PharmaSUG 2012 - Paper AD24

Let's compare two SAS® libraries!

Kavitha Madduri, Boehringer Ingelheim Pharmaceuticals Inc., Ridgefield, CT

ABSTRACT:

Here is a typical scenario that we come across very often – you have twelve datasets in one library and twenty datasets in another library. You want answer to below questions:

a) What datasets are in one library and not the other?
b) For the common datasets:
    i. Is the number of observations the same?
    ii. Are the variable attributes the same?
    iii. Are there any data differences between the datasets, if the number of observations is the same?

One can write SAS® code to answer each of the questions above. But these situations are so common that you do not want to spend time every time to determine the differences between the two SAS® libraries. It would be easy if we have a SAS® macro that would do it automatically for us.

This paper introduces COMPARE_LIB macro to compare datasets in two SAS® libraries and produces reports to answer each question mentioned above. This macro would work on SAS® version 8 and above.

INTRODUCTION

For most of the SAS® programmers, there often will be a situation where they need to compare the data between two libraries. For example, a data snapshot is taken this month and couple of months later another snapshot is taken. In this case we would like to know the data differences between these two snapshots. One way to accomplish this task would be to do a PROC COMPARE individually on each dataset, something like shown below.

```
PROC COMPARE BASE=baselib.dataset1 COMPARE=comparelib.dataset1;
RUN;

PROC COMPARE BASE=baselib.dataset2 COMPARE=comparelib.dataset2;
RUN;
.
.
.
PROC COMPARE BASE=baselib.datasetn COMPARE=comparelib.datasetn;
RUN;
```

It is a laborious and time consuming process to go over the PROC COMPARE results individually and then summarize the differences between the two libraries. This paper describes the COMPARE_LIB macro that was developed to make the comparison process easy. As the macro program is proprietary, the code will not be shared. Snippets of code will be provided to give an idea of the functionality of the macro.

DISCUSSION

COMPARE_LIB macro compares the datasets in given two libraries and produces four different reports to help summarize the differences. The following sections describe the inputs, error checks, reports, and limitations of the macro.

Table 1 describes the COMPARE_LIB macro input parameters and the ones highlighted are the default values.

Macro parameter	Parameter Description	Valid values	Example
BASEPATH	Path of the base library	A UNIX or WINDOWS path	C:\Documents and Settings\data
COMPAREPATH	Path of the compare library	A UNIX or WINDOWS path	C:\Documents and Settings\Project\data
UNQVARNM	Name of the unique variable that is commonly present in all the datasets in base and compare libraries	Name of the most commonly present variable name in the datasets	PTNO or USUBJID
EXCLLIST	List of datasets that should be excluded from the comparison from both base and compare libraries	Comma delimited quoted list of dataset names	%str('AE', 'PATD')
INCLLIST	List of datasets that should be included in the comparison from both base and compare libraries	Comma delimited quoted list of dataset names	%str('TTM', 'DEMOG')
REPORT	Report types	*DATASET,* *OBSERVATION*, ATTRIBUTE, DATA	Observation or Attribute Observation Data
PCOMPARE_OPTS	Any of the PROC COMPARE options	Valid PROC COMPARE procedure options	Maxprint or Listall
SORDER	Sort order for the datasets in base and compare libraries. This sort order is used when comparing the datasets	Space delimited variable names	STUDY PTNO or STUDYID USUBJID
DELETEDS	Deletes temporary datasets created by the macro, deassigns the library references assigned within the macro, and resets the titles	*Yes*, No	Yes

Table 1 Macro input parameters

ERROR CHECKS

The COMPARE_LIB macro does comprehensive checks on the values given to the parameters before processing the compare request. The macro will take either the inclusion or the exclusion list of dataset(s) and not both of them. Table 2 below describes the error checks performed by the macro. The errors are sent to the log for any issues identified during the error check process. The error messages in the log are descriptive enough to help the user identify the problem running the macro.

Macro parameter	Parameter Description	Error checks
BASEPATH	Path of the base library	Check to see if the base path exists
COMPAREPATH	Path of the compare library	Check to see if the compare path exists. If both the base and compare paths exist, check to make sure the data type is the same in these libraries.
UNQVARNM	Name of the unique variable that is commonly present in all the datasets in base and compare libraries	Check if the unique variable exists in the datasets from both base and compare library
EXCLLIST	List of datasets that should be excluded from the comparison from both base and compare libraries	Check whether the dataset(s) given in the exclusion list exists in the base and compare library. At least one dataset from the exclusion should be present in both the libraries. Also checks INCLLIST is null
INCLLIST	List of datasets that should be included in the comparison from both base and compare libraries	Check whether the dataset(s) given in the inclusion list exists in the base and compare library. At least one dataset from the inclusion should be present in both the libraries. Also checks EXCLLIST is null
REPORT	Report types	Check to see if valid values (see Table 1) are given to this parameter
PCOMPARE_OPTS	Any of the PROC COMPARE options	No checks
SORDER	Sort order for the datasets in base and compare libraries. This sort order is used when comparing the datasets	Check to see if the variable name(s) given for the sort order exist in the datasets from both the libraries
DELETEDS	Deletes temporary datasets created by the macro, deassigns the library references assigned within the macro, and resets the titles	Check to see if valid values (see Table 1) are given to this parameter

Table 2 Error checks

REPORTS

To help summarize the differences between the datasets in the two libraries, the COMPARE_LIB macro produces four different reports. The macro assigns report titles internally and resets them by default after completion of the macro. The reports are produced by the macro for only those datasets that contain the unique variable name identified in &UNQVARNM.

Report 1 list out the dataset(s) that are in one library and not in the other. Reports 2 thru Report 4 are produced only for common datasets in both base and compare library i.e., they exclude the datasets from Report 1. This is because we are not interested in knowing either the number of observations or the number of variables or the data differences for the datasets that are present in only one library.

Report 2 produces a list of dataset(s) with the number of observations and the number of variables. Report 3 lists the attribute differences between the datasets. Report 4 produces the PROC COMPARE output for the datasets that have the same number of observations. The datasets are sorted by the variable(s) given to &SORDER parameter.

In the pharmaceutical industry, an example of a unique variable name would be the patient number. A majority of the trial datasets will have this variable. The sort order example would be study number and patient number.

COMPARE_LIB takes the base and compare paths as input and library references are declared within the macro as below. The library references get deassigned by default at the end of the macro completion.

```
LIBNAME baselib    "&BASEPATH.";
LIBNAME comparelib "&COMPAREPATH.";
```

Let us see how these reports help us answer the typical questions one would have when comparing the data between two libraries.

Q1: WHAT DATASETS ARE IN ONE LIBRARY AND NOT THE OTHER?

This is the first question that comes to mind when comparing two libraries. The COMPARE_LIB macro by default produces a report to answer this exact question. First part of the report lists the datasets that are in the base library and not in compare library. And second part lists the datasets in the compare library and not in the base library.

The COMPARE_LIB macro uses the SASHELP.VCOLUMN dictionary table to get the dataset names from base and compare libraries. Only those datasets that have the unique variable name are extracted from the dictionary table.

```
PROC SQL;
      CREATE TABLE base_ds AS
            SELECT DISTINCT memname
            FROM sashelp.vcolumn
            WHERE  libname eq "BASELIB" AND
                   memtype eq "DATA" AND
                   name contains ("&unqvarnm.")
            ORDER BY memname;

      CREATE TABLE compare_ds AS
            SELECT DISTINCT memname
            FROM sashelp.vcolumn
            WHERE  libname eq "COMPARELIB" AND
                   memtype eq "DATA" AND
                   name contains ("&unqvarnm.")
            ORDER BY memname;
QUIT;
```

Using the PROC SQL set-operator EXCEPT the datasets in one library but not the other is derived. These datasets are used to generate the report.

```
PROC SQL;
      %**** datasets in base library and not in compare ****;
      CREATE TABLE inbase_notcompare AS
            SELECT memname
            FROM base_ds
            EXCEPT
            SELECT memname
            FROM compare_ds;

      %**** datasets in compare library and not in base ****;
      CREATE TABLE incompare_notbase AS
            SELECT memname
            FROM compare_ds
            EXCEPT
            SELECT memname
            FROM base_ds;
QUIT;
```

Here is the sample macro call to produce the default dataset report and it is shown in Output 1:

```
%COMPARE_LIB(basepath   = Z:\Documents and Settings\PSUG2012\base,
             comparepath= Z:\Documents and Settings\PSUG2012\compare,
             unqvarnm   = ptno,
             sorder     = study ptno
            );
```

```
Datasets in Base Library NOT in Compare Library

Base library   : Z:\Documents and Settings\PSUG2012\base
Compare library: Z:\Documents and Settings\PSUG2012\compare


_________________________

  Name of Dataset
_________________________

  ADM
  AEAEA
  TTM
```

Output 1 Dataset Report (BASE)

```
List of Datasets in Compare Library NOT in Base Library

Base library   : Z:\Documents and Settings\PSUG2012\base
Compare library: Z:\Documents and Settings\PSUG2012\compare


_________________________

  Name of Dataset
_________________________

  TERM
  VDT
```

Output 1 Dataset Report (COMPARE)

Q2: IS THE NUMBER OF OBSERVATIONS THE SAME?

After we know the datasets that are in one library and not the other, the next question would be to know if there are any differences in the number of observations and number of variables for the datasets that are present in both the two libraries. The COMPARE_LIB macro uses the SASHELP.VCOLUMN to retrieve the list of datasets and number of datasets from each library. The macro loops through the datasets and runs PROC CONTENTS on them to get the number of observations, number of variables and the variable attributes. The counts are used to produce Report 2 and the attributes are used to produce Report 3.

```
PROC SQL NOPRINT;
       SELECT DISTINCT(memname), COUNT(DISTINCT(memname))
               INTO  :dom_lst separated by '~',
                     :dom_cnt
       FROM sashelp.vcolumn
       WHERE  libname eq "BASELIB" AND
              memtype eq "DATA" AND
              name contains ("&unqvarnm.")
       ORDER BY memname;
QUIT;
```

```
%**** loop through the datasets to get the attributes and counts ****;
%DO i = 1 %TO &dom_cnt;
       %LET dom_nm = %SCAN(&dom_lst., &i., ~);
       PROC CONTENTS DATA=work.&dom_nm. OUT=base_attrib NOPRINT;
       RUN;
%END;
```

Most of the times there are quite a number of datasets in each library. To help identify the datasets that have differences in the number of observations and/or the number of variables, Report 2 has a column 'Any differences?'. This column will have a value 'YES' to identify the dataset that has differences.

Here is the sample macro call to produce the default observation report and it is shown in Output 2:

```
%COMPARE_LIB(basepath    = Z:\Documents and Settings\PSUG2012\base,
             comparepath = Z:\Documents and Settings\PSUG2012\compare,
             unqvarnm    = ptno,
             sorder      = study ptno
            );
```

Report of Number of Observations and Columns for Common Views/Datasets

Base library : Z:\Documents and Settings\PSUG2012\base
Compare library: Z:\Documents and Settings\PSUG2012\compare

	No. of Observations			No. of Variables		
Name of Dataset	base	compare	any differences?	base	compare	any differences?
ADMB	253	253		35	35	
AE	683	683		54	54	
BCOND	618	618		21	21	
LAB2	16454	16389	YES	42	42	
BIOM	51	51		32	32	
CT	962	962		29	29	
DAS	250	250		20	20	
ECGECG	758	758		20	20	
ECGEECGE	752	752		63	63	
ECGRECGR	2250	2250		47	47	
ELIG	62	62		20	20	
ELRC	115	115		18	18	
E_ARISG	83	83		28	28	
E_TRTEXP	618	618		18	18	
GIC	50	50		18	18	
GICW	47	47		18	18	
GS	0	0		18	18	
HOSP	55	55		18	18	
INEXEXCL	1232	1232		17	17	
INEXINCL	392	392		17	17	
LABL	1361	1356	YES	18	18	
LABLLABG	98	98		16	16	
LABLLABL	16356	16291	YES	21	21	
PATD	208	208		45	44	YES

Output 2 Observation Report

Q3: ARE THE VARIABLE ATTRIBUTES THE SAME?

After identifying the differences in the number of observations and the variables, the next thing we want to know is if there are any variable attribute differences between the common datasets in two libraries. COMPARE_LIB macro checks for these four variable attributes - type, length, label and format.

As described under question 2, the COMPARE_LIB macro loop through the dataset list, from SASHELP.VCOLUMN dictionary table, and retrieves the attributes for each dataset from PROC CONTENTS.

The attributes for datasets from base library are stored in base_attrib dataset and the ones from compare are stored in compare_attrib. These two attribute datasets are compared for differences and Report 3 is generated. This report has a 'Comment' column which gives information when there is an extra variable in one dataset and not the other. A variable listed in Report 3 can have one or more attribute differences. Even if a variable has just one attribute difference, all the four attributes are displayed. The user will have to check the variable row to identify the attribute that has the difference. For example, in Output 3 variable RACEI has only format difference.

Here is the sample macro call to produce the attribute report and it is shown in Output 3.

```
%COMPARE_LIB(basepath    = Z:\Documents and Settings\PSUG2012\base,
             comparepath = Z:\Documents and Settings\PSUG2012\compare,
             report      = attribute,
             unqvarnm    = ptno,
             sorder      = study ptno
            );
```

Report of Attribute Comparison for Common Views/Datasets

Base library : Z:\Documents and Settings\PSUG2012\base
Compare library: Z:\Documents and Settings\PSUG2012\compare

Dataset Name	Variable Name	Data Type		Length		Format		Comment
		Base	Compare	Base	Compare	Base	Compare	
PATD	BSAU	Char	Num	20	8		UNITF	
	BSAU1	Num	.	8	.	UNIT1F		BSAU1 not in COMPARE.PATD
	CENTRE	Num	Num	8	8	10	10	
	PTNO	Num	Num	8	5	10	10	
	RACEI	Num	Num	8	8	8	RACEF	
	SEX	Num	Num	2	8	SEX1F	SEXF	
	SMOKCD	Num	Num	8	8	8	SMOKCDF	

Dataset Name	Variable Name	Label	
		Base	Compare
PATD	BSAU		Body surface area unit
	BSAU1	Body surface area unit	
	CENTRE	Investigator Centre	Centre
	PTNO	Universal Patient Number	Patient number
	RACEI	Race	Race
	SEX	Sex	Sex
	SMOKCD	Smoking history	Smoking history

Output 3 Attribute Report

Q4: ARE THERE ANY DATA DIFFERENCES BETWEEN THE DATASETS, IF THE NUMBER OF OBSERVATIONS IS THE SAME?

After knowing the observation, variable, and attribute differences the last thing we want to know is the data-value differences between the common datasets from two libraries.

One limitation for this report is that it produces the data report only if the number of observations is the same for the common datasets from two libraries. The reason being one would not want to compare the two datasets knowing already that the number of observations is different. It is important to look at those dataset(s) separately to identify the reason for the different observations and then use the COMPARE_LIB macro. For example, consider a dataset that has thousands of observations. Even for a single observation difference, PROC COMPARE would produce large amount of output. Sometimes the reason for the different observations could just be that one dataset has more patients or visits than other. This difference is good enough to investigate further.

COMPARE_LIB macro takes any of the valid PROC COMPARE options when it comes to the data comparison between the datasets. The macro internally creates temporary datasets for the common datasets from the two libraries where they have same number of observations. These temporary datasets are sorted by the variables given in the &SORDER parameter. They are named base_s for the base library datasets and compare_s for the compare library datasets. In the data report, the name of the dataset being compared is displayed. The temporary datasets are deleted by default at the end of the macro completion. By default the COMPARE_LIB macro does the EXACT comparison between the datasets.

Here is a sample macro call to produce the data report and it is shown in Output 4:

```
%COMPARE_LIB(basepath    = Z:\Documents and Settings\PSUG2012\base,
             comparepath = Z:\Documents and Settings\PSUG2012\compare,
             report      = data,
             unqvarnm    = ptno,
             sorder      = study ptno
             );
```

```
Report of Data Comparison for Common Views/Datasets with Same Number of Observations


Base library   : Z:\Documents and Settings\PSUG2012\base
Compare library: Z:\Documents and Settings\PSUG2012\compare

----------------------------
DOMAIN NAME: ADMB
----------------------------

The COMPARE Procedure
Comparison of WORK.BASE_S with WORK.COMPARE_S
(Method=EXACT)

Data Set Summary

Dataset                   Created          Modified  NVar    NObs

WORK.BASE_S     15MAR12:22:06:02  15MAR12:22:06:02    35     253
WORK.COMPARE_S  15MAR12:22:06:03  15MAR12:22:06:03    35     253


Variables Summary

Number of Variables in Common: 35.


Observation Summary

Observation      Base  Compare

First Obs           1        1
```

```
Last  Obs         253      253

Number of Observations in Common: 253.
Total Number of Observations Read from WORK.BASE_S: 253.
Total Number of Observations Read from WORK.COMPARE_S: 253.

Number of Observations with Some Compared Variables Unequal: 0.
Number of Observations with All Compared Variables Equal: 253.

NOTE: No unequal values were found. All values compared are exactly equal.
```

Output 4 Data Report

The COMPARE_LIB macro gives the user flexibility to select the report(s) of interest. Any combination or all of the above discussed reports can be used in the macro call. For example, here is the sample macro call to produce all the reports discussed above.

```
%COMPARE_LIB(basepath    = Z:\Documents and Settings\PSUG2012\base,
             comparepath = Z:\Documents and Settings\PSUG2012\compare,
             report      = dataset observation attribute data,
             unqvarnm    = ptno,
             sorder      = study ptno
            );
```

There will be situations where there may not be any differences in the datasets between the two libraries. They may both have the same number of identical datasets. In this case, the macro produces a note for the dataset report. In another case there may not be any attribute differences between the common datasets. So the macro produces a note for the attribute report. For the data report, if there are no common datasets with equal number of observations then a note for the data report is generated by the macro. All these notes for the reports are shown in Output 5. These reports are useful for documentation purposes.

```
              ***** FOR DATASET REPORT *****
 The dataset count and dataset names are equal in Base and Compare library


              ***** FOR DATA REPORT *****
 There are no common datasets with same number of observations in base and compare library


              ***** FOR ATTRIBUTE REPORT *****
 There is no attribute discrepancy between the datasets in Base and Compare library
```

Output 5 Reports produced by the COMPARE_LIB macro when there are no differences or certain conditions are not met

LIMITATION

The type of data compared in the two libraries should be the same. For example you cannot have member type as view in one library and data in another library.

CONCLUSION

Whenever there is a need to compare two libraries, it would be laborious and time consuming to do a PROC COMPARE on each individual dataset and then review the results to summarize the differences. The COMPARE_LIB macro discussed here is efficient and easy to use for comparison purposes and is highly recommended. There is no need to manually document the differences. The reports produced by the macro can be used for documentation purposes. The dataset, observation, and attribute reports generated by this macro are useful and they are most of the time one page long. A quick glance at these reports gives the user an overview of the differences between the two libraries.

CONTACT INFORMATION:

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

Name:	**Kavitha Madduri**
Enterprise:	Boehringer Ingelheim Pharmaceuticals Inc.
Address:	900 Ridgebury Road
City, State ZIP:	Ridgefield, CT 06877
Work Phone:	203-791-6208
E-mail:	Kavitha.Madduri@Boehringer-Ingelheim.com
	Kavitha.Madduri@gmail.com