TECHNICAL NOTE

Richard W. Waggoner, Jr.,[1] *Ph.D.*

A Simple Procedure That Allows Software to Determine if a GC/MS Blank Injection Is Contaminated

REFERENCE: Waggoner, R. W., Jr., **"A Simple Procedure That Allows Software to Determine if a GC/MS Blank Injection Is Contaminated."** *Journal of Forensic Sciences,* JFSCA, Vol. 41, No. 4, July 1996, pp. 681–684.

ABSTRACT: One problem commonly encountered by operators of GC/MS systems is the injection of solvent blanks which are contaminated. The sources of this contamination are many and are often of an undetermined origin. The contaminated blank is an especially troublesome problem when the operator is running a sequence of samples using an autoinjector. The injection of a contaminated blank during a sequence of samples could jeopardize the data collected for any ensuing samples. A simple macro program, described here, will allow the Hewlett-Packard MS ChemStation software to decide if a blank is contaminated. If contamination is present then the software will direct the autoinjector to make the necessary reinjections.

KEYWORDS: forensic science, gas chromatography/mass spectrometry, autosamplers, macro programs

The development and routine use of autosamplers have increased productivity in all scientific fields. One common use of autosamplers is the analysis of multiple samples by gas chromatography. In a gas chromatographic analysis the sample is introduced into a column that has been exposed to previous samples. In order to prevent contamination of the sample being examined, a blank injection is made before introducing the current sample into the gas chromatograph. Under normal circumstances, if this blank injection shows no chromatographic peaks then it is safe to proceed with the sample injection. However, if the blank injection shows significant peaks then the gas chromatograph (injector, column, or detector) is considered contaminated. Usually repeating the blank injection corrects this contamination problem. If the contamination is still present after several blank injections then a more serious problem may be present.

The Hewlett-Packard HP G1034C MS ChemStation software (1) allows the operator to perform intelligent sequencing by setting up simple sequences and using various keywords. By using the intelligent sequencing functions of the MS ChemStation software and incorporating a simple macro program, the operator can direct the MS ChemStation software to examine the chromatograph obtained from the blank injection and make decisions regarding future injections.

Instruments

The GC/MS system is a Hewlett-Packard 5890 Series II gas chromatograph interfaced to a 5970B Mass Selective Detector (MSD). The autosampler is a Hewlett-Packard 7673 injector with a Hewlett-Packard 7673 Controller. The instrument is controlled by a Digital DECpc LPx$^+$ 466d2 with a 540 MB hard disk drive and 16 MB RAM. In addition to the Hewlett-Packard HP G1034C MS ChemStation Version C.02.03 software, the computer has MS-DOS Version 6.21 and Microsoft Windows with the Work Group Add-On Version 3.11 installed.

Methods

Two simple modifications need to be made in the HP G1034C MS ChemStation software before entering the macro program. The sequences of code that need to be modified are present in a file named TOPSEQ.MAC that is present in the MSEXE directory. Before making these modifications it is recommended that the TOPSEQ.MAC file be backed up (copy TOPSEQ.MAC to TOPSEQ.OLD).

The first modification is made by loading the TOPSEQ.MAC file into the Microsoft Windows Notepad and searching for the comment line titled: !Test for Keyword "RETURN." This comment line begins the submacro program shown in Fig. 1. Present in the code are three endif lines stacked consecutively (see the arrow in the figure). Insert the word "return" between the first and second endif lines as shown below:

```
    endif
    return
  endif
endif
```

then save the corrected TOPSEQ file as TOPSEQ.MAC in the MSEXE directory.

The second modification does not need to be made unless the blank injection vial is to be incremented in the sequence and the sequence may increment the blank injection vial past position 100 on the autoinjector tray. This modification is performed in the submacro program that begins with the comment line, !Test for Keyword "MOD_LINE" (Fig. 2). After the evaluate a$ command

[1] Chemist, North Carolina State Bureau of Investigation, Crime Laboratory, Raleigh, NC.

Received for publication 10 Aug. 1995; revised manuscript received 20 Nov. 1995; accepted for publication 21 Dec. 1995.

```
    !Test for keyword "RETURN"
     If (1 =INSTR(_MethFile$,"RETURN.M"))
                              ! Restore old parameters
      POP
i,RE_INJTIMES,BACK,MAX_BACK,ONFLAGLABEL$<AUTOJUMPLABEL$,REINJECTLABEL$

      if (LEN(_DataName$) = 0)     ! Parse jumpback number
        I = 0
      else
        I = val(_DataName$)
      endif
      if (I < 0)                ! Test for jumpback
        BACK = BACK + 1
        if (BACK > MAX_BACK)    ! Too many times
          SeqLogWrite "LoopBack limit exceeded!"
          if (LEN(LOOPBACKLABEL$) > 0)   ! Jump to label
            ExecuteSubSequence LOOPBACKLABEL$
          else
            BACK = 0          ! Move ahead
            I = 0
          endif
  →     endif
      endif
      SeqLogWrite "Returning to line "+val$(i+I+1), SEQ_TRACE
      SampleLogTable Position,val$(i+I+1) ! Go back to where we left off
      NESTINGLEVEL = NESTINGLEVEL - 1
      if (NESTINGLEVEL > 0)
        IN_SUBSEQ = 2
      else
        IN_SUBSEQ = 0
      endif
      return
   endif
```

FIG. 1—*HP G1034C MS ChemStation submacro titled Test for Keyword "RETURN."*

(see the arrow in the figure) insert the four following lines of code listed below:

```
if_AlsBottle > 100
  GENERATE_ERROR 99, "Cannot increment vial above 100."
  SeqLogWrite "Cannot increment vial above 100."
endif
```

then save the corrected TOPSEQ file as TOPSEQ.MAC in the MSEXE directory.

The macro program shown in Fig. 3 should be entered into the deuser.mac file for the method that directs the autosampler to make the blank injections (called BLANK in the sequence example that follows). The operator must first determine the maximum number of counts (abundance) that is acceptable for a blank injection to be considered uncontaminated in their specific analysis. The scalar variable Yhigh, which is arbitrarily tested against 10,000 in the example shown, determines if the variable contam is set to a value of one or a value of zero. If any peaks present in the chromatogram obtained from utilizing the method for the blank injections have an abundance of 10,000 counts or greater then contam is set to one; otherwise, contam is set to zero.

Figure 4 shows the printout of a typical sequence that can be utilized where the MS ChemStation software will decide if a blank injection is acceptable. This sequence can be input by selecting either Edit Sample Log Table or Edit Sample Log Table with Barcode from the Sequence menu. For each line in the Sample Log Table select the appropriate sample type. If the sample type is Keyword then the appropriate keyword must be used. If the sample type is blank, sample, or calibration then Vial, Data File, Method, and Sample Name columns should have the necessary data supplied.

In the Fig. 4 example, line 2 injects a solvent blank prior to the analysis of the first sample. Lines 3–7 specify that five separate samples are to be injected. Line one instructs the sequence to jump to the label Run_Blank (line 9) after each sample injection. The LoopBack keyword instructs the sequence to allow the blank to be injected three times (the initial injection and two reinjections). If a fourth injection is attempted, the sequence will jump to the Label Quit (line 17) which causes the sequence to end, and a "loopback limit exceeded" message will be generated. The blank injection is described in line 11.

After injecting the initial solvent blank the sequence shown in Fig. 4 will, jump to the label Run_Blank, and inject a second blank

```
  !Test for keyword "MOD_LINE"
   if (1 = INSTR(_MethFile$,"MOD_LINE.M"))
     local oldline
     i = INSTR(_DataName$,",")
     I = LEN(_DataName$)
     a$ = _DataName$[i+1:I]
     oldline = _SeqLine                    ! save return point
     if (i = 1)                       ! no line number given
       SampleLogTable Position,val$(LAST_INJ_LINE)
     else
       SampleLogTable Position,_DataName$[1:i-1]
     endif
     SampleLogTable Advance
     SampleLogTable Getentry              ! get the information
     varlock off
→    evaluate a$                    ! evaluate expression
     varlock on
     SeqLogWrite "Modifying line: "+val$(_SeqLine)+" "+a$
     SampleLogTable PutEntry               ! Put in changes
     SampleLogtable Postion,val$(oldline+1)! return to old point
     return
   endif

   return
```

FIG. 2—*HP G1034C MS ChemStation submacro titled Test for Keyword "MOD_LINE."*

placed in position 10 on the autoinjector tray. After collecting the data from this blank injection, the sequence will instruct the MS ChemStation to compare the largest peak in the blank chromatogram to a value of 10,000 counts. If the blank chromatogram has a peak that is larger than 10,000 counts then the contam variable is set to a value of one, and the If_Flag keyword in line 12 will instruct the sequence to jump to the Rep_Blank label to repeat the blank injection. After repeating the blank injection it is tested again as previously described. If the blank injection is acceptable then the sequence proceeds to inject the first sample (Sample 1) and the sequence continues. However, if the blank injection is unacceptable then the blank is injected again. After three contaminated blank injections the sequence jumps to the label Quit and the sequence ceases. If an uncontaminated blank has been obtained prior to running the sequence then the operator may omit line 2 from the sequence (if this line is omitted then note that line 15 should modify line 10 instead of line 11 as all the line numbers are shifted one place except for line 1).

```
! File   : deuser.mac
! Macros : CustomAnalysis

NAME CustomAnalysis
   local contam
   file _datapath$+_datafile$
   tic ,r0
   getscalars ,r0
   if Yhigh > 10000 then
     contam = 1
   else
     contam = 0
   endif
   topdot contam
   REMOVE CustomAnalysis
RETURN
```

FIG. 3—*Macro program that sets the value for the variable contam.*

```
Sequence Name: C:\HPCHEM\1\SEQUENCE\EXAMPLE.S
      Comment: Example sequence (Figure 4.)
     Operator: Waggoner
    Data Path: C:\HPCHEM\1\Data\
  Pre-Seq Cmd:
 Post-Seq Cmd:

Method Sections To Run        On A Barcode Mismatch
  (X) Full Method               (X) Inject Anyway
  ( ) Reprocessing Only         ( ) Don't Inject

Line Type        Vial  DataFile   Method     Sample Name
-------------------------------------------------------------------------------

  1) Auto_Sub -  Run_Blank
  2) Blank          1   INITBLNK  BLANK      Initial Solvent Blank
  3) Sample         2   SAMPLE1   METHOD1    Sample 1
  4) Sample         3   SAMPLE2   METHOD2    Sample 2
  5) Sample         4   SAMPLE3   METHOD3    Sample 3
  6) Sample         5   SAMPLE4   METHOD4    Sample 4
  7) Sample         6   SAMPLE5   METHOD5    Sample 5
  8) End
  9) Label      -  Run_Blank
 10) LoopBack   -  2,Quit
 11) Blank         10   ----------- BLANK    Solvent Blank
 12) If_Flag    -  Contam,Rep_Blank
 13) Return
 14) Label      -  Rep_Blank
 15) Mod_Line   -  11,_AlsBottle=_AlsBottle
 16) Return     -  -2
 17) Label      -  Quit
 18) End
```

FIG. 4—*Printout of a typical example sequence from the HP G1034C MS ChemStation.*

If a sequence simulation is attempted with the sequence shown in Fig. 4 then a "Mismatched Sub-Sequence LABEL/RETURN pairs found! Edit Sample Log Table?" message will be present. This error seems to cause no problems in the execution of the sequence; therefore, it appears that this message can be ignored.

If the operator wishes to set up the sequence to inject a different blank vial(s), should the initial blank vial be contaminated, then line 15 could be modified by changing the _AlsBottle =_AlsBottle command to _AlsBottle =_AlsBottle+1. This change would cause the sequence to increment by one space the sample tray location where the sequence looks for the blank vial. If this change is enacted then the blank vials should be placed on the tray following the sample vials, so that if the blank vial number is incremented too far then the sequence will pause to ask for a vial, rather than possibly injecting a sample as a blank. If the last blank vial is in position 100 on the sample tray then the second correction to the TOPSEQ.MAC file described earlier must be made; otherwise, the autoinjector will continue to inject vials beginning again at position 1 on the autoinjector tray.

Conclusions

A simple procedure has been outlined where the operator of a Hewlett-Packard MS ChemStation can direct the Hewlett-Packard HP G1034C MS ChemStation software to examine the chromatogram obtained from a blank injection. After this examination the software, without operator intervention, can instruct the autoinjector to reinject a blank or proceed with the injection of the next sample.

Acknowledgments

The author wishes to thank Mr. Bill Price of Hewlett-Packard Company for his valuable assistance in the modification of the macro programs.

All figures were provided courtesy of Hewlett-Packard Company.

Reference

(1) HP G1034C MS ChemStation (DOS Series) Version C.02.03, Hewlett-Packard Company, 1994.

Address request for reprints or additional information to
Richard W. Waggoner, Jr.
North Carolina State Bureau of Investigation
Crime Laboratory
3320 Garner Road, P. O. Box 29500
Raleigh, NC 27626-0500