignment method were lower than those obtained using MCR-ALS-ssel, the semi-automated alignment method did produce better calculated concentrations in comparison to the application of four-way PARAFAC analysis to the unaligned data set.

6.5 Conclusion

The semi-automated alignment method was successfully used to align and subsequently quantify peaks within four simulated and two experimental data sets. The semi-automated alignment method consistently provided % recoveries closest to 100 % compared to summing MCR-ALS-ssel components. While the % RSD for peak 11 from the urine data set was not as low as previous manual integration results from Bailey et al. and Tistaert et al., the % RSD for peak 12 was slightly lower than the previously reported literature values. Also, the % RSDs obtained were reproducible and because the user was not required to manually draw a baseline under the peaks. The semi-automated alignment method was also successfully demonstrated to be able to divide a single component into two spectrally similar components without a significant
degradation of the results. The semi-automated alignment method, however, was found to experience difficulty when analyzing data sets containing coeluting peaks with highly similar spectra. This difficulty was able to be overcome with careful use of the second dimension and spectral selectivity constraints. Because the peaks were able to be resolved into individual components, the calculated concentrations obtained from the semi-automated alignment method were within the error range of the LC-LC MS/MS results. The results obtained from the analysis of the phenytoin data set by the semi-automated alignment method were also found to be reproducible assuming that the selected peak positions when using the within sample alignment were reproducible. Small differences in the selection of the peak positions resulted in only small changes to the final concentrations (less than 0.5 ppb).
CHAPTER 7: Comparison of Three PARAFAC Based Methods

The focus of the work in this chapter is to compare the method developed in Chapter 6, the semi-automated alignment method (SAAM) [94], with methods already described in the literature, PARAFAC2 [4] and SCP [5], which are readily available to interested researchers.

7.1 Introduction

One of the alternatives to pre-aligning chromatographic data sets prior to using GRAM or PARAFAC, which has been previously described in Chapter 3, section 3.4, is PARAFAC2 [4, 85]. PARAFAC2 allows peaks to shift between chromatograms by relaxing the multilinearity requirement on the dimension containing the shifting data. Skov *et al.* compared the performance of PARAFAC (after retention time alignment by linear shift followed by cross correlation) and PARAFAC2 for GC×GC-TOFMS data [86]. Skov determined that PARAFAC was more robust for peaks with lower signal-to-noise ratios and lower concentrations. PARAFAC2, however, did eliminate the need for retention time alignment prior to analysis. That study focused on peaks that were fully resolved and did not study overlapped peaks. Van Mispelaar *et al.* also compared PARAFAC, PARAFAC2 and integration by summing directly for GC×GC-FID chromatograms [138]. Van Mispelaar aligned the chromatograms prior to PARAFAC analysis by aligning the chromatograms to a standard injection and determined the proper degree of retention shift by calculating the inner product correlation. Van Mispelaar determined that PARAFAC2 overestimated the concentrations when compared to PARAFAC.
Also, van Mispelaar concluded that differences in peak shapes were more detrimental to the PARAFAC2 analysis.

Aside from the studies described in this chapter, SCP has not yet been applied to chromatographic data. However, during the development of SFA, Hong and Harshman [89] applied SFA to the same chromatographic data set used by Bro et al. [85] during the development of PARAFAC2. The chromatographic data utilized by Hong and Harshman was a four-way LC-fluorescence data set consisting of a chromatographic dimension, an excitation dimension, an emission dimension, and a sample dimension. Non-negativity was applied to every dimension but the chromatographic dimension. Non-negativity was performed using the fast non-negativity constraint by Bro and De Jong [121], which implements non-negativity during the ALS step of the algorithm.

While Hong and Harshman did not include a figure showing a direct comparison between the resolved chromatographic profiles from PARAFAC2 and SFA, they included in their paper a figure showing the differences in the resolved spectral (both excitation and emission) profiles, shown in Fig. 39. As can be seen in Fig. 39, the four-way SFA approach resulted in two components possessing the same shape for both the excitation and emission spectra. Hong and Harshman attribute the inability of SFA to match the PARAFAC2 profiles to the possibility that the solutions were local minima. Two explanations were given to explain the possible convergence to local minima. First, the number of random starts may have been insufficient to allow for complete convergence to the global minimum. Second, the convergence criterion for SFA may have been set too shallow and a more strict convergence criterion may have allowed for the correct profiles to be obtained. However, Hong and Harshman did not try a stricter
convergence criterion due to the lengthy analysis time required to reach the results shown in Fig 39.

![Excitation and Emission Spectra](image)

**Fig. 39.** Plots of the excitation (left column) and emission (right column) spectra from the four-way data set used by Hong and Harshman to compare SFA to PARAFAC2. The symbols F1-4 indicate the component numbers. Reproduced from reference [89] with permission from Wiley.

In order to account for these restrictions, Hong and Harshman reran SFA using the previously determined shifts and obtained results analogous to the PARAFAC2 results. This indicated that the previous results were indeed simply due to local minima. Therefore, Hong and Harshman concluded that SFA could produce the same results as PARAFAC2 provided a sufficient convergence criterion is used.

While PARAFAC2 and SFA have been applied to 1D chromatographic data, they have not yet been applied to 2D chromatographic data with the possibility of shifts in two dimensions. Therefore, the following chapter will examine the ability of SAAM, PARAFAC2, and SCP
(instead of SFA due to time and memory limitations) to handle both simulated and real world 2D chromatographic data.

7.2 Experimental

All calculations and data analysis were carried out using Matlab version 7.12.0.635 (R2011a, Mathworks, Inc., Natick, MA) on a Lenovo Win 7 PC laptop with an Intel Core i5-2410 M @ 2.30 GHz and 6.00 GB of RAM. The SAAM function was an in-house developed function as described in Chapter 6. The PARAFAC2 function was purchased from Eigenvector Research as part of the PLS_toolbox (release 6.5.2). The SCP function was downloaded from www.erpwavelab.org on Feb 22, 2012.

7.2.1 PARAFAC based methods

SAAM, PARAFAC2, and SCP were performed using a maximum of 500 iterations with a convergence criterion of $1 \times 10^{-6}$. SAAM was performed by making full use of the constraints available within the design of the approach. PARAFAC2 was performed unconstrained due to the discovery that the order of components changed throughout the course of the analysis. For example, a given peak might begin as component 2 but by the end of the analysis this peak would be found in component 5. As such, this rearrangement of component orders made it difficult to correctly apply the constraints to the data. SCP was also performed unconstrained due to the discovery that if the built in non-negativity constraint was utilized then it appeared that SCP would enter into an infinite loop between the first and second iterations. In addition except for certain simulations, PARAFAC2 and SCP were both configured to allow the second dimension to shift between samples.
Instead of using the pre-built options for generating the initial guesses, MCR-ALS was used to generate the initial guesses for PARAFAC2 and SCP since the same approach is utilized for SAAM. MCR-ALS was initialized using IKSFA, if background correction had not been performed prior to analysis and by IOPA if background correction had been performed prior to analysis. The rationale behind this was the determination that IOPA-initialized MCR-ALS resulted in less over fitting in the resolved chromatograms compared to IKSFA, if the background had been previously removed.

7.2.2 Background correction

Background correction was performed using the OBGC technique with the median filter function. The filter window width was set to 45 as was previously determined in Chapter 4.

7.2.3 Data sets

7.2.3.1 Phenytoin

The same phenytoin data set analyzed in Chapter 6 to validate SAAM was also analyzed by PARAFAC2 and SCP. This experimental data set included two calibration curves, a direct calibration, consisting of four levels, and a standard addition calibration containing an unknown amount of phenytoin in a waste water treatment sample as well as four standard addition levels. PARAFAC2 and SCP were conducted assuming that a seven component model was sufficient to correctly explain the data, as was shown in Chapter 6 with SAAM.
7.2.3.2 Simulations

A series of simulations were conducted to test the capability of SAAM, SCP, and PARAFAC2 to handle retention time shifting in either one or two dimensions. The variables (peak area, degree of retention time shift, peak widths, and background) used to generate the simulations were based on the phenytoin data set obtained from the Stoll research group at Gustavus Adolphus College in Saint Peter, MN and previously analyzed in Chapter 6. The phenytoin peak was observed to shift in the first and second dimensions between samples, as well as within the second dimension for individual samples. In order to best emulate the experimental data set, the simulations were constructed using the same type and order for the samples, i.e., a direct calibration curve followed by a standard addition calibration curve. The analysis time for the first dimension was 12 s with a sampling time of 2 s for a total of six second dimension injections. The analysis time of the second dimension was 20 s although the time scale was shortened to 11.25 s in accordance with the region previously analyzed by SAAM.

Since the experimental data set only provided two blanks samples to use as backgrounds for the simulations, twenty backgrounds were constructed by proportionally (with the sum of the proportions equal to 1) combining the two experimental backgrounds to make a single simulated background. The simulated experimental backgrounds then provided a sufficient amount of chemical noise so that random noise was not introduced into the simulation. However, this method of generating the background is unable to account for the matrix effect of the actual waste water treatment samples. Given that phenytoin was the analyte of interest in the experimental data set, the simulations used the spectrum of phenytoin to produce the two-dimensional peak. Likewise, the sigmas of the Gaussian distribution in both the first (1.23 s) and second (0.72 s) dimensions resulted in a peak width comparable to phenytoin in the experimental
data set. From this basic simulation setup, illustrated in Fig. 40, four groups of simulations were investigated: a) no shifting in either the first or second dimension between samples, b) shifting in only the second dimension between samples, c) shifting in only the first dimension between samples, and d) shifting in both the first and second dimension between samples. The maximum amount of shifting (with the actual amount of shifting occurring determined through the use of the rand function in Matlab) allowed between samples varied from 0 s to 1.00 s in 0.25 s increments in both dimensions. Fig. 41A illustrates the mean Euclidian distance from the non-shifted simulation for each combination of first and second dimension shifts. The mean Euclidian distance was calculated according to Eq. 49,

$$d_k = \frac{\sum_k \left[ (1 \cdot t_{R,ref} - 1 \cdot t_{R,k})^2 + (2 \cdot t_{R,ref} - 2 \cdot t_{R,k})^2 \right]}{K}$$

(49)

where \(d_k\) is the average Euclidian distance for the \(k^{th}\) simulation, \(1 \cdot t_{R,ref}\) is the \(1 \cdot t_R\) of the non-shifted simulation, \(2 \cdot t_{R,ref}\) is the \(2 \cdot t_R\) of the non-shifted simulation, \(1 \cdot t_{R,k}\) is the \(1 \cdot t_R\) of the \(k^{th}\) sample for a given simulation, and \(2 \cdot t_{R,k}\) is the \(2 \cdot t_R\) of the \(k^{th}\) sample for a given simulation. In addition, to the impact of the maximum degree of shifting for a given simulation relative to the non-shifted simulation on the shape of the peak was determined. The correlation coefficient between the non-shifted simulation and a given simulation was calculated for both the first and second dimension peaks. The products of the obtained first and second dimension correlation coefficients were then averaged to produce a mean correlation coefficient, as shown in Fig. 41B. As can be seen in Fig. 41B, the mean correlation coefficient remains relatively constant if shifting only occurs in the second dimension. However, when the first dimension is also allowed to shift, the mean correlation coefficient rapidly changes. This is due to the undersampling of the first dimension and the resulting change in apparent peak shape, as described in Chapter 5.
Fig. 40: Schematic setup of the simulations performed in order to compare the ability of SAAM, SCP, and PARAFAC2 to handle retention time shifting. The letters a-d denote: a) non-shifted simulation; b) shifting only in the second dimension; c) shifting only in the first dimension; d) shifting in both the first and second dimensions.

Fig. 41: A) A contour plot of the mean Euclidian distance from 4.29 s in the first dimension and 14.35 s in the second dimension. The mean Euclidian distance was calculated from the first and second dimension retention times for eighteen samples for each simulation. B) A contour plot of the mean correlation coefficient calculated as described in the text.
In addition to the shifting in the first and second between samples, shifting within the second for a given sample was also investigated. The previously described four simulation experiments were repeated allowing for the possibility of a maximum of 0.2 s shift either earlier or later (in an oscillating pattern to account for the shunt typically found in the sampling device, as explained in Chapter 2) for each 2D chromatogram within a sample.

7.2.3.3 Quantitation study

An experimental quantitation study was conducted using the instrument with the splitter sampling device design, described in Chapter 2 section 2.6, at the Carr research lab at the University of Minnesota.

7.2.3.3.1 Chemicals

Chromatographic grade water was from Sigma-Aldrich (St. Louis, MO) and acetonitrile was obtained from J.T Baker (Phillipsburg, NJ, USA). Reagent grade perchloric acid was purchased from Mallinkrodt Baker (Paris, Kentucky, USA). All materials were used as received. All mobile phases and dilution solutions were prepared gravimetrically (± 0.01 g) and used without any further filtration. All standards were purchased from Sigma-Aldrich as described previously by Stoll et al. and Filgueira et al. [46, 151].

7.2.3.3.2 Sample preparation

The data consisted of two sets of calibration curves, consisting of seven points each, and four quality control (Q1-Q4) samples. Each point in the calibration curve was injected in quadruplicate and the quality control samples were injected in quintuplicate. The samples were
prepared by spiking an appropriate amount of stock or working standard into a 0.2 mL vial insert and then filling to 0.2 mL using a 50/50 mixture of water/ACN. The compounds analyzed and their corresponding concentration ranges were 5-hydroxy-tryptamine (4.2 to 42 µg/mL), 3-chloropyridine (20 to 200 µg/mL), anthranilic acid (6.78 to 67.8 µg/mL), 2-bromopyridine (27.5 to 275 µg/mL), indole-3-acetyl-L-lysine (10.8 to 108 µg/mL), indole-3-acetyl-L-glycine (8.8 to 88 µg/mL), 4-indolyl acetate (10.8 to 108 µg/mL), and indole-3-acetonitrile (7.5 to 75 µg/mL). These compounds were chosen due to potential chromatographic overlap (5-hydroxy-tryptamine/3-chloropyridine and 2-bromo pyridine/indole-3-acetyl-L-lysine) or spectral overlap (indole-3-acetyl-L-lysine/indole-3-acetyl-L-glycine and 4-indolyl acetate/indole-3-acetonitrile). A chromatogram at 220 nm is shown in Fig. 42, with the position of each compound indicated.

**Fig. 42:** A contour plot of the chromatogram of the first replicate of the lowest calibration standard at 220 nm of the quantitation data set. The red numbers indicate the peaks that were quantified. The numbers correspond to the following compounds: 1) 5-hydroxy-tryptamine; 2) 3-chloropyridine; 3) anthranilic acid; 4) 2-bromopyridine; 5) indole-3-acetyl-L-lysine; 6) indole-3-acetyl-L-glycine; 7) 4-indolyl acetate; and 8) indole-3-acetonitrile.
7.2.3.3 Instrument configuration

The mobile phase used for the first dimension was a 10 mM perchloric acid solution in water for mobile phase A and neat ACN for mobile phase B. The mobile phase used for the second dimension was a 10 mM perchloric acid/10 mM phosphoric acid solution in water for mobile phase A and neat ACN for mobile phase B. A 2.1 mm × 100 mm Zorbax SB-C3 column packed with 3.5 µm particles was used in the first dimension. The 1D gradient was 5 % B at 0 min, 29 % B at 8 min, 0 % B at 8.01 min, with a total analysis time of 12 min. The 1D flow rate was 0.5 mL/min, and the column was maintained at a temperature of 40 °C. A 2.1 mm × 33 mm in-house packed column with 2.1 µm superficially porous carbon clad silica particles was used in the second dimension. The 2D gradient was 0 % B at 0 min, 100 % B at 0.15 min, 0 % B at 0.16 min, with a total analysis time of 0.2 min. The 2D flow rate was 3 mL/min and the column was maintained at a temperature of 85 °C. A splitter pump was used, as shown in Fig. 7 in Chapter 2, at a flow rate of 0.1 mL/min resulting in 20 % of the 1D effluent being transferred to the second dimension.

7.2.3.4 Maize recovery

An experimental study was developed to test the ability of MCR-ALS, SAAM, PARAFAC2, and SCP to accurately and precisely resolve a standard compound from a complex sample. The complex sample was the same maize sample previously used in Chapter 4 for the background correction study.
7.2.3.4.1 Chemicals

The chemicals used to prepare the mobile phases and dilution solutions were the same as prepared in section 7.2.3.3.1. In addition the mobile phases and dilution solutions were also prepared as described in section 7.2.3.3.2.

7.2.3.4.2 Sample preparation

A four point calibration curve was prepared in water for thirteen standard compounds (5-hydroxy-tryptamine, indole-3-acetyl-L-glutamic acid, tryptophan, anthranilic acid, indole-3-acetyl-L-glycine, indole-3-acetyl-L-lysine, indole-3-carboxylic acid, indole-3-propionic acid, indole-3-ethanol, indole-3-butyric acid, indole-3-acetonitrile, indole-5-carbonitrile). In addition, the recovery samples (designated as M1-M4) were prepared by spiking 50 µL of a given calibration solution into 50 µL of maize seedling extract. A dilute maize sample (designated as M0) was prepared by diluting 50 µL of the maize extract with 50 µL of a 50/50 water/ACN solution. While thirteen compounds were spiked into the sample, anthranilic acid is the only compound that will be discussed in this chapter, and the corresponding chromatographic region is shown in Fig. 43. This is due to the observation that the rest of the standard compounds did not elute near maize peaks, and were therefore not interesting to investigate. The anthranilic acid standard was prepared using a concentration range from 8.5 to 33.9 µg/mL. Each calibration sample was injected in quadruplicate and each maize sample was injected in triplicate.
Fig. 43: Chromatograms of the maize spiked standards used in the maize recovery study. A) Low level anthranilic acid calibration point from the first replicate; B) Diluted maize sample from the first replicate; C) Resulting low level anthranilic acid spiked maize sample from the first replicate.

7.2.3.4.3 Instrument configuration

The mobile phases used in this data set were the same mobile phases previously described in section 7.2.3.3.2. A 2.1 mm × 300 mm Zorbax SB-C3 column (achieved by placing a 250 mm and 50 mm column in series) packed with 3.5 µm particles was used in the first dimension. The 1D gradient was 0 % B at 0 min, 50 % B at 25 min, 0 % B at 25.01 min, with a total analysis time of 30 min. The 1D flow rate was 0.38 mL/min and the column was maintained at a temperature of 40 °C. A 2.1 mm × 33 mm in-house packed column with 2.1 µm superficially porous carbon clad silica particles was used in the second dimension. The 2D gradient was 0 % B at 0 min, 100 % B at 0.2 min, 0 % B at 0.21 min, with a total analysis time of 0.25 min. The 2D flow rate was 3 mL/min and the column was maintained at a temperature of 90 °C. A splitter pump was used, as shown in Fig. 7 in Chapter 2, at a flow rate of 0.1 mL/min resulting in 26.3% of the 1D effluent being transferred to the second dimension.
7.3 Results and Discussion

7.3.1 Phenytoin

In the previous chapter, the phenytoin data set was used as a real world validation of SAAM to ensure that the results produced by SAAM were both accurate and precise. However, the calculated concentrations from both the direct and standard addition calibration curves were lower than the calculated concentrations reported by Bailey et al. [150]. The concentrations from SAAM were within the experimental range given by LC/MS analysis of the phenytoin samples. As such, there was still some doubt which was the best estimate of the true concentration. Therefore, the phenytoin data set was also analyzed using PARAFAC2 and SCP. Table 13 shows the results for the direct calibration and standard addition based analysis for SAAM, PARAFAC2, and SCP. The resolved intensities for SAAM, PARAFAC2, and SCP for the phenytoin component were normalized prior to linear regression by dividing each value by the associated maximum value from each method. Given that the SAAM results were discussed in the previous chapter in comparison to the MCR-ALS results, the PARAFAC2 and SCP results in Table 13 will be discussed in comparison to the SAAM results.

Looking at the linear regression parameters column, the slopes from the direct calibration and standard addition curves for SCP are very similar to those from the SAAM analysis. However, the slopes from PARAFAC2 are a lot less than those for SAAM. The end result of this trend can be seen in the concentration column. SCP produces concentrations very similar to SAAM while PARAFAC2 produces concentrations outside the range of the original LC/MS estimate. The reason for this large increase for PARAFAC2 can be seen in Fig. 44A and 44B. Unlike the SAAM results shown in Figs. 39C and 39D, or the SCP results shown in Figs. 44C and 44D, PARAFAC2 was not able to separate the interferent peak from phenytoin. This
inability to resolve the two peaks into two separate components explains why the calculated concentrations are much higher since they contain both the primary interferent as well as a smaller secondary interferent. An attempt was made to reduce the number of components on the chance that overfitting may be occurring but PARAFAC2 was still not able to resolve the two peaks into two different components. Likewise, increasing the number of components did not produce a successful result. While the phenytoin component was resolved from the interferent, the addition of the extra components resulted in the phenytoin component being represented by more than one component. Therefore a series of simulations were conducted to determine if the spectral similarity between the interferent and phenytoin was responsible for the poor resolution results.

The simulations were carried out using the same general procedure as described in section 7.2.3.2 except that the peaks were not allowed to shift. The spectra used for the interferent and phenytoin peaks were obtained from the four-way PARAFAC analysis by SAAM as described in Chapter 6. The resolution was allowed to vary from 0 to 1.5. Initial guesses for PARAFAC2 were generated by IKSFA, because IOPA failed to provide adequate initial guesses for $R_s$ values less than 0.4. The simulations showed that PARAFAC2 did not have any trouble separating the two peaks into the proper components regardless of the $R_s$. However, this series of simulations did not contain the same background in the so-called “waste water treatment plant effluent” samples as was found in the original phenytoin data set. It may be that the influence of the waste water background caused PARAFAC2 to resolve both peaks into a single component. Additional simulations need to be considered to see if retention time shifting may have contributed to this result. In addition, an experiment needs to be devised to check whether the shape preserving constraint in PARAFAC2 is responsible for these results.
Table 13: Calculated concentrations (with associated errors given according to a 95% confidence interval) for the phenytoin data set by SAAM, PARAFAC2, and SCP\(^a\)

<table>
<thead>
<tr>
<th>Direct calibration</th>
<th>Linear regression parameters</th>
<th>(R^2)</th>
<th>Standard error ((s_y))</th>
<th>Concentration (ppb)</th>
</tr>
</thead>
<tbody>
<tr>
<td>SAAM with Gaussian fitting(^b)</td>
<td>(m = 0.0060\pm0.0003) (b = 0.00\pm0.03)</td>
<td>0.976</td>
<td>0.055</td>
<td>23(\pm)5</td>
</tr>
<tr>
<td>SAAM with EMG fitting(^b)</td>
<td>(m = 0.0060\pm0.0003) (b = 0.00\pm0.02)</td>
<td>0.979</td>
<td>0.051</td>
<td>28(\pm)5</td>
</tr>
<tr>
<td>PARAFAC2 (7 component model)</td>
<td>(m = 0.0043\pm0.0002) (b = 0.02\pm0.01)</td>
<td>0.987</td>
<td>0.029</td>
<td>93(\pm)4</td>
</tr>
<tr>
<td>PARAFAC2 (4 component model)</td>
<td>(m = 0.00460\pm0.0005) (b = 0.001\pm0.004)</td>
<td>0.999</td>
<td>0.009</td>
<td>74(\pm)1</td>
</tr>
<tr>
<td>SCP</td>
<td>(m = 0.0061\pm0.0004) (b = -0.02\pm0.03)</td>
<td>0.964</td>
<td>0.068</td>
<td>26(\pm)6</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Standard addition</th>
<th>Linear regression parameters</th>
<th>(R^2)</th>
<th>Standard error ((s_y))</th>
<th>Concentration (ppb)</th>
</tr>
</thead>
<tbody>
<tr>
<td>SAAM with Gaussian fitting(^b)</td>
<td>(m = 0.0056\pm0.0002) (b = 0.14\pm0.02)</td>
<td>0.989</td>
<td>0.03</td>
<td>26(\pm)3</td>
</tr>
<tr>
<td>SAAM with EMG fitting(^b)</td>
<td>(m = 0.0054\pm0.0002) (b = 0.17\pm0.02)</td>
<td>0.989</td>
<td>0.03</td>
<td>31(\pm)3</td>
</tr>
<tr>
<td>PARAFAC2 (7 component model)</td>
<td>(m = 0.0038\pm0.0002) (b = 0.40\pm0.01)</td>
<td>0.981</td>
<td>0.07</td>
<td>105(\pm)4</td>
</tr>
<tr>
<td>PARAFAC2 (4 component model)</td>
<td>(m = 0.0037\pm0.0004) (b = 0.34\pm0.04)</td>
<td>0.924</td>
<td>0.06</td>
<td>91(\pm)8</td>
</tr>
<tr>
<td>SCP</td>
<td>(m = 0.0058\pm0.0002) (b = 0.14\pm0.01)</td>
<td>0.995</td>
<td>0.02</td>
<td>24(\pm)2</td>
</tr>
</tbody>
</table>

\(^a\) calculated concentrations were compared to the LC-LC MS/MS concentration of 42\(\pm\)19 ppb obtained by Groskreutz et al. [48]

\(^b\) results previously reported by Allen and Rutan [94].
Fig. 4: Contour plots of the resolved 2D chromatograms for A) the first replicate of the unspiked waste water treatment sample using a seven component PARAFAC2 model (phenytoin component); B) the first replicate of the unspiked waste water treatment sample using a four component PARAFAC2 model (phenytoin component); C) the phenytoin component from a seven component SCP model; and D) the interferent component from a seven component SCP model.

7.3.2 Simulations

7.3.2.1 Inter-sample shifting only

A series of simulations were conducted to determine the ability of SAAM, PARAFAC2, and SCP to handle retention time shifting in either the second dimension, the first dimension, or both. Since the simulations were originally designed to mimic the phenytoin data set, the results obtained by the simulations were calculated according to both a direct calibration curve and a standard addition calibration using two replicate “unknown” samples. In addition, the areas used to generate the 1D peaks were used in place of actual concentrations for each of the calibration curves. As such, the 1D peak areas were compared to the reference 1D peak area value of 42 (the
area used to generate the “unknown” samples). The minimum, maximum, mean, and median calculated $^1D$ peak areas and standard deviations from the twenty-five simulations illustrated in Fig. 40 are shown in Table 14, for both the direct and standard addition calibration curves. The results produced by the direct calibration curve indicated that SAAM, PARAFAC2, and SCP were able to produce approximately the same $^1D$ peak area regardless of the amount of shifting present in the first and second dimensions. However, the standard deviations produced by the three methods were impacted by the degree of retention time shifting. While SAAM and SCP produced low standard deviations ($s_x$ was equal to or less than 1.0), PARAFAC2 produced a standard deviation of 4.6 during the simulation with the greatest allowed degree of retention time shifting in both the first and second dimensions. The increase in the standard deviation from PARAFAC2 matches a previous observation by van Mispelaar [138]. van Mispelaar observed that PARAFAC2 was very susceptible to peak shape changes. In these simulations, the $^1D$ peak shape changed rapidly due to the small $t_s$ used relative to the $^1D$ retention time shifting. Therefore, it is highly unlikely that the same degree of $^1D$ peak shape change would occur in normal LC×LC-DAD data. In the standard addition calibration curve, a very similar trend is observed between SAAM, PARAFAC2, and SCP with slight differences due to different retention time shifting patterns. SAAM provided the most consistent performance, while SCP was slightly more erratic than before. PARAFAC2 once again resulted in the largest standard deviation and greatly underestimated the $^1D$ peak area during the simulation with the largest allowed shift in both dimensions.
Table 14: The calculated simulated phenytoin $^1$D peak area and corresponding standard deviation from calibration curves with inter-sample shifting after analysis by SAAM, PARAFAC2, or SCP

<table>
<thead>
<tr>
<th>Method</th>
<th>Direct Calibration Curve</th>
<th>Standard Addition Calibration Curve</th>
</tr>
</thead>
<tbody>
<tr>
<td>Calculated $^1$D peak area</td>
<td>Min.$^a$</td>
<td>Max.$^b$</td>
</tr>
<tr>
<td>SAAM</td>
<td>41.9</td>
<td>42.5</td>
</tr>
<tr>
<td>PARAFAC2</td>
<td>41.8</td>
<td>42.3</td>
</tr>
<tr>
<td>SCP</td>
<td>41.8</td>
<td>42.4</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SAAM</td>
<td>42.1</td>
<td>42.7</td>
</tr>
<tr>
<td>PARAFAC2</td>
<td>32.9</td>
<td>42.9</td>
</tr>
<tr>
<td>SCP</td>
<td>41.3</td>
<td>44.9</td>
</tr>
</tbody>
</table>

7.3.2.2 Inter- and intra-sample shifting

To further test the limits of SAAM, PARAFAC2, and SCP, intra-sample shifting was introduced in addition to the same shifting patterns previously investigated. The intra-sample shifting was not corrected prior to chemometric analysis. The minimum, maximum, mean, and median calculated $^1$D peak areas and standard deviations from the twenty-five simulations illustrated in Fig. 40 incorporating intra-sample shifting are shown in Table 15, for both the direct and standard addition calibration curves. The same trends are observed as previously seen section 7.3.2.1 except the trends are further accentuated. The most probable explanation is that the intra-sample shifting led to chemometrically induced broadening of the $^2$D peaks. This perceived broadening is a result of how the intra-sample shifting was generated. The intra-sample shifting was designed to mimic the oscillation in the second dimension caused by the use of the shunt on the sampling device. As such, the profiles generated by SAAM, PARAFAC2, and SCP incorporated the intra-sample shifting by broadening the optimized resolved profiles.
Table 15: The calculated $^1$D peak area and corresponding error from calibration curves with inter- and intra-sample shifting after analysis by SAAM, PARAFAC2, or SCP

<table>
<thead>
<tr>
<th>Method</th>
<th>Direct Calibration Curve</th>
<th>Standard Addition Calibration Curve</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Min.</td>
<td>Max.</td>
</tr>
<tr>
<td>SAAM</td>
<td>40.3</td>
<td>43.6</td>
</tr>
<tr>
<td>PARAFAC2</td>
<td>40.8</td>
<td>42.4</td>
</tr>
<tr>
<td>SCP</td>
<td>41.5</td>
<td>42.2</td>
</tr>
<tr>
<td>SAAM</td>
<td>42.7</td>
<td>46.3</td>
</tr>
<tr>
<td>PARAFAC2</td>
<td>27.5</td>
<td>57.3</td>
</tr>
<tr>
<td>SCP</td>
<td>41.6</td>
<td>45.0</td>
</tr>
</tbody>
</table>

7.3.3 Quantitation study

Regions, illustrated in Fig. 45, were selected around each of the peak clusters shown in Fig. 42. Each region was designed to ensure that the peaks present in the region were not cut off in any of the samples. Each region was also selected to ensure that sufficient space surrounded to the peaks to allow for resolution of the background from the peaks. The % recoveries for 3-chloropyridine and 2-bromopyridine are reduced for Q1-Q3, possibly due to a precipitation of the analyte. Due to inconsistent results from SAAM, PARAFAC2, SCP, and MCR-ALS analysis, the peaks corresponding to 4-indolyl acetate and indole-3-acetonitrile (Region 3) are not included in this discussion.
Fig. 45: Illustration of the two regions of the quantitation study analyzed by SAAM, PARAFAC2, and SCP. Region 1 contained the peaks corresponding to 5-hydroxy-tryptamine, 3-chloropyridine, and anthranilic acid. Region 2 contained the peaks corresponding to 2-bromopyridine, indole-3-acetyl-L-lysine, and indole-3-acetyl-L-glycine. Region 3 contained the peaks corresponding to 4-indolyl-acetate and indole-3-acetonitrile.

7.3.3.1 Region 1

The calculated percent recoveries and associated standard deviations for 5-hydroxy-tryptamine, 3-chloropyridine, and anthranilic acid are shown in Tables 16 and 17. PARAFAC2 and SCP were found to produce comparable recoveries with low standard deviations, regardless of whether or not background correction was performed prior to chemometric analysis. However, SAAM was found to experience some problems with resolving the 5-hydroxy-tryptamine and 3-chloropyridine in samples when both compounds were present. Using the standard spectral matching threshold of 0.9, a linear calibration curve (with a $R^2$ value of 0.290) was produced. The primary reason for the low $R^2$ value was that 5-hydroxy-tryptamine was only detected in the third and fourth replicates of the C4 and C5 samples. An attempt was made to see if adjusting the spectral matching threshold would allow for a better resolution during four-way analysis. Sadly, this was not the case. While the recoveries improved, these values are deceptive because there are errors in the resolved spectral profiles, as shown in Fig. 46. Fig. 46A shows the sequenced $^2$D chromatograms from MCR-ALS for the first replicate from the first low
calibration point. As can be seen, the calculated $R_s$ is very small indicating that the two peaks might be considered linearly dependent [152]. It is a result of this linear dependency that causes the second dimension to be considered rank deficient. When MCR-ALS is performed on all samples simultaneously, the resolved spectra shown in Fig. 46B are obtained. However, when three-way PARAFAC is performed on the single sample, the resolved spectra in Fig. 46C are obtained. According to Bro et al, the impact of the linear dependency is that the dimension that is not linearly dependent begins to incorporate “noise” into the resolved profile. This might explain why the spectrum of 5-hydroxy-tryptamine (green line), Fig. 46C, is degraded compared Fig. 46B. In addition, because the second dimension is close to linearly dependent, the three-way PARAFAC loses its ability to generate a unique solution.

![Fig. 46](image)

**Fig. 46:** A) Plot of the resolved sequential 2D chromatograms obtained from MCR-ALS analysis of the first replicate from the low calibration point. The resolution between the two peaks was estimated as 0.13. B) Plot of the true spectra for 5-hydroxy-tryptamine and 3-chloropyridine with the calculated correlation coefficient for the two spectra. C) Plot of the 3-way PARAFAC analysis resolved spectra for 5-hydroxy-tryptamine and 3-chloropyridine with the calculated correlation coefficient from the first replicate of the low calibration point. The blue lines in all three panels correspond to 3-chloropyridine and the green lines in all three panels correspond to 5-hydroxy-tryptamine.

In order to better understand the poor resolution results from SAAM, a simulated data set was constructed to control the amount of $R_s$ between the two peaks. The $R_s$ was varied from 0 to 1.5. The simulations were conducted using the general procedure listed in section 7.2.3.2 except
that the peaks were not allowed to experience inter- or intra-sample shifting. A plot showing the recovered $^1$D peak areas is shown in Fig. 47. As expected, SAAM cannot resolve the peaks correctly around a $R_s$ of 0.1. However, this may not be a firm threshold. In the simulated data set, the two peaks were of comparable height and area. In the experimental data set, one of the two peaks (3-chloropyridine) absorbed much more strongly than the other and had a much wider base due to tailing. As a result, while the $R_s$ wasn’t very different than 0.2, the 5-hydroxytriptamine was not visible on the shoulder of the 3-chloropyridine peak. A similar trend regarding the inability of SAAM to resolve the two peaks was found after background correction was performed. In this case, the order of the components changed so some of the 5-hydroxytriptamine was matched to the anthranilic acid at the lower spectral matching thresholds. This accounts for the decreased recovery for the Q3 samples since anthranilic acid was not present in this sample. This suggests that the % recoveries obtained by SAAM prior to background correction being performed were simply due to the fact that the 5-hydroxy-triptamine components had not been matched yet. If the order had been reversed, it is entirely possible (indeed it is very probable) that the anthranilic acid components may have been assigned to the 5-hydroxy-triptamine reference spectra.

The other important conclusion to draw from Tables 16 and 17 is the impact of background correction on the recoveries of 5-hydroxy-triptamine using the simple summation method after MCR-ALS analysis. Due to the fact that 5-hydroxy-triptamine absorbs across the entire spectrum, it is not possible to utilize a spectral selectivity constraint. As a result, a significant amount of background is included in the calculated areas from the resolved chromatograms. This is shown by the middle panel of the left column of Fig. 48. However, after background correction by OBGC was performed, the background contributions to 5-hydroxy-
tryptamine are no longer present, as shown in the middle panel of the right column of Fig. 48. This indicates that the use of OBGC prior to MCR-ALS may prove beneficial if the simple summation method is used to quantify a given peak. However, there is a price to pay for performing OBGC prior to MCR-ALS analysis. As indicated in Fig. 49B, the resolved spectrum for 5-hydroxy-tryptamine does not have the same shape as the resolved spectra if background correction is not performed prior to MCR-ALS analysis, shown in Fig. 49A. The arrows indicate contributions from the 3-chloropyridine and anthranilic acid spectra into the 5-hydroxy-tryptamine spectra. The left arrow points to contamination from anthranilic acid and the right arrow points to contamination from 3-chloropyridine. Further research needs to be conducted to see if the manner in which OBGC is performed can be optimized to reduce or eliminate these errors.
Table 16: Percent recoveries and associated standard deviations for 5-hydroxy-tryptamine, 3-chloropyridine, and anthranilic acid without background correction prior to chemometric analysis

<table>
<thead>
<tr>
<th>Method</th>
<th>5-hydroxy tryptamine</th>
<th>3-chloropyridine</th>
<th>Anthranilic acid</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Q1</td>
<td>Q3</td>
<td>Q4</td>
</tr>
<tr>
<td>SAAM\textsuperscript{a}</td>
<td>236.7±0.9</td>
<td>96±0</td>
<td>48±0</td>
</tr>
<tr>
<td>SAAM\textsuperscript{b}</td>
<td>105.3±0.9</td>
<td>109±2</td>
<td>99.6±0.6</td>
</tr>
<tr>
<td>SAAM\textsuperscript{c}</td>
<td>88±1</td>
<td>93±3</td>
<td>94.5±0.6</td>
</tr>
<tr>
<td>PARAFAC2</td>
<td>100±1</td>
<td>103.4±0.8</td>
<td>101.2±0.8</td>
</tr>
<tr>
<td>SCP</td>
<td>94±1</td>
<td>93±2</td>
<td>96±1</td>
</tr>
<tr>
<td>MCR-ALS</td>
<td>43±6</td>
<td>40±4</td>
<td>79.9±0.7</td>
</tr>
</tbody>
</table>

\textsuperscript{a} using a spectral matching threshold of 0.9. $R^2=0.290$, $s_y=0.3$

\textsuperscript{b} using a spectral matching threshold of 0.8. $R^2=0.956$, $s_y=0.08$

\textsuperscript{c} using a spectral matching threshold of 0.7. $R^2=0.986$, $s_y=0.04$
Table 17: Percent recoveries and associated standard deviations for 5-hydroxy-tryptamine, 3-chloropyridine, and anthranilic acid with background correction prior to chemometric analysis

<table>
<thead>
<tr>
<th>Method</th>
<th>5-hydroxy tryptamine</th>
<th>3-chloropyridine</th>
<th>Anthranilic acid</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Q1</td>
<td>Q3</td>
<td>Q4</td>
</tr>
<tr>
<td>SAAM(^a)</td>
<td>411±1</td>
<td>80±0</td>
<td>40±0</td>
</tr>
<tr>
<td>SAAM(^b)</td>
<td>137.9±0.8</td>
<td>63±0</td>
<td>113.3±0.5</td>
</tr>
<tr>
<td>SAAM(^c)</td>
<td>93±1</td>
<td>7±0</td>
<td>95.2±0.8</td>
</tr>
<tr>
<td>SAAM(^d)</td>
<td>87±1</td>
<td>0±0</td>
<td>93.8±0.9</td>
</tr>
<tr>
<td>PARAFAC2</td>
<td>100±1</td>
<td>104.4±0.8</td>
<td>101.5±0.9</td>
</tr>
<tr>
<td>SCP</td>
<td>95±1</td>
<td>94±2</td>
<td>96±1</td>
</tr>
<tr>
<td>MCR-ALS</td>
<td>106.3±0.8</td>
<td>103±4</td>
<td>101.8±0.9</td>
</tr>
</tbody>
</table>

\(^a\) using a spectral matching threshold of 0.9. \(R^2=0.108, s_y=0.3\)
\(^b\) using a spectral matching threshold of 0.8. \(R^2=0.837, s_y=0.2\)
\(^c\) using a spectral matching threshold of 0.7. \(R^2=0.982, s_y=0.05\)
\(^d\) using a spectral matching threshold of 0.6. \(R^2=0.984, s_y=0.04\)
**Fig. 47:** A plot of the calculated $^1$D peak areas after 4-way PARAFAC analysis from SAAM using simulated data. The simulation was conducted using a $^1$D peak area of 42.

**Fig. 48:** Mesh plots of the resolved chromatograms from MCR-ALS analysis for 5-hydroxytryptamine, 3-chloropyridine, and anthranilic acid. The left column is the result obtained if background correction was not performed prior to MCR-ALS analysis. The right column is the result obtained if background correction was performed prior to MCR-ALS analysis.
Fig. 49: Plots of the resolved spectra if A) background correction was not performed prior to MCR-ALS analysis; B) background correction was performed prior to MCR-ALS analysis; C) An overlay of the resolved 5-hydroxy-tryptamine spectra if background correction was not performed prior to MCR-ALS analysis (solid line) and if background correction was performed prior to MCR-ALS analysis (dashed line). The colors in A, B, and C correspond to blue (3-chloropyridine), red (anthranilic acid), and purple (5-hydroxy-tryptamine). The arrows in (B) indicate points of difference in the resolved spectrum compared to (A).

7.3.3.2 Region 2

The calculated percent recoveries and associated standard deviations for indole-3-acetyl-L-lysine, 2-bromopyridine, and indole-3-acetyl-L-glycine are shown in Table 18. PARAFAC2 and SCP were found to produce comparable recoveries with low standard deviations regardless of whether or not background correction was performed for 2-bromopyridine. However, PARAFAC2 and SCP were not able to resolve indole-3-1-acetyl-lysine and indole-3-1-acetyl-glycine into separate components. This is illustrated in Fig. 50. In both cases, a portion of the 2-bromopyridine peak is present in the resolved chromatogram for indole-3-acetyl lysine and indole-3-acetyl-L-glycine. In addition, the use of OBGC prior to chemometric analysis was found to not greatly affect the calculated recoveries.

In contrast, SAAM experiences difficulty handling the background in the quantitation study in the case where OBGC was not performed. This is illustrated in Fig. 51. Fig 51A is a plot of the resolved 2D profiles (without implementing unimodality) if component splitting is not
performed. The resulting plot is very similar to the plot obtained when using SCP in Fig. 50B. When the component splitting is applied to every sample, the resolved chromatograms in Fig. 51B are obtained. The presence of two maxima for the indole-3-acetyl-L-lysine component is due to unimodality not being imposed in the final 4-way PARAFAC step during SAAM analysis. Unimodality was not imposed during this final step due to a fear that the background may have compromised the resolution in the final 4-way PARAFAC profile (i.e., background levels may have been higher than the indole-3-acetyl-L-lysine peak). In contrast, the indole-3-acetyl-L-glycine peak has been split into two components (green and cyan lines). This split is due to how SAAM matches components between samples. If a component was split, the spectral matching algorithm matches the spectra of the original component to the reference spectra. Since the background interfered with the spectra of the original component in some of the samples, the sample matching algorithm was unable to correctly match the indole-3-acetyl-L-glycine peak for some of the samples. Therefore, during the 4-way PARAFAC analysis step of SAAM, the imposition of sample selectivity on the indole-3-acetyl-L-glycine component resulted in some samples being transferred to the background component. As a result of these problems, the % recoveries for indole-3-acetyl-L-lysine and indole-3-acetyl-L-glycine were rather poor (results not shown).

In order to accommodate the presence of the background, an additional component was included in the MCR-ALS and 3-way PARAFAC analysis for the impacted samples (in essence these samples were deliberately being overfitted). The resulting profiles shown in Fig. 51C are the result of adding an extra component into the analysis of those samples that were impacted by the background. The % recoveries from this attempt are shown in Table 18 in the section where background correction was not performed prior to SAAM analysis. As shown in Fig. 51C, the
resulting 2D profiles for all three compounds exhibit reasonable chromatographic peak shapes. However, what if the need to overestimate the number of components could have been circumvented by performing OBGC prior to SAAM analysis? An improvement was already established in performing background correction prior to MCR-ALS analysis for 5-hydroxytryptamine. Might a similar improvement be found for the indole-3-acetyl-L-lysine and indole-3-acetyl-L-glycine peaks? The resulting optimized 2D profiles are shown in Fig. 51D when background correction is performed prior to SAAM analysis. While the 2-bromopyridine and indole-3-acetyl-L-glycine peaks appear to be nicely shaped, the indole-3-acetyl-L-lysine peak is slightly distorted. This was also seen in Fig. 52C to lesser degree. A possible reason for this might be the inability of the unimodality constraint to correctly remove the 2-bromopyridine contributions from the indole-3-acetyl-L-lysine component. However, the % recoveries for indole-3-acetyl-L-lysine have somewhat improved if background correction is performed prior to SAAM analysis compared to simply adding extra components. Finally, as expected, the % recoveries for indole-3-acetyl-L-lysine, 2-bromopyridine, and indole-3-lacteal-glycine are improved if background correction is performed prior to MCR-ALS analysis compared to if background correction is not performed prior to MCR-ALS analysis. This indicates that background correction should be performed prior to MCR-ALS if simple summation is used for quantification.
Table 18: Percent recoveries and associated standard deviations for indole-3-acetyl-L-lysine, 2-bromopyridine, and indole-3-acetyl-L-glycine before and after background correction

<table>
<thead>
<tr>
<th>Method</th>
<th>Indole-3-acetyl-L-lysine</th>
<th>2-bromopyridine</th>
<th>Indole-3-acetyl-L-glycine</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Q1</td>
<td>Q3</td>
<td>Q4</td>
</tr>
<tr>
<td>Without background correction</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SAAM</td>
<td>108±9</td>
<td>116±7</td>
<td>94±34</td>
</tr>
<tr>
<td>PARAFAC2</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>SCP</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>MCR-ALS</td>
<td>124±5</td>
<td>116±4</td>
<td>105±2</td>
</tr>
<tr>
<td>With background correction</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SAAM</td>
<td>93±11</td>
<td>103±7</td>
<td>105±7</td>
</tr>
<tr>
<td>PARAFAC2</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>SCP</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>MCR-ALS</td>
<td>95±5</td>
<td>98±2</td>
<td>97±4</td>
</tr>
</tbody>
</table>
**Fig. 50:** Plots of the resolved 2-bromopyridine component (blue) and indole-3-acetyl-L-lysine and indole-3-acetyl-L-glycine component (red) for A) PARAFAC2 and B) SCP.

**Fig. 51:** Plots of the optimized resolved 2D profiles from SAAM analysis. A) Profiles obtained if component splitting is not used. B) Profiles obtained if component splitting is used on all samples. C) Profiles obtained if extra components are added to the samples impacted by background while component splitting was being performed. D) Profiles obtained after OBGc was performed prior to SAAM. The blue peak corresponds to 3-chloropyridine. The red peak corresponds to indole-3-acetyl-L-lysine. The green peak corresponds to indole-3-acetyl-L-glycine. The cyan peak in B corresponds to indole-3-acetyl-L-glycine that was placed into a component that also includes background contributions. Quantitative results shown in Table 16.
7.3.4 Maize recovery

The percent recoveries for the four levels spikes added to the diluted maize sample are shown in Table 19. These percent recoveries are all based on assuming that only four components are chemically relevant in the data set. However, the number of chemically relevant components may be greater, as shown in Fig. 52. The log plots of the singular values obtained by SVD analysis of the maize data set shows that it is difficult to visually estimate the true number of components if background correction was not performed. The log plot of the singular values from SVD analysis does become simpler if background correction was performed. A four component model, if background correction was not performed, was chosen based on the observation that SAAM analysis produced recoveries close to 100 %. If additional components are included in the model, the percent recoveries obtained by SAAM analysis decrease.

The ability of the four analysis methods to resolve the anthenilic acid peak from the data set is shown in Fig. 53. MCR-ALS analysis after background correction is performed is least affected by the choice of the number of components. MCR-ALS is the only method to produce identical results between the calibration samples and the maize spiked samples. SAAM is the second least susceptible method to the number of components chosen. However, SAAM produced normalized intensities for the highest spiked maize sample that were widely different. After looking at the data produced by SAAM, an obvious reason behind this discrepancy has not been found. It is highly probable that some of the maize peaks surrounding the anthenilic acid peak are contributing to the discrepancy. Further work will need to be conducted to see if SAAM can successfully extract a single peak of interest from a data set containing multiple peaks. However, unlike SAAM, both PARAFAC2 and SCP underestimate the anthenilic component within the maize spikes. This underestimation leads to the reduced percent
recoveries shown in Table 19. Once again, it may be that the presence of the adjacent peaks is causing some of the anthranilic acid peak to be placed in other components. Given that the same trend is seen for both PARAFAC2 and SCP (and SAAM to a much lesser extent), it may be that PARAFAC based methods have difficulty correctly resolving the overlapping peaks. Once again, this is most likely due to an incorrect number of components chosen for chemometric analysis.

Table 19: Percent recoveries and associated standard deviations for anthranilic acid spiked into diluted maize extracts after SAAM, PARAFAC2, SCP, and MCR-ALS analysis

<table>
<thead>
<tr>
<th>Method</th>
<th>M1^a</th>
<th>M2^a</th>
<th>M3^a</th>
<th>M4^a</th>
</tr>
</thead>
<tbody>
<tr>
<td>SAAM^b</td>
<td>95±2</td>
<td>95±1</td>
<td>95±6</td>
<td>96±13</td>
</tr>
<tr>
<td>PARAFAC2^b</td>
<td>84±1</td>
<td>79±2</td>
<td>90±5</td>
<td>86±9</td>
</tr>
<tr>
<td>SCP^b</td>
<td>73±2</td>
<td>75±3</td>
<td>87±5</td>
<td>84±8</td>
</tr>
<tr>
<td>MCR-ALS^c</td>
<td>107±5</td>
<td>98±2</td>
<td>99±2</td>
<td>101±2</td>
</tr>
</tbody>
</table>

^a M1-M4 designate the maize 1 to maize 4 samples  
^b background correction by OBGC was not performed prior to analysis  
^c background correction by OBGC was performed prior to analysis

Fig. 52: Plots of the log (base 10) of the singular values from SVD of the maize recovery data set if A) background correction has not been performed; or B) if background correction has been performed.
**Fig. 53:** Plots of the normalized intensity versus the concentration for A) Background correction performed prior to MCR-ALS analysis; B) SAAM analysis; C) PARAFAC2 analysis; D) SCP analysis. The blue diamonds correspond to anthranilic acid samples consisting of water and MeOH. The red squares correspond to the different levels of anthranilic acid spiked into the dilute maize samples. The red square calibration standard zero corresponds to the diluted maize sample without anthranilic acid.

### 7.4 Conclusions

SAAM was compared to SCP and PARAFAC2 using both simulated and experimental data sets. SAAM and SCP were found to produce comparable concentrations for the phenytoin data set. However, PARAFAC2 was not able to resolve the phenytoin peak from the coeluting interferent peak. During the analysis of the simulation data sets, SCP was found to be the least affected by inter- and intra-sample retention time shifting. Although SAAM was found to be somewhat affected by intra-sample retention time shifting, the ability of SAAM to correct for intra-sample retention time shifting was not investigated. PARAFAC2 was found to be the method most vulnerable to inter- and intra-sample retention time shifting, particularly as the $^1$D peak shape changed. During the quantitation study, it was discovered that SAAM has difficulty
handling co-eluting peaks with an R_s less than 0.2. A possible solution to this might the addition of the PARALIND [152] function into the three-way PARAFAC analysis step of SAAM. Finally, SAAM was able to accurately retrieve the anthranilic peak from the spiked maize samples. However, SCP and PARAFAC2 experienced difficulty in recovering the anthranilic acid peak. Their difficulty may have been due to an inadequate number of components chosen for analysis.
8.1 Reflections on Chapter 4

In Chapter 4, several background correction techniques were studied as a means of removing background artifacts, and the subsequent impacts on peak detection and quantitation were characterized. The AWLS technique was found to be the only background correction technique to completely remove the injection and re-equilibration background artifacts. However, the AWLS technique was also found to result in approximately 25%, an attenuation of the peak heights, as shown in Fig. 26A,. Although the weighting factor can be varied, a determination was made that intensity loss occurred regardless of the weighting value used, as shown in Fig. 26C. The alternative background correction technique, OBGC, proposed by Filgueira et al. [46], was found to only completely remove one of the background artifacts. However, the number of peaks detected from the injection ridge decreased after a simple alignment to correct for the oscillation caused by the use of a shunt in the sampling device. This indicates that if this oscillation can be removed then it may be possible for OBGC to remove the injection ridge during background correction.

Filgueira proposed in his doctoral thesis a design of a sampling device that utilizes a concentric design to eliminate the need for a shunt when performing LC×LC [153]. The use of this sampling device would eliminate the oscillations currently found in LC×LC chromatograms, by forcing the flow from both loops to follow the exact same pathway. If the use of this sampling device proves insufficient to allow for complete removal of the injection ridge when
performing OBGC, then an alternative approach could be attempted. Very short trapping columns could be investigated in place of the loops to allow the 1D effluent to pass to waste. The trapping columns would need to be of the same material as the 2D column to ensure that retention is achieved and analytes are not lost. In addition, the 1D effluent can be diluted by water prior to the trapping column to aid in the retention of the analytes at the head of the 2D column, as is currently done by the Stoll research group [47, 48]. This would prevent the injection of solvent containing high amounts of organic modifier from being injected onto the 2D column. Therefore, the oscillations should be minimized and, furthermore, the change in refractive index that causes the injection ridge would be lessened (but not eliminated).

In terms of peak detection, the derivative-based approach was found to detect more of the poorly resolved peaks than the drain algorithm. As such, I would recommend that the derivative-based approach be used but with some minor modifications. As it is currently implemented, the derivative based approach does not restrict the degree of drift allowed in $t_R$ between successive 2D peaks. The approach is currently based only on the degree of overlap between the detected start and stop of each prospective 2D peak. An improvement that can be made is to add an additional constraint to restrict the allowed difference in $t_R$ between successive 2D chromatograms to ensure that peaks with broad tails are not mistakenly merged. Another modification that could be implemented is to assess the correlation of the spectra of the two prospective 2D peaks prior to merging. However, as shown in Fig. 25, the impact of using a spectral correlation constraint to merge peaks for DAD data is currently minimal unless high correlation values are used.

In terms of quantitation, an improvement was not found from performing manual integration versus the drain algorithm in LC Image. However, quantitation was not performed
using the derivative-based approach since the version of software being used squared the signal. An additional comparison should be conducted by calculating the peak volumes using the derivative approach but not using the square function.

8.2 Reflections on Chapter 5

In Chapter 5, five interpolation strategies (linear interpolation, piecewise cubic Hermite interpolating polynomial, cubic spline, Fourier zero-filling, and Gaussian fitting) were compared based on the ability to recreate the unsampled $t_R$ and the calculated improvement in % RSD after alignment and PARAFAC analysis. The Gaussian fitting technique was found to produce the closest $t_R$ to the unsampled $t_R$ during the analysis of simulated data. However, it should be noted that the simulated data was generated using Gaussians functions for the peaks. In contrast, a statistical difference could not be found between the different interpolation techniques when applied to experimental data. A postulation was made that the reason the Gaussian fitting technique did not perform better was that the fitting was performed on the raw data. It was further postulated that if Gaussian fitting had been performed on the PARAFAC resolved $1^D$ chromatogram then the resulting interpolated peak would not be as impacted by the background. This reasoning led directly into the work carried out and described in Chapter 6.

8.3 Reflections on Chapter 6

In Chapter 6, a method was developed that allowed for alignment of first and second dimension retention times between samples based on spectral matching. The method relied on the use of three-way PARAFAC to individually resolve each sample into the corresponding $2^D$, $1^D$, and spectral component profiles. A component matching scheme was developed to associate
the spectral profiles between samples. Gaussian fitting was used to interpolate the sampled \(^1\text{D}\) peak shape before alignment occurred. Alignment was performed individually for each peak resulting in a different reference point for each analyte. After alignment, four-way PARFAC analysis is performed to obtain the concentration profiles for each component. These concentration profiles are then utilized to provide quantitative information. The method was successfully shown to produce results similar to or better than MCR-ALS-based methods. However, as the method currently stands, significant user intervention is still required throughout the method. Therefore, if this method is to be of use in the future, further automation needs to be developed.

The first portion of the automation the needs to be developed is the determination of the correct number of components present in the overall data and in each individual sample. While I have experimented with various means of automating this step, a comprehensive study has yet to be attempted for LC×LC-DAD that utilizes the wide variety of methods already present in the literature. The second portion of the automation involves the determination of whether a given component is either a peak or a background. At this point, I think that the easiest way of accomplishing this is to make use of the drain algorithm employed in LCImage. Unlike the raw data, poorly resolved peaks should not prove a problem since they will have already been resolved into separate components (assuming they have dissimilar spectra). As with LCImage, the boundaries of the peak with the highest intensity would be determined. The number of data points encompassed by this peak would then be calculated and compared against a validated threshold. If the number of data points is greater, then the algorithm would proceed to the next largest peak. If the number of data points is less, then the threshold then the algorithm would
terminate. If a peak was not determined to be present, then the component would be labeled as background.

The use of the drain algorithm would also provide benefits beyond component determination. Since the drain algorithm determines the boundaries of a peak, the drain algorithm could be used to allow for non-linear splitting of components prior to three-way PARAFAC. This would be an improvement over the current approach which relies on a clear linear delineation between peaks to exist. By extension, the drain algorithm could be used to automate the generation of the chromatography selectivity constraint since the region occupied by the peak will have already been defined.

8.4 Reflections on Chapter 7

In Chapter 7, SAAM was directly compared to PARAFAC2 and SCP through a variety of simulated and experimental data sets. The first data set examined was the phenytoin data set obtained from the Stoll research group and previously examined in Chapter 6 by SAAM and MCR-ALS. The analysis of the phenytoin data set was expanded to include the PARAFAC2 and SCP methods. While SCP generated a concentration for phenytoin that was very similar to the concentration generated by SAAM, PARAFAC2 was not able to successfully resolve the phenytoin component from the interferent component. Initial work into this problem has indicated that the similarity in the spectra between phenytoin and the interferent is most likely not the cause. However, a spectral similarity cannot be ruled out given the limitations of the simulations used to reach this conclusion. A more robust set of simulations should be conducted to determine the reason behind why PARAFAC2 was unable to resolve the two peaks into different components.
The simulated data sets revealed that SCP was the least affected by the introduction of first and second dimension inter-sample retention time shifting and intra-sample shifting. However, SAAM held up fairly well with only a slight more impact from the intra-sample shifting compared to SCP. PARAFAC2 was found to experience the greatest difficulty in handling the increased inter-sample retention time shifts in the first dimension in addition to the retention time shifting introduced in the second dimension. In addition, the problems encountered by PARAFAC2 were exacerbated by the introduction of intra-sample shifting in the second dimension. Once again, while these initial simulations begin to show a trend, I think that a more robust series of simulations would better show the capability of SAAM to handle inter- and intra-sample shifting compared to SCP and PARAFAC2. In addition, the ability of SAAM to correct for intra-sample shifting during these simulations was not investigated. If further simulations are to be conducted, then the intra-sample alignment tool should be included to gauge its accuracy.

Two problems were discovered during the analysis of regions 1 and 2 of the quantitation study. The first problem involved the ability of SAAM to handle very poorly resolved peaks (Rs less than 0.2). A possible solution is to modify the PARAFAC algorithms used in this method to make use of the parallel profiles with linear dependence (PARALIND) model [152]. PARALIND allows for PARAFAC to handle situations in which two components have the same shape in one dimension. The use of PARALIND may enable better resolution of the components as well as reduce the possibility of crosstalk between components. The second problem is the influence of unresolved background contributions on the component splitting tool. As such, I would recommend that SAAM be performed only after background correction has been performed. This is based on the observation that the removal of background contributions did
not significantly change the resulting optimal analyte components profiles or resulting percent recoveries.

Finally, the results obtained from the maize data highlight the importance of finding an automated means of correctly determining the number of chemically relevant components. Every quantification method studied in this dissertation relies on the correct number of components to have been chosen prior to analysis. While some methods (MCR-ALS) may be less impacted by the number of components than others (SCP and PARAFAC2), I do not feel that we can truly trust the results of the chemometric methods to accurately and precisely quantify all compounds within a complex sample unless we have \textit{a priori} knowledge of the number of compounds present.

8.5 Final Thoughts

While not providing definite solutions to the problems identified by Bailey and Rutan [1], the work presented in this dissertation does begin to rule out possible solutions while providing hints of further avenues of research. In particular, a first step towards the automated chemometric analysis of LC×LC-DAD samples has been made. As such, it may be possible to greatly simplify the user training and the time required to obtain quantitative and qualitative information from LC×LC-DAD samples. With time, it may eventually become possible to realize the goal of using LC×LC-DAD as a screening tool for biological samples. In the early work on LC×LC in the Carr research group, Stoll \textit{et. al.} showed that it was possible to resolve complex mixtures in approximately thirty minutes [154]. While this particular separation was on peptides, the same principles have been applied to metabolomic samples [92, 155-157]. With the capability of LC×LC to separate complex samples and chemometric techniques to resolve
overlapped peaks, the analysis of metabolomics samples can be used to diagnose diseases. This is the stated eventual goal of both the Carr and Rutan research groups.
List of References
List of References


[77] S.A. Bortolato, J.A. Arancibia, G.M. Escandar, A.C. Olivieri, Time-alignment of bidimensional chromatograms in the presence of uncalibrated interferences using parallel factor


Appendix A

The mathematical algorithms for a three-way PARAFAC analysis, a four-way PARAFAC analysis, a three-way PARAFAC2 analysis, and a three-way SCP analysis are listed.

A.1 Three-way PARAFAC Algorithm

The three-way PARAFAC algorithm was obtained from reference [66].

1) Initialize $B$ and $C$
2) $Z = (C \odot B)$
3) $A = X_{laj} Z (Z^T Z)^{-1}$
4) $Z = (C \odot A)$
5) $B = X_{jkl} Z (Z^T Z)^{-1}$
6) $Z = (B \odot A)$
7) $C = X_{jkl} Z (Z^T Z)^{-1}$
8) Return to step 2 and iterate

where $A_{KxN}$, $B_{JxN}$, and $C_{JxN}$ are the two-way arrays corresponding to the dimensions of $X_{(IxJxL)}$ and $Z$ is a temporary two-way array.
A.2 Four-way PARAFAC algorithm

The four-way PARAFAC algorithm was extended from the three-way PARAFAC algorithm from reference [66].

1) Initialize B, C, and D
2) \( Z = (D \odot C \odot B) \)
3) \( A = X_{KIJK} Z(Z^T Z)^{-1} \)
4) \( Z = (D \odot C \odot A) \)
5) \( B = X_{JKKL} Z(Z^T Z)^{-1} \)
6) \( Z = (D \odot B \odot A) \)
7) \( C = X_{JIKL} Z(Z^T Z)^{-1} \)
8) \( Z = (C \odot B \odot A) \)
9) \( D = X_{LJKK} Z(Z^T Z)^{-1} \)
10) Return to step 2 and iterate

where \( A_{KxN} \), \( B_{JxN} \), \( C_{JxN} \), and \( D_{LxN} \) are the two-way arrays corresponding to the dimensions of \( X \) \((I\times J\times K\times L)\) and \( Z \) is a temporary two-way array.
A.3 Three-way PARAFAC2 algorithm

The three-way PARAFAC2 algorithm was obtained from reference [158].

1) Initialize A, B, and C
2) For every k, k = 1,..., K
   \[ H_k = X_k^T A D_k B^T \]
   \[ P_k = H_k (H_k^T H_k)^{-1} \]
3) For every k, k = 1,..., k
   \[ Y_k = X_k P_k \]
4) Determine A, B, and C from one iteration of PARAFAC
5) Return to step 2 and iterate

where \( A_{I \times N} \), \( B_{L \times N} \), and \( C_{K \times N} \) are the two-way arrays corresponding to the dimensions of \( X \) (IxLxK), \( H_k \) is the constraint that preserves the shape of the shifting dimension between samples, \( P_k \) is the pseudoinverse of \( H_k \), and \( Y_k \) is the optimized profile for the shifting dimension.
A.4 Three-way SCP algorithm

The three-way SCP algorithm was obtained from reference [5].

1) Initialize $A$, $B$, $C$, and $\tau$

2) $\tilde{A}_f = \tilde{X}_{F,KL} \cdot (C \exp \left[ -i2\pi \left( \frac{f-1}{L} \right) \cdot \tau \right] \odot B)^r$

3) $E_{k,il,sh}^{n'} = X_{k,il,sh} - \sum_{n'=n} c_{k,n'} (A_n^{th} \odot B_s)^r$

4) $s_{k,n'}(i) = \sum_{i=n'}^t e_{i,l,k}^n b_{l,n'}$

5) $\tilde{v}_{k,n'}(f) = \tilde{s}_{k,n'}(f) \tilde{a}_{f,n'}$

6) $\hat{\tau}_{k,n'} = \text{argmax} |v_{k,n'}(\tau)|$

7) $\tau_{k,n'} = \hat{\tau}_{k,n'} - (I + 1)$

8) $Z_{i+k(I-1),n} = c_{k,n} a_{i,n}^h$

9) $B = X_{LxIK} Z^r$

10) $C_k = X_{KxIL} (A^k) \odot B)^r$

11) Return to step 2 and iterate

where $A_{KxN}$, $B_{KxN}$, and $C_{JxN}$ are the two-way arrays corresponding to the dimensions of $X$ ($IxJxL$) and $\tau$ is a two-way array containing the degree of shifts for each sample for the shifting dimension.
Appendix B

The .m files and data for Chapter 4 can be found in R:\CHEM\Rutan_lab\Robert\Chapter4. The .m code used to generate the results in Chapter 4 is listed below. In addition to the functions and scripts listed below, the xxx.csv files from Chemstation, the corresponding xxx.mat files from Matlab, and the scripts used to generate the background corrected data in Chapter 4 are included in the listed directory. The .mat files are organized according to the naming system: IndoleX.mat (the standards replicates), MaizeX.mat (the maize replicates), BlankX.mat (the blank replicates), _Direct (background corrected by the Direct technique), _SVD (background corrected by the SVD-BC technique), _asym (background corrected by the AWLS technique), _wolf (for analysis by the Stevenson Mathematica algorithm), and bc_5 (OBGC using a 5 point median filter window).

The matrix_2Dto3Dv2 function is used to adjust the shape of the raw .csv file obtained from Chemstation into a three-way array with dimensions corresponding to the second dimension, first dimension, and UV-Vis spectrum. The inputs for the function are the .csv file (matrix_2D_raw), the injection delay time in the first dimension (time_injection_delay), the data collection frequency of the DAD detector (sampling_freq), and the t_s (time_cycle). The output (matrix_3D) is a .mat variable.

function [matrix_3D] = matrix_2Dto3Dv2(matrix_2D_raw, time_injection_delay, sampling_freq, time_cycle)
% Number of data points for 2D delayed injection
points_delayed_injection = time_injection_delay * sampling_freq;

207
% Gets the wavelengths exported from the datafile
wavelengths = matrix_2D_raw(1,2:end);
% Trims the first rows to correct for delayed injection on the 2D and
% removes the time column
matrix_2D = matrix_2D_raw(points_delayed_injection+1:end,2:end);
% % Rounds down to avoid complains about using a real as an index and removes
% % points from the beginning of the vector
% % vec(1:floor(points_delayed_injection,:)) = [];
% Number of points in a single 2D run
points_2D_cycle = time_cycle*sampling_freq;
% Get the integer values of 1D slices
samples_1D = floor(size(matrix_2D,1)/points_2D_cycle);
% takes only a multiple number the points_2D_cycle
matrix_2D = matrix_2D(1:samples_1D*points_2D_cycle,:);
% now we can reshape to a 3D matrix
matrix_3D = reshape(matrix_2D, points_2D_cycle, samples_1D, size(wavelengths,2));

The background_subtractionv2 function was used to perform SVD-BC on the each standard and
maize sample. The inputs were the sample of interest (data) and the truncated V from SVD from
the augmented blank sample (blank_v).

function [bkgdsub]=background_subtractionv2(data,blank_v)
[second,first,spectra]=size(data);
sampletrim=reshape(data,second*first,spectra); %reshaping for SVD
chrom=sampletrim(:,85:spectra)/blank_v(85:spectra,:)'; %chromatogram of first N components, 85 refers to the data point of the chosen wavelength cutoff point
backfit=chrom*blank_v'; bkgdsub=sampletrim-backfit; %subtracting background from sample
bkgdsub=reshape(bkgdsub,second,first,spectra);

The asymbackv2 function was used to perform AWLS on the each standard and maize sample.
The inputs were the sample of interest (sample), the truncated V from SVD from the augmented
blank sample (P), and the positive residual weight value (p).

function subsam=asymbackv2(sample,P,p)
[second,first,spectra]=size(sample);
sample=reshape(sample,second*first,spectra);'
subsam=zeros(size(sample));
for m=1:size(sample,2)
The script, contained in the file obgc.m, used to perform OBGC on a single sample follows. The value d refers to the size of the filter window, matrix_3D is the sample undergoing background correction, bg is the estimated background for each wavelength and bgCorr is the final background corrected sample.

```matlab
%% oBGc
d=45;
filter_window_median = d;
[a b c] = size(matrix_3D);
bg = zeros(a,b,c);
for wavelength=1:size(wavelengths,2)...
```
bg(:,:,wavelength) = medfilt2(matrix_3D(:,:,wavelength), [1 filter_window_median]);
end
% Subtract filtered background
bgCorr = matrix_3D - bg;

The script, contained in the file countingpeaksv4.m, used to manually pass through each LC×LC-DAD sample to allow the user manually determines the number of 2D peaks present.

MaizeA_peakcount=cell(1,158); % The MaizeA refers to the sample and 158 is the size of the first dimension.
r=1;
for a=1:158
    count=1;
    y=zeros(960,1); % 960 refers to the size of the second dimension.
    for b=1:960
        y(b,1)=b;
    end
    x=zeros(1,960);
    for b=1:960
        x(1,b)=a;
    end
    subplot(2,1,1); contour(MaizeA(:,:,1),150);
    hold
    line(x,y,'Color',[1 0 0]);
    subplot(2,1,2);
    plot(MaizeA(:,a,11)) % 11 refers to the selected wavelength.
    disp(['Are there peaks present for slice ' num2str(a) '?']);
while r==1
    choice=input('0 if no, 1 if yes: ');
    if isempty(choice)==1
        disp('Inappropriate choice. Please enter 0 or 1.');
    else
        choice=round(choice);
        if choice>1 || choice<0
            disp('Inappropriate choice. Please enter 0 or 1.);
        else
            break;
        end
    end
end
close all;
if choice==1
    for b=101:100:801 % The range in the second dimension being looked at.
        subplot(2,1,1);plot(MaizeA(:,a,11));
title(['1D point ' num2str(a)]);

hold
[val,index]=max(MaizeA(:,a,11));
y=zeros(floor(val),1);
x2=zeros(1,floor(val));
x1=zeros(1,floor(val));
for c=1:floor(val)
    y(c,1)=c;
    x1(c,1)=b;
    x2(1,c)=b+99;
end
line(x1,y,'Color',[1 0 0]);
line(x2,y,'Color',[1 0 0]);
subplot(2,1,2);plot(MaizeA(b:b+99,a,11));
set(gca,'XTickLabel',{num2str(b-1);num2str(b+10);num2str(b+20);num2str(b+30);num2str(b+40);num2str(b+50);num2str(b+60);num2str(b+70);num2str(b+80);num2str(b+90);num2str(b+99)});
title(['Between ' num2str(b) ' and ' num2str(b+99) ' in the 2D']);
while r==1
    peak_count=input('How many peaks are present: ');
    if isempty(peak_count)==1
        disp('Inappropriate choice. Please select a number of peaks between 0 and inf. ')
    else
        peak_count=round(peak_count);
        if peak_count<0
            disp('Inappropriate choice. Please select a number of peaks between 0 and inf. ')
        else
            break;
        end
    end
end
for c=1:peak_count
    MaizeA_peakcount{1,a}(count,1)=a;
    [x,y]=ginput(1);
    x=round(x);
    MaizeA_peakcount{1,a}(count,2)=x+b;
    MaizeA_peakcount{1,a}(c,3)=MaizeA(MaizeA_peakcount{1,a}(c,2),MaizeA_peakcount{1,a}(c,1),11);
    count=count+1;
end
close all
The following script, contained in the file merge.m, is used to merge the 2D peaks detected by the previous section of script.

```matlab
d=0;
peak=zeros(960,158);
for a=1:size(MaizeK_peakcount,2) % MaizeK refers to the sample
    b=size(MaizeK_peakcount{1,a},1);
    if b>0
        for c=1:b
            peak(MaizeK_peakcount{1,a}(c,2),MaizeK_peakcount{1,a}(c,1))=MaizeK(MaizeK_peakcount{1,a}(c,2),MaizeK_peakcount{1,a}(c,1),11);
            d=d+1;
        end
    end
end
[m,n]=find(peak>0);
p=zeros(d,3);
for a=1:d
    p(a,1)=n(a,1);
p(a,2)=m(a,1);
p(a,3)=MaizeK(m(a,1),n(a,1),11);
end
spec_corr=zeros(d,d);
for a=1:d
    for b=1:d
        r=corrcoef(squeeze(MaizeK(p(a,2),p(a,1),:)),squeeze(MaizeK(p(b,2),p(b,1),:)));
        spec_corr(a,b)=r(1,2);
    end
end
for a=1:d
    spec_corr(a,a)=0;
end
for a=1:d-1
    if p(a,1)==p(a+1,1)
        spec_corr(a,a+1)=0;
    end
end
[m_c,n_c]=find(spec_corr>=0.0); %Spectral matching threshold. Currently set to 0.
map=zeros(d,d);
for a=1:size(m_c,1)
    map(m_c(a,1),n_c(a,1))=1;
end
```
flag=zeros(d,1);
peak_merge=cell(1,d);
merge_pointer=1;
merge_intensity=zeros(1,d);
for a=1:d
    if flag(a,1)==0
        flag(a,1)=1;
        [m_s,n_s]=find(map(a,:)==1);
        uni=0;
        first=p(a,1);
        second=p(a,2);
        intensity=p(a,3);
        pointer_trail=[first;second;intensity];
        for b=1:size(n_s,2)
            if p(n_s(1,b),1)==(first+1)
                if p(n_s(1,b),2)>=
                    (second-10)&&p(n_s(1,b),2)<=(second+10) % The degree of drift
                        allowed between successive 2D peaks. Currently set to 10 data points.
                        if p(n_s(1,b),3)>intensity && uni==0
                            pointer_trail=[pointer_trail [p(n_s(1,b),1);p(n_s(1,b),2);p(n_s(1,b),3)]];
                            first=p(n_s(1,b),1);
                            second=p(n_s(1,b),2);
                            flag(n_s(1,b),1)=1;
                            intensity=p(n_s(1,b),3);
                        elseif p(n_s(1,b),3)<intensity && uni==0
                            pointer_trail=[pointer_trail [p(n_s(1,b),1);p(n_s(1,b),2);p(n_s(1,b),3)]];
                            first=p(n_s(1,b),1);
                            second=p(n_s(1,b),2);
                            flag(n_s(1,b),1)=1;
                            intensity=p(n_s(1,b),3);
                            uni=1;
                        elseif p(n_s(1,b),3)<intensity && uni==1
                            pointer_trail=[pointer_trail [p(n_s(1,b),1);p(n_s(1,b),2);p(n_s(1,b),3)]];
                            first=p(n_s(1,b),1);
                            second=p(n_s(1,b),2);
                            flag(n_s(1,b),1)=1;
                            intensity=p(n_s(1,b),3);
                        elseif p(n_s(1,b),3)>intensity && uni==1
                            pointer_trail=[pointer_trail [p(n_s(1,b),1);p(n_s(1,b),2);p(n_s(1,b),3)]];
                            first=p(n_s(1,b),1);
                            second=p(n_s(1,b),2);
                            flag(n_s(1,b),1)=1;
                            intensity=p(n_s(1,b),3);
                        end
                    end
                elseif p(n_s(1,b),3)<intensity && uni==0
                    pointer_trail=[pointer_trail [p(n_s(1,b),1);p(n_s(1,b),2);p(n_s(1,b),3)]];
                    first=p(n_s(1,b),1);
                    second=p(n_s(1,b),2);
                    flag(n_s(1,b),1)=1;
                elseif p(n_s(1,b),3)>intensity && uni==0
    end
end
peak_merge{1,merge_pointer}=pointer_trail;
merge_intensity(1,merge_pointer)=max(peak_merge{1,merge_pointer}(3,:));
merge_pointer=merge_pointer+1;
end
end
count_total=merge_pointer-1;
[m_h,n_h]=find(merge_intensity>=6.7);  % 6.7 Refers to the high detection threshold used in this
chapter for the non-background corrected data.
count_high=size(n_h,2);
[m_l,n_l]=find(merge_intensity>=5.925);  % 5.925 Refers to the low detection threshold used in
this chapter for the non-background corrected data.
count_low=size(n_l,2);
Appendix C

The .m files and data for Chapter 5 can be found in R:\CHEM\Rutan_lab\Robert\Chapter5. The .m code used to generate the results in Chapter 5 is listed below. Xsmall refers to the original data with the dimensions: 1) second dimension; 2) first dimension; 3) sample; and 4) UV-Vis spectrum. The scripts are contained in the file interpolation.m. The data for Chapter 5 are named in the following fashion: the Simulation1.mat file contains the simulated data set, X50to60 is peak 1, X30to45 is peak 2, X23to32 is peak 3, X14to25 is peak 4, X9to15 is peak 5.

The interpolation script contains the following interpolation methods:

```matlab
% index of first dimension
Xa = squeeze(Xsmall(45,:,:,:,:)); % resizes Xsmall from 151,16,6,126 to 16,6 and calls it Xa
Xsmall = Xsmall(151,16,6,126); % resizes Xsmall from 151,16,6,126 to 16,6 and calls it Xa
time = [1:8]; % creates original first dimension time points
timex = [1:0.1:8]; % creates new first dimension time points
for n = 1:35 % cycles through injections
    Xs(:,n) = spline(time,Xa(:,n),timex); % perform spline interpolation on Xa
    Xh(:,n) = pchip(time,Xa(:,n),timex); % perform hermite interpolation on Xa
end
Xavg = (0.0*Xs+1.0*Xh); % the average of the spline and hermite interpolation
[value, index] = max(Xavg); % returns the values and positions of the highest point
clear Xfinal; % deletes Xfinal
% index of second dimension
Xb = squeeze(Xfirst(4,:,:,:,:)); % resizes Xsmall from 151,16,6,126 to 6,151 and calls it Xb
Xfirst = Xfirst(151,16,6,126); % resizes Xsmall from 151,16,6,126 to 151,6
timey = [1:150]; % creates the second dimension time points
[valueb, indexb] = max(Xb); % returns the values and positions of the highest point
clear Xfinal; % deletes Xfinal
% first dimension alignment
for a = 1:150 % cycle through second dimension
    for b = 1:126 % cycle through wavelengths
        clear Xa; % deletes Xa
```
Xa = squeeze(Xsmall(a,:,b,:)); % resized Xsmall from 151, 16, 6, 126 to 16, 6 and calls it Xa
Xa = Xa'; % transposes Xa so that it becomes 6, 16

clear Xs; % deletes Xs
clear Xh; % deletes Xh

for n = 1:35 % goes through each injection
    Xs(:,n) = spline(time, Xa(n,:), timexx); % perform spline interpolation on Xa
    Xh(:,n) = pchip(time, Xa(n,:), timexx); % perform hermite interpolation on Xa
end

clear Xavg; % deletes Xavg
Xavg = (0.0*Xs + 1.0*Xh); % the average of the spline and hermite interpolation

for n = 1:35 % cycles through injections
    Xnew(:,n) = Xavg((index(n) - min(index) + 1):(size(timexx,2) + index(n) - max(index)), n)';
    % moves the data around to correspond to the peak maxima
    Xfirst(:,n,a,b) = Xnew(:,n);
    Xfirst(:,n,a,b) = Xnew([1:9:end], n); % incorporates the data into a new matrix
end
end

% second dimension alignment
for a = 1:7 % cycle through second dimension
    for b = 1:126 % cycle through wavelengths
        for n = 1:35 % cycles through injections
            clear Xnew; % deletes Xnew
            Xnew = squeeze(Xfirst(a,n,:,b,:)); % reduces Xfirst to just the second RT points
            clear Xsecond; % deletes Xsecond
            Xsecond(:,n) = Xnew((indexb(n) - min(indexb) + 1):(size(timeyy,2) + indexb(n) - max(indexb)), 1);
            % moves the data around to correspond to the peak maxima
            Xfinal(:,n,a,b) = Xsecond(:,n); % incorporates the data into a new matrix
        end
    end
end

% Reordering Xfinal
Xfinalp = permute(Xfinal, [1, 3, 2, 4]);

% Revised Fourier code by Sarah Rutan to account for the incorrect time axis
% on 7 Feb 2011
% Fourier transform code`
for a = 1:size(Xsmall,3)
    Xa = squeeze(Xsmall(54,:,a,5));
    Size = size(Xa,2)/2;
    if rem(Size,1) == 0
        time = [1:size(Xsmall,2)-1];
        timexx = [1:0.1:size(Xsmall,2)-0.1];
        ftd1 = fft(Xa, size(Xa,2)-1);
    end
end
else
    time=[1:size(Xsmall,2)];
    timexx=[1:0.1:size(Xsmall,2)+0.9];
    ftd1=fft(Xa);
end
N=size(timexx,2)-size(time,2);
Median=ceil(size(ftd1,2)/2);
ftd1int=[ftd1(1:Median) zeros(1,N) ftd1(Median+1:size(ftd1,2))];
d1int=ifft(ftd1int);
X(:,a)=d1int(1,:)';
end
[value,index]=max(X);
Factor=size(timexx,2)/size(time,2);
for a=1:size(Xsmall,4)
    for b=1:size(Xsmall,3)
        for c=1:size(Xsmall,1)
            Xa=squeeze(Xsmall(c,:,b,a));
            if rem(Size,1)==0
                time=[1:size(Xsmall,2)-1];
                timexx=[1:0.1:size(Xsmall,2)-0.1];
                ftd1=fft(Xa,size(Xa,2)-1);
            else
                time=[1:size(Xsmall,2)];
                timexx=[1:0.1:size(Xsmall,2)+0.9];
                ftd1=fft(Xa);
            end
            ftd1int=[ftd1(1:Median) zeros(1,N) ftd1(Median+1:size(ftd1,2))];
d1int=ifft(ftd1int);
            X(:,b)=d1int(1,:)';
            X(:,b)=X(:,b)*Factor;
            Xnew(:,b)=X((index(b)-min(index)+1):(size(timexx,2)+index(b)-max(index)),b)';
            Xfirst(:,b,a,c)=Xnew([1:9:end],b);
        end
    end
end
Xb=squeeze(Xfirst(4,:,5,:));
Xb=Xb';
timeyy=[1:size(Xfirst,4)];
[valueb,indexb]=max(Xb);
for a=1:size(Xfirst,1)
    for b=1:size(Xfirst,3)
        for c=1:size(Xfirst,2)
clear Xnew2;
Xnew2=squeeze(Xfirst(a,c,b,:));
clear Xsecond;
Xsecond(:,1)=Xnew2(1:(indexb(c)-min(indexb))+1):((size(timeyy,2)+indexb(c)-max(indexb)),1);
Xfinal(:,c,a,b)=Xsecond(:,1);
end
clear a;
clear b;
clear c;
Xfinalp=permute(Xfinal,[1 3 2 4]);

%Revised Gaussian code by Dr. Rutan on 7Feb2011
%New Gaussian method

time=1:size(Xsmall,2);
timexx=1:0.05:size(Xsmall,2);
options=optimset;
options.MaxFunEvals=1000;
options.MaxIter=100;
options.TolFun=1e-8;
options.TolX=1e-4;
options.DiffMinChange=1e-6;
options.MaxTime=60;
params=zeros(size(Xsmall,3),5);
for a=1:size(Xsmall,3)
    Xa=squeeze(Xsmall(rtf(a,1),:,a,5))';
    [Value,Index]=max(Xa);
    %Determining a starting value for the peak area of each component
    if Index~=1&&Index~=size(Xsmall,2)
        Peakarea=Xa(Index-1,1)+Xa(Index,1)+Xa(Index+1,1)-1.5*(Xa(1,1)+Xa(end,1));
    end
    if Index==1
        Peakarea=abs(Xa(Index,1))+abs(Xa(Index+1,1));
    end
    if Index==size(Xsmall,2)
        Peakarea=abs(Xa(Index,1))+abs(Xa(Index-1,1));
    end
    %Determining a starting value the retention time of each component
    Xs=spline(time,Xa,timexx);
    [value,index]=max(Xs);
    Position=index*0.05+1;
    %Assuming that the broadened first dimension sigma is 0.59623
    Sigma=0.6;
    params(a,:)=[Peakarea Position Sigma 1 1];
\[ Xg(a, :) = \text{lsqnonlin(@emgfitv2, params(a,:), [0.5*Peakarea 0 0.4 0.25 -1e100], [1.5*Peakarea 1e100 0.8 1.75 1e100], options, Xa, time); } \]

\[ \text{fitpeak(:,a)} = \text{emggenv2}(Xg(a,:), \text{timexx})'; \]

\[ \text{clear Peakarea Xs Position Sigma} \]

\[ [\text{value, index}] = \text{max(fitpeak)}; \]

\[ \text{clear Index Value Xa a fitpeak options params value} \]

\[ \text{finalpeak} = \text{zeros(size(timexx,2),size(Xsmall,1),size(Xsmall,3),size(Xsmall,4));} \]

\[ \text{for c=}1:1:\text{size(Xsmall,4)} \]
\[ \text{for a=}1:1:\text{size(Xsmall,3)} \]
\[ \text{for b=}1:1:\text{size(Xsmall,1)} \]
\[ \text{Xa}=\text{squeeze(Xsmall(b,::,a,c));} \]
\[ \text{x}=\text{gaussb2}(Xg(a,:), \text{time});} \]
\[ \text{x}=[\text{x' ones(size(x,2),1)}]; \]
\[ \text{fitpeak} = \text{gaussb2}(Xg(a,:), \text{timexx});} \]
\[ \text{fitpeak2} = [\text{fitpeak'} \text{ ones(size(fitpeak,2),1)}]; \]
\[ \text{factors} = \text{x}\text{Xa'}; \]
\[ \text{fitpeak3} = \text{fitpeak2*factors}; \]
\[ \text{finalpeak(:,b,a,c)} = \text{fitpeak3}; \]
\[ \text{end} \]
\[ \text{end} \]
\[ \text{end} \]

\[ \text{Xindex} = \text{permute(finalpeak,[2 1 3 4]);} \]
\[ [\text{value, index}] = \text{max(Xindex(40,:,:,:,:))}; \]

\[ \text{for a=}1:1:\text{size(Xsmall,3)} \]
\[ \text{for b=}1:1:\text{size(Xsmall,4)} \]
\[ \text{for c=}1:1:\text{size(Xsmall,1)} \]
\[ \text{finalpeak2} = \text{squeeze(Xindex(c,::,b));} \]
\[ \text{Xnew(:,a)} = \text{finalpeak2((index(a)-min(index)+1):(size(timexx,2)+index(a)-max(index)),a)}; \]
\[ \text{Xfirst(:,a,c,b)} = \text{Xnew([1:9:end],a)}; \]
\[ \text{end} \]
\[ \text{end} \]
\[ \text{end} \]

\[ \text{Xb} = \text{squeeze(Xfirst(3,::,:,:))}; \]
\[ \text{Xb} = \text{Xb'}; \]
\[ \text{timeyy} = [1:1:\text{size(Xfirst,3)}]; \]
\[ [\text{valueb, indexb}] = \text{max(Xb)}; \]

\[ \text{clear Xsecond;} \]

\[ \text{for a=}1:1:\text{size(Xfirst,1)} \]
\[ \text{for b=}1:1:\text{size(Xfirst,4)} \]
\[ \text{for c=}1:1:\text{size(Xfirst,2)} \]
\[ \text{clear Xnew2;} \]
\[ \text{Xnew2} = \text{squeeze(Xfirst(a,c,::,b))}; \]
\[ \text{clear Xsecond;} \]
\[ \text{Xsecond(:,1)} = \text{Xnew2((indexb(c)-min(indexb)+1):(size(timeyy,2)+indexb(c)-max(indexb)),1)}'; \]

219
Xfinal(:,c,a,b)=Xsecond(:,1);
end
end
end
Xfinalp=permute(Xfinal,[1 3 2 4]);
Appendix D

The .m files and data for Chapter 6 can be found in R:\CHEM\Rutan_lab\Robert\Chapter6. The .m code used to generate the results in Chapter 6 is listed below. The primary SAAM function, analyze_2DLCv18, and other associated functions are listed below. The target_revised.mat is the phenytoin data set, the urine_data is the urine data set, the SimulationX is the first, second, and third simulated data set, and the christophesimulation5 is the fourth simulated data set.

The analyze_2DLCv18 function can be run either as a function or individual script if the results from the intermediate steps are desired.

```matlab
function [opt,t_analysis,total_fit]=analyze_2DLCv18(X)
%Function designed to align multiple LCxLC-DAD chromatograms.
%Inputs:
%X = A four dimensional data set consisting of the second chromatographic
dimension, the first chromatographic dimension, the sample dimension, and
the spectral dimension.
%Output:
%opt is a cell array consisting of the individual decomposed matrices
%corresponding to the four original dimensions of X
%t_analysis is the analysis time (in seconds)

t_initial=cputime;
[second,first,inj,spectra]=size(X);
Xrs=reshape(X,second*first*inj,spectra);
 [~,s]=svd(Xrs,0);
r=1;
while r==1
```

221
Would you like to manually determine the number of components or use the automatic process?

comp_choice=input('Enter 0 for manual and 1 for automatic: ');
comp_choice=round(comp_choice);
if isempty(comp_choice)==1
    disp('You have entered an invalid choice.');
    disp('Please select either 0 or 1.');
else
    if comp_choice<=1&&comp_choice>=0
        break;
    else
        disp('You have entered an invalid choice.');
        disp('Please select either 0 or 1.');
    end
end
if comp_choice==0
    r=1;
    while r==1
        inj_choice=input('Which injection would you like to view for component determination? ');
        inj_choice=round(inj_choice);
        if isempty(inj_choice)==1
            disp('You have entered an invalid injection choice.');
            disp(['Please select an injection between 1 and ' num2str(inj) '.']);
        else
            if inj_choice<=inj&&inj_choice>0
                break;
            else
                disp('You have entered an invalid injection choice.');
                disp(['Please select an injection between 1 and ' num2str(inj) '.']);
            end
        end
        while r==1
            spec_choice=input(['Which wavelength would you like to view injection ' num2str(inj_choice) ' at? ']);
            spec_choice=round(spec_choice);
            if isempty(spec_choice)==1
                disp('You have entered an invalid wavelength choice.');
                disp(['Please select a wavelength between 1 and ' num2str(spectra) '.']);
            else
                if spec_choice<=spectra&&spec_choice>0
                    break;
                else
                    disp('You have entered an invalid wavelength choice.');
                    disp(['Please select a wavelength between 1 and ' num2str(spectra) '.']);
                end
            end
        end
    end
end
while r==1
    NoCX=input('How many components are in the raw data set: ');  
    NoCX=round(NoCX);  
    if isempty(NoCX)==1  
        disp('You have entered an invalid number of components.');  
        disp(['Please select a number of components between 1 and ',num2str(spectra),'.']);  
    else  
        if NoCX>0&&NoCX<=spectra  
            break;  
        else  
            disp('You have entered an invalid number of components.');  
            disp(['Please select a number of components between 1 and ',num2str(spectra),'.']);  
        end  
    end  
end
disp('  ');  
close all;
NoC=zeros(1,inj);
for a=1:inj  
    Xrs=reshape(squeeze(X(:,:,a,:)),second*first,spectra);  
    [~,s]=svd(Xrs,0);  
    subplot(1,2,1); contour(X(:,:,a,spec_choice),50);  
    title(["Injection 'num2str(inj_choice) at Wavelength' num2str(spec_choice)]);  
    xlabel('First dimension');  
    ylabel('Second dimension');  
    subplot(1,2,2); plot(log10(diag(s)), '*');  
    title('SVD plot');  
    xlabel('SVD number');  
    ylabel('log10 SVD');  
end
while r==1
    comp=input(["Select the number of components for injection 'num2str(a) ': ']);  
    comp=round(comp);  
    if isempty(comp)==1  
        disp('You have entered an invalid number of components.');  
        disp(['Please enter a number of components between 1 and ',num2str(spectra),'.']);  
    else  
        NoC=[NoC;comp];  
    end  
end
else
    if comp>0&&comp<=spectra
        NoC(1,a)=comp;
        break;
    else
        disp('You have entered an invalid number of components. ');
        disp([ 'Please enter a number of components between 1 and ' num2str(spectra) '.']);
    end
end
end
if NoC(1,a)>NoCX
    disp('WARNING: You have selected a number of components greater than the number of components');
    disp('chosen for the raw data set. ');
    disp('Do you wish to use the number of components for the raw data set instead? ');
    while r==1
        comp_choice=input('0 for no, 1 for yes: ');
        comp_choice=round(comp_choice);
        if isempty(comp_choice)==1
            disp('You have entered an inappropriate choice. Please enter 0 or 1. ');
        else
            if comp_choice==0||comp_choice==1
                break;
            else
                disp('You have entered an inappropriate choice. Please enter 0 or 1. ');
            end
        end
    end
    if comp_choice==1
        disp([ 'The number of components for injection ' num2str(a) ' has been set to ' num2str(NoCX) ]);
        NoC(1,a)=NoCX;
    end
end
close all;
end
disp(' ');
disp('Det ermined the number of components for each injection.'稍微');
disp(' ');
else
    [NoCX,NoC]=NoCdetv6(X);
end
Xrs=reshape(X,second*first*inj,spectra);
disp('Do you wish to use IKSFA or OPA to generate the initial guesses? ');
while r==1
    IG_choice=input('0 if IKSFA, 1 if OPA: ');
IG_choice=round(IG_choice);
if isempty(IG_choice==1)
    disp('You have entered an inappropriate choice. Please enter 0 or 1.);
else
    if IG_choice==0||IG_choice==1
        break;
    else
        disp('You have entered an inappropriate choice. Please enter 0 or 1.);
    end
end
if IG_choice==0
    disp('Performing IKSFA on the raw data set.');
    disp('');
    [brow]=iksfa(Xrs,NoCX);
    IGX=Xrs(brow,:);
else
    disp('Performing OPA on the raw data set.');
    disp('');
    [brow]=opa(Xrs,NoCX);
    IGX=Xrs(brow,:);
end
if IG_choice==0
    disp('Performing MCR-ALS with non-negativity on the chromatographic dimension on');
    disp('the raw data set.');</
else
    disp('Performing MCR-ALS with non-negativity on the chromatographic dimension on');
end
disp('Required ' num2str(it) ' iterations with a fit error of ' num2str(fit) '%%.');
disp('');
raw_type=zeros(1,NoCX);
MCR=reshape(r_X,second,first,inj,NoCX);
for a=1:NoCX
    plot_inj(squeeze(MCR(:,:,,:,a)),inj);
disp([Are there peaks present for component ' num2str(a) ' 0 if no, 1 if yes: ]);}
while r==1
    choice=input((Blue peaks indicate background components): ');
    choice=round(choice);
    if isempty(choice)==1
        disp('You have entered an inappropriate choice. Please enter 0 or 1.);
    else
        if choice==0||choice==1
            break;
        else
            disp('You have entered an inappropriate choice. Please enter 0 or 1.);
        end
    end
end
end
if choice ==1
    raw_type(1,a)=1;
end
close all;
end
disp("");
bkgdX=find(raw_type==0);
[NN,Ssel]=ALS_constraints(NoCX,spectra,raw_type,s_X);
disp(’Performing MCR-ALS with selected contraints on the raw data set.’);
[~,s_X,~,~,fit]=als_Xv2(Xrs,IGX,NN,Ssel);
disp("");
disp(’Performed MCR-ALS with selected constraints on the raw data.’);
disp([’Required ’ num2str(it) ’ iterations with a fit error of ’ num2str(fit) ’%.’]);
disp("");
ALS=cell(2,inj);
it_ALS=zeros(2,inj);
error_ALS=zeros(2,inj);
for a=1:inj
    Xrs=reshape(squeeze(X(:,:,a,:)),second*first,spectra);
    if IG_choice==0
        [brow]=iksfa(Xrs,NoC(1,a));
        IG=Xrs(brow,:)’;
    else
        [brow]=opa(Xrs,NoC(1,a));
        IG=Xrs(brow,:)’;
    end
    [ALS{1,a},ALS{2,a},it_ALS(1,a),error_ALS(1,a)]=align_als_v2(Xrs,IG);
disp([’Performed MCR-ALS on injection ’ num2str(a) ’ out of ’ num2str(inj) ’ injections.’]);
disp([’Required ’ num2str(it_ALS(1,a)) ’ iterations, with a fit error of’ num2str(error_ALS(1,a)) ’%.’]);
disp("");
end
type=cell(1,inj);
for a=1:inj
    MCR=reshape(ALS{1,a},second,first,NoC(1,a));
type{1,a}=zeros(1,NoC(1,a));
plot_MCRv2(MCR,NoC(1,a));
while r==1
    choice=input([’Are there peak(s) present for injection ’ num2str(a) ’? 0 if no, 1 if yes: ’]);
    choice=round(choice);
    if isempty(choice)==1
        disp(’You have entered an inappropriate choice. Please enter 0 or 1.’);
    else
        if choice==1||choice==0
...
...
...
...
end
end
if choice ==1
    raw_type(1,a)=1;
end
break;
else
    disp('You have entered an inappropriate choice. Please enter 0 or 1.');</end
end
end
if choice ==1
    for b=1:NoC(1,a)
        disp(['Are there peaks present for component ' num2str(b) '? 0 if no, 1 if yes: ']);
        while r==1
            peak=input('(Blue peaks indicate background components): ');
            peak=round(peak);
            if isempty(peak)==1
                disp('You have entered an inappropriate choice. Please enter 0 or 1.');
            else
                if peak==1||peak==0
                    break;
                else
                    disp('You have entered an inappropriate choice. Please enter 0 or 1.');
                end
            end
        end
        if peak==1
            type{1,a}(1,b)=1;
        end
    end
end
disp(' ');close all;
end
opt_b=cell(1,inj);
split=cell(1,inj);
for a=1:inj
    MCR=reshape(ALS{1,a},second,first,NoC(1,a));
    split{1,a}=cell(1,NoC(1,a));
    Csel2=NaN(second,NoC(1,a));
    Csel1=NaN(first,NoC(1,a));
    c=1;
    split_done=zeros(2,NoC(1,a));
    disp(['Injection ' num2str(a) ':']);
    while c<=NoC(1,a)
        if type{1,a}(1,c)==1
            subplot(1,3,1);
            contour(MCR(:,:,c),50);
            title(['Component ' num2str(c) ' for injection ' num2str(a)]);
            xlabel('First dimension');
ylabel('Second dimension');
subplot(1,3,2);
plot(mean(squeeze(MCR(:,:,c)),2));
ymin=min(mean(squeeze(MCR(:,:,c)),2));
ymax=max(mean(squeeze(MCR(:,:,c)),2));
xmin=1;
xmax=second;
axis([xmin xmax ymin ymax]);
title('Mean second dimension');
subplot(1,3,3);
plot(mean(squeeze(MCR(:,:,c)),1));
ymin=min(mean(squeeze(MCR(:,:,c)),1));
ymax=max(mean(squeeze(MCR(:,:,c)),1));
xmin=1;
xmax=first;
axis([xmin xmax ymin ymax]);
title('Mean first dimension');
if split_done(1,c)==0&&split_done(2,c)==0
   disp(['Do you wish to split component ' num2str(c) ' into multiple components and if so how?']);
   while r==1
      choice_split=input('0 if no, 1 for only the 1D, 2 for only the 2D, 3 for both: ');
      choice_split=round(choice_split);
      if isempty(choice_split)==1
         disp('You have entered an inappropriate choice. Please enter 0, 1, 2, or 3.');
      else
         if choice_split>=0&&choice_split<=3
            break;
         else
            disp('You have entered an inappropriate choice. Please enter 0, 1, 2, or 3.');
         end
      end
   end
   if choice_split==2
      while r==1
         comp_increase=input(['How many components do you wish to split component ' num2str(c) ' into? ']);
         comp_increase=round(comp_increase);
         if isempty(comp_increase)==1
            disp('Please select a number of components greater than 1.');
         else
            if comp_increase>=2
               break;
            else
               disp('Please select a number of components greater than 1.');
            end
         end
      end
   end
split\_done(2,c)=1;
split\{1,a\}\{1,c\}=zeros(comp\_increase,1);
split\{1,a\}\{1,c\}(1,1)=c;
comp\_increase=comp\_increase-1;
split\_2D=input([''At which point(s) do you wish to split component ' num2str(c) ' in the second dimension?' ]); start\_2D=1;
Csel2(:,c)=0;
Csel2(start\_2D:spli2\_2D(1,1),c)=NaN;
MCRnew=zeros(second,first,NoC(1,a)+comp\_increase);
MCRnew(:,:,1:NoC(1,a))=MCR;
for b=1:comp\_increase
    type\{1,a\}(1,NoC(1,a)+b)=type\{1,a\}(1,c);
    ALS\{2,a\}(:,NoC(1,a)+b)=ALS\{2,a\}(:,c);
    Csel2(:,NoC(1,a)+b)=0;
    Csel1(:,NoC(1,a)+b)=NaN;
    MCRnew(:,:,NoC(1,a)+b)=MCR(:,:,c);
    if start\_2D==1
        MCRnew(split\_2D(1,b),:,:)=MCRnew(split\_2D(1,b),:,:).*0.5;
        MCRnew(split\_2D(1,b)+1:end,:,:)=0;
    end
    start\_2D=start\_2D(b);
    if b==size(split\_2D,2)
        Csel2(start\_2D:end,NoC(1,a)+b)=NaN;
        MCRnew(start\_2D,:,NoC(1,a)+b)=MCRnew(start\_2D,:,NoC(1,a)+b).*0.5;
        MCRnew(1:start\_2D-1,:,NoC(1,a)+b)=0;
    else
        Csel2(start\_2D:spli2\_2D(1,b+1),NoC(1,a)+b)=NaN;
        MCRnew(1:start\_2D-1,:,NoC(1,a)+b)=0;
        MCRnew(split\_2D(1,b+1)+1:end,:,NoC(1,a)+b)=0;
    end
    split\{1,a\}\{1,c\}(b+1,1)=NoC(1,a)+b;
split\_done(1,NoC(1,a)+b)=0;
split\_done(2,NoC(1,a)+b)=1;
end
MCR=MCRnew;
NoC(1,a)=NoC(1,a)+comp\_increase;
elseif choice\_split==1
    while r==1
        comp\_increase=input([''How many components do you wish to split component ' num2str(c) ' into? ' ]); comp\_increase=round(comp\_increase);
        if isempty(comp\_increase)==1
            disp('Please select a number of components greater than 1.');
else
    \text{if} \ \text{comp\_increase} \geq 2
        \text{break;}
    \text{else}
        \text{disp('Please select a number of components greater than 1.');}
    \text{end}
\text{end}
\text{end}
\text{split\_done}(1,c)=1;
split\{1,a\}\{1,c\} = \text{zeros}(\text{comp\_increase},1);
split\{1,a\}\{1,c\}(1,1)=c;
\text{comp\_increase} = \text{comp\_increase}-1;
\text{split\_1D}=\text{input}(['At which point(s) do you wish to split component ' \text{num2str(c)} ' in the first dimension?']);
\text{start\_1D}=1;
Csel1(:,c)=0;
Csel1(\text{start\_1D}:\text{split\_1D}(1,1),c)=\text{NaN};
\text{MCRnew}=\text{zeros}(\text{second,first,NoC(1,a)+comp\_increase});
\text{MCRnew}(::,1:NoC(1,a))=\text{MCR};
\text{for} \ b=1:\text{comp\_increase}
    \text{type\{1,a\}(1,NoC(1,a)+b)=type\{1,a\}(1,c);
    \text{ALS\{2,a\}(::NoC(1,a)+b)=ALS\{2,a\}(::,c);
    Csel1(:,NoC(1,a)+b)=0;
    Csel2(:,NoC(1,a)+b)=\text{NaN};
    \text{MCRnew}(::,NoC(1,a)+b)=\text{MCR}(::,c);
    \text{if} \ \text{start\_1D}=1
        \text{MCRnew}(::,\text{split\_1D}(1,b),c)=\text{MCRnew}(::,\text{split\_1D}(1,b),c).*0.5;
        \text{MCRnew}(::,\text{split\_1D}(1,b)+1:end,c)=0;
    \text{end}
\text{start\_1D}=\text{split\_1D}(1,b);
\text{if} \ b==\text{size}(\text{split\_1D},2)
    \text{Csel1(\text{start\_1D}:end,NoC(1,a)+b)=\text{NaN};}
    \text{MCRnew(\::,\text{start\_1D},NoC(1,a)+b)=MCRnew(\:,\text{start\_1D},NoC(1,a)+b).*0.5;}
    \text{MCRnew(\:,1:\text{start\_1D}-1,NoC(1,a)+b)=0;}
\text{else}
    \text{Csel1(\text{start\_1D}:end,NoC(1,a)+b)=\text{NaN};}
    \text{MCRnew(\:,1:\text{start\_1D}-1,NoC(1,a)+b)=0;}
    \text{MCRnew(\:,\text{split\_1D}(1,b+1)+1:end,NoC(1,a)+b)=0;}
\text{end}
\text{split\{1,a\}\{1,c\}(b+1,1)=NoC(1,a)+b;
\text{split\_done}(1,NoC(1,a)+b)=1;
\text{split\_done}(2,NoC(1,a)+b)=0;
\text{end}
\text{MCR=\text{MCRnew};}
\text{NoC(1,a)=NoC(1,a)+\text{comp\_increase};}
\text{elseif choice\_split==3
while r==1
    comp_increase=input(['How many components do you wish to split component ' num2str(c) ' into? ']);
    comp_increase=round(comp_increase);
    if isempty(comp_increase)==1
        disp('Please select a number of components greater than 1.);
    else
        if comp_increase>=2
            break;
        else
            disp('Please select a number of components greater than 1.);
        end
    end
    split_done(:,c)=1;
    split{1,a}{1,c}=zeros(comp_increase,1);
    split{1,a}{1,c}(1,1)=c;
    comp_increase=comp_increase-1;
    split_2D=input(['At which point(s) do you wish to split component ' num2str(c) ' in the second dimension? ']);
    split_1D=input(['At which point(s) do you wish to split component ' num2str(c) ' in the first dimension? ']);
    start_2D=1;
    start_1D=1;
    Csel2(:,c)=0;
    Csel2(start_2D:split_1D(1,1),c)=NaN;
    Csel1(:,c)=0;
    Csel1(start_1D:split_2D(1,1),c)=NaN;
    MCRnew=zeros(second,first,NoC(1,a)+comp_increase);
    MCRnew(:,:,1:NoC(1,a))=MCR;
    for b=1:comp_increase
        type{1,a}(1,NoC(1,a)+b)=type{1,a}(1,c);
        ALS{2,a}(1,NoC(1,a)+b)=ALS{2,a}(1,c);
        Csel1(:,NoC(1,a)+b)=0;
        Csel2(:,NoC(1,a)+b)=0;
        MCRnew(:,:,NoC(1,a)+b)=MCR(:,:,c);
        if start_2D==1
            MCRnew(split_2D(1,b)+1:end,:,c)=0;
        end
        if start_1D==1
            MCRnew(:,split_1D(1,b)+1:end,c)=0;
        end
        start_2D=split_2D(1,b);
        start_1D=split_1D(1,b);
        if b==size(split_2D,2)
            Csel2(start_2D:end,NoC(1,a)+b)=NaN;
        end
    end
MCRnew(1:start_2D-1,:,NoC(1,a)+b)=0;
else
    Csel2(start_2D:split_2D(1,b+1),NoC(1,a)+b)=NaN;
    MCRnew(1:start_2D-1,:,NoC(1,a)+b)=0;
    MCRnew(split_2D(1,b+1)+1:end,:,NoC(1,a)+b)=0;
end
if b==size(split_1D,2)
    Csel1(start_1D:end,NoC(1,a)+b)=NaN;
    MCRnew(:,1:start_1D-1,NoC(1,a)+b)=0;
else
    Csel1(start_1D:end,NoC(1,a)+b)=NaN;
    MCRnew(:,1:start_1D-1,NoC(1,a)+b)=0;
    MCRnew(:,split_1D(1,b+1)+1:end,NoC(1,a)+b)=0;
end
split{1,a}{1,c}(b+1,1)=NoC(1,a)+b;
split_done(:,NoC(1,a)+b)=1;
end
MCR=MCRnew;
NoC(1,a)=NoC(1,a)+comp_increase;
end
end
close all;
contour(MCR(:,:,c),50);
title(['Component ' num2str(c) ' for injection ' num2str(a)]);
xlabel('First dimension');
ylabel('Second dimension');
disp(['Do you wish to align the peak within the component for component ' num2str(c) ' ?'])
while r==1
    choice_comp_align=input('0 if no, 1 if yes: ');
    choice_comp_align=round(choice_comp_align);
    if isempty(choice_comp_align)==1
        disp('You have entered an inappropriate choice. Please enter 0 or 1. ');
    else
        if choice_comp_align==0||choice_comp_align==1
            break;
        else
            disp('You have entered an inappropriate choice. Please enter 0 or 1. ');
        end
    end
end
if choice_comp_align==1
    while r==1
        range_1D_start=input('At which first dimension point does the peak start? ');
        range_1D_start=round(range_1D_start);
        if isempty(range_1D_start)==1

232
disp(['You have entered an inappropriate start point. Please choose a point between 1 and ' num2str(first-1) '.']);
else
    if range_1D_start>0&&range_1D_start<first
        break;
    else
        disp(['You have entered an inappropriate start point. Please choose a point between 1 and ' num2str(first-1) '.']);
    end
end
end
end
while r==1
    range_1D_end=input('At which first dimension point does the peak end? ');
    range_1D_end=round(range_1D_end);
    if isempty(range_1D_end)==1
        disp(['You have entered an inappropriate end point. Please choose a point between ' num2str(range_1D_start+1) ' and ' num2str(first) '.']);
    else
        if range_1D_end>range_1D_start&&range_1D_end<=first
            break;
        else
            disp(['You have entered an inappropriate end point. Please choose a point between ' num2str(range_1D_start+1) ' and ' num2str(first) '.']);
        end
    end
end
range_1D=range_1D_start:range_1D_end;
close all;
range_2D=zeros(1,range_1D_end-range_1D_start+1);
spot=1;
for d=range_1D_start:range_1D_end
    plot(MCR(:,d,c));
    title(['Second dimension chromatogram from data point ' num2str(d) ' in the 1D']);
    while r==1
        disp('Which second dimension point is the location of the peak maxima? ');
        [range_2D_input,trash]=ginput(1);
        disp(['You selected data point ' num2str(range_2D_input) ' for second dimension chromatogram from data point ' num2str(d) ' in the 1D']);
        range_2D_input=round(range_2D_input);
        if isempty(range_2D_input)==1
            disp(['You have selected a location outside the range of the second dimension. Please select a location between 1 and ' num2str(second) '.']);
        else
            if range_2D_input>=1&&range_2D_input<=second
                range_2D(1,spot)=range_2D_input;
                break;
            end
        end
    end
else
disp(['You have selected a location outside the range of the second
dimension.  Please select a location between 1 and ' num2str(second) '.']);
end
end

spot=spot+1;
close all;
end

MCR(:,range_1D,c)=comp_shift(MCR(:,range_1D,c),range_2D,second);
end

if split_done(1,c)==1&&split_done(2,c)==1
elseif split_done(1,c)==0&&split_done(2,c)==0
subplot(1,3,1);
contour(MCR(:,c),50);
title(['Component ' num2str(c) ' for injection ' num2str(a)]);
xlabel('First dimension');
ylabel('Second dimension');
subplot(1,3,2);
plot(mean(squeeze(MCR(:,c)),2));

ymin=min(mean(squeeze(MCR(:,c)),2));
ymax=max(mean(squeeze(MCR(:,c)),2));
xmin=1;
xmax=second;
axis([xmin xmax ymin ymax]);
title('Mean second dimension');
subplot(1,3,3);
plot(mean(squeeze(MCR(:,c)),1));

ymin=min(mean(squeeze(MCR(:,c)),1));
ymax=max(mean(squeeze(MCR(:,c)),1));
xmin=1;
xmax=first;
axis([xmin xmax ymin ymax]);
title('Mean first dimension');
disp(['Do you wish to implement chromatographic selectivity for component ' num2str(c) '?'])

while r==1
    choice_csel=input('0 if no, 1 for only the 1D, 2 for only the 2D, 3 for both: ');
    choice_csel=round(choice_csel);
    if isempty(choice_csel)==1
        disp('You have entered an inappropriate choice.  Please enter 0, 1, 2, or 3.');
    else
        if choice_csel>=0&&choice_csel<=3
            break;
        else
        end
    end
disp('You have entered an inappropriate choice. Please enter 0, 1, 2, or 3.'); end
end
end
if choice_csel==2
    disp(['There are ', num2str(second), ' data points in the second dimension.']);
    while r==1
        start=input('What is the start point of the component in the 2D? ');
        start=round(start);
        if isempty(start)==1
            disp(['You have entered an invalid start point. Please enter a number between 1 and ', num2str(second-1), '.']);
        else
            if start>=1&&start<=second-1
                break;
            else
                disp(['You have entered an invalid start point. Please enter a number between 1 and ', num2str(second-1), '.']);
            end
        end
    end
    while r==1
        stop=input('What is the stop point of the component in the 2D? ');
        stop=round(stop);
        if isempty(stop)==1
            disp(['You have entered an invalid stop point. Please enter a number between ', num2str(start), ' and ', num2str(second), '.']);
        else
            if stop>=1&&stop<=second&&stop>start
                break;
            else
                disp(['You have entered an invalid stop point. Please enter a number between ', num2str(start), ' and ', num2str(second), '.']);
            end
        end
        Csel2(1:start-1,c)=0;
        Csel2(stop+1:end,c)=0;
        disp('');
    end
elseif choice_csel==1
    disp(['There are ', num2str(first), ' data points in the first dimension.']);
    while r==1
        start=input('What is the start point of the component in the 1D? ');
        start=round(start);
        if isempty(start)==1
disp('You have entered an invalid start point. Please enter a number between 1 and ' num2str(first-1) '.');
else
    if start>=1&&start<=first-1
        break;
    else
        disp('You have entered an invalid start point. Please enter a number between 1 and ' num2str(first-1) '.');
    end
end
end
end
while r==1
    stop=input('What is the stop point of the component in the 1D? ');
    stop=round(stop);
    if isempty(stop)==1
        disp('You have entered an invalid stop point. Please enter a number between ' num2str(start) ' and ' num2str(first) '.');
    else
        if stop>=1&&stop<=first&&stop>start
            break;
        else
            disp('You have entered an invalid stop point. Please enter a number between ' num2str(start) ' and ' num2str(first) '.');
        end
    end
end
Cset1(1:start-1,c)=0;
Cset1(stop+1:end,c)=0;
disp('');
elseif choice_cset==3
    disp('There are ' num2str(second) ' data points in the second dimension.');
    while r==1
        start=input('What is the start point of the component in the 2D? ');
        start=round(start);
        if isempty(start)==1
            disp('You have entered an invalid start point. Please enter a number between 1 and ' num2str(second-1) '.');
        else
            if start>=1&&start<=second-1
                break;
            else
                disp('You have entered an invalid start point. Please enter a number between 1 and ' num2str(second-1) '.');
            end
        end
    end
end
while r==1
    stop=input('What is the stop point of the component in the 2D? ');
    stop=round(stop);
    if isempty(stop)==1
        disp('You have entered an invalid stop point. Please enter a number between ' num2str(start) ' and ' num2str(second) '.');
    else
        if stop>=1&&stop<=second&&stop>start
            break;
        else
            disp('You have entered an invalid stop point. Please enter a number between ' num2str(start) ' and ' num2str(second) '.');
        end
    end
end
Csel2(1:start-1,c)=0;
Csel2(stop+1:end,c)=0;
disp('');
disp(['There are ' num2str(first) ' data points in the first dimension.']);
while r==1
    start=input('What is the start point of the component in the 1D? ');
    start=round(start);
    if isempty(start)==1
        disp('You have entered an invalid start point. Please enter a number between 1 and ' num2str(first-1) '.');
    else
        if start>=1&&start<=first-1
            break;
        else
            disp('You have entered an invalid start point. Please enter a number between 1 and ' num2str(first-1) '.');
        end
    end
end
while r==1
    stop=input('What is the stop point of the component in the 1D? ');
    stop=round(stop);
    if isempty(stop)==1
        disp('You have entered an invalid stop point. Please enter a number between ' num2str(start) ' and ' num2str(first) '.');
    else
        if stop>=1&&stop<=first&&stop>start
            break;
        else
            disp('You have entered an invalid stop point. Please enter a number between ' num2str(start) ' and ' num2str(first) '.');
        end
    end
end
elseif split_done(1,c)==1&&split_done(2,c)==0
    subplot(1,2,1);
    contour(MCR(:,:,c),50);
    title(['Component ' num2str(c) ' for injection ' num2str(a)]);
    xlabel('First dimension');
    ylabel('Second dimension');
    subplot(1,2,2);
    plot(mean(squeeze(MCR(:,:,c)),2));
    ymin=min(mean(squeeze(MCR(:,:,c)),2));
    ymax=max(mean(squeeze(MCR(:,:,c)),2));
    xmin=1;
    xmax=second;
    axis([xmin xmax ymin ymax]);
    title('Mean second dimension');
    disp(['Do you wish to implement chromatographic selectivity for component ' num2str(c) ' in the 2D?']);
    while r==1
        choice_csel=input('0 if no, 1 if yes: ');
        choice_csel=round(choice_csel);
        if isempty(choice_csel)==1
            disp('You have entered an inappropriate choice. Please enter 0 or 1. ');
        else
            if choice_csel>=0&&choice_csel<=3
                break;
            else
                disp('You have entered an inappropriate choice. Please enter 0 or 1. ');
            end
        end
    end
    if choice_csel==1
        disp(['There are ' num2str(second) ' data points in the second dimension. ']);
        while r==1
            start=input('What is the start point of the component in the 2D? ');
            start=round(start);
            if start>=1&&start<=second-1
                break;
            else
                disp('You have entered an invalid start point. Please enter a number between 1 and ' num2str(second-1) '. ');
            end
        end
    else
        disp('Do you wish to implement chromatographic selectivity for component ' num2str(c) ' in the 2D? ');
        while r==1
            choice_csel=input('0 if no, 1 if yes: ');
            choice_csel=round(choice_csel);
            if isempty(choice_csel)==1
                disp('You have entered an inappropriate choice. Please enter 0 or 1. ');
            else
                if choice_csel>=0&&choice_csel<=3
                    break;
                else
                    disp('You have entered an inappropriate choice. Please enter 0 or 1. ');
                end
            end
        end
    end
elseif split_done(1,c)==1&&split_done(2,c)==0
    subplot(1,2,1);
    contour(MCR(:,:,c),50);
    title(['Component ' num2str(c) ' for injection ' num2str(a)]);
    xlabel('First dimension');
    ylabel('Second dimension');
    subplot(1,2,2);
    plot(mean(squeeze(MCR(:,:,c)),2));
    ymin=min(mean(squeeze(MCR(:,:,c)),2));
    ymax=max(mean(squeeze(MCR(:,:,c)),2));
    xmin=1;
    xmax=second;
    axis([xmin xmax ymin ymax]);
    title('Mean second dimension');
    disp(['Do you wish to implement chromatographic selectivity for component ' num2str(c) ' in the 2D?']);
    while r==1
        choice_csel=input('0 if no, 1 if yes: ');
        choice_csel=round(choice_csel);
        if isempty(choice_csel)==1
            disp('You have entered an inappropriate choice. Please enter 0 or 1. ');
        else
            if choice_csel>=0&&choice_csel<=3
                break;
            else
                disp('You have entered an inappropriate choice. Please enter 0 or 1. ');
            end
        end
    end
    if choice_csel==1
        disp(['There are ' num2str(second) ' data points in the second dimension. ']);
        while r==1
            start=input('What is the start point of the component in the 2D? ');
            start=round(start);
            if start>=1&&start<=second-1
                break;
            else
                disp('You have entered an invalid start point. Please enter a number between 1 and ' num2str(second-1) ' ');
while r==1
    stop=input('What is the stop point of the component in the 2D? ');
    stop=round(stop);
    if stop>=1&&stop<=second&&stop>start
        break;
    else
        disp(['You have entered an invalid stop point. Please enter a number between ' num2str(start) ' and ' num2str(second) '.']);
    end
    Csel2(1:start-1,c)=0;
    Csel2(stop+1:end,c)=0;
    disp('');
end
elseif split_done(1,c)==0&split_done(2,c)==1
    subplot(1,2,1);
    contour(MCR(:,:,c),50);
    title(['Component ' num2str(c) ' for injection ' num2str(a)]);
    xlabel('First dimension');
    ylabel('Second dimension');
    subplot(1,2,2);
    plot(mean(squeeze(MCR(:,:,c)),1));
    ymin=min(mean(squeeze(MCR(:,:,c)),1));
    ymax=max(mean(squeeze(MCR(:,:,c)),1));
    xmin=1;
    xmax=first;
    axis([xmin xmax ymin ymax]);
    title('Mean first dimension');
    disp(['Do you wish to implement chromatographic selectivity for component ' num2str(c) ' in the 1D?']);
    while r==1
        choice_csel=input('0 if no, 1 if yes: ');
        choice_csel=round(choice_csel);
        if isempty(choice_csel)==1
            disp('You have entered an inappropriate choice. Please enter 0 or 1.');
        else
            if choice_csel>=0&choice_csel<=1
                break;
            else
                disp('You have entered an inappropriate choice. Please enter 0 or 1.');
            end
        end
    end
    if choice_csel==1
        if isempty choice_csel==1
            disp('You have entered an inappropriate choice. Please enter 0 or 1.');
        else
            if choice_csel>=0&choice_csel<=1
                break;
            else
                disp('You have entered an inappropriate choice. Please enter 0 or 1.');
            end
        end
        if choice_csel==1
            disp('')
        end
    end
end
disp(['There are ' num2str(first) ' data points in the first dimension.']);
while r==1
    start=input('What is the start point of the component in the 1D? ');
    if isempty(start)==1
        disp(['You have entered an invalid start point. Please enter a number between 1 and ' num2str(first-1) '.']);
    else
        start=round(start);
        if start>=1&&start<=first-1
            break;
        else
            disp(['You have entered an invalid start point. Please enter a number between 1 and ' num2str(first-1) '.']);
        end
    end
end
while r==1
    stop=input('What is the stop point of the component in the 1D? ');
    if isempty(stop)==1
        disp(['You have entered an invalid stop point. Please enter a number between ' num2str(start+1) ' and ' num2str(first) '.']);
    else
        stop=round(stop);
        if stop>=1&&stop<=first&&stop>start
            break;
        else
            disp(['You have entered an invalid stop point. Please enter a number between ' num2str(start+1) ' and ' num2str(first) '.']);
        end
    end
end
Csel1(1:start-1,c)=0;
Csel1(stop+1:end,c)=0;
disp(' ');
end
c=c+1;
end
close all;
ALS_recon=ALS_reconstructv2(MCR,ALS{2,a});
[IG_1,IG_2]=align_3_init(MCR);
[NN,U,Ssel]=PARAFAC_constraintsv2(NoC(1,a),spectra,type{1,a},ALS{2,a});
disp(' ');
disp(['Performing 3way PARAFAC on injection ' num2str(a) '.']);
disp("");
[opt_b{1,a},it_ALS(2,a),error_ALS(2,a)]=align_fPARAFAC3v2(permute(ALS_recon,[3 1 2]),NoC(1,a),IG_1,IG_2,NN,U,Ssel,Csel2,Csel1);
disp('Performed 3way PARAFAC on injection ' num2str(a) ' out of ' num2str(inj) ' injections.');
disp('Required ' num2str(it_ALS(2,a)) ' iterations, with a fit error of ' num2str(error_ALS(2,a)) '%%.');
disp('');
end
Co_s=cell(1,inj);
for a=1:inj
    [Co_s{1,a}]=correlate_spec(s_X,opt_b{1,a}{1},NoCX,NoC(1,a));
end
disp('Correlated raw spectra to spectra of each injection.');
disp('');
spec_match=cell(1,inj);
while r==1
    threshold=input('What threshold (between -1 and 1) will be used for the spectral match? ');
    if isempty(threshold)==1
        disp('You have entered an inappropriate threshold. Please select a threshold between -1 and 1.');
    else
        if threshold>=-1&&threshold<=1
            break;
        else
            disp('You have entered an inappropriate threshold. Please select a threshold between -1 and 1.);
        end
    end
end
disp('');
for a=1:inj
    spec_match{1,a}=spec_matchv4(Co_s{1,a},raw_type,type{1,a},split{1,a},threshold);
end
disp('Matched each injections spectra to the raw data spectra.');
disp('');
for a=1:inj
    for b=1:NoC(1,a)
        if type{1,a}{1}(b)==1
            opt_b{1,a}{2}(;,:)=whitsm(opt_b{1,a}{2}(;,:),9);
        end
    end
end
chrom2d_position=cell(1,inj);
for a=1:inj
    chrom2d_position{1,a}=find_position(opt_b{1,a}{2});
end
disp('Determined the maximum point of each second dimension chromatogram.');
disp('');
disp('What threshold will be used for the second chromatographic dimension match? ');
while r==1
    threshold=round(abs(input('(How many data points +/- from trend line are allowed.): ')));
    if isempty(threshold)==0
        break;
    else
        disp('You have entered an inappropriate threshold.');
    end
end
[chrom2d_trail,chrom2d_split]=chrom2d_matchv8(spec_match,chrom2d_position,raw_type,type,NoCX,NoC,split,threshold);
disp('Matched up the second dimension chromatograms between each injection.');
disp('');
fitpeak=cell(1,inj);
parameters=cell(1,inj);
residual=cell(1,inj);
disp('Interpolation is performed with 0.05 data point increments.');
end
end
for a=1:inj
    if fit_choice==0
        [fitpeak{1,a},parameters{1,a},residual{1,a}]=gaussinterpv3(opt_b{1,a}{1,3},NoC(1,a),type{1,a},first);
    elseif fit_choice==1
        [fitpeak{1,a},parameters{1,a},residual{1,a}]=gaussinterpv2(opt_b{1,a}{1,3},NoC(1,a),type{1,a},first);
    end
end
end
disp('');
disp('Interpolated first chromatographic dimension for each injection.');
disp('');
chrom1d_position=cell(1,inj);
for a=1:inj
    chrom1d_position{1,a}=find_position(fitpeak{1,a});
end
disp('Determined the maximum point of each first dimension chromatogram.');
disp('');
disp('What threshold will be used for the first chromatographic dimension match? ');
while r==1
    threshold=round(abs(input('(How many data points +/- from trend line are allowed.): ')));
    if isempty(threshold)==0
        break;
    else
        disp('You have entered an inappropriate threshold.');
    end
end
disp('');
chrom1d_trail=chrom1d_matchv5(chrom2d_trail,chrom1d_position,chrom2d_split,threshold);
disp('Matched up the first dimension chromatograms between each injection.');
disp('');
align_second=align_chrom2dv3(opt_b,chrom1d_trail,inj,chrom2d_split);
disp('Aligned second dimension chromatograms.');
disp('');
align_first=align_chrom1dv3(fitpeak,chrom1d_trail,chrom2d_split);
disp('Aligned first dimension chromatograms.');
disp('');
disp('Resampling first dimension chromatograms.');
disp('');
sampled_first=resample(align_first);
disp('Resampled first dimension chromatograms. ');
disp('');
disp('Reconstructing data set from individual components.');
disp('');
Xreconstruct=reconstructv2(align_second,sampled_first,opt_b,NoC);
disp('Reconstructed data set from individual components.');
disp('');
[second,first,inj,spectra]=size(Xreconstruct);
Xrs=reshape(Xreconstruct,second*first*inj,spectra);
[NN,Ssel]=ALS_constraints(NoCX,spectra,raw_type,IGX);
disp('');
disp('Performing MCR-ALS on reconstructed data set.');
disp('');
[r_Xrecon,s_Xrecon]=als_X(Xrs,IGX,NN,Ssel);
disp('Performed MCR-ALS on reconstructed data set.');
disp(' ');
MCRX=reshape(r_Xrecon,second,first,inj,NoCX);
raw_split=cell(1,NoCX);
Csel2=NaN(second,NoCX);
Csel1=NaN(first,NoCX);
split_done=zeros(2,size(chrom2d_split,2));
a=1;
NoCX_orig=NoCX;
while a<=NoCX
    if raw_type(1,a)==1
        plot_injv3(squeeze(MCRX(:,:,:,a)),inj)
        if chrom2d_split(1,a)==1
            if a<=NoCX_orig
                disp(['Component ' num2str(a) ' was previously split into ' num2str(size(chrom1d_trail{1,a},2)) ' components.'])
                disp(['How would you like to split component ' num2str(a) '??']);
                while r==1
                    choice_split=input('1 for only the 1D, 2 for only the 2D, 3 for both: ');
                    choice_split=round(choice_split);
                    if isempty(choice_split)==1
                        disp('You have entered an inappropriate choice. Please enter 1, 2, or 3.');
                    else
                        if choice_split>=1&&choice_split<=3
                            break;
                        else
                            disp('You have entered an inappropriate choice. Please enter 1, 2, or 3.');
                        end
                    end
                    comp_increase=size(chrom1d_trail{1,a},2);
                    if choice_split==2
                        split_done(2,a)=1;
                        raw_split{1,a}=zeros(comp_increase,1);
                        raw_split{1,a}(1,1)=a;
                        comp_increase=comp_increase-1;
                        split_2D=input(['At which point(s) do you wish to split component ' num2str(a) ' in the second dimension? ']);
                        start_2D=1;
                        Csel2(:,a)=0;
                        Csel2(start_2D:split_2D(1,1),a)=NaN;
                        MCRnew=zeros(second,first,inj,NoCX+comp_increase);
                        MCRnew(:,:,1:NoCX)=MCRX;
                        for b=1:comp_increase
                            raw_type(1,NoCX+b)=raw_type(1,a);
                            s_Xrecon(:,NoCX+b)=s_Xrecon(:,a);
                        end
                    elseif choice_split==3
                        split_done(2,a)=1;
                        raw_split{1,a}=zeros(comp_increase,1);
                        raw_split{1,a}(1,1)=a;
                        comp_increase=comp_increase-1;
                        split_2D=input(['At which point(s) do you wish to split component ' num2str(a) ' in the second dimension? ']);
                        start_2D=1;
                        Csel2(:,a)=0;
                        Csel2(start_2D:split_2D(1,1),a)=NaN;
                        MCRnew=zeros(second,first,inj,NoCX+comp_increase);
                        MCRnew(:,:,1:NoCX)=MCRX;
                        for b=1:comp_increase
                            raw_type(1,NoCX+b)=raw_type(1,a);
                            s_Xrecon(:,NoCX+b)=s_Xrecon(:,a);
                        end
                    end
                end
            else
                disp(['Component ' num2str(a) ' was previously split into ' num2str(size(chrom1d_trail{1,a},2)) ' components.'])
                disp(['How would you like to split component ' num2str(a) '??']);
                while r==1
                    choice_split=input('1 for only the 1D, 2 for only the 2D, 3 for both: ');
                    choice_split=round(choice_split);
                    if isempty(choice_split)==1
                        disp('You have entered an inappropriate choice. Please enter 1, 2, or 3.');
                    else
                        if choice_split>=1&&choice_split<=3
                            break;
                        else
                            disp('You have entered an inappropriate choice. Please enter 1, 2, or 3.');
                        end
                    end
                end
            else
                disp(['Component ' num2str(a) ' was previously split into ' num2str(size(chrom1d_trail{1,a},2)) ' components.'])
                disp(['How would you like to split component ' num2str(a) '??']);
                while r==1
                    choice_split=input('1 for only the 1D, 2 for only the 2D, 3 for both: ');
                    choice_split=round(choice_split);
                    if isempty(choice_split)==1
                        disp('You have entered an inappropriate choice. Please enter 1, 2, or 3.');
                    else
                        if choice_split>=1&&choice_split<=3
                            break;
                        else
                            disp('You have entered an inappropriate choice. Please enter 1, 2, or 3.');
                        end
                    end
                end
            end
        else
            disp(['Component ' num2str(a) ' was previously split into ' num2str(size(chrom1d_trail{1,a},2)) ' components.'])
            disp(['How would you like to split component ' num2str(a) '??']);
            while r==1
                choice_split=input('1 for only the 1D, 2 for only the 2D, 3 for both: ');
                choice_split=round(choice_split);
                if isempty(choice_split)==1
                    disp('You have entered an inappropriate choice. Please enter 1, 2, or 3.');
                else
                    if choice_split>=1&&choice_split<=3
                        break;
                    else
                        disp('You have entered an inappropriate choice. Please enter 1, 2, or 3.');
                    end
                end
            end
        end
    else
        disp(['Component ' num2str(a) ' was previously split into ' num2str(size(chrom1d_trail{1,a},2)) ' components.'])
        disp(['How would you like to split component ' num2str(a) '??']);
        while r==1
            choice_split=input('1 for only the 1D, 2 for only the 2D, 3 for both: ');
            choice_split=round(choice_split);
            if isempty(choice_split)==1
                disp('You have entered an inappropriate choice. Please enter 1, 2, or 3.');
            else
                if choice_split>=1&&choice_split<=3
                    break;
                else
                    disp('You have entered an inappropriate choice. Please enter 1, 2, or 3.');
                end
            end
        end
    end
end

Csel2(:,NoCX+b)=0;
Csel1(:,NoCX+b)=NaN;
MCRnew(:,;:,NoCX+b)=MCRX(:,;:,a);
if start_2D==1
    MCRnew(start_2D(:,1)+1:end,;:,a)=0;
end
start_2D=start_2D(1:b);
if b==size(split_2D,2)
    Csel2(start_2D:end,2)=NaN;
    MCRnew(1:start_2D-1,;:,2)=0;
    MCRnew(1:start_2D-1,;:,2)=0;
else
    Csel2(start_2D:split_2D(1,b+1),2)=NaN;
    MCRnew(1:start_2D-1,;:,2)=0;
end
raw_split{1,a}(b+1,1)=NoCX+b;
Ssel(:,NoCX+b)=Ssel(:,a);
end
MCRX=MCRnew;
chrom2d_split(1,2)=NoCX+1:NoCX+comp_increase;
split_done(1,2)=0;
split_done(2,2)=1;
NoCX=NoCX+comp_increase;
elseif choice_split==1
    split_done(1,a)=1;
    raw_split{1,a}=zeros(comp_increase,1);
    raw_split{1,a}(1,1)=a;
    comp_increase=comp_increase-1;
    split_1D=input(['At which point(s) do you wish to split component ' num2str(a) ' in the first dimension?']);
    start_1D=1;
    Csel1(:,a)=0;
    Csel1(start_1D:split_1D(1,1),a)=NaN;
    MCRnew=zeros(second,first,inj,NoCX+comp_increase);
    MCRnew(:,;:,1:NoCX)=MCRX;
    for b=1:comp_increase
        raw_type(1,2)=raw_type(1,a);
        s_Xrecon(:,NoCX+b)=s_Xrecon(:,a);
        Csel1(:,NoCX+b)=0;
        Csel2(:,NoCX+b)=NaN;
        MCRnew(:,;:,2)=MCRX(:,;:,a);
        if start_1D==1
            MCRnew(:,;:,2)=0;
        end
        start_1D=start_1D+1:
        if b==size(split_1D,2)
Csel1(start_1D:end,NoCX+b)=NaN;
MCRnew(:,1:start_1D-1,:,NoCX+b)=0;

else
    Csel1(start_1D:end,NoCX+b)=NaN;
    MCRnew(:,1:start_1D-1,:,NoCX+b)=0;
    MCRnew(:,split_1D(1,b)+1:end,:,NoCX+b)=0;
end
raw_split{1,a}(b+1,1)=NoCX+b;
Ssel(:,NoCX+b)=Ssel(:,a);

end
MCRX=MCRnew;
NoCX=NoCX+comp_increase;
chrom2d_split(1,NoCX+1:NoCX+comp_increase)=1;
split_done(1,NoCX+1:NoCX+comp_increase)=1;
split_done(2,NoCX+1:NoCX+comp_increase)=0;
elseif choice_split==3
    split_done(:,a)=1;
    raw_split{1,a}=zeros(comp_increase,1);
    raw_split{1,a}(1,1)=a;
    comp_increase=comp_increase-1;
    split_2D=input(['At which point(s) do you wish to split component ' num2str(a) ' in the second dimension? ']);
    split_1D=input(['At which point(s) do you wish to split component ' num2str(a) ' in the first dimension? ']);
    start_2D=1;
    start_1D=1;
    Csel2(:,a)=0;
    Csel2(start_2D:split_1D(1,1),a)=NaN;
    Csel1(:,a)=0;
    Csel1(start_1D:split_2D(1,1),a)=NaN;
    MCRnew=zeros(second,first,inj,NoCX+comp_increase);
    MCRnew(:,:,1:NoCX)=MCRX;
    for b=1:comp_increase
        raw_type(1,NoCX+b)=raw_type(1,a);
        s_Xrecon(:,NoCX+b)=s_Xrecon(:,a);
        Csel1(:,NoCX+b)=0;
        Csel2(:,NoCX+b)=0;
        MCRnew(:,:,NoC(1,a)+b)=MCRX(:,:,a);
        if start_2D==1
            MCRnew(split_2D(1,b)+1:end,:,a)=0;
        end
        if start_1D==1
            MCRnew(:,split_1D(1,b)+1:end,:,a)=0;
        end
        start_2D=split_2D(1,b);
        start_1D=split_1D(1,b);
if b==size(split_2D,2)
    Csel2(start_2D:end,NoCX+b)=NaN;
    MCRnew(1:start_2D-1,:,:,NoCX+b)=0;
else
    Csel2(start_2D:spli_2D(1,b+1),NoCX+b)=NaN;
    MCRnew(1:start_2D-1,:,:,NoCX+b)=0;
    MCRnew(split_2D(1,b)+1:end,:,:NoCX+b)=0;
end
if b==size(split_1D,2)
    Csel1(start_1D:end,NoCX+b)=NaN;
    MCRnew(:,1:start_1D-1,:,:NoCX+b)=0;
else
    Csel1(start_1D:end,NoCX+b)=NaN;
    MCRnew(:,1:start_1D-1,:,:NoCX+b)=0;
    MCRnew(:,split_1D(1,b)+1:end,:,:NoCX+b)=0;
end
raw_split{1,a}(b+1,1)=NoCX+b;
Ssel(:,NoCX+b)=Ssel(:,a);
end
MCRX=MCRnew;
NoCX=NoCX+comp_increase;
chrom2d_split(1,NoCX+1:NoCX+comp_increas)=1;
split_done(1,NoCX+1:NoCX+comp_increase)=1;
split_done(2,NoCX+1:NoCX+comp_increase)=1;
end
end
end
plot_injv3(squeeze(MCRX(:,:,a)),inj)
if split_done(1,a)==1&&split_done(2,a)==1
elseif split_done(1,a)==1&&split_done(2,a)==0
    disp(['Do you wish to implement chromatographic selectivity for component ' num2str(a) '?'])
    while r==1
        choice_csel=input('0 if no, 1 for the 2D: ');
        choice_csel=round(choice_csel);
        if isempty(choice_csel)==1
            disp('You have entered an inappropriate choice. Please enter 0 or 1.');
        else
            if choice_csel==0||choice_csel==1
                break;
            else
                disp('You have entered an inappropriate choice. Please enter 0 or 1.');
            end
        end
    end
end
if choice_csel==1
    disp(['There are ' num2str(second) ' data points in the second dimension.']);
while r==1
    start=input('What is the start point of the component in the 2D? ');
    start=round(start);
    if isempty(start)==1
        disp(['You have entered an invalid start point. Please enter a number between 1 and ' num2str(second-1) '.']);
    else
        if start>=1&&start<=second-1
            break;
        else
            disp(['You have entered an invalid start point. Please enter a number between 1 and ' num2str(second-1) '.']);
        end
    end
end
while r==1
    stop=input('What is the stop point of the component in the 2D? ');
    stop=round(stop);
    if isempty(stop)==1
        disp(['You have entered an invalid stop point. Please enter a number between ' num2str(start) ' and ' num2str(second) '.']);
    else
        if stop>=1&&stop<=second&&stop>start
            break;
        else
            disp(['You have entered an invalid stop point. Please enter a number between ' num2str(start) ' and ' num2str(second) '.']);
        end
    end
    Csel2(1:start-1,a)=0;
    Csel2(stop+1:end,a)=0;
    disp('');
end
elseif split_done(1,a)==0&&split_done(2,a)==1
    disp(['Do you wish to implement chromatographic selectivity for component ' num2str(a) '?']);
while r==1
    choice_csel=input('0 if no, 1 for the 1D: ');
    choice_csel=round(choice_csel);
    if isempty(choice_csel)==1
        disp('You have entered an inappropriate choice. Please enter 0 or 1.');
    else
        if choice_csel==0||choice_csel==1

break;
else
    disp('You have entered an inappropriate choice. Please enter 0 or 1. ');
end
end
end
if choice_csel==1
    disp(['There are ' num2str(first) ' data points in the first dimension. ']);
while r==1
    start=input('What is the start point of the component in the 1D? ');
    start=round(start);
    if isempty(start)==1
        disp(['You have entered an invalid start point. Please enter a number between 1 and ' num2str(first-1) ' .']);
    else
        if start>=1&&start<=first-1
            break;
        else
            disp(['You have entered an invalid start point. Please enter a number between 1 and ' num2str(first-1) ' .']);
        end
    end
end
while r==1
    stop=input('What is the stop point of the component in the 1D? ');
    stop=round(stop);
    if isempty(stop)==1
        disp(['You have entered an invalid stop point. Please enter a number between ' num2str(start) ' and ' num2str(first) ' .']);
    else
        if stop>=1&&stop<=first&&stop>start
            break;
        else
            disp(['You have entered an invalid stop point. Please enter a number between ' num2str(start) ' and ' num2str(first) ' .']);
        end
    end
end
Csel1(1:start-1,a)=0;
Csel1(stop+1:end,a)=0;
disp(' ');
elseif split_done(1,a)==0&&split_done(2,a)==0
    disp(['Do you wish to implement chromatographic selectivity for component ' num2str(a) '?'])
while r==1


choice_csel=input('0 if no, 1 for the 1D, 2 for the 2D, or 3 for both: ');
choice_csel=round(choice_csel);
if isempty(choice_csel)==1
    disp('You have entered an inappropriate choice. Please enter 0, 1, 2, or 3. ');
else
    if choice_csel>=0||choice_csel<=3
        break;
    else
        disp('You have entered an inappropriate choice. Please enter 0, 1, 2, or 3. ');
    end
end
end
if choice_csel==1
    disp(['There are ' num2str(first) ' data points in the first dimension.']);
    while r==1
        start=input('What is the start point of the component in the 1D? ');
        start=round(start);
        if isempty(start)==1
            disp(['You have entered an invalid start point. Please enter a number between 1 and ' num2str(first-1) '.']);
        else
            if start>=1&&start<=first-1
                break;
            else
                disp(['You have entered an invalid start point. Please enter a number between 1 and ' num2str(first-1) '.']);
            end
        end
    end
    while r==1
        stop=input('What is the stop point of the component in the 1D? ');
        stop=round(stop);
        if isempty(stop)==1
            disp(['You have entered an invalid stop point. Please enter a number between ' num2str(start) ' and ' num2str(first) '.']);
        else
            if stop>=1&&stop<=first&&stop>start
                break;
            else
                disp(['You have entered an invalid stop point. Please enter a number between ' num2str(start) ' and ' num2str(first) '.']);
            end
        end
    end
end
Csel1(1:start-1,a)=0;
Csel1(stop+1:end,a)=0;
disp('');
end
if choice_csel==2
    disp(['There are ' num2str(second) ' data points in the second dimension.']);
    while r==1
        start=input('What is the start point of the component in the 2D? ');
        start=round(start);
        if isempty(start)==1
            disp(['You have entered an invalid start point. Please enter a number between 1 and ' num2str(second-1) '.']);
        else
            if start>=1&&start<=second-1
                break;
            else
                disp(['You have entered an invalid start point. Please enter a number between 1 and ' num2str(second-1) '.']);
            end
        end
    end
    while r==1
        stop=input('What is the stop point of the component in the 2D? ');
        stop=round(stop);
        if isempty(stop)==1
            disp(['You have entered an invalid stop point. Please enter a number between ' num2str(start) ' and ' num2str(second) '.']);
        else
            if stop>=1&&stop<=second&&stop>start
                break;
            else
                disp(['You have entered an invalid stop point. Please enter a number between ' num2str(start) ' and ' num2str(second) '.']);
            end
        end
    end
    Csel2(1:start-1,a)=0;
    Csel2(stop+1:end,a)=0;
    disp('');
end
if choice_csel==3
    disp(['There are ' num2str(second) ' data points in the second dimension.']);
    while r==1
        start=input('What is the start point of the component in the 2D? ');
        start=round(start);
        if isempty(start)==1
            disp(['You have entered an invalid start point. Please enter a number between 1 and ' num2str(second-1) '.']);
        end
    end
end
else
  if start>=1&&start<=second-1
    break;
  else
    disp(['You have entered an invalid start point. Please enter a number between 1 and ' num2str(second-1) '.']);
  end
end
while r==1
  stop=input('What is the stop point of the component in the 2D? ');
  stop=round(stop);
  if isempty(stop)==1
    disp(['You have entered an invalid stop point. Please enter a number between ' num2str(start) ' and ' num2str(second) '.']);
  else
    if stop>=1&&stop<=second&&stop>start
      break;
    else
      disp(['You have entered an invalid stop point. Please enter a number between ' num2str(start) ' and ' num2str(second) '.']);
    end
  end
end
Csel2(1:start-1,a)=0;
Csel2(stop+1:end,a)=0;
disp(' ');
disp(['There are ' num2str(first) ' data points in the first dimension.']);
while r==1
  start=input('What is the start point of the component in the 1D? ');
  start=round(start);
  if isempty(start)==1
    disp(['You have entered an invalid start point. Please enter a number between 1 and ' num2str(first-1) '.']);
  else
    if start>=1&&start<=first-1
      break;
    else
      disp(['You have entered an invalid start point. Please enter a number between 1 and ' num2str(first-1) '.']);
    end
  end
end
while r==1
  stop=input('What is the stop point of the component in the 1D? ');
  stop=round(stop);
if isempty(stop)==1
    disp(['You have entered an invalid stop point. Please enter a number between ' num2str(start) ' and ' num2str(first) '.']);
else
    if stop>=1&&stop<=first&&stop>start
        break;
    else
        disp(['You have entered an invalid stop point. Please enter a number between ' num2str(start) ' and ' num2str(first) '.']);
    end
end
end
Csel1(1:start-1,a)=0;
Csel1(stop+1:end,a)=0;
disp('');
end
a=a+1;
end
[IG_1,IG_2]=align_4_init(MCRX);
[U,Isel,NN]=PARAFACrecon_constrainsv2(NoCX,inj,bkgdX,chrom1d_trail,chrom2d_split,raw_split,NoCX_orig);
disp('');
disp('Performing 4way PARAFAC on reconstructed data set.');
disp('');
[opt,it_PARAFAC4,error_PARAFAC4]=align_fPARAFAC4v3(permute(Xreconstruct,[3 1 2 4]),NoCX,IG_1,IG_2,s_Xrecon,NN,U,Isel,Csel2,Csel1,Ssel);
disp('Performed 4way PARAFAC on the reconstructed data set.');
disp(['Required ' num2str(it_PARAFAC4) ' iterations, with a fit error of ' num2str(error_PARAFAC4) '%.']);
total_fit=100;
for a=1:inj
    total_fit=total_fit-error_ALS(1,a)/inj-error_ALS(2,a)/inj;
end
total_fit=total_fit-error_PARAFAC4;
t_final=cputime;
t_analysis=(t_final-t_initial);

The align_3_init function generates the initial guesses for a single LC×LC-DAD sample for use in a 3-way PARAFAC analysis.
The `align_4_init` function generates the initial guesses for multiple LC×LC-DAD samples for use in a 4-way PARAFAC analysis.
IG_1D(:,r)=mean(temp,1)';
end

The align_als_v2 function performs MCR-ALS.

function [r_opt,s_opt,IT,fit,res_opt]=align_als_v2(data,ig)
[size_C,size_S]=size(data);
comp=size(ig,2);
S=ig;
md=3;
osq=inf;
bsq=inf;
dc=0;
ssqX=sum(sum(data.^2));
for IT=1:200
    C=data*pinv(S');
    for a=1:comp
        C(:,a)=constrain(C(:,a),1,0,0,NaN(size_C,1));
    end
    S=(pinv(C)*data)';
    for a=1:comp
        S(:,a)=constrain(S(:,a),0,0,0,NaN(size_S,1));
    end
    T=C*S';
    res=data-T;
    ssq=sum(sum(res.^2));
    imp=(osq-ssq)/osq;
    if IT==1
        imp=0;
        r_opt=C;
        s_opt=S;
        bsq=ssq;
        res_opt=res;
    end
    if ssq<bsq
        dc=0;
        r_opt=C;
        s_opt=S;
        bsq=ssq;
        res_opt=res;
    end
    if ssq<osq
        dc=0;
    end
    if ssq>osq
dc=dc+1;
end
osq=ssq;
if dc>md
    break
end
if abs(imp<1e-4)&imp>0
    break
end
end
for a=1:comp
    r_opt(:,a)=r_opt(:,a)*norm(s_opt(:,a));
    s_opt(:,a)=s_opt(:,a)/norm(s_opt(:,a));
end
fit=100*sqrt(bsq/ssqX);

The align_chrom1dv3 function aligns 1D peaks between samples.

function chrom1d=align_chrom1dv3(chrom1d_peak,trail,chrom_split)
comp=size(trail,2);
chrom1d=chrom1d_peak;
for a=1:comp
    if isempty(trail{1,a})~=1
        if chrom_split(1,a)==0
            left=min(trail{1,a}(:,4));
            right=max(trail{1,a}(:,4));
            total=size(chrom1d_peak{1,1}(:,1),1);
            middle=round(median(1:total));
            dev_left=abs(middle-left);
            dev_right=abs(middle-right);
            if dev_left>dev_right
                for b=1:size(trail{1,a},1)
                    peak=chrom1d_peak{1,trail{1,a}(b,1)}(:,trail{1,a}(b,2));
                    peak=peak((trail{1,a}(b,4)+1-left):end,1);
                    if size(peak,1)==total
                        spacer=zeros(total-size(peak,1),1);
                        for f=1:size(spacer,1)
                            spacer(f,1)=peak(end,1);
                        end
                        peak=peak';
                        peak=[peak spacer']';
                        chrom1d{1,trail{1,a}(b,1)}(:,trail{1,a}(b,2))=peak;
                    end
                end
            elseif dev_right>dev_left
                for b=1:size(trail{1,a},1)
                    peak=chrom1d_peak{1,trail{1,a}(b,1)}(:,trail{1,a}(b,2));
                    peak=peak((trail{1,a}(b,4)+1-right):end,1);
                    if size(peak,1)==total
                        spacer=zeros(total-size(peak,1),1);
                        for f=1:size(spacer,1)
                            spacer(f,1)=peak(end,1);
                        end
                        peak=peak';
                        peak=[peak spacer']';
                        chrom1d{1,trail{1,a}(b,1)}(:,trail{1,a}(b,2))=peak;
                    end
                end
            end
        elseif chrom_split(1,a)==1
            left=min(trail{1,a}(:,4));
            right=max(trail{1,a}(:,4));
            total=size(chrom1d_peak{1,1}(:,1),1);
            middle=round(median(1:total));
            dev_left=abs(middle-left);
            dev_right=abs(middle-right);
            if dev_left>dev_right
                for b=1:size(trail{1,a},1)
                    peak=chrom1d_peak{1,trail{1,a}(b,1)}(:,trail{1,a}(b,2));
                    peak=peak((trail{1,a}(b,4)+1-left):end,1);
                    if size(peak,1)==total
                        spacer=zeros(total-size(peak,1),1);
                        for f=1:size(spacer,1)
                            spacer(f,1)=peak(end,1);
                        end
                        peak=peak';
                        peak=[peak spacer']';
                        chrom1d{1,trail{1,a}(b,1)}(:,trail{1,a}(b,2))=peak;
                    end
                end
            elseif dev_right>dev_left
                for b=1:size(trail{1,a},1)
                    peak=chrom1d_peak{1,trail{1,a}(b,1)}(:,trail{1,a}(b,2));
                    peak=peak((trail{1,a}(b,4)+1-right):end,1);
                    if size(peak,1)==total
                        spacer=zeros(total-size(peak,1),1);
                        for f=1:size(spacer,1)
                            spacer(f,1)=peak(end,1);
                        end
                        peak=peak';
                        peak=[peak spacer']';
                        chrom1d{1,trail{1,a}(b,1)}(:,trail{1,a}(b,2))=peak;
                    end
                end
            end
        end
    end
end
for b=1:size(trail{1,a},1)
    peak=chrom1d_peak{1, trail{1,a}(b,1)}(:, trail{1,a}(b,2));
    peak=peak(1: (trail{1,a}(b,4)+total-right));
    if size(peak,1)==total
        spacer=zeros(total-size(peak,1),1);
        for f=1:size(spacer,1)
            spacer(f,1)=peak(1,1);
        end
        peak=peak';
        peak=[spacer' peak']';
        chrom1d{1, trail{1,a}(b,1)}(:, trail{1,a}(b,2))=peak;
    end
end
elseif dev_right==dev_left
    for b=1:size(trail{1,a},1)
        peak=chrom1d_peak{1, trail{1,a}(b,1)}(:, trail{1,a}(b,2));
        if trail{1,a}(b,4)==middle
            chrom1d{1, trail{1,a}(b,1)}(:, trail{1,a}(b,2))=peak;
        elseif trail{1,a}(b,4)<middle
            index=trail{1,a}(b,4);
            peak=peak(1: (trail{1,a}(b,4)+total-index));
            if size(peak,1)==total
                spacer=zeros(total-size(peak,1),1);
                for f=1:size(spacer,1)
                    spacer(f,1)=peak(1,1);
                end
                peak=peak';
                peak=[spacer' peak']';
                chrom1d{1, trail{1,a}(b,1)}(:, trail{1,a}(b,2))=peak;
            end
        elseif trail{1,a}(b,4)>middle
            index=trail{1,a}(b,4);
            peak=peak((trail{1,a}(b,4)+1-index):end,1);
            if size(peak,1)==total
                spacer=zeros(total-size(peak,1),1);
                for f=1:size(spacer,1)
                    spacer(f,1)=peak(end,1);
                end
                peak=peak';
                peak=[peak spacer']';
                chrom1d{1, trail{1,a}(b,1)}(:, trail{1,a}(b,2))=peak;
            end
        end
    end
else
for c=1:size(trail{1,a},2)
    left=min(trail{1,a}{1,c}(:,4));
    right=max(trail{1,a}{1,c}(:,4));
    total=size(chrom1d_peak{1,1}{:,1},1);
    middle=round(median(1:total));
    dev_left=abs(middle-left);
    dev_right=abs(middle-right);
    if dev_left>dev_right
        for b=1:size(trail{1,a}{1,c},1)
            peak=chrom1d_peak{1,trail{1,a}{1,c}(b,1)}{1,c}(b,1);
            peak=peak((trail{1,a}{1,c}(b,4)+1-left):end,1);
            if size(peak,1)~=total
                spacer=zeros(total-size(peak,1),1);
                for f=1:size(spacer,1)
                    spacer(f,1)=peak(end,1);
                end
                peak=peak';
                peak=[peak spacer']';
                chrom1d{1,trail{1,a}{1,c}(b,1)}{1,c}(b,1)=peak;
            end
        end
    elseif dev_right>dev_left
        for b=1:size(trail{1,a}{1,c},1)
            peak=chrom1d_peak{1,trail{1,a}{1,c}(b,1)}{1,c}(b,1);
            peak=peak(1:(trail{1,a}{1,c}(b,4)+total-right));
            if size(peak,1)~=total
                spacer=zeros(total-size(peak,1),1);
                for f=1:size(spacer,1)
                    spacer(f,1)=peak(1,1);
                end
                peak=peak';
                peak=['spacer' peak'];
                chrom1d{1,trail{1,a}{1,c}(b,1)}{1,c}(b,1)=peak;
            end
        end
    elseif dev_right==dev_left
        for b=1:size(trail{1,a},1)
            peak=chrom1d_peak{1,trail{1,a}{1,c}(b,1)}{1,c}(b,1);
            if trail{1,a}{1,c}(b,4)==middle
                chrom1d{1,trail{1,a}{1,c}(b,1)}{1,c}(b,2)=peak;
            elseif trail{1,a}{1,c}(b,4)<middle
                index=trail{1,a}{1,c}(b,4);
                peak=peak((trail{1,a}{1,c}(b,4)+total-index));
                if size(peak,1)~=total
                    spacer=zeros(total-size(peak,1),1);
                    for f=1:size(spacer,1)
                        spacer(f,1)=peak(end,1);
                    end
                    peak=peak';
                    peak=[peak spacer']';
                    chrom1d{1,trail{1,a}{1,c}(b,1)}{1,c}(b,1)=peak;
                end
            end
        end
    end
end
spacer(f,1)=peak(1,1);
end
peak=peak';
peak=[spacer' peak'];
chrom1d{1,trail{1,a}{1,c}(b,1)}(:,trail{1,a}{1,c}(b,2))=peak;
end
elseif trail{1,a}{1,c}(b,4)>middle
index=trail{1,a}{1,c}(b,4);
peak=peak((trail{1,a}{1,c}(b,4)+1-index):end,1);
if size(peak,1)~=total
  spacer=zeros(total-size(peak,1),1);
  for f=1:size(spacer,1)
      spacer(f,1)=peak(end,1);
  end
  peak=peak';
  peak=[peak spacer']';
end
chrom1d{1,trail{1,a}{1,c}(b,1)}(:,trail{1,a}{1,c}(b,2))=peak;
end
end
end
end
end
end
end
end
end
end

The align_chrom2dv3 aligns the 2D peaks between samples.

function chrom2d=align_chrom2dv3(chrom2d_peak,trail,samp,chrom_split)
comp=size(trail,2);
chrom2d=cell(1,samp);
for a=1:samp
    chrom2d{1,a}=chrom2d_peak{1,a}{2};
end
for a=1:comp
    if isempty(trail{1,a})~=1
        if chrom_split(1,a)==0
            left=min(trail{1,a}(;,:3));
            right=max(trail{1,a}(;,:3));
            total=size(chrom2d_peak{1,1}{2}(;,1),1);
            middle=round(median(1:total));
            dev_left=abs(middle-left);
            dev_right=abs(middle-right);
            if dev_left>dev_right
                for b=1:size(trail{1,a},1)
peak=chrom2d_peak{1,trail{1,a}(b,1)}{2}(;trail{1,a}(b,2));
peak=peak((trail{1,a}(b,3)+1-left):end,1);
if size(peak,1)==total
    spacer=zeros(total-size(peak,1),1);
    for f=1:size(spacer,1)
        spacer(f,1)=peak(end,1);
    end
    peak=peak';
    peak=[peak spacer']';
    chrom2d{1,trail{1,a}(b,1)}(:,trail{1,a}(b,2))=peak;
end
elseif dev_right>dev_left
    for b=1:size(trail{1,a},1)
        peak=chrom2d_peak{1,trail{1,a}(b,1)}{2}(;trail{1,a}(b,2));
        peak=peak(1:(trail{1,a}(b,3)+total-right));
        if size(peak,1)==total
            spacer=zeros(total-size(peak,1),1);
            for f=1:size(spacer,1)
                spacer(f,1)=peak(1,1);
            end
            peak=peak';
            peak=[spacer' peak']';
            chrom2d{1,trail{1,a}(b,1)}(:,trail{1,a}(b,2))=peak;
        end
    end
elseif dev_right==dev_left
    for b=1:size(trail{1,a},1)
        peak=chrom2d_peak{1,trail{1,a}(b,1)}{2}(;trail{1,a}(b,2));
        if trail{1,a}(b,3)==middle
            chrom2d{1,trail{1,a}(b,1)}(:,trail{1,a}(b,2))=peak;
        elseif trail{1,a}(b,3)<middle
            index=trail{1,a}(b,3);
            peak=peak(1:(trail{1,a}(b,3)+total-index));
            if size(peak,1)==total
                spacer=zeros(total-size(peak,1),1);
                for f=1:size(spacer,1)
                    spacer(f,1)=peak(1,1);
                end
                peak=peak';
                peak=[spacer' peak']';
                chrom2d{1,trail{1,a}(b,1)}(:,trail{1,a}(b,2))=peak;
            end
        elseif trail{1,a}(b,3)>middle
            index=trail{1,a}(b,3);
            peak=peak((trail{1,a}(b,3)+1-index):end,1);
        end
    end
else
if size(peak,1)~=total
    spacer=zeros(total-size(peak,1),1);
    for f=1:size(spacer,1)
        spacer(f,1)=peak(end,1);
    end
    peak=peak';
    peak=[peak spacer']';
    chrom2d{1,trail{1,a}(b,1)}(:,trail{1,a}(b,2))=peak;
end
end
end
else
    for c=1:size(trail{1,a},2)
        left=min(trail{1,a}{1,c}(:,3));
        right=max(trail{1,a}{1,c}(:,3));
        total=size(chrom2d_peak{1,1}{2}(:,1),1);
        middle=round(median(1:total));
        dev_left=abs(middle-left);
        dev_right=abs(middle-right);
        if dev_left>dev_right
            for b=1:size(trail{1,a}{1,c},1)
                peak=chrom2d_peak{1,trail{1,a}{1,c}(b,1)}{2}(:,trail{1,a}{1,c}(b,2));
                peak=peak((trail{1,a}{1,c}(b,3)+1-left):end,1);
                if size(peak,1)~=total
                    spacer=zeros(total-size(peak,1),1);
                    for f=1:size(spacer,1)
                        spacer(f,1)=peak(end,1);
                    end
                    peak=peak';
                    peak=[peak spacer']';
                    chrom2d{1,trail{1,a}{1,c}(b,1)}(:,trail{1,a}{1,c}(b,2))=peak;
                end
            end
        end
        elseif dev_right>dev_left
            for b=1:size(trail{1,a}{1,c},1)
                peak=chrom2d_peak{1,trail{1,a}{1,c}(b,1)}{2}(:,trail{1,a}{1,c}(b,2));
                peak=peak(1:(trail{1,a}{1,c}(b,3)+total-right));
                if size(peak,1)~=total
                    spacer=zeros(total-size(peak,1),1);
                    for f=1:size(spacer,1)
                        spacer(f,1)=peak(1,1);
                    end
                    peak=peak';
                    peak=[spacer' peak']';
                    chrom2d{1,trail{1,a}{1,c}(b,1)}(:,trail{1,a}{1,c}(b,2))=peak;
                end
            end
        end
    end
else
end
The function align_fPARAFAC3v5 performs 3-way PARAFAC analysis on a single LC×LC-DAD sample.

function [opt,IT,fit,res_opt]=align_fPARAFAC3v5(data,comp,ig_1,ig_2,nn,u,S_1,S_2,S_3)
[size_A, size_B, size_C] = size(data);

opt = cell(1, 3);
opt_A = randn(size_A, comp);
opt_B = randn(size_B, comp);
opt_C = randn(size_C, comp);

B = ig_1;
C = ig_2;
osq = Inf;
bsq = Inf;
dc = 0;
md = 3;
ssqX = 0;

for a = 1:size_C
    ssqX = ssqX + sum(sum(data(:,:,a).^2));
end

for IT = 1:2000
    Z = zeros(size_B*size_C, comp);
    Xrs = reshape(data, size_A, size_B*size_C);
    for r = 1:comp
        Z(:, r) = kron(C(:, r), B(:, r));
    end
    A = Xrs * Z * (pinv(Z'*Z));
    for r = 1:comp
        A(:, r) = constrain(A(:, r), nn(1, r), u(1, r), u(1, comp+1), S_1(:, r));
    end
    clear Xrs Xp Z
    Z = zeros(size_A*size_C, comp);
    Xp = permute(data, [2 1 3 4]);
    Xrs = reshape(Xp, size_B, size_A*size_C);
    for r = 1:comp
        Z(:, r) = kron(C(:, r), A(:, r));
    end
    B = Xrs * Z * (pinv(Z'*Z));
    for r = 1:comp
        B(:, r) = constrain(B(:, r), nn(2, r), u(2, r), u(2, comp+1), S_2(:, r));
        if r == 5
            plot(B(:, r))
        end
    end
    clear Xrs Xp Z
end

Z = zeros(size_A*size_B, comp);
Xp = permute(data, [3 1 2 4]);
Xrs = reshape(Xp, size_C, size_A*size_B);
for r = 1:comp
    Z(:, r) = kron(B(:, r), A(:, r));
end

263
C = Xrs*Z*(pinv(Z'*Z));
for r=1:comp
    C(:,r)=constrain(C(:,r),nn(3,r),u(3,r),u(3,comp+1),S_3(:,r));
end
clear Xrs Xp Z
T_sum=0;
T=zeros(size_A,size_B,size_C);
for a=1:size_A
    for b=1:size_B
        for c=1:size_C
            for r=1:comp
                T_sum=T_sum+A(a,r)*B(b,r)*C(c,r);
            end
            T(a,b,c)=T_sum;
            T_sum=0;
        end
    end
end
clear res
res=data-T;
clear T;
ssq=0;
for a=1:size_C
    ssq=ssq+sum(sum(res(:,:,a).^2));
end
imp=(osq-ssq)/osq;
if IT==1
    imp=0;
    opt_A=A;
    opt_B=B;
    opt_C=C;
    bsq=ssq;
    res_opt=res;
end
if ssq<bsq
    dc=0;
    opt_A=A;
    opt_B=B;
    opt_C=C;
    bsq=ssq;
    res_opt=res;
end
if ssq<osq
    dc=0;
end
if ssq>osq
The function `align_fPARAFAC4v3` performs 4-way PARAFAC analysis on multiple LC×LC-DAD samples.

```matlab
function [opt,IT,fit]=align_fPARAFAC4v3(data,comp,ig_1,ig_2,ig_3,nn,u,S_1,S_2,S_3,S_4)
    [size_A,size_B,size_C,size_D]=size(data);
    opt=cell(1,4);
    opt_A=randn(size_A,comp);
    opt_B=randn(size_B,comp);
    opt_C=randn(size_C,comp);
    opt_D=randn(size_C,comp);
    B=ig_1;
    C=ig_2;
    D=ig_3;
    osq=Inf;
    bsq=Inf;
    dc=0;
    md=3;
    ssqX=0;
    for a=1:size_D
        for b=1:size_C
            ssqX=ssqX+sum(sum(data(:,:,b,a).^2));
        end
    end
end
```
for IT=1:500
    Z=zeros(size_B*size_C*size_D,comp);
    Xrs=reshape(data,size_A,size_B*size_C*size_D);
    for r=1:comp
        Z(:,r)=kron(D(:,r),kron(C(:,r),B(:,r)));
    end
    clear A
    A=Xrs*Z*(pinv(Z'*Z));
    for r=1:comp
        A(:,r)=constrain(A(:,r),nn(1,r),u(1,r),u(1,comp+1),S_1(:,r));
    end
    clear Xrs Xp Z
    Z=zeros(size_A*size_C*size_D,comp);
    Xp=permute(data,[2 1 3 4]);
    Xrs=reshape(Xp,size_B,size_A*size_C*size_D);
    for r=1:comp
        Z(:,r)=kron(D(:,r),kron(C(:,r),A(:,r)));
    end
    clear B
    B=Xrs*Z*(pinv(Z'*Z));
    for r=1:comp
        B(:,r)=constrain(B(:,r),nn(2,r),u(2,r),u(2,comp+1),S_2(:,r));
    end
    clear Xrs Xp Z
    Z=zeros(size_A*size_B*size_D,comp);
    Xp=permute(data,[3 1 2 4]);
    Xrs=reshape(Xp,size_C,size_A*size_D*size_B);
    for r=1:comp
        Z(:,r)=kron(D(:,r),kron(B(:,r),A(:,r)));
    end
    clear C
    C=Xrs*Z*(pinv(Z'*Z));
    for r=1:comp
        C(:,r)=constrain(C(:,r),nn(3,r),u(3,r),u(3,comp+1),S_3(:,r));
    end
    clear Xrs Xp Z
    Z=zeros(size_A*size_D*size_B,comp);
    Xp=permute(data,[4 1 2 3]);
    Xrs=reshape(Xp,size_D,size_A*size_C*size_B);
    for r=1:comp
        Z(:,r)=kron(D(:,r),kron(B(:,r),A(:,r)));
    end
    clear D
    D=Xrs*Z*(pinv(Z'*Z));
    for r=1:comp
        D(:,r)=constrain(D(:,r),nn(4,r),u(4,r),u(4,comp+1),S_4(:,r));
    end
end
end
clear Xrs, Xp, Z
T_sum = 0;
T = zeros(size_A, size_B, size_C, size_D);
for a = 1:size_A
    for b = 1:size_B
        for c = 1:size_C
            for d = 1:size_D
                for r = 1:comp
                    T_sum = T_sum + A(a, r)*B(b, r)*C(c, r)*D(d, r);
                end
                T(a, b, c, d) = T_sum;
                T_sum = 0;
            end
        end
    end
end
clear res
res = data - T;
clear T;
ssq = 0;
for a = 1:size_C
    ssq = ssq + sum(sum(res(:, :, a).^2));
end
imp = (osq - ssq) / osq;
if IT == 1
    imp = 0;
    opt_A = A;
    opt_B = B;
    opt_C = C;
    opt_D = D;
    bsq = ssq;
end
if ssq < bsq
    dc = 0;
    opt_A = A;
    opt_B = B;
    opt_C = C;
    opt_D = D;
    bsq = ssq;
end
if ssq < osq
    dc = 0;
end
if ssq > osq
    dc = dc + 1;
end
osq=ssq;
if (dc>md)
  break
end
if (abs(imp<1e-6)&&imp>0)
  break
end
end
for a=1:comp
  opt_A(:,a)=opt_A(:,a)*norm(opt_B(:,a))*norm(opt_C(:,a))*norm(opt_C(:,a));
  opt_B(:,a)=opt_B(:,a)/norm(opt_B(:,a));
  opt_C(:,a)=opt_C(:,a)/norm(opt_C(:,a));
  opt_D(:,a)=opt_D(:,a)/norm(opt_D(:,a));
end
opt{1}=opt_A;
opt{2}=opt_B;
opt{3}=opt_C;
opt{4}=opt_D;
fit=100*sqrt(bsq/ssqX);

The function ALS_constraints sets up the constraints for MCR-ALS analysis.

function [NN,ssel]=ALS_constraints(comp,wave,type,IG)
NN=ones(2,comp);
ssel=NaN(wave,comp);
plot(IG)
r=1;
for a=1:comp
  if type(1,a)==1
    while r==1
      user=input([('At what data point do you wish to implement spectral selectivity for component ' num2str(a) ')? ']);
      user=round(user);
      if isempty(user)==1
        disp([('You have selected an inappropriate data point. Please select a data point between 1 and ' num2str(wave) ').']);
      else
        if user>=1&&user<=wave
          break;
        else
          disp([('You have selected an inappropriate data point. Please select a data point between 1 and ' num2str(wave) ').']);
        end
      end
    end
  end
end
end

else
    NN(2,a)=0;
end
end

function recon=ALS_reconstructv2(chrom,wave)
second=size(chrom,1);
first=size(chrom,2);
comp=size(chrom,3);
spec=size(wave,1);
chrom=reshape(chrom,second*first,comp);
recon=chrom*wave';
recon=reshape(recon,second,first,spec);

The function als_X performs MCR-ALS analysis.

function [r_opt,s_opt,IT,fit]=als_X(data,ig,nn,ssel)
size_C=size(data,1);
comp=size(ig,2);
S=ig;
md=3;
soq=inf;
bsq=inf;
dc=0;
ssqX=sum(sum(data.^2));
for IT=1:200
    C=data*pinv(S');
    for a=1:comp
        C(:,a)=constrain(C(:,a),nn(1,a),0,0,NaN(size_C,1));
    end
    S=(pinv(C)*data');
    for a=1:comp
        S(:,a)=constrain(S(:,a),nn(2,a),0,0,ssel(:,a));
    end
    T=C*S';
    res=data-T;
    ssq=sum(sum(res.^2));
    imp=(osq-ssq)/osq;
    if IT==1
        imp=0;
        r_opt=C;
        s_opt=S;
bsq=ssq;
end
if ssq<bsq
dc=0;
r_opt=C;
s_opt=S;
bsq=ssq;
end
if ssq<osq
dc=0;
end
if ssq>osq
dc=dc+1;
end
osq=ssq;
if dc>md
break
end
if abs(imp<1e-4)&&imp>0
break
end
end
fit=100*sqrt(bsq/ssqX);
for a=1:comp
    r_opt(:,a)=r_opt(:,a)*norm(s_opt(:,a));
s_opt(:,a)=s_opt(:,a)/norm(s_opt(:,a));
end

The als_Xv2 function performs MCR-ALS.

function [r_opt,s_opt,IT,fit,res_opt]=als_Xv2(data,ig,nn,ssel)
size_C=size(data,1);
comp=size(ig,2);
S=ig;
md=3;
osq=inf;
bsq=inf;
dc=0;
ssqX=sum(sum(data.^2));
for IT=1:200
    C=data*pinv(S');
    for a=1:comp
        C(:,a)=constrain(C(:,a),nn(1,a),0,0,NaN(size_C,1));
    end
    S=(pinv(C)*data)';
end
for a=1:comp
    S(:,a)=constrain(S(:,a),nn(2,a),0,0,ssel(:,a));
end
T=C*S';
res=data-T;
ssq=sum(sum(res.^2));
imp=(osq-ssq)/osq;
if IT==1
    imp=0;
    r_opt=C;
    s_opt=S;
    bsq=ssq;
    res_opt=res;
end
if ssq<bsq
    dc=0;
    r_opt=C;
    s_opt=S;
    bsq=ssq;
    res_opt=res;
end
if ssq<osq
    dc=0;
end
if ssq>osq
    dc=dc+1;
end
osq=ssq;
if dc>md
    break
end
if abs(imp<1e-4)&imp>0
    break
end
for a=1:comp
    r_opt(:,a)=r_opt(:,a)*norm(s_opt(:,a));
    s_opt(:,a)=s_opt(:,a)/norm(s_opt(:,a));
end
fit=100*sqrt(bsq/ssqX);

The function chrom1d_matchv5 eliminates false matches from the component matching step of SAMM based on the position of each peak in the 1D.
function [trail]=chrom1d_matchv5(trail_2d,position,trail2d_split,thr)
comp=size(trail_2d,2);
trail=cell(1,comp);
for a=1:comp
    if isempty(trail_2d{1,a})==1
        if trail2d_split(1,a)==0
            samp=size(trail_2d{1,a},1);
            y=zeros(samp,2);
            x=ones(samp,2);
            for b=1:samp
                y(b,1)=position{1,trail_2d{1,a}(b,1)}(1,trail_2d{1,a}(b,2));
                y(b,2)=trail_2d{1,a}(b,2);
                x(b,2)=trail_2d{1,a}(b,1);
            end
            reg=1;
            [m,n]=find(y==0);
            y(m,:)=[];
            x(m,:)=[];
            if isempty(y)==1
                while reg==1
                    lin=x*y;
                    reg=0;
                    dev=zeros(1,size(y,1));
                    for d=1:size(y,1)
                        dev(1,d)=abs((y(d,1)-(lin(2,1)*x(d,2))-lin(1,1)));
                    end
                    [dev_max,dev_index]=max(dev);
                    if dev_max>thr
                        reg=1;
                        y(dev_index,:)=[];
                        x(dev_index,:)=[];
                    end
                end
            trail{1,a}=zeros(size(y,1),4);
            for c=1:size(y,1)
                trail{1,a}(c,1)=x(c,2);
                trail{1,a}(c,2)=y(c,2);
                trail{1,a}(c,4)=y(c,1);
                trail{1,a}(c,3)=trail_2d{1,a}(c,3);
            end
        end
    else
        trail{1,a}=cell(1,size(trail_2d{1,a},2));
        for b=1:size(trail_2d{1,a},2)
            samp=size(trail_2d{1,a}{1,b},1);
        end
    end
end
y=zeros(samp,2);
x=ones(samp,2);
for c=1:samp
    y(c,1)=position{1.trail_2d{1,a}{1,b}(c,1)}{1.trail_2d{1,a}{1,b}(c,2)};
    y(c,2)=trail_2d{1,a}{1,b}(c,2);
    x(c,2)=trail_2d{1,a}{1,b}(c,1);
end
reg=1;
[m,n]=find(y==0);
y(m,:)=[];
x(m,:)=[];
if isempty(y)==1
    while reg==1
        lin=x\y;
        reg=0;
        dev=zeros(1,size(y,1));
        for d=1:size(y,1)
            dev(1,d)=(y(d,1)-(lin(2,1)*x(d,2))-lin(1,1));
        end
        [dev_max,dev_index]=max(dev);
        if dev_max>thr
            reg=1;
            y(dev_index,:)=[];
            x(dev_index,:)=[];
        end
    end
end
trail{1,a}{1,b}=zeros(size(y,1),4);
for c=1:size(y,1)
    trail{1,a}{1,b}(c,1)=x(c,2);
    trail{1,a}{1,b}(c,2)=y(c,2);
    trail{1,a}{1,b}(c,4)=y(c,1);
    trail{1,a}{1,b}(c,3)=trail_2d{1,a}{1,b}(c,3);
end
end
end
end
end

The function chrom2d_matchv8 eliminates false matches from the component matching step of SAMM based on the position of each peak in the $^2$D.
function [trail, trail_split] = chrom2d_matchv8(spec, position, X_type, inj_type, raw_comp, inj_comp, inj_split, thr)
samp = size(spec, 2);
trail = cell(1, raw_comp);
trail_split = zeros(1, raw_comp);
comp_done = cell(1, samp);
for a = 1: samp
    comp_done{1, a} = zeros(1, inj_comp{1, a});
end
for a = 1: samp
    for b = 1: inj_comp{1, a}
        if inj_type{1, a}(1, b) == 0
            comp_done{1, a}(1, b) = 1;
        end
    end
end
for a = 1: raw_comp
    if X_type{1, a} == 1
        comp_split = 0;
        size_split = 0;
        for b = 1: samp
            if isempty(spec{1, b}{a, 1}) ~= 1
                comp = spec{1, b}{a, 1}(1, 1);
                if size(inj_split{1, b}{1, comp}, 1) == 0
                    comp_split = 1;
                    if size(spec{1, b}{a, 1}, 2) > size_split
                        size_split = size(spec{1, b}{a, 1}, 2);
                    end
                end
            end
        end
        if comp_split == 0
            y = zeros(samp, 2);
            x = ones(samp, 2);
            for b = 1: samp
                if isempty(spec{1, b}{a, 1}) ~= 1 && comp_done{1, b}(1, spec{1, b}{a, 1}) == 0
                    x(b, 2) = b;
                    y(b, 1) = position{1, b}(1, spec{1, b}{a, 1});
                    y(b, 2) = spec{1, b}{a, 1};
                end
            end
            reg = 1;
            [m, n] = find(y == 0);
            y(m, :) = [];
            x(m, :) = [];
        end
    end
end
274
if isempty(y)~=1
    while reg==1;
        lin=x*y(:,1);
        reg=0;
        dev=zeros(1,size(y,1));
        for d=1:size(y,1)
            dev(d)=abs(y(d,1)-(lin(2,1)*x(d,2))-lin(1,1));
        end
        [dev_max,dev_index]=max(dev);
        if dev_max>thr;
            reg=1;
            y(dev_index,:)=[];
            x(dev_index,:)=[];
        end
    end
end
trail{1,a}=zeros(size(y,1),3);
for c=1:size(y,1)
    trail{1,a}(c,1)=x(c,2);
    trail{1,a}(c,2)=y(c,2);
    trail{1,a}(c,3)=y(c,1);
    comp_done{1,x(c,2)}(1,y(c,2))=1;
end
end
else
    trail{1,a}=cell(1,size_split);
    trail_split(1,a)=1;
    b=1;
    while b<=size_split
        y=zeros(samp,2);
        x=ones(samp,2);
        for c=1:samp
            if b<=size(spec{1,c}{a,1},2)
                comp=spec{1,c}{a,1}(1,b);
                if isempty(spec{1,c}{a,1})~=1 && comp_done{1,c}(1,comp)==0 && b<=size(spec{1,c}{a,1},2)
                    x(c,2)=c;
                    y(c,1)=position{1,c}(1,spec{1,c}{a,1}(1,b));
                    y(c,2)=spec{1,c}{a,1}(1,b);
                    comp_done{1,c}(1,spec{1,c}{a,1}(1,b))=1;
            end
        end
    end
    for d=1:b
        if d<=size(spec{1,c}{a,1},2)
            if comp_done{1,c}(1,spec{1,c}{a,1}(1,d))=0
                x(c,2)=c;
                y(c,1)=position{1,c}(1,spec{1,c}{a,1}(1,d));
            end
        end
    end
end
The function comp_shift is used to correct for intra-sample retention time shifting.

function output=comp_shift(MCR_inj,shift_range,size_2D)
output=MCR_inj;
for a=1:size(shift_range,2)
    peak=squeeze(MCR_inj(:,a));
    peak=peak((shift_range(1,a)-min(shift_range)+1):end,1);
    spacer=zeros(1,size_2D-size(peak,1));
    for f=1:size(spacer,2)
        spacer(1,f)=peak(end,1);
    end
    peak=peak';
    peak=[peak spacer]';
    output(:,a)=peak;
end

The function constrainv2 is used to implement constraints during MCR-ALS and PARAFAC analysis.

function
output=constrainv2(input,nonnegativity,unimodality,dir,spec_sel,chrom2_sel,chrom1_sel)
rows=size(input,1);
%NoCc=size(input,2);
output=input;
% implementation of selectivity
%if min(selectivity.*selectivity==selectivity)    % original implementation of selectivity (only zeros, ones and Inf squared the same)
%    output(:,:)=output(:,:).*selectivity;
%else                           % new implementation
%    for scan=1:rows,
%        for comp=1:NoCc,
%            if not(selectivity(scan,comp)~=selectivity(scan,comp))     % only replace non NAN values
%                output(scan,comp)=selectivity(scan,comp);
%            end
%        end
%    end
% implementation of non_negativity
%for comp=1:NoCc,
%    if nonnegativity(comp)>0
%        output(:,comp)=non_negativity(output(:,comp));
%    end
%end
% implementation of unimodality
%for comp=1:NoCc,
%    if unimodality(comp)~=0                        % apply unimodality when a tolerance is given
The function correlate_spec is used to match the reference spectra to the resolved spectra for each sample from 3-way PARAFAC analysis.
function [Cospec_r]=correlate_spec(spec_X,spec_inj,comp_X,comp_inj)
Cospec_r=cell(comp_X,comp_inj);
for b=1:comp_X
    for c=1:comp_inj
        Cospec_r{b,c}=pearson(spec_X(:,b)',spec_inj(:,c)');
    end
end

The function emgfitv2 fits an EMG model to the resolved 1D component profile.

function [fx]=emgfitv2(params,data,time)
area=params(1,1);
position=params(1,2);
sigma=params(1,3);
tau=params(1,4);
bkgd=params(1,5);
peak=area/(2*tau)*exp(sigma^2/(2*tau^2)+(position-time)/sigma).*erf((time-position)/(2^0.5*sigma)-sigma/(2^0.5*tau))+1);
peak=peak+bkgd;
fx=data-peak';

The function emggenv2 generates a EMG curve based on the fitting parameters previously obtained by emgfitv2.

function [peak]=emggenv2(params,time)
area=params(1,1);
position=params(1,2);
sigma=params(1,3);
tau=params(1,4);
bkgd=params(1,5);
peak=area/(2*tau)*exp(sigma^2/(2*tau^2)+(position-time)/sigma).*erf((time-position)/(2^0.5*sigma)-sigma/(2^0.5*tau))+1);
peak=peak+bkgd;

The function find_position finds the maximum position for each 2D and 1D resolved component profile.

function position=find_position(chrom_1d)
position=zeros(1,size(chrom_1d,2));
for b=1:size(chrom_1d,2)
The function gaussb2 generates a Gaussian curve.

```matlab
function [peak]=gaussb2(params,time)
area=params(1,1);
position=params(1,2);
sigma=params(1,3);
bkgd=params(1,4);
peak=area/(sigma*(2*pi)^0.5)*exp((-position-time).^2./(2*sigma^2));
peak=peak+bkgd;
```

The function gaussfitb2 fits a Gaussian function to the resolved 1D component profiles.

```matlab
function [fx]=gaussfitb2(params,data,time)
area=params(1,1);
position=params(1,2);
sigma=params(1,3);
bkgd=params(1,4);
peak=area/(sigma*(2*pi)^0.5)*exp((-position-time).^2./(2*sigma^2));
peak=peak+bkgd;
fx=data-peak';
```

The function gaussinterpv2 fits and then generates an EMG curve.

```matlab
function [gaussfit,Xg,res]=gaussinterpv2(data,comp,comp_type,first)
time=1:first;
timexx=1:0.05:first;
options=optimset;
options.MaxFunEvals=1000;
options.MaxIter=100;
options.TolFun=1e-8;
options.TolX=1e-4;
options.DiffMinChange=1e-6;
options.MaxTime=60;
[Value,Index]=max(data);
Peakarea=zeros(1,comp);
for b=1:comp
    if Index(1,b)==1&&Index(1,b)==size(data,1)
        Peakarea(1,b)=data(Index(1,b)-1,b)+data(Index(1,b),b)+data(Index(1,b)+1,b);
    end
end
```
if Index(1,b) == 1
    Peakarea(1,b) = data(Index(1,b),b) + data(Index(1,b)+1,b);
end
if Index(1,b) == size(data,1)
    Peakarea(1,b) = data(Index(1,b),b) + data(Index(1,b)-1,b);
end
end
Xs = zeros(size(timexx,2),comp);
for n = 1:comp
    Xs(:,n) = spline(time, data(:,n), timexx);
end
[value, index] = max(Xs);
Position = zeros(1,comp);
for b = 1:comp
    Position(1,b) = index(1,b)*0.05 + 1;
end
Sigma = 0.6;
temp_peak = zeros(size(timexx,2),comp);
Xg = zeros(comp,5);
res = zeros(comp,1);
for n = 1:comp
    if comp_type(1,n) == 1
        params = [Peakarea(1,n) Position(1,n) Sigma 1 min(data(:,n))];
        Xsub = data(:,n);
        [Xg(n,:), res(n,1)] = lsqnonlin(@emgfitv2, params, [0.5*Peakarea(1,n) 0.75*Position(1,n) 0.4 .25 1e-30], [1.5*Peakarea(1,n) 1.25*Position(1,n) 0.8 1.75 1e-29], options, Xsub, time);
        temp_peak(:,n) = emgenv2(Xg(n,:), timexx)';
    else
        Xsub = data(:,n);
        temp_peak(:,n) = interp1(time, Xsub', timexx)';
    end
end
gaussfit = temp_peak;

The function gaussinterpv3 fits and then generates a Gaussian curve.

function [gaussfit, Xg, res] = gaussinterpv3(data, comp, comp_type, first)
time = 1:first;
timexx = 1:0.05:first;
options = optimset;
options.MaxFunEvals = 1000;
options.MaxIter = 100;
options.TolFun = 1e-8;
options.TolX = 1e-4;
options.DiffMinChange=1e-6;
options.MaxTime=60;
[Value,Index]=max(data);
Peakarea=zeros(1,comp);
for b=1:comp
    if Index(1,b)==1&&Index(1,b)==size(data,1)
        Peakarea(1,b)=data(Index(1,b),b)+data(Index(1,b)+1,b);
    end
    if Index(1,b)==1
        Peakarea(1,b)=data(Index(1,b),b)+data(Index(1,b)+1,b);
    end
    if Index(1,b)==size(data,1)
        Peakarea(1,b)=data(Index(1,b),b)+data(Index(1,b)-1,b);
    end
end
Xs=zeros(size(timexx,2),comp);
for n=1:comp
    Xs(:,n)=spline(time,data(:,n),timexx);
end
[value,index]=max(Xs);
Position=zeros(1,comp);
for b=1:comp
    Position(1,b)=index(1,b)*0.05+1;
end
Sigma=0.59623;
temp_peak=zeros(size(timexx,2),comp);
Xg=zeros(comp,4);
res=zeros(comp,1);
for n=1:comp
    if comp_type(1,n)==1
        params=[Peakarea(1,n) Position(1,n) Sigma min(data(:,n))];
        Xsub=data(:,n);
        [Xg(n,:),res(n,1)]=lsqnonlin(@gaussfitb2,params,[0.5*Peakarea(1,n) 0.75*Position(1,n) 0.4 1e-30],[1.5*Peakarea(1,n) 1.25*Position(1,n) 0.8 1e-29],options,Xsub,time);
        temp_peak(:,n)=gaussb2(Xg(n,:),timexx)';
    else
        Xsub=data(:,n);
        temp_peak(:,n)=interp1(time,Xsub',timexx)';
    end
end
gaussfit=temp_peak;

The function iksfa performs IKSFA.

function [brow,maxdet]=iksfa(d,num)
[r,c]=size(d);
if num>c
    fprintf('Too many factors');
    return;
end
[u,s]=svd(d,'econ');
val=zeros(num,1);
row=zeros(num,1);
nu=zeros(r,num);
for i=1:r
    for k=1:num
        den=0;
        for j=1:num
            den=den+u(i,j)^2*s(j,j)^2;
        end
        den=den^0.5;
        nu(i,k)=u(i,k)*s(k,k)/den;
    end
end
[val(1,1),row(1,1)]=max(abs((nu(:,1))));
detnu=zeros(r,1);
for i=2:num
    for j=1:r
        detnu(j,1)=det([nu(row(1:i-1),1:i);nu(j,1:i)]);
    end
    [val(i),row(i)]=max(detnu);
end
brow=row;
for z=1:1000
    arow=brow;
    for i=1:num
        trow=brow;
        for j=1:r
            trow(i)=j;
            detnu(j,1)=det(nu(trow(1:num),1:num));
            if val(num)<(detnu(j,1))
                val(num)=(detnu(j,1));
                brow=trow;
            end
        end
    end
    irow=arow;
if irow==brow, break; end
end
maxdet=det(nu(brow(1:num),1:num));

The function NoCdetv6 estimates the number of components by fitting a trend line to the
diagonal elements of \( S \).

```matlab
function [NoCX2,NoC2]=NoCdetv6(X)
[second,first,inj,spectra]=size(X);
Xrs=reshape(X,second*first*inj,spectra);
 [~,s]=svd(Xrs,0);
 x=ones(spectra,2);
y=zeros(spectra,1);
m=log10(diag(s));
for a=1:spectra
 x(a,2)=a;
 y(a,1)=m(a);
end
lin=x\y;
dev=zeros(spectra,1);
for d=1:spectra
 dev(d,1)=(y(d,1)-(lin(2,1)*x(d,2))-lin(1,1))^2;
end
dev_std=std(dev);
a=1;
b=1;
while a==1
 if dev(b,1)<dev_std
  a=0;
  NoCX2=b-1;
 end
  b=b+1;
end
NoC2=zeros(1,inj);
for a=1:inj
 Xrs=reshape(squeeze(X(:,:,a,:)),second*first,spectra);
 [~,s]=svd(Xrs,0);
 x=ones(spectra,2);
y=zeros(spectra,1);
m=log10(diag(s));
for t=1:spectra
 x(t,2)=t;
 y(t,1)=m(t);
end
lin=x\y;
```

284
dev=zeros(spectra,1);
for d=1:spectra
    dev(d,1)=(y(d,1)-(lin(2,1)*x(d,2))-lin(1,1))^2;
end
dev_std=std(dev);
t=1;
b=1;
while t==1
    if dev(b,1)<dev_std
        t=0;
        NoC2(1,a)=b-1;
    end
    b=b+1;
end

The function non-negativity applies non-negativity to a component.

function vector_out=non_negativity(vector_in);

% This function returns the non_negative part of an input vector
% Usage output=non_negativity(input);
%
% The negative numbers in a vector are replaced by zeros, by subtracting
% the absolute vector from itself and dividing the answer by two.

vector_out=(vector_in+abs(vector_in))/2;
vector_out=vector_out+1e-30;

The function opa performs OPA.

function [ind_opt,ind_opt_norm,di,di_norm,maxdet]=opa(Xrs,NoC)
ind_opt_norm=0;
di_norm=0;
[r,c]=size(Xrs);
Xrs_norm=zeros(r,c);
di=zeros(r,c);
R=mean(Xrs,1);
ind=zeros(NoC,1);
val=zeros(NoC,1);
for a=1:r
    denom=0;
    for b=1:c
denom=denom+Xrs(a,b)^2;
end
Xrs_norm(a,:)=Xrs(a,:)/(sqrt(denom));
end
for n=1:NoC
  for a=1:r
    Xm=[R;Xrs(a,:)];
    di(a,n)=det(Xm*Xm');
  end
  [val(n,1),ind(n,1)]=max(di(:,n));
  if n==1
    R=Xrs(ind(n,1),:);
  else
    R=[R;Xrs(ind(n,1),:)];
  end
end
ind_start=ind;
for z=1:1000
  ind_opt=ind_start;
  for a=1:NoC
    ind_test=ind_start;
    for b=1:r
      ind_test(a,1)=b;
      Xm=Xrs(ind_test,:);
      di_test=det(Xm*Xm');
      if val(a,1)<di_test
        val(a,1)=di_test;
        ind_start=ind_test;
      end
    end
  end
  if ind_opt==ind_start, break; end
end
maxdet=det(Xrs(ind_opt,:)*Xrs(ind_opt,:));

The function PARAFAC_constraintsv2 sets up the constraints for three-way PARAFAC analysis.

function [NN,U,ssel]=PARAFAC_constraintsv2(comp,wave,comp_type,ALS)
NN=ones(3,comp);
ssel=NaN(wave,comp);
U=zeros(3,comp+1);
U(2:3,1:comp)=1;
U(2:3,comp+1)=3;

r=1;

for b=1:size(comp_type,2)
    if comp_type(1,b)==1
        while r==1
            plot(ALS(:,b))
            user=input(['At what data point do you wish to implement spectral selectivity for component ' num2str(b) '? ']);
            if isempty(user)==1
                disp(['You have entered an incorrect spectra point. Please enter a number between 1 and ' num2str(wave)]);
            else
                user=round(user);
                if user<=0 || user>wave
                    disp(['You have entered an incorrect spectra point. Please enter a number between 1 and ' num2str(wave)]);
                else
                    break
                end
            end
        end
        ssel(user:wave,b)=0;
        while r==1
            disp(['Do you wish to implement unimodality on component ' num2str(b) '? '])
            user=input('0 for no, 1 for yes: ');
            if isempty(user)==1
                disp('You have entered an inappropriate choice. Please enter 0 or 1.')
            else
                user=round(user);
                if user<0 || user>1
                    disp('You have entered an inappropriate choice. Please enter 0 or 1.')
                else
                    if user==1
                        U(2:3,b)=1;
                    end
                    break
                end
            end
        end
    end
    if comp_type(1,b)==0
        NN(1,b)=0;
    end
end
The function PARAFACrecon_constraints sets up the constraints for four-way PARAFAC analysis.

```matlab
function [u,isel,nn]=PARAFACrecon_constraints(comp,samp,bkgd,presence)
    u=zeros(4,comp+1);
    u(2:3,1:comp)=1;
    u(2:3,comp+1)=3;
    isel=zeros(samp,comp);
    nn=ones(4,comp);
    for a=1:size(bkgd,2)
        u(2:3,bkgd(1,a))=0;
        isel(:,bkgd(1,a))=NaN;
        nn(4,bkgd(1,a))=0;
    end
    for a=1:comp
        if isempty(presence{1,a})~=1
            for b=1:size(presence{1,a},1)
                isel(presence{1,a}(b,1),a)=NaN;
            end
        end
    end
end
```

The function pearson calculates the Pearson correlation coefficient.

```matlab
function [r]=pearson(X,Y)
    mean_X=mean(X);
    mean_Y=mean(Y);
    num=0;
    n=size(X,2);
    for a=1:n
        num=num+(X(1,a)-mean_X)*(Y(1,a)-mean_Y);
    end
    den_1=0;
    for a=1:n
        den_1=den_1+(X(1,a)-mean_X)^2;
    end
    den_2=0;
    for a=1:n
        den_2=den_2+(Y(1,a)-mean_Y)^2;
    end
    r=num/(sqrt(den_1)*sqrt(den_2));
end
```

The function plot_inj plots the resolved MCR-ALS chromatograms after reconstruction.

```matlab
function plot_inj(MCR_inj,inj)
```

288
figure
vmax = max(max(max(MCR_inj(:, :, :))));
 vmin = min(min(min(MCR_inj(:, :, :))));
if inj == 1
    contour(MCR_inj(:, :, 1), 100)
    title(['Injection ' num2str(inj)]);
    colorbar('southoutside', 'Position', [0.130 0.050 0.780 0.03])
caxis([vmin vmax])
h = gca;
set(h, 'xticklabel', '')
set(h, 'yticklabel', '')
endif inj == 2
for b = 1:size(MCR_inj, 3)
    subplot(2, 1, b); contour(MCR_inj(:, :, b), 100);
    title(['Injection ' num2str(b)]);
    h = gca;
    set(h, 'xticklabel', '')
    set(h, 'yticklabel', '')
caxis([vmin vmax])
end
colorbar('southoutside', 'Position', [0.130 0.050 0.780 0.03])
elseif inj <= 4
for b = 1:size(MCR_inj, 3)
    subplot(2, 2, b); contour(MCR_inj(:, :, b), 100);
    title(['Injection ' num2str(b)]);
    h = gca;
    set(h, 'xticklabel', '')
    set(h, 'yticklabel', '')
caxis([vmin vmax])
end
colorbar('southoutside', 'Position', [0.130 0.050 0.780 0.03])
elseif inj <= 6
for b = 1:size(MCR_inj, 3)
    subplot(3, 2, b); contour(MCR_inj(:, :, b), 100);
    title(['Injection ' num2str(b)]);
    h = gca;
    set(h, 'xticklabel', '')
    set(h, 'yticklabel', '')
caxis([vmin vmax])
end
colorbar('southoutside', 'Position', [0.130 0.050 0.780 0.03])
elseif inj <= 9
for b = 1:size(MCR_inj, 3)
    subplot(3, 3, b); contour(MCR_inj(:, :, b), 100);
    title(['Injection ' num2str(b)]);
    h = gca;
end
colorbar('southoutside', 'Position', [0.130 0.050 0.780 0.03])
elseif inj <= 9
for b = 1:size(MCR_inj, 3)
    subplot(3, 3, b); contour(MCR_inj(:, :, b), 100);
    title(['Injection ' num2str(b)]);
    h = gca;
end
colorbar('southoutside', 'Position', [0.130 0.050 0.780 0.03])
```matlab
set(h,'xticklabel','')
set(h,'yticklabel','')
caxis([vmin vmax])
end
colorbar('southoutside','Position',[0.130 0.050 0.780 0.03])
elseif inj<=12
    for b=1:size(MCR_inj,3)
        subplot(4,3,b);contour(MCR_inj(:,:,b),100);
        title(['Injection ' num2str(b)]);
        h=gca;
        set(h,'xticklabel','')
        set(h,'yticklabel','')
        caxis([vmin vmax])
    end
colorbar('southoutside','Position',[0.130 0.050 0.780 0.03])
elseif inj<=16
    for b=1:size(MCR_inj,3)
        subplot(4,4,b);contour(MCR_inj(:,:,b),100);
        title(['Injection ' num2str(b)]);
        h=gca;
        set(h,'xticklabel','')
        set(h,'yticklabel','')
        caxis([vmin vmax])
    end
colorbar('southoutside','Position',[0.130 0.050 0.780 0.03])
elseif inj<=20
    for b=1:size(MCR_inj,3)
        subplot(5,4,b);contour(MCR_inj(:,:,b),100);
        title(['Injection ' num2str(b)]);
        h=gca;
        set(h,'xticklabel','')
        set(h,'yticklabel','')
        caxis([vmin vmax])
    end
colorbar('southoutside','Position',[0.130 0.050 0.780 0.03])
elseif inj<=25
    for b=1:size(MCR_inj,3)
        subplot(5,5,b);contour(MCR_inj(:,:,b),100);
        title(['Injection ' num2str(b)]);
        h=gca;
        set(h,'xticklabel','')
        set(h,'yticklabel','')
        caxis([vmin vmax])
    end
colorbar('southoutside','Position',[0.130 0.050 0.780 0.03])
elseif inj<=30
```

290
for b=1:size(MCR_inj,3)
    subplot(6,5,b);contour(MCR_inj(:,:,b),100);
    title(['Injection ' num2str(b)]);
    h=gca;
    set(h,'xticklabel','
    set(h,'yticklabel','
    caxis([vmin vmax])
end

elseif inj<=36
for b=1:size(MCR_inj,3)
    subplot(6,6,b);contour(MCR_inj(:,:,b),100);
    title(['Injection ' num2str(b)]);
    h=gca;
    set(h,'xticklabel','
    set(h,'yticklabel','
    caxis([vmin vmax])
end

elseif inj<=42
for b=1:size(MCR_inj,3)
    subplot(7,6,b);contour(MCR_inj(:,:,b),100);
    title(['Injection ' num2str(b)]);
    h=gca;
    set(h,'xticklabel','
    set(h,'yticklabel','
    caxis([vmin vmax])
end

elseif inj<=49
for b=1:size(MCR_inj,3)
    subplot(7,7,b);contour(MCR_inj(:,:,b),100);
    title(['Injection ' num2str(b)]);
    h=gca;
    set(h,'xticklabel','
    set(h,'yticklabel','
    caxis([vmin vmax])
end

end

The function plot_MCRv2 plots the resolved MCR-ALS chromatograms.

function plot_MCRv2(MCR_inj,inj)
figure
vmax=max(max(max(MCR_inj(:,:,1))));
vmin=min(min(min(MCR_inj(:,:,1))));
if inj==1
    contour(MCR_inj(:,:,1),100)
    title(['Component ' num2str(inj)]);
    colorbar('southoutside','Position',[0.130 0.050 0.780 0.03])
    caxis([vmin vmax])
    h=gca;
    set(h,'xticklabel','' )
    set(h,'yticklabel','' )
else if inj==2
    for b=1:size(MCR_inj,3)
        subplot(2,1,b);contour(MCR_inj(:,:,b),100);
        title(['Component ' num2str(b)]);
        h=gca;
        set(h,'xticklabel','' )
        set(h,'yticklabel','' )
        caxis([vmin vmax])
    end
    colorbar('southoutside','Position',[0.130 0.050 0.780 0.03])
else if inj<=4
    for b=1:size(MCR_inj,3)
        subplot(2,2,b);contour(MCR_inj(:,:,b),100);
        title(['Component ' num2str(b)]);
        h=gca;
        set(h,'xticklabel','' )
        set(h,'yticklabel','' )
        caxis([vmin vmax])
    end
    colorbar('southoutside','Position',[0.130 0.050 0.780 0.03])
else if inj<=6
    for b=1:size(MCR_inj,3)
        subplot(3,2,b);contour(MCR_inj(:,:,b),100);
        title(['Component ' num2str(b)]);
        h=gca;
        set(h,'xticklabel','' )
        set(h,'yticklabel','' )
        caxis([vmin vmax])
    end
    colorbar('southoutside','Position',[0.130 0.050 0.780 0.03])
else if inj<=9
    for b=1:size(MCR_inj,3)
        subplot(3,3,b);contour(MCR_inj(:,:,b),100);
        title(['Component ' num2str(b)]);
        h=gca;
        set(h,'xticklabel','' )
    end
    colorbar('southoutside','Position',[0.130 0.050 0.780 0.03])
else if inj<=12
    for b=1:size(MCR_inj,3)
        subplot(4,3,b);contour(MCR_inj(:,:,b),100);
        title(['Component ' num2str(b)]);
        h=gca;
        set(h,'xticklabel','' )
        set(h,'yticklabel','' )
        caxis([vmin vmax])
    end
    colorbar('southoutside','Position',[0.130 0.050 0.780 0.03])
else if inj<=18
    for b=1:size(MCR_inj,3)
        subplot(6,3,b);contour(MCR_inj(:,:,b),100);
        title(['Component ' num2str(b)]);
        h=gca;
        set(h,'xticklabel','' )
        set(h,'yticklabel','' )
        caxis([vmin vmax])
    end
    colorbar('southoutside','Position',[0.130 0.050 0.780 0.03])
else if inj<=27
    for b=1:size(MCR_inj,3)
        subplot(9,3,b);contour(MCR_inj(:,:,b),100);
        title(['Component ' num2str(b)]);
        h=gca;
        set(h,'xticklabel','' )
        set(h,'yticklabel','' )
        caxis([vmin vmax])
    end
    colorbar('southoutside','Position',[0.130 0.050 0.780 0.03])
else
    for b=1:size(MCR_inj,3)
        subplot(18,3,b);contour(MCR_inj(:,:,b),100);
        title(['Component ' num2str(b)]);
        h=gca;
        set(h,'xticklabel','' )
        set(h,'yticklabel','' )
        caxis([vmin vmax])
    end
    colorbar('southoutside','Position',[0.130 0.050 0.780 0.03])
end
```matlab
set(h,'yticklabel','')
caxis([vmin vmax])
end
elseif inj<=12
    for b=1:size(MCR_inj,3)
        subplot(4,3,b);contour(MCR_inj(:,:,b),100);
        title(['Component ' num2str(b)]);
        h=gca;
        set(h,'xticklabel','')
        set(h,'yticklabel','')
        caxis([vmin vmax])
    end
colorbar('southoutside','Position',[0.130 0.050 0.780 0.03])
caxis([vmin vmax])
elseif inj<=16
    for b=1:size(MCR_inj,3)
        subplot(4,4,b);contour(MCR_inj(:,:,b),100);
        title(['Component ' num2str(b)]);
        h=gca;
        set(h,'xticklabel','')
        set(h,'yticklabel','')
        caxis([vmin vmax])
    end
colorbar('southoutside','Position',[0.130 0.050 0.780 0.03])
elseif inj<=20
    for b=1:size(MCR_inj,3)
        subplot(5,4,b);contour(MCR_inj(:,:,b),100);
        title(['Component ' num2str(b)]);
        h=gca;
        set(h,'xticklabel','')
        set(h,'yticklabel','')
        caxis([vmin vmax])
    end
colorbar('southoutside','Position',[0.130 0.050 0.780 0.03])
elseif inj<=25
    for b=1:size(MCR_inj,3)
        subplot(5,5,b);contour(MCR_inj(:,:,b),100);
        title(['Component ' num2str(b)]);
        h=gca;
        set(h,'xticklabel','')
        set(h,'yticklabel','')
        caxis([vmin vmax])
    end
colorbar('southoutside','Position',[0.130 0.050 0.780 0.03])
elseif inj<=30
    for b=1:size(MCR_inj,3)
        subplot(5,5,b);contour(MCR_inj(:,:,b),100);
        title(['Component ' num2str(b)]);
        h=gca;
        set(h,'xticklabel','')
        set(h,'yticklabel','')
        caxis([vmin vmax])
    end
colorbar('southoutside','Position',[0.130 0.050 0.780 0.03])
end
```
The function `reconstructv2` reconstructs the data back into a 4-way array.

```matlab
function recon=reconstructv2(chrom2d,chrom1d,wave,comp)
    Xsum=0;
    second=size(chrom2d{1,1},1);
    first=size(chrom1d{1,1},1);
    inj=size(comp,2);
    spec=size(wave{1,1}{1},1);
    recon=zeros(second,first,inj,spec);
    for a=1:second
        for b=1:first
            for c=1:inj
                for d=1:spec
                    for r=1:comp(1,c)
                        Xsum=Xsum+chrom1d{1,c}(b,r)*chrom2d{1,c}(a,r)*wave{1,c}{1}(d,r);
                    end
                    recon(a,b,c,d)=Xsum;
                    Xsum=0;
                end
            end
        end
    end
end
```

The function `resample` resamples the interpolated 1D profiles.
function sampled=resample(chrom1d)
sampled=cell(1,size(chrom1d,2));
time=1:20:size(chrom1d{1,1},1);
sampled_time=size(time,2);
for a=1:size(chrom1d,2)
    sampled{1,a}=zeros(sampled_time,size(chrom1d{1,a},2));
    for b=1:size(chrom1d{1,a},2)
        sampled{1,a}(:,b)=chrom1d{1,a}(1:20:end,b);
    end
end

The function spec_matchv4 matches the reference spectra to each sample’s three-way PARAFAC resolve spectra.

function [trail]=spec_matchv4(Co,raw_type,inj_type,inj_split,thr)
[comp_raw,comp_inj]=size(Co);
match=zeros(comp_raw,comp_inj);
trail=cell(comp_raw,1);
for a=1:comp_raw
    trail{a,1}=zeros(1,comp_inj);
end
Z=zeros(comp_raw*comp_inj,3);
comp_inj=size(inj_split,2);
comp_done=1;
for a=1:size(raw_type,2)
    if raw_type(1,a)==0
        match(a,:)=-1;
    end
end
for a=1:size(inj_type,2)
    if inj_type(1,a)==0
        match(:,a)=-1;
    end
end
while comp_done==1
    maxcorr=-3;
    comp_1=0;
    comp_2=0;
    comp_done=0;
    for a=1:comp_raw
        for b=1:comp_inj
            if Co{a,b}>thr&&match(a,b)==0&&Co{a,b}>maxcorr
                
295
\[
\begin{align*}
\text{maxcorr} &= \text{Co}\{a,b\}; \\
\text{comp}_1 &= a; \\
\text{comp}_2 &= b; \\
\end{align*}
\]

\[
\begin{align*}
\text{if} \quad \text{maxcorr} &= -3 \& \& \text{comp}_1 \neq 0 \& \& \text{comp}_2 \neq 0 \\
\quad \text{match}(\text{comp}_1,:) &= -1; \\
\quad \text{match}(,\text{comp}_2) &= -1; \\
\quad \text{match}(\text{comp}_1,\text{comp}_2) &= 1; \\
\quad \text{comp}\_\text{done} &= 1; \\
\quad \text{if} \quad \text{isempty}(\text{inj}\_\text{split}\{1,\text{comp}_2\}) &= 1 \\
\quad \quad \text{for} \quad a &= 1:\text{size}(\text{inj}\_\text{split}\{1,\text{comp}_2\},1) \\
\quad \quad \quad \text{match}(\text{comp}_1,\text{inj}\_\text{split}\{1,\text{comp}_2\}(a,1)) &= 1; \\
\quad \quad \end{align*}
\]

\[
\begin{align*}
\text{[comp}\_\text{raw},\text{comp}_\text{inj}] &= \text{size}(\text{Co}); \\
\text{counter} &= 1; \\
\text{for} \quad a &= 1:\text{comp}\_\text{raw} \\
\quad \text{for} \quad b &= 1:\text{comp}_\text{inj} \\
\quad \quad \text{Z}(\text{counter},1) &= a; \\
\quad \quad \text{Z}(\text{counter},2) &= b; \\
\quad \quad \text{Z}(\text{counter},3) &= \text{match}(a,b); \\
\quad \quad \text{counter} &= \text{counter} + 1; \\
\quad \end{align*}
\]

\[
\begin{align*}
\text{[m,n]} &= \text{find}(\text{Z} == -1); \\
\text{Z}(m,:,:) &= []; \\
\text{[m,n]} &= \text{find}(\text{Z} == 0); \\
\text{Z}(m,:) &= []; \\
\text{for} \quad a &= 1:\text{size}(\text{Z},1) \\
\quad \quad \text{trail}\{\text{Z}(a,1)\}(1,\text{Z}(a,2)) &= \text{Z}(a,2); \\
\quad \end{align*}
\]

The function unimodal applies unimodality to a component.

\[
\begin{align*}
\text{function} \quad \text{output} &= \text{unimodal}(\text{vector},\text{tolerance},\text{direction}); \\
\% \quad \text{This is a subroutine of the ALS4d program} \\
\% \quad \text{It assures the returning vector only has one maximum.} \\
\% \quad \text{Usage: output} = \text{unimodal}(\text{input},\text{tolernaces},\text{direction})
\end{align*}
\]

296
tolerance > 1 allows for a small rise in the signal after the maximum.
% direction is either 1, 2 or 3
% in the case of 1 the second peak is cropped to the baseline as soon as the signal rises
% in the case of 2 the second peak is cropped horizontally until the signal drops below this point again
% in the case of 3 the second peak is halved in intensity and will be removed in multiple iterations

nil = min(vector); % cut signal to lowest signal, not necessarily zero!
vector_length = max(size(vector)); % correct for whether vector is columns or rows
[maximum_signal, position_of_maximum] = max(vector);
valley = maximum_signal;
position = position_of_maximum + 1; % shift position up from maximum (forward looking)
while position <= vector_length
    signal = vector(position);
    if signal < valley
        valley = signal;
    end % no valley
    if signal > valley
        if signal > valley * tolerance
            if direction == 1
                signal = nil; % cut the rest of the profile to zero
                while position < vector_length
                    vector(position) = nil;
                    position = position + 1;
                end
            end
            if direction == 2
                signal = valley * tolerance;
            end % cut top of peak
            if direction == 3
                signal = (signal + valley * tolerance) / 2;
            end % average peak
            end
        elseif direction == 2
            signal = valley * tolerance;
        end % cut top of peak
        elseif direction == 3
            signal = (signal + valley * tolerance) / 2;
        end % average peak
    end
    end

valley = maximum_signal;
position = position_of_maximum - 1; % shift position up from maximum (backward looking)
while position > 0
    signal = vector(position);
    if signal < valley
        valley = signal;
    end % no valley
    if signal > valley
        if signal > valley * tolerance
            if direction == 1
                while position > 1
                    vector(position) = nil;
                    position = position - 1;
                end
            end
            if direction == 2
                signal = valley * tolerance;
            end % cut top of peak
            if direction == 3
                signal = (signal + valley * tolerance) / 2;
            end % average peak
        end % is valley deep enough?
    end % bottom of the valley
end % no valley
end
vector(position)=signal;
position=position-1;
end
output=vector;

The function `whitsm` performs the Whitaker smoother on the resolved 2D profiles.

```matlab
function [z, cve, h] = whitsm(y, lambda, d)
% Whittaker smoother
% Input:
%   y: data series, assumed to be sampled at equal intervals
%   lambda: smoothing parameter; large lambda gives smoother result
%   d: order of differences (default = 2)
% Output:
%   z: smoothed series
%   cve: RMS leave-one-out prediction error
%   h: diagonal of hat matrix
%
% Paul Eilers, 2003
% Default order of differences
if nargin < 3
    d = 2;
end
% Smoothing
m = length(y);
E = speye(m);
D = diff(E, d);
C = chol(E + lambda * D' * D);
z = C \ (C' \ y);
% Computation of hat diagonal and cross-validation
if nargout > 1
    if m <= 100       % Exact hat diagonal
        H = inv(E + lambda * D' * D);
        h = diag(h);
    else              % Map to diag(H) for n = 100
        n = 100;
        E1 = speye(n);
        D1 = diff(E1, d);
        lambda1 = lambda * (n / m) ^ (2 * d);
        H1 = inv(E1 + lambda1 * D1' * D1);
        h1 = diag(H1);
        u = zeros(m, 1);
        k = floor(m / 2);
        k1 = floor(n / 2);
        u(k) = 1;
    end
end
```
\[ v = C \backslash (C' \backslash u); \]
\[ h k = v(k); \]
\[ f = \text{round}((1:m)' - 1) \ast (n - 1)/(m - 1) + 1; \]
\[ h = h1(f) \ast v(k)/h1(k1); \]
end

\[ r = (y - z)/(1 - h); \]
\[ \text{cve} = \sqrt{(r' \ast r/m)}; \]
end
Appendix E

The .m files and data for Chapter 7 can be found in R:\CHEM\Rutan_lab\Robert\Chapter7. The .m code used to generate the results in Chapter 7 is listed below. The scripts are located in the alternative.m file. The .csv files are from chemstation. The corresponding CalX.mat and QCX.mat files are the calibration and quality control samples used for the quantitation study. The QuantCodeX.m scripts indicate the regions used to analyze each cluster of peaks. The SimulationSetup2.mat and SimulationGeneration.m were used to generate the simulated comparison data sets.

Script used to setup the SCP function developed by Mørup.

\begin{verbatim}
t_init=cputime;
opts.FACT{1,1}=IG_4; %Sample initial guess.
opts.FACT{1,2}=IG_1; %Second dimension initial guess.
opts.FACT{1,3}=IG; %Spectral initial guess.
opts.FACT{1,4}=IG_2; %First dimension initial guess.
opts.FACT{1,5}=zeros(48,4); %Tau initial guess with dimensions sample by the number of components.
opts.TauW=zeros(4,2);
opts.TauW([1 2],1)=-30; %Degree of forward shifting allowed.
opts.TauW([1 2],2)=30; %Degree of backward shifting allowed.
[FACT,T,nLogP,varexpl,Lambda,RemoveFrequencies,TauW] = ShiftCP(permute(Xsub,[3 1 4 2]),4,opts);
t_final=cputime;
t_analysis=t_final-t_init;
\end{verbatim}

Script used to the PARAFAC2 function from Eigenvector Research.

\begin{verbatim}
t_init=cputime;
\end{verbatim}
IG_model.loads{2,1}=IG_2;  \textit{% First dimension initial guess.}
IG_model.loads{3,1}=IG;  \textit{% Spectral initial guess.}
IG_model.loads{4,1}=IG_4;  \textit{% Sample initial guess.}
IG_model.detail.options.constraints{1,1}.type='unconstrained';
IG_model.detail.options.constraints{2,1}.type='unconstrained';
IG_model.detail.options.constraints{3,1}.type='unconstrained';
IG_model.detail.options.constraints{4,1}.type='unconstrained';
[model_constr]=parafac2(permute(Xsub,[1 2 4 3]),IG_model.loads,IG_model.detail.options);
t_final=cputime;
t_noconstr=t_final-t_init;
Vita

Robert Craig Allen was born in Dunoon, Scotland on August 15, 1979. He graduated from Matthew Fontaine Maury High School in Norfolk, Virginia. He received a B.S. in chemistry from George Mason University in 2001 and a M.S. in chemistry from George Mason University in 2004. Immediately after graduating, he moved to Richmond, VA to begin working for Analytics in the metals extraction room. He left Analytics in July 2005 to take a position at PPD as a Scientist. He was promoted to LC/MS instrument specialist in August 2007. He left PPD in July 2008 to begin working on a doctorate in chemistry in August 2008. He joined Dr. Rutan’s research group in October 2008 and has been a member since. He spent eleven months (August 2011 to July 2012) visiting and working with a collaborator group (the Carr research group) at the University of Minnesota, Minneapolis campus. He is the first author on three publications and has presented his research at five scientific meetings.

Publications


Tistaert, C.; Bailey, H.P.; Allen, R.C.; Heyden, Y.V.; Rutan, S.C.; Resolution of spectrally rank deficient multivariate curve resolution - alternating least squares components in LCxLC-DAD analysis, J. Chemom., 26 (2012) 474-486


Presentations
Allen, R. C.; Rutan, S. C.; Overview of Quantitation in Comprehensive Two-Dimensional Chromatography; Minnesota Chromatography Forum, Minneapolis, MN, May 2012

Allen, R. C.; Rutan, S. C.; The Development of an Interpolation-Based Approach to the Alignment of Fast LCxLC-DAD Chromatograms, FACSS, Raleigh, NC, October 2010

**Posters**


Allen, R. C.; Rutan, S. C.; Alignment of 2D LC-DAD chromatograms by interpolation, American Chemical Society, Washington, D.C., August 2009