<u>Supporting Information</u>

Learning to Read Spectra: Teaching Decomposition with Excel in a Scientific Writing Course

Andrew W. Muelleman and Rainer E. Glaser*

Department of Chemistry, University of Missouri, Columbia, Missouri 65211, United States

TABLE OF CONTENTS

Supplied as Separate Files:

<u>SI-01_A03.zip contains the following files:</u>

A03 Handouts for students used in SP16 (PDF, DOC) and SP17 (PDF, DOC)

A03 Rubrics for students used in SP16 (PDF, DOC) and SP17 (PDF, DOC)

<u>SI-01_A04.zip contains the following files:</u>

A04 Handouts for students used in SP16 (PDF, DOC) and SP17 (PDF, DOC)

A04 Rubrics for students used in SP16 (PDF, DOC) and SP17 (PDF, DOC)

Excel workbooks for A04 in SP16 (XLSX) and SP17 (XLSX)

<u>SI-01_A04-Video Demonstration files:</u>

An animation in two file formats (MOV, AVI) to demonstrate the organization and functionality of the

XLSX workbook

Adjusting *Solver* Options

The *Solver* process requires initial values for the variables and it is a good practice to set constraints on the values of the variables by specification of reasonable. For example, if two peaks are used to deconvolute the feature of the line at pH =12 at 410 nm in Figure 1, one might constrain the center of the peak to 395 nm ≤ λ_{max} ≤ 420.

The error is measured at discrete intervals and therefore wide peaks will always benefit from more optimization. The wide band at *ca.* 410 nm, for example, will be fit very well because information about this feature is contained in *ca.* 25 digitized points. Conversely, small features may be reflected in only a much smaller number of digitized points and, for example, the shoulder at *ca.* 300 nm in Figure 1 is such a case and only affects a few points of the digitized spectrum. Instead of finding all of the optimal parameters in one simulation over the entire spectrum range, a much better solution can be obtained by step-wise deconvolution. In the first step, only the parameters of the three peaks in the low-λ region of the spectrum (i.e., 260 nm ≤ λ_{max} ≤ 360 nm) are optimized based on the error function for that range only. With those three bands now set, in the second step we optimized the parameters of the remaining three peaks in the high-λ region (360 nm ≤ λ_{max}) based on the error function generated with the digitized points of the entire spectrum.

The recommendation to use 50 – 100 data points between the left and right bounds is based on the authors' experience. Using fewer than 50 points (in the case of Figure 1 a step size of 10 nm) created a decent first approximation solution, but failed to give enough credibility to smaller features. Using more than 100 points (step sizes of 2, 1, or 0.5 nm) did not aid the solution in any noticeable way. In fact, when using considerably more points the *Solver* solution took longer and seemed to be more likely to find a poor solution.

Worksheet Organization

As was pointed out by the reviewers, we note that the templates laid out in Figures 3 and 6 are neither the only nor the shortest ways to approach those problems.

In Figure 3, the maximum of any normal curve is a function of its standard deviation, and therefore only two columns are necessary to define each primitive (one needs only define the

wavelengths once). The layout in Figure S3 uses the fact that the Excel formula for the maximum

height of a normal curve is given by the expression 1/(sqrt(2*pi())*σ). While this would simplify the

worksheet illustrated in Figure 3, we believe that the stepwise process is easier for students to

understand.

Likewise in Figure 6, column **AB** is not necessary to complete this problem. One could use the

sumxmy2() function in Excel to achieve the same result with two columns instead of three. This

function takes the data from two columns (columns **Z** and **AA** in our case) and returns the sum of

squares of the difference at each corresponding value. The use of the sumxmy2() function does reduce

the number of columns, but again, we believe that it is more instructive to present the method

stepwise to an audience with varying mathematical skills.

Figure S3: Shortcut Version of Figure 3

	A	**B**
1	λ_{max} (Center)	255
2	Height	0.4
3	Width (std. dev.)	7.9
4		
5	**λ (nm)**	**Normal Curve**
6	260	=($**B**$**2***sqrt(2*pi())*$**B**$**3**)*NORM.DIST(**A6**, **B**$**1**, **B**$**3**, FALSE)
7	=**A6** + 5	=($**B**$**2***sqrt(2*pi())*$**B**$**3**)*NORM.DIST(**A7**, **B**$**1**, **B**$**3**, FALSE)
8	=**A7** + 5	=($**B**$**2***sqrt(2*pi())*$**B**$**3**)*NORM.DIST(**A8**, **B**$**1**, **B**$**3**, FALSE)
...	...	...
74	600	=($**B**$**2***sqrt(2*pi())*$**B**$**3**)*NORM.DIST(**A74**, **B**$**1**, **B**$**3**,FALSE)

Figure S6: Shortcut Version of Figure 6

	Z	**AA**	**...**	**AC**
5	Sum of Normed	Digitized		Sum of Error Squared
6	=SUM($n_i(\lambda)$, i=1,2,...,n)	from digitizer		=SUMXMY2(**Z6:Z74**, **AA6:AA74**)
7	=SUM($n_i(\lambda)$, i=1,2,...,n)	...		
...	...	...		
73	=SUM($n_i(\lambda)$, i=1,2,...,n)	...		
74	=SUM($n_i(\lambda)$, i=1,2,...,n)	...		