PharmaSUG 2019 - Paper ST-149

Application of R® Functions in SAS® to Estimate Dose Limiting Toxicity Rates for Early Oncology Dose Finding

Huei-Ling Chen, Zhen Zeng, Merck & Co., Inc., Kenilworth, NJ, USA

ABSTRACT

Pool-adjacent-violators algorithm (PAVA) is a solution to isotonic regression used to estimate dose limiting toxicity (DLT) rates in early oncology trials. In the current version of SAS, there is no ready-to-use statistical procedure to implement PAVA. However, there are multiple R packages available for PAVA estimation. In most pharmaceutical companies, SAS is the mainstream working environment for statistical analysis and reporting. It would be very helpful to exploit the bridge between SAS and R to take advantage of both worlds: R's capability in versatile statistical analysis tools, and SAS's well validated existing statistical procedures and mature reporting system. This paper presents a SAS macro to accomplish this collaborative task by embedding the available R PAVA function into a SAS program. This paper provides details on three important features used to streamline the process of integration. The first feature is how to prepare a SAS environment which enables SAS code to communicate with R. The second feature, how to import and export data between SAS and R. Key PROC IML syntaxes will be provided for demonstration. The Third feature, how to make R package recognize a SAS macro parameter value so that the R package can easily become a nested sub-macro inside a SAS macro. This macro has been implemented in real-life to support the reporting of DLT summary statistics.

KEYWORDS

PROC IML, R, PAVA, DOSE FINDING, DOSE LIMITING TOXICITY, ONCOLOGY

INTRODUCTION

Finding the maximum tolerable dose (MTD) based on dose limiting toxicity (DLT) is critical to early oncology development. Current oncology dose finding methods include 3+3 design, modified toxicity probability interval (mTPI), continual reassessment method (CRM), and Bayesian optimal interval design (BOIN). MTD is usually determined from an estimated dose level–DLT rate curve by isotonic regression using a pool-adjacent-violators algorithm (PAVA) on the data collected from these studies. Although PAVA is not a ready-to-use function available in SAS, it has already been implemented in multiple R packages. Furthermore, with the recent trend in immuno-oncology to develop combinations of established therapies with new agents to either overcome drug resistance or achieve synergic effects, as well as the need of exploring safety/efficacy profile of the new agent alone, many trials were designed to have two arms of monotherapy of the new agent and its combination with an established immuno-oncology therapy in parallel. In these trials, beside the monotone relationship between dose level and toxicity rate, the combination arm should not be less toxic than the monotherapy arm with same dose level of the new agent, which imposes another dimension of constraint. Ignoring such constraint may result in selecting MTD that is contradictory to common sense due to the relatively small size of phase I studies. Two-dimensional PAVA (2D-PAVA) can be used in this case to fit two monotonic dose level–toxicity rate curves, with the one for combination therapy arm always not lower than the monotherapy curve. MTDs of both arms are then determined from two curves. Thus, the probability of selecting the true MTD may be improved and the selection of contradictory MTDs between the two arms can be avoided. Therefore, we implemented 2D-PAVA in an R function. A SAS macro was then developed to call PAVA and 2D-PAVA functions in R from SAS, so that these existing tools in R can be readily used while maintaining the standard analysis environment and enjoying other amenities of SAS reporting.

PREPARING A SAS ENVIRONMENT TO WORK WITH R

- Install R in PC
- For the first-time user, open R and install a R package 'Iso'. This is for PAVA estimate.

```
install.packages("Iso")
```

- Set SAS to run R (reference: https://communities.sas.com/t5/General-SAS-Programming/Run-R-code-inside-SAS-easily/td-p/210116)
  1) Find SAS CFG file "sasv9.cfg".

     e.g. C:\Program Files\SAS94 Enterprise Guide 7.1\x86\SASFoundation\9.4\nls\en
  2) Locate your R location.

     e.g. C:\Program Files\R\R-3.4.3
  3) Add 3 lines of command at the end of "sasv9.cfg" file. Configure "sasv9.cfg" location, according to your SAS version. The R location in "R_HOME" also should be consistent with your R version and location.

```
-RLANG
-config "C:\Program Files\SAS94 Enterprise Guide
7.1\x86\SASFoundation\9.4\nls\en\sasv9.cfg"
-SET R_HOME "C:\Program Files\R\R-3.4.3"
```

  4) SAS is now ready to run R.

IMPORT AND EXPORT DATA BETWEEN SAS AND R

Here we introduce key syntax in PROC IML statement to work with R statistical programming language.

- SUBMIT/ENDSUBMIT
- Export SAS data to R
- Import R data to SAS

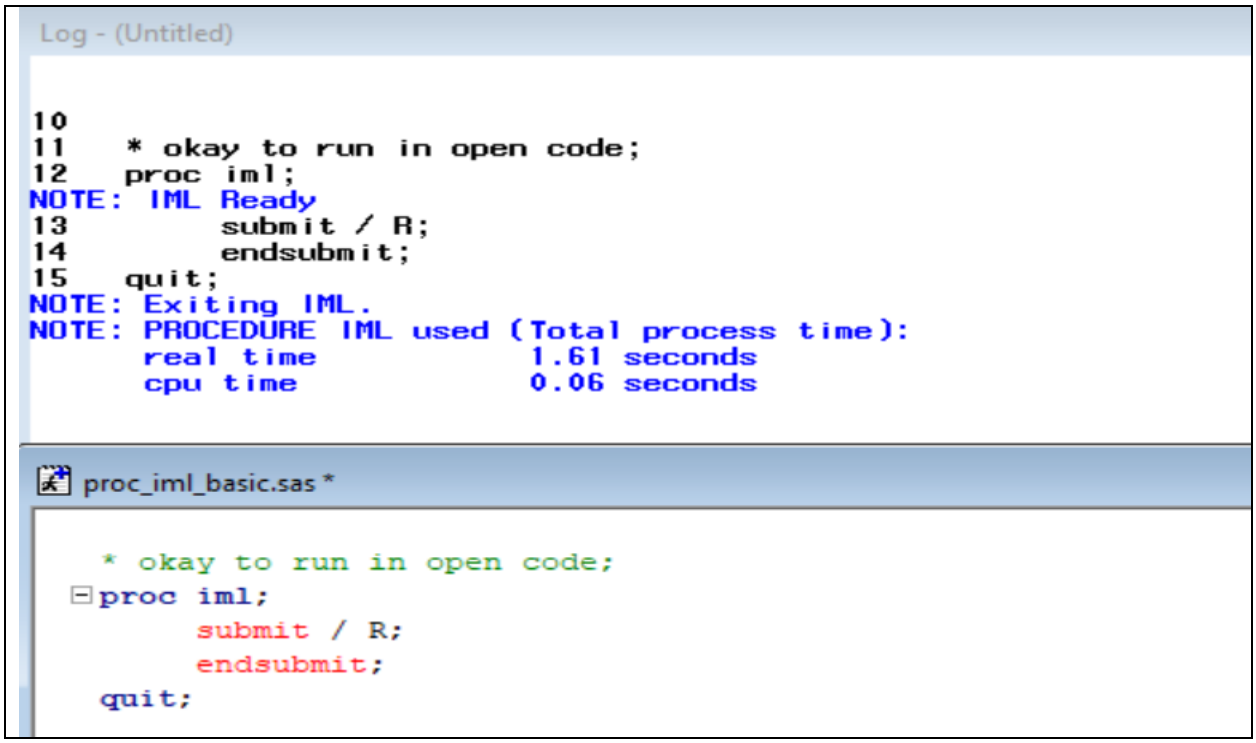
KEY SYNTAX 1: SUBMIT/ENDSUBMIT

SUBMIT/ENDSUBMIT statement provides an interface to submit R codes to R statistical programming language within the PROC IML program. To connect to R, specify the "R" option in the SUBMIT statement.

```
proc iml;
        submit / R;
        < R code>
        endsubmit;
quit;
```

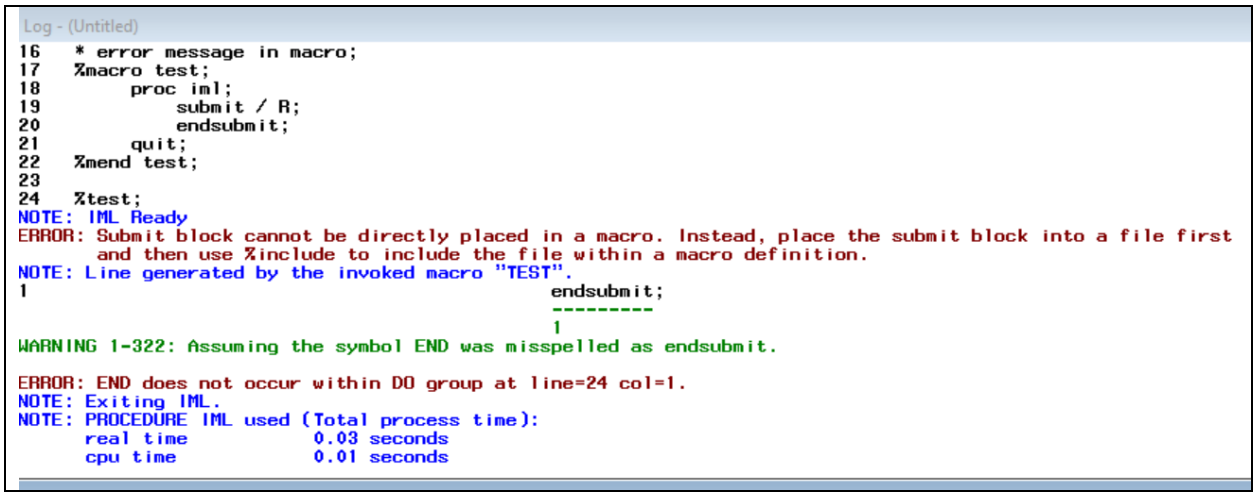
The syntax is simple and straightforward in open code, but not so in SAS macro. Reason is the SUBMIT/ENDSUBMIT statements are not allowed in SAS macro.

The SUBMIT/ENDSUBMIT statements has clean log message.



Display 1. PROC IML SUBMIT in open code with clean log message

Yet in a macro setting, the SUBMIT/ENDSUBMIT statements creates error log message.



Display 2. PROC IML SUBMIT in macro with error message

When writing a macro, we suggest saving the SUBMIT/ENDSUBMIT codes into a SAS code and use %include statement to include the file.

In this paper, we added SUBMIT and ENDSUBMIT statement before and after the R syntax, respectively, and saved the mixed SAS and R code to a SAS file and used %include statement to include the file. Below are snapshots of PAVA and 2D-PAVA R functions masked in SAS file.

```
submit / R;

      #_______Start of R code_________________
      #Study:          Oncology Standards
      #Program:        r0code0pava.sas
      #Function:       Produce the Summary of Dose Limiting Toxicity - PAVA (mono therapy)
      #Comments:       sub-macro of sas code or0sa0dlt0summary.sas
      #Author:         Zhen Zeng
      #Version Date:   01DEC2017

      (n <- TOTAL_IN_R$DENOM);
      (x <- TOTAL_IN_R$COUNT);
      (p <- P_IN_R$P);

      #PAVA
      library(Iso)

      PAVA <- function(n = c(7, 18, 6, 3, 0, 0), x = c(1, 4, 1, 2, 0, 0), pT = 0.3) {

       . . .
      }

      result <- PAVA(n, x, p);

      #_______End of R code_________________
endsubmit;
```

Display 3: R code r0code0pava.sas (PAVA function - Mono therapy)

```
submit / R;

      #_______Start of R code_________________
      #Study:          Oncology Standards
      #Program:        r0code0pava2d.sas
      #Function:       Produce the Summary of Dose Limiting Toxicity - 2D-PAVA (mono vs combo therapy)
      #Comments:       sub-macro of sas code or0sa0dlt0summary.sas
      #Author:         Zhen Zeng
      #Version Date:   06MAR2018

      (n_m <- TOTAL_IN_R_MONO$DENOM);
      (x_m <- TOTAL_IN_R_MONO$COUNT);
      (n_c <- TOTAL_IN_R_COMBO$DENOM);
      (x_c <- TOTAL_IN_R_COMBO$COUNT);
      (pT <- P_IN_R$P);

      #2D-PAVA
      PAVA.2D <- function(n_m, x_m, n_c, x_c, pT) {

       . . .

      }

      result <- PAVA.2D(n_m, x_m, n_c, x_c, pT);

      #_______End of R code_________________
endsubmit;
```

Display 4: R code r0code0pava2d.sas (2D-PAVA function - Mono therapy vs. Combo therapy)

KEY SYNTAX 2: EXPORT SAS DATA TO R

PROC IML can export SAS dataset to R data frame.

```
proc iml;
     run ExportDataSetToR("WORK.SASdatasetName", "RdatasetName");
quit;
```

KEY SYNTAX 3: IMPORT R DATA TO SAS

PROC IML can export R data frame to SAS dataset.

```
proc iml;
     run ImportDataSetFromR("WORK.SASdatasetName ", "RdatasetName" );
quit;
```

PASS A SAS MACRO PARAMETER VALUE TO R CODE

R software cannot recognize SAS macro parameter. To make R code a nested sub-macro inside a SAS macro, a walk-around strategy adopted here is a two-steps process. First, the macro parameters were saved into a SAS dataset. Next PROC IML was used to convert that SAS dataset with macro parameter values to a R data. R software can then read the macro parameter values from the data and do the analysis accordingly. Below is example code to pass SAS macro parameters to R.

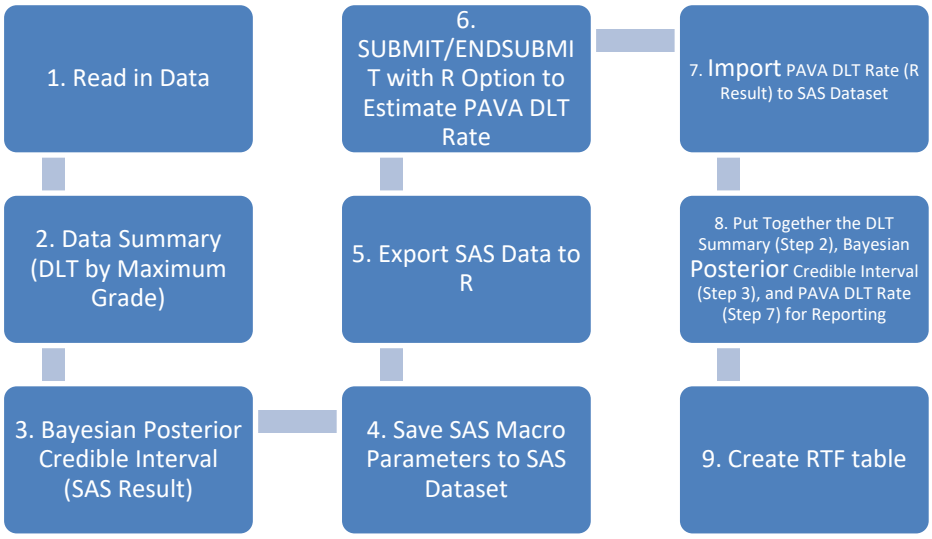
First, save the macro parameters into a SAS dataset.

```
data temp;
       variablename1=&macroparameter1;
       variablename2=&macroparameter2;
       …
   run;
```

Next use PROC IML to convert the SAS dataset with macro parameter values to a R data.

```
proc iml;
     run ExportDataSetToR("WORK.temp", "RdatasetName");
quit;
```

SAS MACRO PROGRAM FLOW TO SUPPORT THE REPORTING OF DLT SUMMARY STATISTICS.



Display 5: SAS macro %or0sa0dlt0summary program flow

STEP 1: READ IN DATA

- ADSL: ADaM Subject Level Analysis Dataset

USUBJID	DLTFL	TRT01AN	TRT01A
9999-001_000100001	Y	3	MK-9999 7 mg
9999-001_000100002	Y	3	MK-9999 7 mg
9999-001_000100003	Y	4	MK-9999 21 mg
9999-001_000100004	Y	8	MK-9999 7 mg + AAA 200 mg
9999-001_000100005	Y	4	MK-9999 21 mg
9999-001_000100006	Y	5	MK-9999 70 mg
9999-001_000100007	Y	9	MK-9999 21 mg + AAA 200 mg
9999-001_000100008	Y	6	MK-9999 210 mg

- Adverse Event Dataset (either ADAE or AE)
  - ADAE: ADaM Dataset
  - AE: SDTM Dataset

USUBJID	AEDECOD	AETOXGR
9999-001_000100008	Decreased appetite	2
9999-001_000100008	Gastric stenosis	3
9999-001_000100009	Dry mouth	1
9999-001_000100009	Skin infection	2
9999-001_000100009	Pyrexia	1
9999-001_000100009	Fatigue	1
9999-001_000100009	Pruritus	1
9999-001_000100009	Dermatitis acneiform	1
9999-001_000100010	Infusion related reaction	1
9999-001_000100010	Influenza like illness	1
9999-001_000100010	Fatigue	2
9999-001_000100010	Cough	1
9999-001_000100010	Photosensitivity reaction	1

STEP 2. DATA SUMMARY (DLT BY MAXIMUM GRADE)

1) Sort the treatment group arm code by user-specified dosage level order. Create a pseudo arm code based on the user-specified dosage level order.

TRT01AN	Pseudo Arm Code
3	1
4	2
5	3
6	4
7	5
8	101
9	102
10	103
11	104
12	105

2) Calculate count of DLT Evaluable population per pseudo arm code.

3) If a subject has multiple dose limiting toxicity events with the same AE terms but different grades,

only the one with the highest grade will be kept.

4) Calculate frequency of each dose limiting toxicity grade (AETOXGR value: 1, 2, 3, 4, 5) per pseudo arm code and AEDECOD.
5) Calculate count of DLT subjects (with at least one AEDECOD records from step 3) per pseudo arm code and AEDECOD.
6) Calculate count of DLT subjects (with at least one AEDECOD records from step 3) per pseudo arm code.
7) Calculate percentage of DLT subjects per pseudo arm code by dividing number from step 5 by number from step 2.

STEP 3. BAYESIAN POSTERIOR CREDIBLE INTERVAL

```
DATA BCI;
     Npt=6;
     Ndlt=1;
     level=0.8;
     a=1;
     b=1;
     e=0.001;
     Maxstart=quantile('BETA', 1-level, a+Ndlt, b+Npt-Ndlt);
     do BCIstart = 0 to Maxstart by e;
        BCIend=quantile('BETA', level+cdf('BETA', BCIstart, a+Ndlt,
                                b+Npt-Ndlt), a+Ndlt, b+Npt-Ndlt);
        BDIlength=BCIend-BCIstart;
        output;
     end;

     keep BCIstart BCIend BDIlength;
     RUN;

PROC SORT DATA=BCI OUT=BCI_sort ;
      BY BDIlength;
RUN;

PROC PRINT DATA=BCI_sort(obs=1);
      VAR BCIstart BCIend;
RUN;
```

Display 6: Snapshot of %or0sa0dlt0summary for Bayesian Credible Interval calculation

STEP 4 TO STEP 7. PAVA DLT RATE ESTIMATION

```
data p;
     p=&pava_target;
run;

proc iml;
    title "Statistic in R (integration with SAS)";
    title2 "PAVA estimate";
    run ExportDataSetToR("WORK.TOTAL", "TOTAL_IN_R");

    run ExportDataSetToR("WORK.P", "P_IN_R");

    %inc &rcode_fileref(r0code0pava.sas);

    run ImportDataSetFromR( "WORK._RRESULT", "result" );
```

```
quit;
```

Display 7: Snapshot of %or0sa0dlt0summary for PAVA DLT Rate Estimation

OUTPUT

Summary of Dose Limiting Toxicity
(DLT Evaluable Population)

Treatment Group	N	Dose Limiting Toxicity	Subject with DLT			DLT by Maximum Grade†				
			n	% (80% CI)‡	PAVA[I] Estimate (%)	Grade 1	Grade 2	Grade 3	Grade 4	Grade 5
Treatment Group 1 (Dose Level 1)	##	TOTAL	##	## (##, ##)	##					
		Fatigue	##	##		##	##	##	##	##
Treatment Group 2 (Dose Level 2)	##	TOTAL	##	## (##, ##)	##					
		Nausea	##	##		##	##	##	##	##
Treatment Group 3 (Dose Level 3)	##	TOTAL	##	## (##, ##)	##					
		Nausea	##	##		##	##	##	##	##
		Vomiting	##	##		##	##	##	##	##
Treatment Group 4 (Dose Level 4)	##	TOTAL	##	## (##, ##)	##					
		Alanine aminotransferase increased	##	##		##	##	##	##	##
		Aspartate aminotransferase increased	##	##		##	##	##	##	##
		Blood amylase increased	##	##		##	##	##	##	##
		Hypokalaemia	##	##		##	##	##	##	##

DLT=Dose Limiting Toxicity
Adverse event terms are based on MedDRA version XX.X.
Grades are based on NCI CTCAE version X.X.
Dose limiting toxicity (DLT) evaluable population consists of subjects XXXX.
† Only the highest reported grade for a given DLT is counted for the individual subject.
‡ 80% confidence interval is based on the Bayesian posterior credible interval with a prior distribution of Beta (1, 1).
[I] PAVA: Pooled-adjacent-violator algorithm. For trials with both mono-therapy and combination-therapy arms, this is two-dimensional PAVA (2-D PAVA).

(Database cutoff date: DDMMMYYYY)

Output 2. Output from macro %or0sa0dlt0summary

CONCLUSION

This paper exemplified the integration of SAS and R in an analysis and reporting package. We introduced the key syntaxes of passing the data from SAS to R, calling R script in SAS, and bringing data from R back to SAS. A detailed macro code flow was displayed to show how DLT summary table can be automatically generated by integrating SAS and R. Taking advantages of the strengths of both can be a powerful tool for analysis and reporting.

REFERENCES

https://communities.sas.com/t5/General-SAS-Programming/Run-R-code-inside-SAS-easily/td-p/210116
http://support.sas.com/documentation/cdl/en/imlug/63541/PDF/default/imlug.pdf
https://www.analytical-software.de/fileadmin/doc/papers/HMS_-_PhUSE_2011_-_Calling_R-Functions_from_SAS.pdf
Bril, G., et al., Algorithm AS 206: Isotonic Regression in Two Independent Variables. Journal of the Royal Statistical Society. Series C (Applied Statistics), 1984. 33(3): p. 352-357.

ACKNOWLEDGMENTS

The authors would like to thank the management teams for their advice on this paper/presentation.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the authors at:

Huei-Ling Chen
c/o Merck & Co., Inc.
126 Lincoln Avenue
P.O. Box 2000
Rahway, NJ 07065
Phone: 732-594-2287
e-mail: huei-ling_chen@merck.com

Zhen Zeng
c/o Merck & Co., Inc.
126 Lincoln Avenue
P.O. Box 2000
Rahway, NJ 07065
Phone: 732-594-8143
e-mail: zhen_zeng1@merck.com

TRADEMARK