Program II
Section-by-Section Description of Program Logic

The documentation contained in this paper assumes that the reader has a copy of the SAS program source code or the program's flowchart (or preferably, both). If one is using the source code, sections are identified by a comment line (in the form, */* comment */*) that contains the word, "start," preceded by the section number (*e.g.,* " … 1 start …"). The end of each section is likewise identified by a comment line that contains the word, "end" (*e.g.,* "… 1 end …"). If one is using the flowchart, sections are identified by dashed rectangles that encompass the SAS program blocks. These program blocks are parallelograms for SAS data steps and rectangles for SAS procedures. Other blocks, such as rectangles with two vertical inscribed lines for SAS macro calls, diamonds for decisions or loops, and a concave rectangle for variables declaration, are also used. Output that is written to the SAS output window is indicated by a "P" alongside a SAS procedure block or between program blocks. Output that is written to an external file (in ASCII format) is likewise indicated by an "F." Each section is described as follows:

1. This section defines all of the macro variables whose values are assigned here and left unchanged throughout the entire program. Some variables may never need changing, such as **maxs** (the maximum number of spectral data points) and **savgolgap.prn** (the name of the ASCII file that contains the Savitzky-Golay convolution coefficients). Other variables, especially those that pertain to loop limits, may need to be changed with each run of the program. Output file names that are not changed from run to run will result in the program output being appended to the existing file(s). The loop control macro variables are grouped by their appearance in the four SAS macros (**MSCnoSGloop, Sgloop, SNVnoSGloop, and SNVloop**).

   Two other macro variables, **leftcut** and **rightcut**, need explanation. These variables allow the spectral data to be truncated to a smaller (contiguous) wavelength region than that of the original spectra. The number of wavelengths to omit from the left end of the spectrum is specified by **leftcut**. Likewise, omitted wavelengths from the right end are specified by **rightcut**. **Truncation is performed before the application of all spectral pretreatments described herein**. The user is advised that the program does not check for ill-conceived values for these variables, such as negative values or values so large that the size of the truncated spectrum is ridiculously small or mathematically impossible.

   Permissible values for the macro variables that regulate the lower and upper loop limits are shown in the source code, and more specifically, in the comments section to the right of their assignment statements. As with the use of the truncation control variables, there is no error checking of the values for these limit variables. Therefore, the user must be very careful in selecting the values for these variables.

2. Spectral data is read into the program, once and only once. The SAS statements encoded in this section were written for the reading of a Grams (Galactic Industries, Salem, NH) multifile. Further, the multifile must be a specific type, as Grams has many. This type (what Galactic calls the "fixed integer format"), which is probably the most common of all in Grams, is a binary file, specifically having the spectral data stored as 32-bit fixed-

point signed fractions scaled by an exponent value for each spectrum. Wavelength values are absent because it is assumed that the spectral points are uniformly spaced in the wavelength domain. (Starting and ending wavelength values are contained within the 512-byte header of the Grams multifile; however, this program does not read these values into memory.) This type of file is termed by Galactic as their "new" format, which is not to be confused with the older format that Galactic used in versions of their earlier programs, LabCalc and SpectraCalc. Each spectrum must have the same number of data points. The exponent value for each spectrum is stored within a 32-byte subfile header that lies before each spectrum's values.

Getting the spectral data file into this format may present a challenge, as the Grams program has evolved over the years and file converters are added by Galactic all the time, with some storing the data as IEEE 32-bit floating-point values rather than binary fixed integers. Specific details can be found in white papers on Grams file formats that are available for download at Galactic's website (www.galactic.com).

There is an alternative to the reliance on the Grams multifile, albeit at the expense of the users diligence in writing SAS code. The user can write his/her own data step. If this route is taken, the user should bear in mind the essential variables that ensue from this data step: **y1-yn** (where n is the maximum number of spectral data points, as prescribed in Section 1), **numpoints** (the actual number of spectral data points, occupying positions 1 through this number in the y array), and **numrecords** (the number of spectra).

3. Chemical data are read into the program, once and only once. This program is designed to operate on one analyte at a time; therefore the chemical data file must consist of one value per line in an ASCII file. The order of the values, with respect to their spectra, must be the same as the order established by Section 2. The other functions of this section are determining the variance of the chemical value (for later use in variance scaling) and combining the spectral and chemical data into one SAS dataset.

4. Coefficient values from the first seven tables of the original Savitzky-Golay paper [Analytical Chemistry, 36:1627-1639 (1964)], as corrected by Steinier et al. [Analytical Chemistry, 44:1906-1909 (1972)], along with tables for first central difference (table 8), second central difference (table 9), and a running mean smooth (table 10) are contained in an ASCII file that is read during this data step.

5. **This is the first of seven sections (*i.e.*, 5-11) of the macro program, MSCnoSGloop**. This section, along with Sections 6-10, is contained within a loop that controls the application of MSC. In this section, a determination is made on whether multiplicative scatter correction (MSC) is to be applied to each spectrum (depending on the loop limit macro variables, **pathbegin** and **pathend**, set in Section 1). If no MSC is to occur (*i.e.,* the loop control variable, **k**, is zero), the resulting dataset is merely a truncated version of the original. That is, unused array variables are dropped. If MSC is used (*i.e.,* when **k** = 1), the series of procedures and data steps are followed to scatter-correct each spectrum, as defined in Martens and Naes (*Multivariate Calibration*, Wiley & Sons, 1989).

Note: Execution of Sections 5-11 may be bypassed by commenting out the macro call statement (%MSCnoSGloop) in the main program (Section 39).

6. This section, along with Sections 7-10, is contained within a loop that controls the application of variance scaling (as set in Section 1 by the macro variables, **varscalebegin** and **varscaleend**). The SAS PLS procedure is invoked, using either no variance scaling (loop control variable, **m**, is zero) or variance scaling (**m** = 1). One-sample-out cross validation is used. The output datasets (plsout, PRESS_results, and crossval_results) contain information on the predicted residual error sum of squares (PRESS) and on the individual sample cross validation predictions of the model whose number of factors (*e.g.,* factor 1, factor 2 …, factor *i*) produced the smallest PRESS.

7. The SAS datasets generated from the previous section are manipulated to produce printed output (to the SAS output window) on the cross validation error for each set of PLS factors. The number of factors identified with the lowest PRESS value is then summarized in the SAS output window and in an output file (*i.e.,* **SASstyleoutfile**, as specified in Section 1). One line of information is written to this output file. It includes the model conditions (noting that in this macro, "SG_Table_=0" and "SG_Window_=0" reflect the fact that a Savitzky-Golay convolution is not permitted), the number of factors that produced the minimum PRESS value, the root mean of the minimum PRESS, and the R^2 value. Note that the R^2 value is not evaluated from a conglomeration of the one-out cross validation predictions (as done in Grams PLSplus), but rather is determined from the correlation between the predictions from applying the calibration equation (formed from all samples) and the chemical values.

   Additionally, the PRESS values are examined to determine the statistical significance of adding factors in a stepwise manner in the PLS equations. This involves the application of a ratio of variances (F-) test, which is patterned after the procedure used in Grams PLSplus that originated in articles of Haaland and Thomas [Analytical Chemistry 60:1193-1201, 1202-1212 (1988)]. Results of this test are evaluated in Section 9.

8. This section assembles the PRESS values and the optimal PRESS value (by F-test), and performs variance scaling on the PRESS values when the variance scaling option is active.

9. The PRESS values and associated probabilities from the previous two sections are evaluated (starting with the PLS factor number one, then adding one factor at a time) to determine the factor at which the probability that model improvement has occurred in going from *j* factors to (*j*+1) factors drops below the level, predefined by Grams, of 0.75. Once this number of factors is determined, the PLS procedure is run again, this time with the number of factors explicitly stated as the Grams-type optimal number. This rerun of the PLS procedure is necessary to obtain the predicted values from the calibration equation which has used the "optimal" number of factors (as opposed to the predicted values from the original PLS run, which gave predicted values associated with the PLS model that produced the minimum PRESS value).

10. The predicted values from the Grams-type optimal model are correlated to their reference values. The results of this model are summarized in the one line of information written to a second output file (*i.e.,* **Gramsstyleoutfile** as prescribed in Section 1). This line contains the summary statistics on the number of PLS factors that produced the smallest PRESS value, as well as the number that produced the Grams-type optimal model. The values for "F-Ratio" , "F_Test" , and "RMSD" should match those (typically, to 4-5 decimal places) found in the report summary of Grams PLSplus.

11. This section is invoked once all PLS trials within this macro (for a maximum of four trials: 2 MSC loop cycles × 2 variance scaling loop cycles) have been run. The SAS procedure, "datasets," allows for the freeing of computer memory by erasing datasets that are no longer needed.

12. **This is the first of nine sections (*i.e.,* 12-20) of the macro program, SGloop**. Note that this section, along with sections 13-19, is contained within a loop that controls the selection of the Savitzky-Golay table under use. Section 12 results in the selection of a Savitzky-Golay convolution table (choice of ten tables, as determined by a combination of **firstSGtable**, **secondSGtable**, **thirdSGtable** and the loop control variables, **tablebegin** and **tableend**), depending on what was specified in Section 1 and what convolution table loop cycle the program is currently executing.

    Note: Execution of Sections 12-20 may be bypassed by commenting out the macro call statement (%SGloop) in the main program (Section 39).

13. Derivatization or smoothing is performed. Note that this section, along with sections 14-19, is contained within a loop that controls the selection of the Savitzky-Golay window (within a given table) under use. The convolution window, contained within the convolution table that was identified in the previous section, is applied to the spectra. The exact window (1 of 11, corresponding to every odd number from 5 to 25 points) depends on the loop limit macro variables (**windowbegin** and **windowend**) set in Section 1 and on what convolution window loop cycle the program is currently executing.

14. This section is identical to Section 5, with the exception that the incoming SAS dataset has been derivatized or smoothed, as opposed to the raw form used in Section 5. Section 14, as well as Sections 15-19, is contained within a loop that controls the application of MSC. MSC is performed if the loop limit macro variables, **pathbegin** and **pathend**, were set to allow for MSC and, if so, when the loop control variable, **k**, equals one.

15. This section is identical to Section 6. Note that the number of wavelengths used in the PLS procedure have been reduced (compared to Section 6) as a result of the derivatization/smoothing operation in Section 13. Specifically, the number of dropped wavelengths on the left end of the spectrum is one half of one less than the number of points of the convolution window. The same is true for the right end. Section 15, along with Sections 16-19, is contained within a loop that controls the application of variance

scaling (as set in Section 1 by the macro variables, **varscalebegin** and **varscaleend**).

16. This section is identical to Section 7, with the exception that the line of model results information written to the output file (**SASstyleoutfile**) specifies the convolution table and window used in the derivatization/smoothing operation.

17. This section is identical to Section 8.

18. This section is identical to Section 9.

19. This section is identical to Section 10, with the exception that the line of model results information written to the output file (**Gramsstyleoutfile**) specifies the convolution table and window used in the derivatization/smoothing operation.

20. Similar to Section 11, this section is invoked once all PLS trials within this macro (for a maximum of 132 trials: 3 convolution tables × 11 convolution windows per table × 2 MSC loop cycles × 2 variance scaling loop cycles) have been run. Note that while the coefficients for 10 convolution tables are initially read during a program run (see Section 4), it is possible to use a maximum of 3 tables per run. If more than three convolution tables are needed, the program will have to be rerun, having first changed the values of the macro variables (**firstSGtable**, **secondSGtable**, **thirdSGtable**, **tablebegin** and **tableend**) that control program flow. Also note that tables 2, 5, and 7 possess 10 convolution windows, as there are no 5-point convolutions for the tables' corresponding polynomials.

21. **This is the first of eight sections (*i.e.,* 21-28) of the macro program, SNVnoSGloop**. This section is somewhat akin to Section 5, but with two exceptions. First, there is no option for not performing a pathlength correction, and second, the correction itself is the Standard Normal Variate (SNV) transformation (as opposed to the MSC of Section 5). Recalling that the SNV transformation [as described in Barnes et al., Applied Spectroscopy, 43:772-777 (1989)] is performed on each spectrum independently of other spectra (*e.g.,* there is no correction to a mean spectrum), this transformation requires the determination of each spectrum's mean (over all wavelengths) and standard deviation. The procedures and data steps in this section determine the mean and standard deviation of each spectrum, and then apply the SNV.

    Note: Execution of Sections 21-28 may be bypassed by commenting out the macro call statement (%SNVnoSGloop) in the main program (Section 39).

22. This section, along with Sections 23-27, is contained within a loop that allows for the application of "detrending." The SNV-transformed spectra (Section 21) may be detrended [Barnes et al., Applied Spectroscopy, 43:772-777 (1989)], depending on the settings for the macro variables, **detrendbegin** and **detrendend**, specified in Section 1. Recapitulating, the detrend procedure fits a quadratic polynomial to each spectrum by least squares regression. The resulting spectrum from this procedure is the residual

spectrum (*i.e.,* actual minus fitted).

23. This section is identical to Section 6, with the exception that the incoming SAS dataset has been SNV-transformed (Section 21) and possibly detrended (Section 22). Sections 23-27 are contained within a loop that allows for application of variance scaling (as regulated by the macro variables, **varscalebegin** and **varscaleend**, which were set in Section 1).

24. This section is identical to Section 7; however, instead of having fields in the output file that deal with MSC, these fields now deal with the presence or absence of detrending after the SNV transformation.

25. This section is identical to Section 8.

26. This section is identical to Section 9.

27. This section is identical to Section 10. Similar to Section 24, instead of having fields in the output file that deal with MSC, these fields now deal with the presence or absence of detrending after the SNV transformation.

28. Similar to Section 11, this section is invoked once all PLS trials within this macro (for a maximum of four trials: 2 detrend loop cycles × 2 variance scaling loop cycles) have been run.

29. **This is the first of ten sections (*i.e.,* 29-38) of the macro program, SNVloop**. This section is identical to Section 21.

    Note: Execution of Sections 29-38 may be bypassed by commenting out the macro call statement (%SNVloop) in the main program (Section 39).

30. This section is identical to Section 22. It, along with Sections 31-37, is contained within a loop that allows for the application of "detrending."

31. This section is identical to Section 12. It, along with Sections 32-37, is contained within a loop that controls the selection of the Savitzky-Golay table under use.

32. This section is identical to Section 13, with the exception that the incoming SAS dataset has already undergone an SNV transformation with possible detrend. Sections 32-37 are contained within a loop that controls the selection of the Savitzky-Golay window (within a given table) under use.

33. This section is identical to Section 15. Sections 33-37 are contained within a loop that allows for application of variance scaling (as regulated by the macro variables, **varscalebegin** and **varscaleend**, which were set in Section 1).

34. This section is identical to Section 24, with the exception that the line of model results information written to the output file (**SASstyleoutfile**) specifies the convolution table and window used in the derivatization/smoothing operation.

35. This section is identical to Section 8.

36. This section is identical to Section 9.

37. This section is identical to Section 27, with the exception that the line of model results information written to the output file (**Gramsstyleoutfile**) specifies the convolution table and window used in the derivatization/smoothing operation.

38. Similar to Section 20, this section is invoked once all PLS trials within this macro (for a maximum of 132 trials: 2 detrend cycles × 3 convolution tables × 11 convolution windows per table × 2 variance scaling loop cycles) have been run. Note that while the coefficients for 10 convolution tables are initially read during a program run (see Section 4), it is possible to use a maximum of 3 tables per run. If more than three convolution tables are needed, the program will have to be rerun, having first changed the values of the macro variables (**firstSGtable**, **secondSGtable**, **thirdSGtable**, **tablebegin** and **tableend**) that control program flow. Also note that tables 2, 5, and 7 possess 10 convolution windows, as there are no 5-point convolutions for the tables' corresponding polynomials.

39. This section is actually the main program. From this section, the macro programs are called. As stated in bold print in the description of starting section of each PLS-executing macro (**MSCnoSGloop**, **SGloop**, **SNVnoSGloop**, and **SNVloop**), the user has the option of bypassing a macro by simply commenting out the one-line macro call statement in Section 39.

Upon completion of all SAS macros of Section 39, the program returns to the SAS environment. The user has the option to examine the results of the PLS trials in the output window. Alternatively, the user may find it easier to examine the contents of the two output files named in Section 1 (assigned to macro variables, **SASstyleoutfile** and **Gramsstyleoutfile**). Each file contains a one-line summary for each PLS model. A maximum of 272 (= 4 + 132 + 4 + 132, arising from the four PLS-executing macros) lines are written for each program run. The data is column-aligned, with text descriptions preceding the value of every statistical figure of merit. This file format can be directly read into a spreadsheet program, such as Microsoft Excel. Lines within the spreadsheet file can then be sorted by a chosen field (column) (*e.g.,* PRESS, RMSD, R^2), and these values can be used to produce graphs that indicate trends in model pretreatments.