Instruction Manual for "Quantlaser": a Batch Process Macro for Reduction of Quantitative Laser Ablation Data

by W. Ian Ridley[1]

Open-File Report 00-311

2000

U.S. DEPARTMENT OF THE INTERIOR
U.S. GEOLOGICAL SURVEY

[1]Denver, Colorado

INSTRUCTION MANUAL FOR "QUANTLASER": A BATCH PROCESS MACRO FOR REDUCTION OF QUANTITATIVE LASER ABLATION DATA

W. Ian Ridley, USGS, Mineral Resources Program, Box 25046, MS 973, Denver, CO 80225

Introduction.

The USGS Laser Ablation ICP-MS Laboratory in Denver, CO houses a Perkin Elmer Sciex 6000[1] mass spectrometer interfaced with a CETAC LSX 200[1] ultra-violet (UV) laser. The Perkin Elmer (PE) Sciex 6000 software is written for complete on-line reduction of solution ICP-MS data and only supplies a partial on-line reduction of laser ablation ICP-MS data. Consequently, a batch off-line program has been written that will quickly perform a complete data reduction and output quantitative parts per million results together with other relevant information.

Data files generated by PE software during the analysis session are easily imported into an Excel spreadsheet. QUANTLASER is a macro for Excel (Windows platform) written in Excel Visual Basic (a version of Visual Basic For Applications) to reduce quantitative laser ablation ICP-MS data. The program is available on 3.5" floppy disks and a hard copy is reproduced here. The macro has been tested on Excel 97 and Excel 2000.

Principles of Quantitative Laser Ablation Data Reduction

In both solution and laser ablation analysis, it is common to use an internal standard to correct for various matrix and instrumental variations. In laser ablation analysis the internal standard element (or group of elements) is chosen to correct for ablation efficiency**. The group of elements analyzed determines the choice of internal standard; the latter is sometimes a compromise between element availability and similar ablation characteristics. The samples and calibration standards invariably have different concentrations of the internal standard element(s), unlike the usual analytical scheme for solution ICP-MS. The PE software does not allow for this situation, so the ablation efficiency correction, which can be substantial, has to be made off-line. The equation used by the PE software to generate concentration data is:

$$C^{i}_{sample} = C^{i}_{standard} * (cps^{i}_{sample} / cps^{i}_{standard})$$

With an ablation efficiency correction, the equation becomes:

$$C^{i}_{sample} = C^{i}_{standard} * (cps^{i}_{sample} / cps^{i}_{standard}) * (cps^{is}_{standard}/cps^{is}_{sample}) * (C^{is}_{sample}/C^{is}_{standard})$$

Where: i = unknown element; is = internal standard element; C = concentration in ppm; cps = net counts per second.

** Ablation efficiency is a measure of the effectiveness of ablation when the laser energy couples with a solid matrix. It is principally a function of sample color, crystallography, and thermal diffusivity.

In solution ICP-MS, matrix effects between samples and calibration standards are relatively small, because all are dilute solutions that can be closely matrix-matched. Thus, the internal standard provides an excellent measure of short-term instrumental drift, which is factored into the software data reduction. However, in laser ablation ICP-MS the internal standard cannot easily be used to measure instrumental drift from sample to sample because the samples and standards can rarely (if ever) be closely matrix-matched. Thus, the ablation efficiency (matrix) component of variation cannot be separated from the instrumental drift component of variation. To compensate for this problem, the analyst runs a drift sample at regular intervals under the assumption that the ablation efficiency for this sample is constant and any variations between drift samples is due to instrumental variations. The PE software cannot provide this drift correction, which is performed off-line by QUANTLASER.

The data report template "Quant Summary USGS.rop" provides count per second (cps) data and total integrated time per element data, thus total integrated counts can be calculated. This information is the basis for a counting statistic error analysis for each element in each sample. The error analysis is intended as an indication of instrument errors and does not account for errors that are associated with precision and accuracy of the calibration standards.

The analysts can also collect a group of blanks (there is no limit on the number of repeat analyses) analyzed as samples. This data is used by QUANTLASER to calculate detection limits for each element. It should be noted that QUANTLASER tracks blank data (measured as blank, not as sample) and calibration data, and sample data is sorted into appropriate data blocks. Thus, the analyst is free to measure blanks and calibration standards several times during a run. The blank data is used to calculate the number of elements analyzed, the symbol for each element, and as part of the error analysis.

Data Format

A data report template, "Quant Summary USGS..rop" writes data files, "*.rep", in a specific format required by QUANTLASER*. The PE software provides a preliminary data reduction whereby concentrations are calculated relative to a calibration curve for each element analyzed using blank-subtracted counts per second data. This information is part of the "*.rep" data file.

The data file, *.rep, provides data for blanks, calibration standards, and samples, in the ASCII format shown in Figure 1.

Starting the Program and User Input

1. Open *.rep file in an Excel workbook as a comma-delimited ASCII file.

2. Save the file as a *.xls file.

- Macro LQUANT99 can be used to reduce data collected using the standard PE template for quantitative analysis (Quant Summary.rop). However, this macro does not provide a counting error analysis.

3. Open the program QUANTLASER in an Excel workbook. This program contains the required macro.

4. Add the required information to the *.xls file, as described below.

5. In the *.xls file, start the macro by using: Tools ⇒ Macro⇒ Macros⇒ (highlight the macro)⇒ Run

Add to the *.xls file:

1. In column 1, an "End" (no punctuation) after the last line of data.
2. If more than one calibration standard, for standards 2, 3 etc. in column 6, a "*" (no punctuation) for each calibration element measured (see Figure 1). This cell or these cells will normally be empty.

User input is required either in message boxes (requiring yes/no answers), or input boxes (requiring a string, integer, or real value).

Message or Input Box	Required Input
1. "Did You Insert End in Column 1 of Data"	Yes/No. If No, program is terminated. " End "required so program recognizes last data row.
2. "Name of Workbook"	Name of data file being processed. Requires .xls extension in Excel 2000.
3. "Blanks Run For Detection Limits?"	Yes/No.
4. "Number of Calibration Standards"	Integer value. If > 4, program is terminated.
5. "Correction method: 1=Multi-element; 2=Single Element"	Integer value. 1 if more then one internal standard element used, 2 if single internal standard element used.
6. "Name of sample used for drift correction"	String value (case insensitive).
7. "Number of Internal Standard Elements"	Integer, if method=1 in message 4.
8. "Total Concentration of Internal Standard Elements For Samples"	Real number if method =1 in message 5.
9. "Internal Standard Element Conc. For Standard #"	Real number if >1 in message 4.
10. "Total concentration of elements for drift sample"	Real number, if method=1 in message 5.
11. "Internal Standard Element Name #"	String value. A box will appear for each internal element if method=1 in message 5, depending upon the value in message 7.
12 "Internal Standard Element Name"	String value, if method=2 in message 5.
13. "One Internal Standard Element Conc. For All Samples?"	Yes/No.
14. "Internal Standard Element conc. For Samples (ppm)"	Real number, if message 13 is Yes.

15. "Internal Standard Element Conc. For Standard #"	Real number, if message 4 > 1
13. "Internat Standard Element Conc. For Drift Sample"	Real number if message 11 is Yes.
14. "Internal element concentration (ppm) for ##, **"	Real number, if message 13 is No. ## is sample name, ** is internal element name.

Program Steps

A program flow diagram is shown in Figure 2.

Program Output

Results are copied to the origin *.xls worksheet, beginning at column 12. The results consist of five blocks of data (see Figure 3).

First, the element concentrations from the initial input data are reproduced, excepting that any concentrations using Calibration Standards 2, 3 or 4 are corrected for an ablation factor relative to Calibration Standard 1.

Second, the element concentrations are corrected for ablation efficiency differences for each sample relative to Calibration Standard 1.

Third, the element concentrations are corrected for instrumental drift based on repeated measurement of a drift sample. The drift sample is grayed in the output block. This block also shows detection limits for each element if appropriate blank samples were run.

Fourth, measurement errors, based on counting statistics, for each element in each sample are output in absolute values (ppm).

Fifth, the data are transposed from X (sample ID) – Y (element name) to X (element name) – Y (sample ID) as an alternate arrangement for copying to other programs.

```
Option Compare Text
Sub QuantLaser()

'Instructions how to run this macro are found in USGS Open File 00-

'SET VARIABLE DIMENSIONS
Dim pCell As Range
Dim element As String, SampNam(500) As String, Wb As String
Dim Conc(500) As String, elem As String, namelem(10) As String
Dim SaveBook As String, EndNote As String, Std4El(30) As String
Dim Std2El(30) As String, Std3El(30) As String, StandardName(10) As String
Dim NotDet(80, 500) As String

Dim numelem As Integer, DriftNum(50) As Integer, kint(50) As Integer
Dim DriftInt(80) As Integer, intflag(50) As Integer

Dim Econc(500) As Single, DriftFrac(500, 80) As Single
Dim Calib2(30) As Single, Calib3(30) As Single, Calib4(30) As Single
Dim DriftVal(60, 500) As Single, Factor(500) As Single
Dim DriftDiff(500, 80) As Single, IntFactor(500, 80) As Single
Dim BlankTotal(80) As Single, BlankMean(80) As Single
Dim SqMeanDev(80) As Single, SumSq(80) As Single, StdDev(80) As Single
Dim GrossBlank(80, 20) As Single, DetLim(80) As Single
Dim StdError(80) As Single, StdIntError1(50) As Single
Dim StdIntError2(50) As Single, StdIntError3(50) As Single
Dim StdIntError4(50) As Single, MeanError(80) As Single
Dim GrossBlankError(80, 50) As Single, SampleError(500, 80) As Single
Dim StdNetError(80) As Single, DetLimitError(80) As Single
Dim BlankConc(80) As Single, BlankError(80) As Single
Dim SampleNetError(500, 80) As Single, IntSampleNetError As Single
Dim SampleConcError(500, 80) As Single
Dim MeanCounts(80) As Single, DevError(50) As Single
Dim DevMeanSqError(80, 50) As Single, SumDevMeanSqError(80) As Single
Dim TotalCounts(80) As Single, Blank(80, 50) As Single
Dim StdNetCounts(80) As Single, cvalues(80, 500) As Single
Dim BlankNet(80) As Single, Std2Factor(20) As Single
Dim Std3Factor(20) As Single, Std4Factor(20) As Single
Dim IntStdConc(10) As Single

'INITIALIZE VARIABLES
I = 11: ii = 0: k1 = 0: k2 = 0: k3 = 0: k4 = 0: k5 = 0: k6 = 0
k7 = 1: k8 = 0: k9 = 0: k10 = 0: InTotal = 0: Outotal = 0
St2 = 0: St3 = 0: St4 = 0: n = 0: i6 = 0: nn = 0
i5 = 0: p13 = 0: BlankBlank = 0: BlockNumber = 0: StandCount = 0
```

```
'INITIAL MESSAGE BOX'
MsgBox "Written by: Ian Ridley, USGS, Mineral Resources Program", , "PROGRAM
QUANTLASER"

'CHECK FOR "End" IN COLUMN 1
EndNote = MsgBox("Did You Insert End in Column 1 of Data?", vbYesNo)
If EndNote = vbNo Then
   GoTo Out6
End If

'INPUT NAME OF EXCEL DATA WORKBOOK AND ACTIVATE
Wb = InputBox("Name of Workbook")
Workbooks(Wb).Activate
Set WkWb = Workbooks(Wb).ActiveSheet
Set WkSheetC1 = Workbooks(Wb).ActiveSheet.Columns(1).Cells

'CLEAR FINAL DATA AREA AND RESET COLUMN WIDTH
Range("I1:IV500").Select
Selection.Clear
Columns("I:IV").Select
Selection.ColumnWidth = 12

'CHECK FOR BLANKS RUN AS SAMPLES FOR DETECTION LIMITS
Detect = MsgBox("Blanks Run For Detection Limits?", vbYesNo)

'CHECK # CALIBRATION STANDARDS 4 OR LESS
StdNum = InputBox("Number of calibration standards:")
If Val(StdNum) > 4 Then
   StdExceed = MsgBox(" No more than 4 calibration standards allowed!!", vbOKOnly)
   GoTo Out6
End If

'INPUT BASIC DATA REQUIRED FOR CORRECTION FROM KEYBOARD
Method = InputBox("Correction Method: 1= Multi-element; 2= Single Element")
drift = InputBox("Name of sample used for drift correction")
If Method = 1 Then
   numelem = InputBox("Number of Internal Standard Elements: ")
   ConcTotal = InputBox("Total Concentration of Internal Standard Elements For
Samples")
   If StdNum > 1 Then
      For x = 2 To StdNum
         IntStdConc(x) = InputBox("Internal Standard Element Conc. For Standard" & x)
      Next
   End If
   StdConc = InputBox("Total Concentration of Elements For Drift Sample")
    For k = 1 To numelem
```

```
      namelem(k) = InputBox("Internal Element Name #" & k)
    Next
Else
   element = InputBox("Internal Standard Element Name, e.g. Zn: ")
   YesNo = MsgBox("One Internal Standard Element Conc. For All Samples?",
vbYesNo)
   If YesNo = vbYes Then
      elem = InputBox("Internal Standard Element Conc. For Samples(ppm)")
      k1 = 1
   End If
   If StdNum > 1 Then
      For x = 2 To StdNum
      IntStdConc(x) = InputBox("Internal Standard Element Conc. For Standard" & x)
      Next
   End If
   StdElem = InputBox("Internal Standard Element Concentration For Drift Sample
(ppm)")
End If

'ADJUST COLUMN HEADERS AT START OF DATA
Rows("1:3").Select
Selection.Insert Shift:=xlDown
Rows("5:5").Select
Selection.Cut
Rows("1:1").Select
ActiveSheet.Paste
Rows("5:5").Select
Selection.Delete Shift:=xlUp
Rows("6:7").Select
Selection.Cut
Rows("2:3").Select
ActiveSheet.Paste
Rows("6:7").Select
Selection.Delete Shift:=xlUp

'REMOVE DATE AND TIME HEADERS FOR EACH ANALYSIS
For Each pCell In WkSheetC1
   nn = nn + 1
   If WkWb.Cells(nn, 1).Value Like "Standard" Or WkWb.Cells(nn, 1).Value _
    Like "Sample" Then
      Application.Rows(nn + 1).Select
      Selection.Cut
      Application.Rows(nn + 2).Select
      Selection.Cut
      Application.Rows(nn + 2).Select
      Selection.Delete Shift:=xlUp
```

```
      Application.Rows(nn + 1).Select
      Selection.Delete Shift:=xlUp
      Application.Rows(nn - 1).Select
      Selection.Cut
      Application.Rows(nn - 1).Select
      Selection.Delete Shift:=xlUp
   ElseIf WkWb.Cells(nn, 1).Value Like "End" Then
      Exit For
   End If
Next
nn = 0
'USE BLANK TO COUNT # ELEMENTS ANALYSED (i5)
For Each pCell In WkSheetC1
   nn = nn + 1
   If nn = 1 Then GoTo Out90
   If WkWb.Cells(nn, 1).Value Like "Blank" And WkWb.Cells(nn - 1, 1).Value _
    Like "Blank" Then
      BlankBlank = 1
      GoTo Out90
   End If
      If IsEmpty(WkWb.Cells(nn, 3)) And Not IsEmpty(WkWb.Cells(nn, 2)) And _
      BlankBlank = 1 Then
        i5 = i5 + 1
      End If
   If WkWb.Cells(nn, 1).Value Like "Standard" Then
      BlankBlank = 0
      StandCount = StandCount + 1
      StandardName(StandCount) = WkWb.Cells(nn - 1, 1).Value
   ElseIf WkWb.Cells(nn, 1).Value Like "Sample" And WkWb.Cells(nn - 1, 1).Value _
       Like "Blank" Then
      BlankBlank = 0
   ElseIf WkWb.Cells(nn, 1).Value Like "Sample" And Not WkWb.Cells(nn - 1,
1).Value _
     Like "Blank" Then
      Exit For
   End If
Out90:
Next
nn = 0

'MOVE INTEGRATED COUNT TIMES MEASURED IN MICROSECONDS
For Each pCell In WkSheetC1
   nn = nn + 1
   If nn = 1 Then GoTo Out99
   If WkWb.Cells(nn, 1).Value Like "Blank" Or WkWb.Cells(nn, 1).Value _
    Like "Standard" Or WkWb.Cells(nn, 1).Value Like "Sample" Then
```

```
      For x = 1 To i5
         WkWb.Cells(x + nn + i5, 7).Value = WkWb.Cells(x + nn, 2).Value
      Next
   ElseIf WkWb.Cells(nn, 1).Value Like "End" Then
      Exit For
   End If
Out99:
Next

'ERASE INTEGRATED COUNT TIMES AFTER MOVING AND MOVE COLUMNS
UP
j = 0
For Each pCell In WkSheetC1
   j = j + 1
   If j = 1 Then GoTo Out98
   If WkWb.Cells(j, 1).Value Like "Blank" And WkWb. _
    Cells(j - 1, 1).Value Like "Blank" Then
      For x = i5 To 1 Step -1
         Application.Rows(j + x).Select
         Selection.Cut
         Selection.Delete Shift:=xlUp
      Next
   ElseIf WkWb.Cells(j, 1).Value Like "Sample" Or WkWb.Cells(j, 1).Value _
    Like "Standard" Then
      For x = i5 To 1 Step -1
         Application.Rows(j + x).Select
         Selection.Cut
         Selection.Delete Shift:=xlUp
      Next
   ElseIf WkWb.Cells(j, 1).Value Like "End" Then
      Exit For
   End If
Out98:
Next


n = 0
For Each pCell In WkSheetC1
   n = n + 1
   If n = 1 Then GoTo Out1000

'POSITION OF INTERNAL STANDARD ELEMENT FROM STANDARD 1
   If WkWb.Cells(n, 1).Value Like StandardName(1) Then
      BlockNumber = BlockNumber + 1
      If Method = 2 Then
         For x = 1 To i5
```

```
            If WkWb.Cells(n + x + 1, 1).Value = element Then
               k5 = x
            End If
         Next
      End If
   End If
'CHECK COLUMNS FOR STANDARDS 2, 3 AND 4
'DATA FOR STANDARD 2
   If WkWb.Cells(n, 1).Value Like StandardName(2) Then
      St2 = 1
   End If

'DATA FOR INTERNAL STANDARD ELEMENT(S)
'CALCULATE ABLATION FACTOR (Std2Factor)
'IDENTIFY CALIBRATION ELEMENTS IN STANDARD 2 (Std2El)
   If Method = 2 Then
      For x = 1 To i5
         If WkWb.Cells(n + x + 1, 1).Value = element And St2 = 1 Then
            k10 = 1
            Std2Factor(BlockNumber) = IntStdConc(2) / _
              WkWb.Cells(n + x + 1, 6).Value
         ElseIf WkWb.Cells(n + x + 1, 6).Value = "*" And St2 = 1 Then
            k2 = k2 + 1
            Std2El(k2) = WkWb.Cells(n + x + 1, 1).Value
         End If
      Next
End If
If Method = 1 Then
   For k = 1 To numelem
   For x = 1 To i5
      If WkWb.Cells(n + x, 1).Value = namelem(k) And St2 = 1 Then
         k10 = 1
         k9 = k9 + 1
         intflag(k) = x
         InTotal = InTotal + WkWb.Cells(n + x, 6).Value
            If k9 = numelem Then
               Std2Factor(BlockNumber) = IntStdConc(2) / InTotal
               k9 = 0
               InTotal = 0
               GoTo Out550
            End If
      ElseIf WkWb.Cells(n + x, 6) = "*" And St2 = 1 Then
         k2 = k2 + 1
         Std2El(k2) = WkWb.Cells(n + x, 1).Value
      End If
   Next
```

```
        Next
Out550:
    End If
    St2 = 0

'DATA FOR STANDARD 3
    k9 = 0
    If WkWb.Cells(n, 1).Value Like StandardName(3) Then
        St3 = 1
    End If

'DATA FOR INTERNAL STANDARD ELEMENT(S)
'CALCULATE ABLATION FACTOR (Std3Factor)
'IDENTIFY CALIBRATION ELEMENTS IN STANDARD 3 (Std3El)
    If Method = 2 Then
        For x = 1 To i5
            If WkWb.Cells(n + x + 1, 1).Value = element And St3 = 1 Then
                k10 = 1
                Std3Factor(BlockNumber) = IntStdConc(3) / WkWb.Cells _
                  (n + x + 1, 6).Value
            ElseIf WkWb.Cells(n + x + 1, 6) = "*" And St3 = 1 Then
                k3 = k3 + 1
                Std3El(k3) = WkWb.Cells(n + x + 1, 1).Value
            End If
        Next
End If
If Method = 1 Then
    For k = 1 To numelem
    For x = 1 To i5
        If WkWb.Cells(n + x + 1, 1).Value = namelem(k) And St3 = 1 Then
            k10 = 1
            k9 = k9 + 1
            intflag(k) = x
            InTotal = InTotal + WkWb.Cells(n + x + 1, 6).Value
            If k9 = numelem Then
                Std3Factor(BlockNumber) = IntStdConc(3) / InTotal
                k9 = 0
                InTotal = 0
                GoTo Out560
            End If
        ElseIf WkWb.Cells(n + x + 1, 6) = "*" And St3 = 1 Then
            k3 = k3 + 1
            Std3El(k3) = WkWb.Cells(n + x + 1, 1).Value
        End If
    Next
    Next
```

```
Out560:
   End If
St3 = 0

'DATA FOR STANDARD 4
   If WkWb.Cells(n, 1).Value Like "Standard 4" Then
      St4 = 1
   End If

'DATA FOR INTERNAL STANDARD ELEMENT(S)
'CALCULATE ABLATION FACTOR (Std4Factor)
'IDENTIFY CALIBRATION ELEMENTS IN STANDARD 4 (Std4El)
   If Method = 2 Then
      For x = 1 To i5
         If WkWb.Cells(n + x + 1, 1).Value = element And St4 = 1 Then
            k10 = 1
            Std4Factor(BlockNumber) = IntStdConc(4) / WkWb.Cells(n + x, 8).Value
         ElseIf WkWb.Cells(n + x, 6) = "*" And St4 = 1 Then
            k4 = k4 + 1
            Std4El(k4) = WkWb.Cells(n + x, 1).Value
         End If
      Next
   End If
   If Method = 1 Then
      For k = 1 To numelem
      For x = 1 To i5
         If WkWb.Cells(n + x, 1).Value = namelem(k) And St4 = 1 Then
            k10 = 1
            k9 = k9 + 1
            intflag(k) = x
            InTotal = InTotal + WkWb.Cells(n + x, 6).Value
            If k9 = numelem Then
               Std4Factor(BlockTotal) = IntStdConc(4) / InTotal
               k9 = 0
               InTotal = 0
               GoTo Out570
            End If
         ElseIf WkWb.Cells(n + x, 6) = "*" And St4 = 1 Then
            k4 = k4 + 1
            Std4El(k4) = WkWb.Cells(n + x, 1).Value
         End If
      Next
      Next
Out570:
   End If
St4 = 0
```

```
'EXIT AFTER EVALUATING STANDARDS
   If WkWb.Cells(n, 1).Value Like "Sample" And Not WkWb.Cells(n - 1, 1).Value _
    Like "Blank" Then
      Exit For
   End If
Out1000:
Next

'COPY ELEMENT NAMES FROM STANDARD 1 TO RESULTS AREA
For Each pCell In WkSheetC1
   ii = ii + 1
   If (WkWb.Cells(ii, 1).Value Like "Standard") Then
      flag = 1
      For x = 1 To i5
         WkWb.Cells(x + 1, 11).Value = WkWb.Cells(ii + x, 1).Value
         WkWb.Cells(x + 1, 11).Font.Color = RGB(0, 0, 255)
         WkWb.Cells(x + 1, 11).Font.Bold = "True"
      Next
   End If
   If flag = 1 Then
      Exit For
   End If
Next
n = 0
For Each pCell In WkSheetC1
   n = n + 1

'CALCULATE "BLANK AS SAMPLE" GROSS COUNTS (GrossBlank)
   If WkWb.Cells(n, 1).Value Like "Sample" Then
      If WkWb.Cells(n - 1, 1).Value Like "Blank" Then
         i6 = i6 + 1
         For x = 1 To i5
            Blank(x, i6) = Abs(WkWb.Cells(n + x, 3).Value)
            GrossBlank(x, i6) = Abs(WkWb.Cells(n + x, 3).Value) * _
             (WkWb.Cells(n + x, 7).Value / 1000)
            If i6 = 1 Then
               BlankConc(x) = Abs(WkWb.Cells(n + x, 6).Value)
               BlankNet(x) = Abs(WkWb.Cells(n + x, 5).Value)
            End If
         Next
         If i6 = 1 Then
         End If
         GoTo Out800
      ElseIf Not WkWb.Cells(n - 1, 1).Value Like "Blank" Then
      I = I + 1
```

```
'COPY SAMPLE NAMES TO RESULTS AREA
      WkWb.Cells(1, I).Value = WkWb.Cells(n - 1, 1).Value
      WkWb.Cells(1, I).HorizontalAlignment = xlRight
      WkWb.Cells(1, I).Font.Bold = "True"
      WkWb.Cells(1, I).Font.Color = RGB(204, 0, 204)
      SampNam(I) = WkWb.Cells(n - 1, 1).Value

'INTERNAL STANDARD CONCENTRATIONS FOR EACH SAMPLE (Econc)
FROM KEYBOARD
'IF FLAG k1 SET TO ZERO
      If Method = 2 Then
         If k1 = 0 Then
            If SampNam(I) <> drift Then
               Conc(I) = InputBox("Internal Standard Concentration (ppm) For " _
                & SampNam(I) & "Element:  " & element)
               Econc(I) = Val(Conc(I))
            End If
         End If
      End If
      End If
   ElseIf WkWb.Cells(n, 1).Value Like "End" Then
      Exit For
   End If
Out800:
Next

'INITIALIZE VARIABLES
I = 11: i2 = 0: i3 = 1: i4 = 0: j2 = 0: NewVal = 0: BlockTotal = 0

'CPS DATA
For Each pCell In WkSheetC1
   i2 = i2 + 1
   If i2 = 1 Then GoTo Out2
   If WkWb.Cells(i2, 1) Like StandardName(1) Then
      BlockTotal = BlockTotal + 1
   End If
   If WkWb.Cells(i2, 1).Value Like "End" Then
      Exit For
   End If
   If (WkWb.Cells(i2, 1).Value Like "Standard") Then
      j2 = 0
      GoTo Out2
   End If
   If (WkWb.Cells(i2, 1).Value) Like "Sample" And Not (WkWb.Cells(i2 - 1, 1).Value)
_
```

```
      Like "Blank" Then
      i3 = i3 + 1
      i4 = i4 + 1
       I = I + 1
       For x = 1 To i5

'RECALCULATE CONC. FOR CALIBRATION ELEMENTS IN STANDARD 2
'BASED ON ABLATION FACTOR
      For m = 1 To k2
         If WkWb.Cells(i2 + x, 1).Value = Std2El(m) Then
            WkWb.Cells(i2 + x, 6).Value = WkWb.Cells(i2 + x, 6).Value * _
             Std2Factor(BlockNumber)
         End If
      Next

'RECALCULATE CONC. FOR CALIBRATION ELEMENTS IN STANDARD 3
'BASED ON ABLATION FACTOR
      For m = 1 To k3
         If WkWb.Cells(i2 + x, 1).Value = Std3El(m) Then
            WkWb.Cells(i2 + x, 6).Value = WkWb.Cells(i2 + x, 6).Value * _
             Std3Factor(BlockNumber)
         End If
      Next

'RECALCULATE CONC. FOR CALIBRATION ELEMENTS IN STANDARD 4
'BASED ON ABLATION FACTOR
      For m = 1 To k4
         If WkWb.Cells(i2, 1).Value = Std4El(m) Then
            WkWb.Cells(i2, 6).Value = WkWb.Cells(i2, 6).Value * _
             Std4Factor(BlockNumber)
         End If
      Next

'MOVE CONC. DATA FOR SAMPLES TO RESULTS AREA
         If WkWb.Cells(i2 + x, 6).Value <= 0 Then
            WkWb.Cells(x + 1, I).Value = "n.d."
            WkWb.Cells(x + 1, I).HorizontalAlignment = xlRight
         Else
         WkWb.Cells(x + 1, I).Value = WkWb.Cells(i2 + x, 6).Value
         End If
      Next
   End If
Out2:
Next
i2 = 0
```

```
'CALCULATE ABLATION FACTOR (Factor) IF USING MULTI-ELEMENT
INTERNAL STANDARD
If Method = 1 Then
   For xx = 1 To i4
      NewVal = 0
   For i7 = 1 To numelem
   For x = 1 To i5
      If WkWb.Cells(x + 1, 11).Value = namelem(i7) Then
         NewVal = NewVal + WkWb.Cells(x + 1, xx + 11).Value
      End If
   Next
   Next
   If WkWb.Cells(1, xx + 11).Value Like drift Then
      Factor(xx) = Val(StdConc) / NewVal
   Else
      Factor(xx) = Val(ConcTotal) / NewVal
   End If
   Next
End If

'CALCULATE DETECTION LIMIT (DetLim) FOR EACH ELEMENT ANALYSED
If Detect = vbYes Then
   For a = 1 To i5
      BlankTotal(a) = 0
      SumSq(a) = 0
   Next
   For a = 1 To i5
   For b = 1 To i6
      BlankTotal(a) = BlankTotal(a) + Blank(a, b)
   Next
   BlankMean(a) = BlankTotal(a) / i6
   Next
   For a = 1 To i5
   For b = 1 To i6
      SqMeanDev(a) = (Blank(a, b) - BlankMean(a)) ^ 2
      SumSq(a) = SumSq(a) + SqMeanDev(a)
   Next
   Next
   For a = 1 To i5
      StdDev(a) = (Sqr(SumSq(a) / i6)) * BlankConc(a) / BlankNet(a)
      DetLim(a) = 3 * StdDev(a)
   Next
End If
'FORMAT CELLS
WkWb.Cells(i5 + 2, 11).Value = "Corrected Data"
WkWb.Cells(i5 + 2, 11).Font.Bold = "True"
```

```
WkWb.Cells(i5 + 2, 11).Font.Color = RGB(255, 0, 0)

'CORRECTIONS FOR SINGLE INTERNAL ELEMENT
If Method <> 1 Then
For kk = 12 To (11 + i4)
For x = 1 To i5
   kj1 = x

'COPY ELEMENT NAMES AND FORMAT CELLS
   WkWb.Cells(x + i5 + 3, 11).Value = WkWb.Cells(x + 1, 11).Value
   WkWb.Cells(x + i5 + 3, 11).Font.Bold = "True"
   WkWb.Cells(x + i5 + 3, 11).Font.Color = RGB(0, 0, 255)
   If WkWb.Cells(x + 1, kk).Value = "S" Then
      GoTo Out4
   End If

'ADJUST CONC. FOR VARIABLE INTERNAL ELEMENT VALUES FOR SAMPLES
   If Not IsEmpty(WkWb.Cells(x + 1, kk)) And k1 = 0 Then
      If WkWb.Cells(x + 1, kk) = "n.d." Then
         WkWb.Cells(x + i5 + 3, kk).Value = "n.d."
         WkWb.Cells(x + i5 + 3, kk).HorizontalAlignment = xlRight
         GoTo Out5000
      End If
      If WkWb.Cells(1, kk).Value Like drift Then
         WkWb.Cells(x + i5 + 3, kk).Value = WkWb.Cells(x + 1, kk).Value _
          * (Val(StdElem) / WkWb.Cells(k5 + 1, kk).Value)
      Else
         WkWb.Cells(x + i5 + 3, kk).Value = WkWb.Cells(x + 1, kk).Value _
          * (Econc(kk) / WkWb.Cells(k5 + 1, kk).Value)
      End If

' ADJUST CONCENTRATIONS FOR SINGLE INTERNAL ELEMENT VALUE FOR
SAMPLES
   ElseIf Not IsEmpty(WkWb.Cells(x + 1, kk)) And k1 = 1 Then
      If WkWb.Cells(x + 1, kk) = "n.d." Then
         WkWb.Cells(x + i5 + 3, kk).Value = "n.d."
         WkWb.Cells(x + i5 + 3, kk).HorizontalAlignment = xlRight
         GoTo Out5000
      End If
      If WkWb.Cells(1, kk).Value Like drift Then
         WkWb.Cells(x + i5 + 3, kk).Value = WkWb.Cells(x + 1, kk).Value _
          * (Val(StdElem) / WkWb.Cells(k5 + 1, kk).Value)
      Else
      WkWb.Cells(x + i5 + 3, kk).Value = WkWb.Cells(x + 1, kk).Value _
          * (Val(elem) / WkWb.Cells(k5 + 1, kk).Value)
      End If
```

```
Out5000:
   End If

'COPY SAMPLE NAMES AND FORMAT CELLS
   WkWb.Cells(x + 1, kk).NumberFormat = "0.0"
      WkWb.Cells(i5 + 3, kk).Value = WkWb.Cells(1, kk).Value
      WkWb.Cells(i5 + 3, kk).HorizontalAlignment = xlRight
      WkWb.Cells(i5 + 3, kk).Font.Bold = "True"
      WkWb.Cells(i5 + 3, kk).Font.Color = RGB(255, 0, 255)
   WkWb.Cells(x + i5 + 3, kk).NumberFormat = "0.0"
   If WkWb.Cells(x + 1, kk).Value < 0.1 Then
      WkWb.Cells(x + 1, kk).NumberFormat = "0.00"
   End If
   If WkWb.Cells(x + i5 + 3, kk).Value < 0.1 Then
      WkWb.Cells(x + i5 + 3, kk).NumberFormat = "0.00"
   End If
Out4:
Next
Next

'CORRECTIONS FOR MULTI-ELEMENT INTERNAL STANDARD METHOD
Else
For kk = 12 To i4 + 11
For x = 1 To i5
   kj1 = x
   If kk = 12 Then

'COPY ELEMENTS NAMES AND FORMAT CELLS
      WkWb.Cells(x + i5 + 3, 11).Value = WkWb.Cells(x + 1, 11).Value
      WkWb.Cells(x + i5 + 3, 11).Font.Bold = "True"
      WkWb.Cells(x + i5 + 3, 11).Font.Color = RGB(0, 0, 255)
   End If
   If WkWb.Cells(x + 1, kk).Value = "S" Then
      GoTo Out10
   End If

'** IN CELLS FOR ELEMENTS USED AS INTERNAL STANDARDS
'DRIFT SAMPLE VS REGULAR SAMPLE IDENTIFIED BY DIFFERENT COLORS
   If Not IsEmpty(WkWb.Cells(x + 1, kk)) Then
      For i30 = 1 To numelem
         If WkWb.Cells(x + i5 + 3, 11).Value = namelem(i30) Then
            WkWb.Cells(x + i5 + 3, kk).Value = "**"
            If WkWb.Cells(1, kk).Value Like drift Then
               WkWb.Cells(x + i5 + 3, kk).HorizontalAlignment = xlRight
               WkWb.Cells(x + i5 + 3, kk).Font.Color = RGB(0, 255, 0)
               WkWb.Cells(i5 + i5 + 4, 18).Value = _
```

```
                " **    Internal Elements Summed To (ppm):  " & Val(StdConc)
              WkWb.Cells(i5 + i5 + 4, 18).Font.Bold = "True"
              WkWb.Cells(i5 + i5 + 4, 18).Font.Color = RGB(0, 255, 0)
            Else
              WkWb.Cells(x + i5 + 3, kk).HorizontalAlignment = xlRight
              WkWb.Cells(x + i5 + 3, kk).Font.Color = RGB(255, 0, 0)
              WkWb.Cells(i5 + i5 + 4, 12).Value = _
                " **    Internal Elements, Summed To (ppm):  " & Val(ConcTotal)
              WkWb.Cells(i5 + i5 + 4, 12).Font.Bold = "True"
              WkWb.Cells(i5 + i5 + 4, 12).Font.Color = RGB(255, 0, 0)
            End If
          GoTo Out30
          ElseIf WkWb.Cells(x + 1, kk).Value = "n.d." Then
            WkWb.Cells(x + i5 + i3, kk).Value = "n.d."
            WkWb.Cells(x + i5 + i3, kk).HorizontalAlignment = xlRight
          Else

'COPY CONC. VALUES TO RESULTS AREA
            WkWb.Cells(x + i5 + 3, kk).Value = WkWb.Cells(x + 1, kk). _
            Value * Factor(kk - 11)
          End If
        Next
      End If
Out30:

'COPY SAMPLE NAMES AND FORMAT CELLS
    WkWb.Cells(i5 + 3, kk).Value = WkWb.Cells(1, kk).Value
    WkWb.Cells(i5 + 3, kk).HorizontalAlignment = xlRight
    WkWb.Cells(i5 + 3, kk).Font.Bold = "True"
    WkWb.Cells(i5 + 3, kk).Font.Color = RGB(255, 0, 255)
    WkWb.Cells(x + 1, kk).NumberFormat = "0.0"
    WkWb.Cells(x + i5 + 3, kk).NumberFormat = "0.0"
    If WkWb.Cells(x + i5 + 3, kk).Value < 0.1 Then
      WkWb.Cells(x + 1, kk).NumberFormat = "0.00"
    End If
    If WkWb.Cells(x + i5 + 3, kk).Value < 0.1 Then
      WkWb.Cells(x + i5 + 3, kk).NumberFormat = "0.00"
    End If
Out10:
Next
Next
End If

'DRIFT CORRECTION USING DRIFT SAMPLES (drift)
'DRIFT SAMPLE CONC. ADJUSTED TO FIRST DRIFT SAMPLE VALUES
kj2 = kj1 + i5 + 8
```

```
WkWb.Cells(kj2 - 2, 11).Value = "Drift Corrected Data"
WkWb.Cells(kj2 - 2, 11).Font.Bold = "True"
WkWb.Cells(kj2 - 2, 11).Font.Color = RGB(255, 0, 0)
knum = 0
For kk = 12 To I
If WkWb.Cells(1, kk).Value = drift Then
   knum = knum + 1
   DriftNum(knum) = kk
End If
Next
k10 = 0

'NUMBER OF SAMPLES (kint) BETWEEN EACH DRIFT SAMPLE
'NUMBER OF DRIFT SAMPLES = 0 to k10
For kk = 12 To I
For i12 = 1 To knum
   If kk = DriftNum(i12) Then
      k10 = k10 + 1
      kint(k10) = kk
      For x = 1 To i5
         If Method = 1 And WkWb.Cells(x + i5 + 3, kk).Value = "**" Then
         GoTo Out200
         ElseIf Method = 1 And WkWb.Cells(x + i5 + 3, kk).Value = "n.d." Then
         GoTo Out200
         End If
         DriftVal(x + 1, k10) = WkWb.Cells(x + i5 + 3, kk)
Out200:
      Next
      If k10 > 1 Then
         DriftInt(k10 - 1) = kint(k10) - kint(k10 - 1)
      End If
   End If
Next
Next
For x = 1 To i5
For jj = 1 To k10 - 1
For kk = 12 To i4 + 11
   If Method = 1 And WkWb.Cells(x + i5 + 3, kk).Value = "**" Then
      GoTo Out201
   ElseIf Method = 1 And WkWb.Cells(x + i5 + 3, kk).Value = "n.d." Then
      GoTo Out201
   End If
Next

'SAMPLE FRACTIONAL DIFFERENTIAL (DriftFrac)
   DriftFrac(x + 1, jj) = DriftVal(x + 1, jj + 1) / DriftVal(x + 1, 1)
```

```
'SAMPLE ABSOLUTE DIFFERENTIAL (DriftDiff)
   DriftDiff(x + 1, jj) = DriftVal(x + 1, 1) - DriftVal(x + 1, jj + 1)
Out201:
Next
Next

'COPY ELEMENT NAMES
k12 = 0
For mn = kj2 To (kj2 + i5 - 1)
k13 = mn - i5 - 4
k12 = k12 + 1
   WkWb.Cells(kj2 - 1, 10).Value = "D.L."
   WkWb.Cells(kj2 - 1, 10).HorizontalAlignment = xlRight
   WkWb.Cells(kj2 - 1, 10).Font.Color = RGB(255, 0, 255)
   WkWb.Cells(kj2 - 1, 10).Font.Bold = "True"
   WkWb.Cells(mn, 11).Value = WkWb.Cells(k13, 11).Value
   WkWb.Cells(mn, 10).Value = DetLim(k12)
   WkWb.Cells(mn, 10).Interior.Color = RGB(255, 204, 204)
   WkWb.Cells(mn, 10).Interior.Pattern = xlSolid
   WkWb.Cells(mn, 11).Font.Bold = "True"
   WkWb.Cells(mn, 11).Font.Color = RGB(0, 0, 255)
k7 = k7 + 1
If knum = 1 Then
   WkWb.Cells(mn, kint(1)).Value = WkWb.Cells((mn - i5 - 4), kint(jj)).Value
   If WkWb.Cells(mn, kint(1)).Value = "**" Then
      WkWb.Cells(mn, kint(1)).Value = "**"
      WkWb.Cells(mn, kint(1)).HorizontalAlignment = xlRight
      WkWb.Cells(mn, kint(1)).Font.Color = RGB(255, 0, 0)
   ElseIf WkWb.Cells(mn, kint(1)).Value = "n.d." Then
      WkWb.Cells(mn, kint(1)).Value = "n.d."
      WkWb.Cells(mn, kint(1)).HorizontalAlignment = xlRight
   End If
End If
For jj = 1 To k10 - 1
If DriftFrac(k7, jj) < 1 Then
IntFactor(k7, jj) = (1 - DriftFrac(k7, jj)) / DriftInt(jj)
Else
IntFactor(k7, jj) = (Abs(DriftFrac(k7, jj)) - 1) / DriftInt(jj)
End If
For jk = kint(jj) + 1 To kint(jj + 1)
   k6 = k6 + 1

'COPY DATA FOR FIRST DRIFT SAMPLE AND FORMAT CELLS
   If k6 = 1 And jj = 1 Then
      WkWb.Cells(mn, kint(jj)).Value = WkWb.Cells((mn - i5 - 4), kint(jj)).Value
```

```
      If WkWb.Cells(mn, kint(jj)).Value = "**" Then
         WkWb.Cells(mn, kint(jj)).Value = "**"
         WkWb.Cells(mn, kint(jj)).HorizontalAlignment = xlRight
         WkWb.Cells(mn, kint(jj)).Font.Color = RGB(255, 0, 0)
      ElseIf WkWb.Cells(mn, kint(jj)).Value = "n.d." Then
         WkWb.Cells(mn, kint(jj)).Value = "n.d."
         WkWb.Cells(mn, kint(jj)).HorizontalAlignment = xlRight
      End If
   End If

'COPY ALL ** AND N.D. DATA AND FORMAT CELLS
   If DriftDiff(k7, jj) < 0 Then
      If WkWb.Cells((mn - i5 - 4), jk).Value = "**" Then
         WkWb.Cells(mn, jk).Value = "**"
         WkWb.Cells(mn, jk).HorizontalAlignment = xlRight
         WkWb.Cells(mn, jk).Font.Color = RGB(255, 0, 0)
         GoTo Out202
      ElseIf WkWb.Cells((mn - i5 - 4), jk).Value = "n.d." Then
         WkWb.Cells(mn, jk).Value = "n.d."
         WkWb.Cells(mn, jk).HorizontalAlignment = xlRight
         GoTo Out202
      End If

'APPLY DRIFT CORRECTION IF DriftDiff < 0, COPY, AND FORMAT CELLS
      WkWb.Cells(mn, jk).Value = WkWb.Cells((mn - i5 - 4), jk).Value / _
         (1 + (IntFactor(k7, jj) * k6))
   Else
      If WkWb.Cells((mn - i5 - 4), jk).Value = "**" Then
         WkWb.Cells(mn, jk).Value = "**"
         WkWb.Cells(mn, jk).HorizontalAlignment = xlRight
         WkWb.Cells(mn, jk).Font.Color = RGB(255, 0, 0)
         GoTo Out202
      ElseIf WkWb.Cells((mn - i5 - 4), jk).Value = "n.d." Then
         WkWb.Cells(mn, jk).Value = "n.d."
         WkWb.Cells(mn, jk).HorizontalAlignment = xlRight
         GoTo Out202
      End If

'APPLY DRIFT CORRECTION IF DriftDiff > 0, COPY, AND FORMAT CELLS
      WkWb.Cells(mn, jk).Value = WkWb.Cells((mn - i5 - 4), jk).Value / _
         (1 - (IntFactor(k7, jj) * k6))
   End If
Out202:
Next
k6 = 0
Next
```

```
Next
For mn = kj2 To (kj2 + i5 - 1)
For jk = kint(1) To kint(k10)

'COPY SAMPLE NAMES AND FORMAT CELLS
   WkWb.Cells(kj2 - 1, jk).Value = WkWb.Cells(1, jk).Value
   WkWb.Cells(kj2 - 1, jk).HorizontalAlignment = xlRight
   WkWb.Cells(kj2 - 1, jk).Font.Bold = "True"
   WkWb.Cells(kj2 - 1, jk).Font.Color = RGB(255, 0, 255)
   If WkWb.Cells(mn, jk).Value >= 1# Or WkWb.Cells(mn, 10).Value >= 1# Then
      WkWb.Cells(mn, jk).NumberFormat = "0.0"
      WkWb.Cells(mn, 10).NumberFormat = "0.00"
   ElseIf WkWb.Cells(mn, jk).Value = "n.d." Or WkWb.Cells(mn, jk).Value = "**" Then
      WkWb.Cells(mn, jk).HorizontalAlignment = xlRight
   ElseIf WkWb.Cells(mn, jk).Value < 1# Or WkWb.Cells(mn, 10).Value < 1# Then
      WkWb.Cells(mn, jk).NumberFormat = "0.00"
      WkWb.Cells(mn, 10).NumberFormat = "0.00"
   End If

'ADD CELL SHADING FOR ALL DRIFT SAMPLES
   If WkWb.Cells(kj2 - 1, jk) = drift Then
      k8 = 1
      If k8 = 1 Then
         WkWb.Cells(kj2 - 1, jk).Font.Color = RGB(204, 204, 204)
         WkWb.Cells(mn, jk).Interior.Color = RGB(204, 204, 204)
         WkWb.Cells(mn, jk).Interior.Pattern = xlSolid
      End If
   End If
Next
Next
If kint(k10) < I Then
   x = 0
   For mn = kj2 To (kj2 + i5 - 1)
   x = x + 1
   For jk = kint(k10) + 1 To I
      WkWb.Cells(kj2 - 1, jk).Value = WkWb.Cells(1, jk).Value
      WkWb.Cells(kj2 - 1, jk).HorizontalAlignment = xlRight
      WkWb.Cells(kj2 - 1, jk).Font.Bold = "True"
      WkWb.Cells(kj2 - 1, jk).Font.Color = RGB(255, 0, 255)
      WkWb.Cells(mn, jk).Value = WkWb.Cells(x + i5 + 3, jk).Value
      If WkWb.Cells(mn, jk).Value >= 1# Or WkWb.Cells(mn, 10).Value >= 1# Then
         WkWb.Cells(mn, jk).NumberFormat = "0.0"
         WkWb.Cells(mn, 10).NumberFormat = "0.00"
      End If
      If WkWb.Cells(mn, jk).Value = "n.d." Or WkWb.Cells(mn, jk).Value = "**" Then
         WkWb.Cells(mn, jk).HorizontalAlignment = xlRight
```

```
        End If
        If WkWb.Cells(mn, jk).Value < 1# Or WkWb.Cells(mn, 10).Value < 1# Then
           WkWb.Cells(mn, jk).NumberFormat = "0.00"
           WkWb.Cells(mn, 10).NumberFormat = "0.00"
        End If
     Next
     Next
End If

n = 0

'ADD NEW HEADER FOR ERRORS, COPY SAMPLE NAMES AND ELEMENT
NAMES
kj3 = kj2 + i5 + 4
WkWb.Cells(kj3 - 2, 11).Value = "Counting Errors (ppm)"
WkWb.Cells(kj3 - 2, 11).Font.Bold = "True"
WkWb.Cells(kj3 - 2, 11).Font.Color = RGB(255, 0, 0)

'COPY SAMPLE NAMES
For jk = kint(1) To I
   WkWb.Cells(kj3 - 1, jk).Value = WkWb.Cells(1, jk).Value
   WkWb.Cells(kj3 - 1, jk).HorizontalAlignment = xlRight
   WkWb.Cells(kj3 - 1, jk).Font.Bold = "True"
   WkWb.Cells(kj3 - 1, jk).Font.Color = RGB(255, 0, 255)
Next

'COPY ELEMENT NAMES
For x = 1 To i5
   WkWb.Cells(kj3 + (x - 1), 11).Value = WkWb.Cells(kj2 + (x - 1), 11).Value
   WkWb.Cells(kj3 + (x - 1), 11).Font.Bold = "True"
   WkWb.Cells(kj3 + (x - 1), 11).Font.Color = RGB(0, 0, 255)
Next
For Each pCell In WkSheetC1
   n = n + 1
   If n <> 1 Then

'ERROR ON BLANK COUNTS (BlankError)
   If WkWb.Cells(n, 1).Value Like "Blank" And WkWb.Cells(n - 1, 1).Value Like
"Blank" Then
      For x = 1 To i5
         If WkWb.Cells(n + x, 3).Value <= 0 Then
            WkWb.Cells(n + x, 3).Value = 0.1
         End If
         BlankError(x) = (Sqr(WkWb.Cells(n + x, 3).Value * (WkWb.Cells(n + x, 7). _
          Value / 1000)) / (WkWb.Cells(n + x, 3).Value * WkWb.Cells(n + x, 7). _
          Value / 1000)) * WkWb.Cells(n + x, 3).Value
```

```
    Next

'ERROR ON STANDARD COUNTS (StdError)
  ElseIf WkWb.Cells(n, 1).Value Like "Standard" And WkWb.Cells(n - 1, 1).Value Like
_
    StandardName(1) Then
     For x = 1 To i5
        If Method = 2 Then
            If WkWb.Cells(n + x, 5).Value <= 0 Then
              WkWb.Cells(n + x, 5).Value = 1
            End If
            If WkWb.Cells(n + x, 3).Value <= 0 Then
               WkWb.Cells(n + x, 3).Value = 1
            End If
          If WkWb.Cells(n + x, 1).Value = element Then
            IntStdNetCounts = WkWb.Cells(n + x, 5).Value
          End If
            StdNetCounts(x) = WkWb.Cells(n + x, 5).Value
            StdError(x) = (Sqr(WkWb.Cells(n + x, 3).Value * (WkWb.Cells(n + x, 7). _
              Value / 1000)) / (WkWb.Cells(n + x, 3).Value * WkWb.Cells(n + x, 7). _
              Value / 1000)) * WkWb.Cells(n + x, 3).Value
        End If
        If Method = 1 Then
          For y = 1 To numelem
            If WkWb.Cells(n + x, 5).Value <= 0 Then
              WkWb.Cells(n + x, 5).Value = 1
            End If
            If WkWb.Cells(n + x, 3).Value <= 0 Then
              WkWb.Cells(n + x, 3).Value = 1
            End If
            If WkWb.Cells(n + x, 1).Value Like namelem(y) Then
              StdIntError1(y) = (Sqr(WkWb.Cells(n + x, 3).Value * (WkWb.Cells(n +
x, 7). _
                Value / 1000)) / (WkWb.Cells(n + x, 3).Value * WkWb.Cells(n + x, 7). _
                Value / 1000)) * WkWb.Cells(n + x, 3).Value
            Else
              StdNetCounts(x) = WkWb.Cells(n + x, 5).Value
              StdError(x) = (Sqr(WkWb.Cells(n + x, 3).Value * (WkWb.Cells(n + x, 7).
_
                Value / 1000)) / (WkWb.Cells(n + x, 3).Value * WkWb.Cells(n + x, 7). _
                Value / 1000)) * WkWb.Cells(n + x, 3).Value
            End If
          Next
        End If
    Next
```

```
'ERROR ON STANDARD 2 COUNTS
   ElseIf WkWb.Cells(n, 1).Value Like "Standard" And WkWb.Cells(n - 1, 1).Value Like
_
    StandardName(2) Then
     For y = 1 To k2
     For x = 1 To i5
        If Method = 2 Then
             If WkWb.Cells(n + x, 5).Value <= 0 Then
               WkWb.Cells(n + x, 5).Value = 1
             End If
             If WkWb.Cells(n + x, 3).Value <= 0 Then
                WkWb.Cells(n + x, 3).Value = 1
             End If
          If WkWb.Cells(n + x, 1).Value = Std2El(y) Then
             StdNetCounts(x) = WkWb.Cells(n + x, 5).Value
             StdError(x) = (Sqr(WkWb.Cells(n + x, 3).Value * (WkWb.Cells(n + x, 7). _
              Value / 1000)) / (WkWb.Cells(n + x, 3).Value * WkWb.Cells(n + x, 7). _
              Value / 1000)) * WkWb.Cells(n + x, 3).Value
          End If
        End If
     Next
     Next
     For x = 1 To i5
        If Method = 1 Then
          For y = 1 To k2
             If WkWb.Cells(n + x, 5).Value <= 0 Then
               WkWb.Cells(n + x, 5).Value = 1
             End If
             If WkWb.Cells(n + x, 3).Value <= 0 Then
                WkWb.Cells(n + x, 3).Value = 1
             End If
             If WkWb.Cells(n + x, 1).Value = Std2El(y) Then
                StdNetCounts(x) = WkWb.Cells(n + x, 5).Value
                StdError(x) = (Sqr(WkWb.Cells(n + x, 3).Value * (WkWb.Cells(n + x, 7).
_
                 Value / 1000)) / (WkWb.Cells(n + x, 3).Value * WkWb.Cells(n + x, 7). _
                 Value / 1000)) * WkWb.Cells(n + x, 3).Value
             End If
          Next
        End If
     Next

'ERROR ON STANDARD 3 COUNTS
   ElseIf WkWb.Cells(n, 1).Value Like "Standard" And WkWb.Cells(n - 1, 1).Value Like
_
    StandardName(3) Then
```

```
For y = 1 To k3
For x = 1 To i5
   If Method = 2 Then
         If WkWb.Cells(n + x, 5).Value <= 0 Then
           WkWb.Cells(n + x, 5).Value = 1
         End If
         If WkWb.Cells(n + x, 3).Value <= 0 Then
            WkWb.Cells(n + x, 3).Value = 1
         End If
      If WkWb.Cells(n + x, 1).Value = Std3El(y) Then
         StdNetCounts(x) = WkWb.Cells(n + x, 5).Value
         StdError(x) = (Sqr(WkWb.Cells(n + x, 3).Value * (WkWb.Cells(n + x, 7). _
          Value / 1000)) / (WkWb.Cells(n + x, 3).Value * WkWb.Cells(n + x, 7). _
          Value / 1000)) * WkWb.Cells(n + x, 3).Value
      End If
   End If
Next
Next
For x = 1 To i5
   If Method = 1 Then
      For y = 1 To k3
         If WkWb.Cells(n + x, 5).Value <= 0 Then
           WkWb.Cells(n + x, 5).Value = 1
         End If
         If WkWb.Cells(n + x, 3).Value <= 0 Then
            WkWb.Cells(n + x, 3).Value = 1
         End If
         If WkWb.Cells(n + x, 1).Value = Std3El(y) Then
            StdNetCounts(x) = WkWb.Cells(n + x, 5).Value
            StdError(x) = (Sqr(WkWb.Cells(n + x, 3).Value * (WkWb.Cells(n + x, 7).
_
             Value / 1000)) / (WkWb.Cells(n + x, 3).Value * WkWb.Cells(n + x, 7). _
             Value / 1000)) * WkWb.Cells(n + x, 3).Value
         End If
      Next
   End If
Next

'ERROR ON STANDARD 4 COUNTS
   ElseIf WkWb.Cells(n, 1).Value Like "Standard" And WkWb.Cells(n - 1, 1).Value Like
_
    "Standard 4" Then
     For y = 1 To k4
     For x = 1 To i5
        If Method = 2 Then
           If WkWb.Cells(n + x, 5).Value <= 0 Then
```

```
            WkWb.Cells(n + x, 5).Value = 1
          End If
          If WkWb.Cells(n + x, 3).Value <= 0 Then
            WkWb.Cells(n + x, 3).Value = 1
          End If
        If WkWb.Cells(n + x, 1).Value = Std4El(y) Then
          StdNetCounts(x) = WkWb.Cells(n + x, 5).Value
          StdError(x) = (Sqr(WkWb.Cells(n + x, 3).Value * (WkWb.Cells(n + x, 7). _
           Value / 1000)) / (WkWb.Cells(n + x, 3).Value * WkWb.Cells(n + x, 7). _
           Value / 1000)) * WkWb.Cells(n + x, 3).Value
        End If
    End If
Next
Next
For x = 1 To i5
    If Method = 1 Then
      For y = 1 To k4
        If WkWb.Cells(n + x, 5).Value <= 0 Then
          WkWb.Cells(n + x, 5).Value = 1
        End If
        If WkWb.Cells(n + x, 3).Value <= 0 Then
          WkWb.Cells(n + x, 3).Value = 1
        End If
        If WkWb.Cells(n + x, 1).Value = Std4El(y) Then
          StdNetCounts(x) = WkWb.Cells(n + x, 5).Value
          StdError(x) = (Sqr(WkWb.Cells(n + x, 3).Value * (WkWb.Cells(n + x, 7).
_
           Value / 1000)) / (WkWb.Cells(n + x, 3).Value * WkWb.Cells(n + x, 7). _
           Value / 1000)) * WkWb.Cells(n + x, 3).Value
        End If
      Next
    End If
  Next
  ElseIf WkWb.Cells(n + x, 1).Value Like "End" Then
    Exit For
  End If
  End If
Next

'ERROR ON NET STANDARD COUNTS (StdNetError)
For x = 1 To i5
  If StdError(x) = 0 Then GoTo Out900
  StdNetError(x) = Sqr(StdError(x) ^ 2 + BlankError(x) ^ 2)
Out900:
Next
```

```
'ERROR ON NET COUNTS FOR INTERNAL ELEMENT IN
STANDARD(IntStdNetError)
If Method = 2 Then
   IntStdNetError = Sqr(StdError(k5) ^ 2 + BlankError(k5) ^ 2)
End If
   n = 0: m = 0
For Each pCell In WkSheetC1
   If Method = 2 Then
      n = n + 1
      If n > 1 Then
'         If WkWb.Cells(n - 1, 1).Value Like StandardName(1) Then
'           For x = 1 To i5
         If WkWb.Cells(n, 1).Value Like "Sample" And Not WkWb.Cells(n - 1, 1). _
          Value Like "Blank" Then
           m = m + 1

'NET COUNTS FOR INTERNAL ELEMENT IN SAMPLE (IntSampleNetCounts)
'ERROR ON SAMPLE INTERNAL ELEMENT COUNTS (IntSampleError)
'ERROR ON SAMPLE INTERNAL ELEMENT NET COUNTS (IntSampleNetError)
            For x = 1 To i5
               If WkWb.Cells(n + x, 5).Value <= 0 Then
                  WkWb.Cells(n + x, 5).Value = 1
               End If
               If WkWb.Cells(n + x, 3).Value <= 0 Then
                  WkWb.Cells(n + x, 3).Value = 1
               End If
               If WkWb.Cells(n + x, 1).Value = element Then
                  IntSampleNetCounts = WkWb.Cells(n + x, 5).Value
                  IntSampleError = (Sqr(WkWb.Cells(n + x, 3).Value * (WkWb.Cells(n +
x, 7). _
                   Value / 1000)) / (WkWb.Cells(n + x, 3).Value * WkWb.Cells(n + x, 7). _
                   Value / 1000)) * WkWb.Cells(n + x, 3).Value
                  IntSampleNetError = Sqr(IntSampleError ^ 2 + _
                   BlankError(k5) ^ 2)
               End If
            Next

'ERROR ON GROSS SAMPLE COUNTS (SampleError)
'ERROR ON NET SAMPLE COUNTS (SampleNetError)
'FINAL SAMPLE CONCENTRATION ERROR(SampleConcError)
         r = kint(1)
         rr = kint(k10)
         If knum = 1 Or knum = 0 Then
            r = 12
            rr = I
         End If
```

```
For jk = r To rr
For x = 1 To i5
   If WkWb.Cells(n + x, 5).Value <= 0 Then
      WkWb.Cells(n + x, 5).Value = 1
   End If
   If WkWb.Cells(n + x, 3).Value <= 0 Then
      WkWb.Cells(n + x, 3).Value = 1
   End If
   SampleError(x, m) = (Sqr(WkWb.Cells(n + x, 3).Value * (WkWb.Cells(n + x,
7). _
    Value / 1000)) / (WkWb.Cells(n + x, 3).Value * WkWb.Cells(n + x, 7). _
    Value / 1000)) * WkWb.Cells(n + x, 3).Value
   SampleNetError(x, m) = Sqr(SampleError(x, m) ^ 2 + BlankError(x) ^ 2)
   Error1 = (Sqr(((SampleNetError(x, m) / WkWb.Cells _
    (n + x, 5).Value) ^ 2) + ((StdNetError(x) / StdNetCounts(x)) ^ 2))) _
      * (WkWb.Cells(n + x, 5).Value / StdNetCounts(x))
   Error2 = (Sqr(((IntStdNetError / IntStdNetCounts) ^ 2) + _
    ((IntSampleNetError _
    / IntSampleNetCounts) ^ 2))) * (IntStdNetCounts / IntSampleNetCounts)
   Error3 = (Sqr((((Error1 / (WkWb.Cells(n + x, 5).Value / StdNetCounts(x))) ^
2) _
    + (Error2 / (IntStdNetCounts / IntSampleNetCounts)) ^ 2))) * _
    ((WkWb.Cells(n + x, 5).Value / StdNetCounts(x)) / (IntStdNetCounts / _
    IntSampleNetCounts))
   If WkWb.Cells(x + i5 + 3, jk).Value = "n.d." Then GoTo Out7000
   SampleConcError(x, m) = WkWb.Cells(x + i5 + 3, jk).Value * Error3
Out7000:

'WRITE ERROR DATA TO RESULTS AREA
   If WkWb.Cells(kj3 + (x - 1), 11).Value = element Then
      WkWb.Cells(kj3 + (x - 1), jk).Value = "I.S."
      WkWb.Cells(kj3 + (x - 1), jk).HorizontalAlignment = xlRight
      GoTo Out3000:
   End If
   If WkWb.Cells(x + i5 + 3, jk).Value = "n.d." Then
      WkWb.Cells(kj3 + (x - 1), jk).Value = "n.d."
      WkWb.Cells(kj3 + (x - 1), jk).HorizontalAlignment = xlRight
   Else
   WkWb.Cells(kj3 + (x - 1), jk).Value = SampleConcError(x, m)
   End If
   If WkWb.Cells(kj3 + (x - 1), jk).Value >= 1# Or WkWb.Cells(mn, 10).Value
>= 1# Then
      WkWb.Cells(kj3 + (x - 1), jk).NumberFormat = "0.0"
   End If
   If WkWb.Cells(kj3 + (x - 1), jk).Value < 1# Or WkWb.Cells(mn, 10).Value <
1# Then
```

```
                WkWb.Cells(kj3 + (x - 1), jk).NumberFormat = "0.00"
            End If
Out3000:
        Next
        Next
        m = 0
        ElseIf WkWb.Cells(n, 1).Value Like "End" Then
            Exit For
        End If
      End If
   End If
   Next

'TRANSPOSE DATA FROM X-Y TO Y-X MATRIX
kj6 = kj3 + i5 + 4
kj4 = kj6
WkWb.Cells(kj6 - 2, 11).Value = "Transposed Data"
WkWb.Cells(kj6 - 2, 11).Font.Color = RGB(255, 0, 0)
WkWb.Cells(kj6 - 2, 11).Font.Bold = "True"

'TRANSPOSE SAMPLE NAMES
For jk = kint(1) To I
WkWb.Cells(kj4, 11).Value = WkWb.Cells(1, jk).Value
WkWb.Cells(kj4, 11).HorizontalAlignment = xlRight
WkWb.Cells(kj4, 11).Font.Color = RGB(255, 0, 255)
WkWb.Cells(kj4, 11).Font.Bold = "True"
kj4 = kj4 + 1
Next
kj5 = kj3

'TRANSPOSE ELEMENT NAMES
For mn = kint(1) To kint(1) + k7 - 1
WkWb.Cells(kj6 - 1, mn).Value = WkWb.Cells(kj5, 11).Value
WkWb.Cells(kj6 - 1, mn).HorizontalAlignment = xlRight
WkWb.Cells(kj6 - 1, mn).Font.Color = RGB(0, 0, 255)
WkWb.Cells(kj6 - 1, mn).Font.Bold = "True"
kj5 = kj5 + 1
Next
kj4 = 0
kj5 = 0

'CALCULATE X (sample)-Y (element) CONCENTRATION MATRIX (cvalues)
For jk = kint(1) To I
kj4 = kj4 + 1
For mn = kj2 To kj2 + i5 - 1
kj5 = kj5 + 1
```

```
If WkWb.Cells(mn, jk).Value = "n.d." Or WkWb.Cells(mn, jk).Value = "**" Then
   NotDet(kj5, kj4) = WkWb.Cells(mn, jk).Value
Else
cvalues(kj5, kj4) = WkWb.Cells(mn, jk).Value
End If
Next
kj5 = 0
Next
kj4 = 0

'TRANSPOSE TO X (element)- Y (sample) CONCENTRATION MATRIX
For mn = kj6 To kj6 + I - kint(1)
kj4 = kj4 + 1
For jk = kint(1) To kint(1) + i5 - 1
kj5 = kj5 + 1
If NotDet(kj5, kj4) = "n.d." Or NotDet(kj5, kj4) = "**" Then
   WkWb.Cells(mn, jk).Value = NotDet(kj5, kj4)
   WkWb.Cells(mn, jk).HorizontalAlignment = xlRight
Else
   WkWb.Cells(mn, jk).Value = cvalues(kj5, kj4)
   WkWb.Cells(mn, jk).NumberFormat = "0.0"
If WkWb.Cells(mn, jk).Value < 1# Then
   WkWb.Cells(mn, jk).NumberFormat = "0.00"
End If
End If
Next
kj5 = 0
Next
Out6:
End Sub
```

	1	2	3	4	5	6	7	8
1	Blank							
2	Friday, May 26, 2000 10:51:33							
3	Blank							
4	c:\elandata\Dataset\default\Blank.268							
5	c:\elandata\Report Output\ian may 26 00							
6	S	750						
7	Cr	750						
8	Mn	750						
9	Fe	750						
10	Co	750						
11	Ni	750						
12	Cu	750						
13	Zn	750						
14	Ga	750						
15	Ge	750						
16	As	750						
17	Ag	750						
18	Cd	750						
19	Sb	750						
20	Ba	750						
21	Hg	750						
22	Tl	750						
23	Pb	750						
24	Bi	750						
25	S	34	538.141					
26	Cr	52	2896.129					
27	Mn	55	81.208					
28	Fe	57	43.566					
29	Co	59	7.448					
30	Ni	60	3.677					
31	Cu	65	5.334					
32	Zn	66	77.018					
33	Ga	71	4					
34	Ge	74	25.355					
35	As	75	45.795					
36	Ag	107	4					
37	Cd	111	5.334					
38	Sb	121	6.667					
39	Ba	138	11.205					
40	Hg	202	39.585					
41	Tl	203	2.667					
42	Pb	208	29.774					
43	Bi	209	19.316					
44								
45	**Figure 1-a. Raw data for blank analysis. Each analysis consists of two**							
46	**datasets. Rows 6-24 show element name and integrated count time for**							
47	**each element. Rows 25-43 show element name, mass analyzed, and counts**							
48	**per second data.**							

	1	2	3	4	5	6	7
259	Standard 1						
260	Friday, May 26, 2000 10:54:58						
261	Standard						
262	c:\elandata\Dataset\default\Standard 1.274						
263	c:\elandata\Report Output\ian may 26 00						
264	S	750					
265	Cr	750					
266	Mn	750					
267	Fe	750					
268	Co	750					
269	Ni	750					
270	Cu	750					
271	Zn	750					
272	Ga	750					
273	Ge	750					
274	As	750					
275	Ag	750					
276	Cd	750					
277	Sb	750					
278	Ba	750					
279	Hg	750					
280	Tl	750					
281	Pb	750					
282	Bi	750					
283	S	34	1141.151	538.141	603.011		
284	Cr	52	111659	2896.129	108762.9		
285	Mn	55	275594.4	81.208	275513.2		
286	Fe	57	356874.6	43.566	356831		
287	Co	59	126077.9	7.448	126070.5		
288	Ni	60	26506.02	3.677	26502.34		
289	Cu	65	30908.95	5.334	30903.61		
290	Zn	66	24401.58	77.018	24324.57		
291	Ga	71	6157.346	4	6153.345		
292	Ge	74	78462.32	25.355	78436.97		
293	As	75	28343.74	45.795	28297.95		
294	Ag	107	75276.55	4	75272.55		
295	Cd	111	20989.54	5.334	20984.21		
296	Sb	121	157915.1	6.667	157908.4		
297	Ba	138	718711.4	11.205	718700.2		
298	Hg	202	103.382	39.585	63.797		
299	Tl	203	9565.531	2.667	9562.864		
300	Pb	208	534499.2	29.774	534469.4		
301	Bi	209	672186.4	19.316	672167		
302							
303	**Figure 1-b. Raw data for calibration standard 1. Column 3 is**						
304	**gross cps, column 4 is blank cps and column 5 is net cps.**						
305	**Note that S and Hg are not present in this calibration standard**						
306	**and were measured in calibration standards 2 and 3.**						

	1	2	3	4	5	6	7	8
309	Standard 2							
310	Friday, May 26, 2000 11:06:12							
311	Standard							
312	c:\elandata\Dataset\default\Standard 2.275							
313	c:\elandata\Report Output\ian may 26 00							
314	S	3000						
315	Cr	3000						
316	Mn	3000						
317	Fe	3000						
318	Co	3000						
319	Ni	3000						
320	Cu	3000						
321	Zn	3000						
322	Ga	3000						
323	Ge	3000						
324	As	3000						
325	Ag	3000						
326	Cd	3000						
327	Sb	3000						
328	Ba	3000						
329	Hg	3000						
330	Tl	3000						
331	Pb	3000						
332	Bi	3000						
333	S	34	463166	538.141	462627.8	*		
334	Cr	52	4576.449	2896.129	1680.319	7.570199		
335	Mn	55	102735.9	81.208	102654.7	223.5567		
336	Fe	57	2459171	43.566	2459128	316185.5		
337	Co	59	6931.558	7.448	6924.11	24.71515		
338	Ni	60	1990.104	3.677	1986.428	37.47646		
339	Cu	65	17288366	5.334	17288361	279714.2		
340	Zn	66	3270796	77.018	3270719	67230.77		
341	Ga	71	1215.316	4	1211.316	7.874195		
342	Ge	74	344.935	25.355	319.58	1.833457		
343	As	75	11674.02	45.795	11628.23	184.9146		
344	Ag	107	67148.05	4	67144.05	338.9647		
345	Cd	111	15707.97	5.334	15702.64	299.3229		
346	Sb	121	17495.76	6.667	17489.09	52.05469		
347	Ba	138	12856.05	11.205	12844.85	8.936165		
348	Hg	202	16793.95	39.585	16754.36			
349	Tl	203	2018.132	2.667	2015.465	4.63671		
350	Pb	208	17866975	29.774	17866945	15377.48		
351	Bi	209	204671.2	19.316	204651.9	146.1436		
352								
353	**Figure 1-c. Raw data for calibration standard 2. Note the asterisk in**							
354	**column 6 to indicate S was analyzed using this standard. Column**							
355	**6 also shows preliminary concentration data for those elements**							
356	**analyzed using calibration standard 1.**							

	1	2	3	4	5	6	7	8
751	GSE							
752	Friday, May 26, 2000 11:37:03							
753	Sample							
754	c:\elandata\Dataset\default\GSE.285							
755	c:\elandata\Report Output\ian may 26 00							
756	S	750						
757	Cr	750						
758	Mn	750						
759	Fe	750						
760	Co	750						
761	Ni	750						
762	Cu	750						
763	Zn	750						
764	Ga	750						
765	Ge	750						
766	As	750						
767	Ag	750						
768	Cd	750						
769	Sb	750						
770	Ba	750						
771	Hg	750						
772	Tl	750						
773	Pb	750						
774	Bi	750						
775	S	34	836.531	538.141	298.39	224.4565		
776	Cr	52	183826	2896.129	180929.9	815.128		
777	Mn	55	321840.7	81.208	321759.4	700.713		
778	Fe	57	418627.8	43.566	418584.2	53820		
779	Co	59	159659.8	7.448	159652.4	569.8684		
780	Ni	60	34545.74	3.677	34542.07	651.6795		
781	Cu	65	37839.83	5.334	37834.5	612.1371		
782	Zn	66	11255.76	77.018	11178.75	229.783		
783	Ga	71	7263.759	4	7259.758	47.19227		
784	Ge	74	71911.27	25.355	71885.91	412.416		
785	As	75	17300.65	45.795	17254.86	274.3904		
786	Ag	107	98634.62	4	98630.62	497.919		
787	Cd	111	15274.87	5.334	15269.54	291.0672		
788	Sb	121	82050.2	6.667	82043.53	244.1951		
789	Ba	138	448724.6	11.205	448713.4	312.1701		
790	Hg	202	205.127	39.585	165.542	1.185473		
791	Tl	203	10589.08	2.667	10586.42	24.35475		
792	Pb	208	551977.8	29.774	551948.1	475.0433		
793	Bi	209	692030.5	19.316	692011.2	494.1709		
794	End							
795								
796	**Figure 1-d. Raw data for lasr sample analyzed. Note the "End" in column 1.**							
797	**This is also the drift sample GSE.**							

	23	24	25	26	27	28	29	30	31	32
1		GSE	light band	dark band	light band	light band	dark band	light band	dark band	GSE
2	S	436.9	84908.7	147199.2	224807.9	83280.1	9624.2	142799.0	173416.7	183.6
3	Cr	463.2	0.3	0.7	1.1	0.5	-0.45	0.7	0.6	815.1
4	Mn	573.0	0.7	3.3	1.9	0.8	0.2	1.2	2.0	700.7
5	Fe	43845.9	50.6	548.7	127.5	31.7	12.5	73.8	458.2	53820.0
6	Co	426.3	0.04	0.04	0.3	0.08	-0.01	0.09	0.04	569.9
7	Ni	475.5	1.0	0.05	0.5	0.3	0.02	0.3	0.2	651.7
8	Cu	471.4	4.2	10.7	12.3	5.0	1.0	15.1	23.6	612.1
9	Zn	455.4	247924.2	429318.3	639095.7	233992.3	20916.4	405016.6	495579.7	229.8
10	Ga	36.6	0.05	1.0	0.2	0.06	0.03	0.2	0.1	47.2
11	Ge	426.9	4.4	9.7	7.5	2.4	0.5	5.6	4.4	412.4
12	As	403.8	38.6	110.0	86.2	30.2	5.0	53.0	54.2	274.4
13	Ag	355.0	1.2	2.7	12.9	2.8	0.2	3.6	91.3	497.9
14	Cd	371.5	4346.2	4833.3	10381.8	4040.6	316.6	5802.9	6504.6	291.1
15	Sb	442.2	1.7	1.0	5.9	1.9	0.2	4.1	32.6	244.2
16	Ba	471.9	0.5	3.5	1.5	1.0	0.1	0.8	0.9	312.2
17	Hg	0.2	5.2	3.3	3.9	3.1	2.2	3.2	1.9	0.3
18	Tl	20.7	7.0	25.9	18.2	5.9	0.9	10.8	21.3	24.4
19	Pb	440.3	410.3	983.4	1057.9	379.6	57.9	732.9	746.4	475.0
20	Bi	466.0	0.07	0.04	0.06	0.05	0.01	0.04	0.04	494.2
21										
22	**Results Block 1. Raw concentration data from data file *.xls., excepting those elements measured using calibration standard 2 (S)**									
23	**and calibration standard 3 (Hg) that have been abaltion efficiency corrected to calibration standard 1.**									

	23	24	25	26	27	28	29	30	31	32
24	**Corrected Data**									
25		**GSE**	**light band**	**dark band**	**light band**	**light band**	**dark band**	**light band**	**dark band**	**GSE**
26	**S**	527.7	222610.9	222863.7	228643.6	231341.1	299082.6	229174.1	227452.5	439.5
27	**Cr**	559.4	0.7	1.1	1.2	1.5	-13.85	1.2	0.8	1951.1
28	**Mn**	692.0	1.7	5.0	2.0	2.1	5.9	2.0	2.6	1677.2
29	**Fe**	52952.3	132.7	830.7	129.7	88.1	388.6	118.4	600.9	128821.5
30	**Co**	514.8	0.1	0.06	0.3	0.2	-0.35	0.1	0.06	1364.0
31	**Ni**	574.3	2.6	0.08	0.5	0.8	0.6	0.5	0.3	1559.8
32	**Cu**	569.3	11.1	16.2	12.5	13.9	30.1	24.2	31.0	1465.2
33	**Zn**	550.0	650000.0	650000.0	650000.0	650000.0	650000.0	650000.0	650000.0	550.0
34	**Ga**	44.2	0.1	1.5	0.2	0.2	0.8	0.3	0.2	113.0
35	**Ge**	515.6	11.4	14.7	7.6	6.5	14.8	8.9	5.8	987.1
36	**As**	487.7	101.3	166.6	87.7	83.8	155.4	85.1	71.1	656.8
37	**Ag**	428.7	3.1	4.1	13.1	7.9	5.2	5.8	119.7	1191.8
38	**Cd**	448.6	11394.7	7317.8	10559.0	11224.2	9840.0	9312.9	8531.4	696.7
39	**Sb**	534.1	4.3	1.5	6.0	5.2	6.0	6.5	42.7	584.5
40	**Ba**	569.9	1.2	5.3	1.5	2.7	4.2	1.3	1.2	747.2
41	**Hg**	0.2	13.6	5.1	4.0	8.7	67.5	5.1	2.4	0.8
42	**Tl**	25.0	18.3	39.1	18.5	16.5	28.1	17.3	27.9	58.3
43	**Pb**	531.8	1075.6	1488.8	1076.0	1054.4	1799.6	1176.3	978.9	1137.0
44	**Bi**	562.8	0.2	0.06	0.06	0.1	0.3	0.06	0.05	1182.8
45										
46	**Results Biock 2. Data corrected for ablation efficiency differences. Zn used as an internal standard. Drift sample GSE corrected**									
47	**to 550 ppm Zn, samples corrected to 650000 ppm Zn.**									

	23	24	25	26	27	28	29	30	31	32	33
49		Drift Corrected Data									
50	D.L.		GSE	light band	dark band	light band	light band	dark band	light band	dark band	GSE
51	52.19	S	527.7	227360.0	232579.9	243929.0	252432.1	333961.2	262011.1	266404.8	527.7
52	0.47	Cr	559.42	0.57	0.69	0.60	0.67	-5.42	0.40	0.25	559.42
53	0.04	Mn	692.02	1.48	3.69	1.29	1.23	3.11	0.95	1.17	692.02
54	1.61	Fe	52952.3	112.6	611.6	84.4	51.3	205.0	57.1	266.7	52952.3
55	0.03	Co	514.80	0.09	0.04	0.21	0.13	-0.17	0.06	0.02	514.80
56	0.20	Ni	574.27	2.16	0.05	0.30	0.44	0.28	0.23	0.12	574.27
57	0.10	Cu	569.32	9.24	11.65	7.86	7.75	15.19	11.11	13.03	569.32
58	0.77	Zn	550.00	650000.00	650000.00	650000.00	650000.00	650000.00	650000.00	650000.00	550.00
59	0.05	Ga	44.22	0.10	1.09	0.14	0.09	0.41	0.12	0.07	44.22
60	0.11	Ge	515.61	10.25	11.96	5.65	4.49	9.42	5.28	3.23	515.61
61	0.12	As	484.72	97.05	153.27	77.62	71.41	127.73	67.51	54.53	484.72
62	0.05	Ag	428.72	2.55	2.87	7.88	4.17	2.44	2.50	46.81	428.72
63	0.15	Cd	448.65	10658.17	6429.18	8745.76	8793.38	7313.03	6583.21	5749.84	448.65
64	0.02	Sb	534.10	4.28	1.51	5.75	4.94	5.63	6.09	39.49	534.10
65	0.01	Ba	569.86	1.20	4.87	1.32	2.34	3.50	1.08	0.95	569.86
66	0.10	Hg	0.20	9.91	2.89	1.88	3.45	23.41	1.56	0.67	0.20
67	0.02	Tl	24.98	15.66	29.35	12.31	9.88	15.32	8.64	12.86	24.98
68	0.02	Pb	531.75	941.60	1159.01	754.09	671.97	1051.53	634.53	490.45	531.75
69	0.01	Bl	562.80	0.16	0.05	0.04	0.08	0.15	0.03	0.03	562.80
70											
71	**Results Block 3. Data corrected for drift as monitored by drift sample GSE (grayed). D.L. column is detection limits**										
72	**based upon repeated blank analysis.**										

	58	59	60	61	62	63	64	65	66	67
73	**Counting Errors (ppm)**									
74		**GSE**	**light band**	**dark band**	**light band**	**light band**	**dark band**	**light band**	**dark band**	**GSE**
75	**S**	0.02	9.51	9.52	9.77	9.88	12.77	9.79	9.71	0.02
76	**Cr**	6.00	0.01	0.01	0.01	0.02	-0.15	0.01	0.01	20.91
77	**Mn**	5.05	0.01	0.04	0.01	0.02	0.04	0.01	0.02	12.25
78	**Fe**	386.17	0.97	6.06	0.95	0.64	2.83	0.86	4.38	939.47
79	**Co**	4.18	0.00	0.00	0.00	0.00	0.00	0.00	0.00	11.09
80	**Ni**	5.60	0.03	0.00	0.00	0.01	0.01	0.01	0.00	15.21
81	**Cu**	5.11	0.10	0.15	0.11	0.12	0.27	0.22	0.28	13.15
82	**Zn**	2.18	2576.19	2576.19	2576.19	2576.19	2576.19	2576.19	2576.19	2.18
83	**Ga**	0.58	0.00	0.02	0.00	0.00	0.01	0.00	0.00	1.47
84	**Ge**	3.16	0.07	0.09	0.05	0.04	0.09	0.05	0.04	6.05
85	**As**	2.37	0.49	0.81	0.43	0.41	0.76	0.41	0.35	3.19
86	**Ag**	3.72	0.03	0.04	0.11	0.07	0.04	0.05	1.04	10.33
87	**Cd**	2.71	68.91	44.25	63.86	67.88	59.51	56.32	51.59	4.21
88	**Sb**	1.81	0.01	0.01	0.02	0.02	0.02	0.02	0.14	1.98
89	**Ba**	2.20	0.00	0.02	0.01	0.01	0.02	0.01	0.00	2.88
90	**Hg**	0.00	0.03	0.01	0.01	0.02	0.13	0.01	0.00	0.00
91	**Tl**	0.27	0.20	0.42	0.20	0.18	0.30	0.18	0.30	0.62
92	**Pb**	3.40	6.87	9.51	6.87	6.73	11.49	7.51	6.25	7.26
93	**Bi**	3.57	0.00	0.00	0.00	0.00	0.00	0.00	0.00	7.51
94										
95	**Results Biock 4. Measurement errors based on counting statistics, in absolute values (ppm). Precision and accuracy of**									
96	**calibration standards are not included in the error analysis.**									

	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
90	**Transposad Data**																			
91		**S**	**Cr**	**Mn**	**Fe**	**Co**	**Ni**	**Cu**	**Zn**	**Ga**	**Ge**	**As**	**Ag**	**Cd**	**Sb**	**Ba**	**Hg**	**Tl**	**Pb**	**Bi**
92	**GSE**	527.7	559.4	692.0	52952.3	514.8	574.3	569.3	550.0	44.2	515.6	487.7	428.7	448.6	534.1	569.9	0.20	25.0	531.8	562.8
93	**light band**	227360.0	0.57	1.5	112.6	0.09	2.2	9.2	650000.0	0.10	10.2	97.0	2.6	10658.2	4.3	1.2	9.9	15.7	941.6	0.16
94	**dark band**	232579.9	0.69	3.7	611.6	0.04	0.05	11.7	650000.0	1.1	12.0	153.3	2.9	6429.2	1.5	4.9	2.9	29.4	1159.0	0.05
95	**light band**	243929.0	0.60	1.3	84.4	0.21	0.30	7.9	650000.0	0.14	5.7	77.6	7.9	8745.8	5.8	1.3	1.9	12.3	754.1	0.04
96	**light band**	252432.1	0.67	1.2	51.3	0.13	0.44	7.8	650000.0	0.09	4.5	71.4	4.2	8793.4	4.9	2.3	3.5	9.9	672.0	0.08
97	**dark band**	333961.2	-5.42	3.1	205.0	-0.17	0.28	15.2	650000.0	0.41	9.4	127.7	2.4	7313.0	5.6	3.5	23.4	15.3	1051.5	0.15
98	**light band**	262011.1	0.40	0.95	57.1	0.06	0.23	11.1	650000.0	0.12	5.3	67.5	2.5	6583.2	8.1	1.1	1.6	8.6	634.5	0.03
99	**dark band**	266404.8	0.25	1.2	266.7	0.02	0.12	13.0	650000.0	0.07	3.2	54.5	46.8	5749.8	39.5	0.95	0.67	12.9	490.4	0.03
100	**GSE**	527.7	559.4	692.0	52952.3	514.8	574.3	569.3	550.0	44.2	515.6	487.7	428.7	448.6	534.1	569.9	0.20	25.0	531.8	562.8
101																				
102	**Results Block 5. Data are transposed to facilitate importing into other database programs.**																			

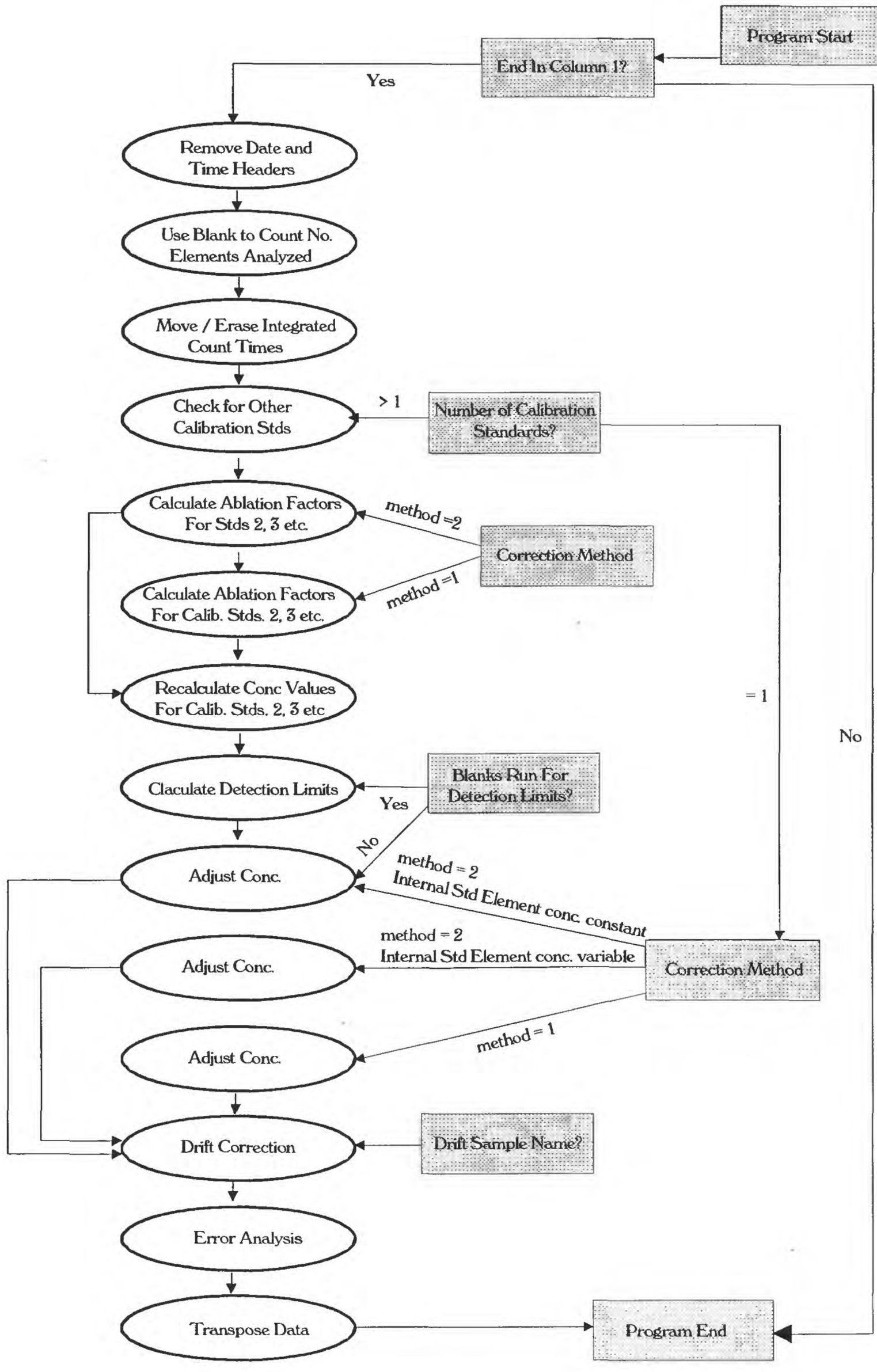

Figure 2. Flow diagram for program QUANTLASER