PharmaSUG 2015 - Paper PO16

Automating biomarker research visualization process

Xiaohui Huang, Gilead Sciences Inc., Foster City, CA
Jigar Patel, Gilead Sciences Inc., Foster City, CA

ABSTRACT

Biomarker assays have become more and more popular in drug discovery clinical trial designs. It is widely used in different therapeutic areas to support decision making regarding drug candidates and accelerate drug development, in addition to reducing costs. Graphic presentation is essential in every Biomarker study report. An analysis of Biomarker data are typically fast paced and facing a tight timeline. A very useful tool is to have a utility macro to facilitate the visualization process and promote standardization to fulfill regulatory requirements. This paper presents a macro toolkit for generating the most commonly used statistical graphics in biomarker data analysis. Instead of relying on the default template, the code is developed in SAS 9.2 and using Graph Template Language (GTL) to take control of the graphic appearance. One feature of this macro is that it can automate symbol assignments to different populations in a consistent way across different plots. The following will describe key elements of the macro functionality and techniques used in code development. This macro tool is not only for programmers but also for clinical scientists who do not necessarily have SAS programming skills but want to do some exploratory research of the biomarker data themselves.

INTRODUCTION

Macro is developed with the graph template language (GTL), PROC TEMPLATE, and PROC SGRENDER. GTL is one part of the SAS Output Delivery System (ODS), to create statistical graphs. It is more powerful and flexible in creating customized graphs than SG (statistical graphics) procedures (SGPLOT, SGSCATTER, and SGPANEL). This paper is going to introduce 4 types of statistical graphs that are commonly used in biomarker research and discuss key elements in building these graphs in GTL. The graphs covered in the paper are:

1. Spaghetti plot of %baseline in CD63 by time point.

2. Box plot of %baseline in CD63 by time and treatment.

3. Mean/SE or Median/Q1Q3 plot of %baseline in CD63 by time and treatment.

4. Scatter Plot over Box plot of %baseline in CD63 by time and treatment.

INPUT DATA STRUCTURE

PROC TEMPLATE contains a set of instructions for SAS to make graphs. PROC SGRENDER is the step for SAS to read in a SAS dataset and draw the graph based on template specification. Biomarker dataset structure is similar to the LB domain. CDISC has developed domains, such as Biospecimen Events (BE), Pharmacogenomics/Genetics Findings (PF) and Pharmacogenomics/Genetics Methods and Supporting Information (PG), which are suitable for different types of biomarker information.

For demonstration purpose, the following is simulated data. The data structure is adapted from CDISC standard PF domain with addition of basic subject and treatment information. This dataset contains subject ID (SUBJID), Treatment group (TRTP), visit, time points (PFTPT) and %baseline CD63 result values (PFSTRESN). Data was simulated to have subjects under three distinct dosing/treatment groups and have visits on Days 1, 5, and 10 at time point 0 hour, 2 hours, 4 hours and 6 hours.

SUBJID	TRTPN	TRTP	PFTESTCD	VISIT	VISITNUM	PFTPT	PFTPTNUM	PFSTRESN
3648-1001	6	Placebo	CD63BASE	Day 1	1	0 hour	0	-37.11
3648-1001	6	Placebo	CD63BASE	Day 1	1	1 hour	1	-87.03
3648-1001	6	Placebo	CD63BASE	Day 1	1	2 hours	2	-1.98
3648-1001	6	Placebo	CD63BASE	Day 1	1	4 hours	4	-30.8
3648-1001	6	Placebo	CD63BASE	Day 1	1	6 hours	6	110.4
3648-1002	1	S1234 0.1 mg	CD63BASE	Day 1	1	0 hour	0	-19.3
3648-1002	1	S1234 0.1 mg	CD63BASE	Day 1	1	1 hour	1	61.21
3648-1002	1	S1234 0.1 mg	CD63BASE	Day 1	1	2 hours	2	-189.46
3648-1002	1	S1234 0.1 mg	CD63BASE	Day 1	1	4 hours	4	167.29
3648-1002	1	S1234 0.1 mg	CD63BASE	Day 1	1	6 hours	6	150.22

Figure 1. Sample input data structure

SPAGHETTI PLOT

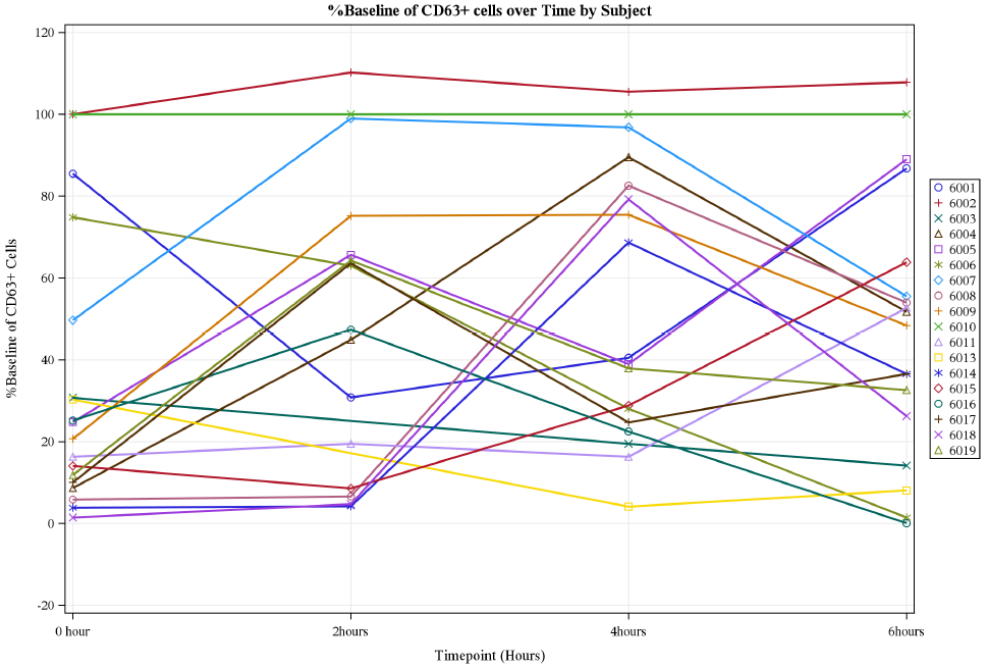

Figure 2. Spaghetti plot of %baseline in CD63 by time point

Figure 2 shows %Baseline CD63 values for each patient, which is grouped by subject ID. In the graphic template, it calls SCATTERPLOT statement to draw the dot at each time point in the x axis and use the SERIESPLOT to draw the lines to connect the dots within each group.

```
/* &xval: variable to display in x axis*/
/* &yval: variable to display in y axis*/
scatterplot x=&xval y=&yval/group = &grp index=inx name="gp";
seriesplot x=&xval y=&yval/group = &grp index=inx lineattrs=(PATTERN=1);
```

When the axis values are numeric values, it is often requested to keep consistent scale ranges for x and y axis with the same test across different plots. The fact is, the default setting for ODS Graphics axis range is data driven. It might end up with one graph which has y axis from 0 to 100, while the other one with the same test parameter in different testing date has y axis from -10 to 120. One way to restrict the y axis to have axis range and tick value display is using ROWAXIS statement with LINEAROPTS and TICKVALUESEQUENCE option.

```
/* &ylabel: y axis label */
/* &ymin: y axis minimum value*/
/* &ymax: y axis maximum value */
/* &yinc: y axis increases unit */
rowaxis / label="&ylabel" griddisplay=&bgrid linearopts=(viewmin=&ymin viewmax=&ymax
tickvaluesequence=(start=&ymin end=&ymax increment=&yinc));
```

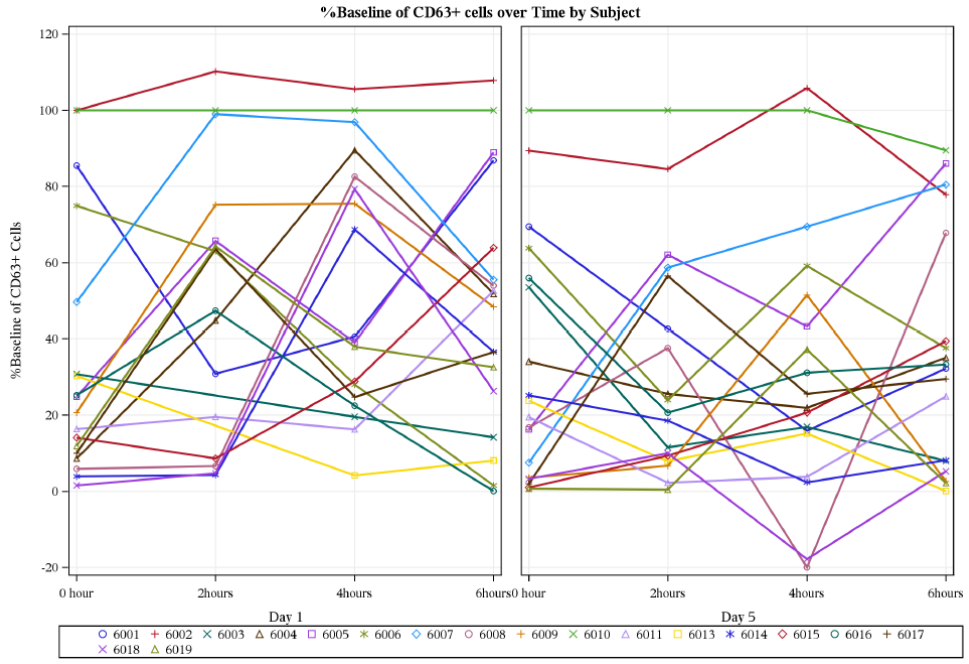

Figure 3. Spaghetti plot of %baseline in CD63 by Day and time point.

Depends on the study design, it is often required to display graphs side by side as show in Figure 3. The template is built under the following structure to add multiple lays into one graph. &panum is number of panels to be presented in one page. Set ROWDATARANGE to UNION is to ensure all y axises to follow the same tick range.

```
proc template;
define statgraph mygraphs.expression;
begingraph/ designwidth=9.5in designheight=6.5in border=off;
/* For multiple layers */
layout lattice / columns=&panum rows=1 rowdatarange=union;
rowaxes;
       rowaxis / label="&ylabel" griddisplay=&bgrid linearopts=(viewmin=&ymin
       viewmax=&ymax tickvaluesequence=(start=&ymin end=&ymax increment=&yinc));
endrowaxes;

/* Draw panels by stacking LAYOUT OVERLAY statement*/
%do j=1 %to &panum.;
/* &&sepl&j: distinct panel grouping variable values */
/* &tk.: tick values in x axis */
/* ytmp_&j: y values for corresponding panel group */
layout overlay /xaxisopts=(label= "&&sepl&j" griddisplay=&bgrid
linearopts=(TICKVALUELIST= (&tk.)));

       scatterplot x=&xval y=ytmp_&j/group = &grp index=inx name="gp";
       seriesplot x=&xval y=ytmp_&j/group = &grp index=inx lineattrs=(PATTERN=1);
endlayout;
%end;
* Legend bar;
sidebar / align=bottom;
       discretelegend "gp";
endsidebar;
endlayout;
endgraph;
end;
run;
```

BOX PLOT

The following box plot displays %baseline CD63 values by treatment groups in different time points.

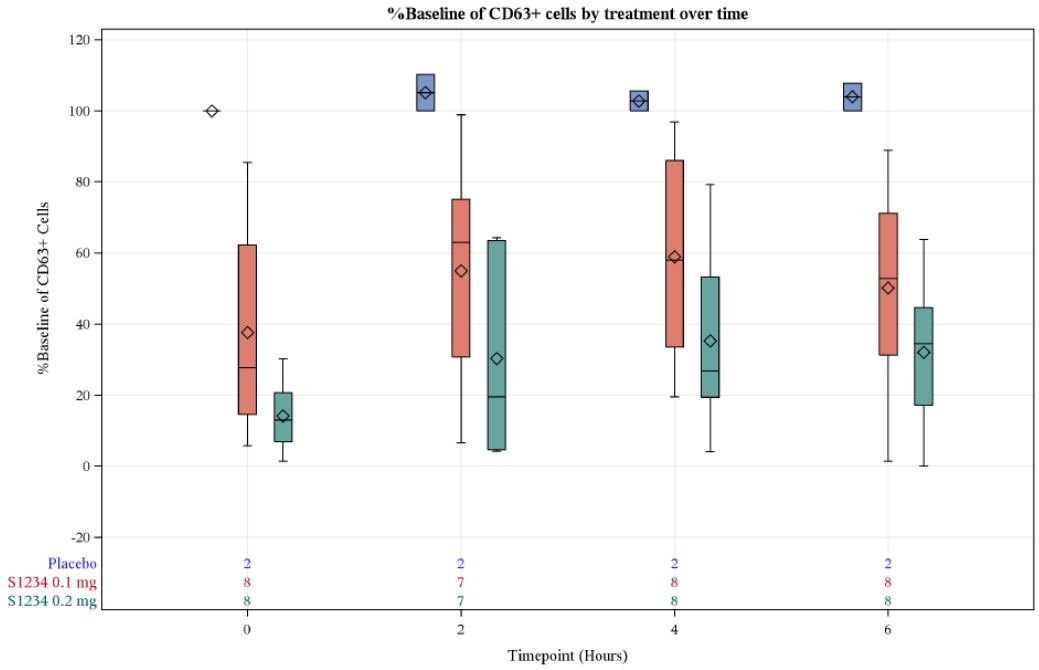

Figure 3. Box plot of %baseline in CD63 by time point.

Since multiple treatments are plotted on the same time point, multiple box bars will be overlaid on one another. A separation between bars is necessary to have a better view on each treatment group. The following code takes number of treatment groups in the dataset and calculates offset value for each box bar. Macro variable Off&i can be used in DISCRETEOFFSET option in the BOXPLOT statement. This option is used to move the box bar position along the discrete axis. It offsets the bar by a fraction of the midpoint spacing. For example, the first box bar is offset to the left of the midpoint by %eval(&off1)*100%. In the meantime, the width of each box bar also needs to be considered. It is defined in the BOXWIDTH option.

```
/* &grp_m: number of treatment groups */
/* &&off&i: offset values for group i */
/* _&I: y axis value in group i */
/* &&tn&i: group value in numeric */
/* &&t&i: group value in character */
data _NULL_;
tmp= int(%eval(&grp_m)/2);
call symputx("it",tmp);
run;

%let off1=%SYSEVALF(-1*1/%eval(&grp_m)*&it);
%let t_p = &off1;
%do i = 2 %to &grp_m;
%let off&i = %SYSEVALF(&t_p+%SYSEVALF(1/%eval(&grp_m)));
%let t_p = &&off&i;
%end;

……

boxplot x=&xval. y=_&I /discreteoffset=&&off&i boxwidth=%SYSEVALF(1/%eval(&grp_m)*0.5)
name="&&tn&i" legendlabel="&&t&i";
```

MEAN/SE OR MEDIAN/Q1Q3 PLOT

The following graph presents the Mean/SE of the %baseline CD63 values for different treatment groups over time.

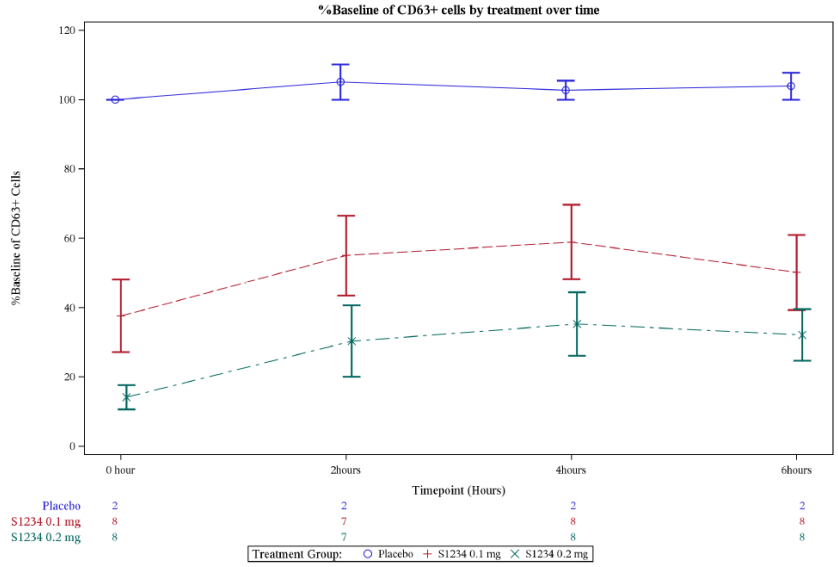

Figure 4. Mean/SE plot of %baseline in CD63 by time point.

There are groups of graph attributes predefined in style elements, named as GraphData1 – GraphDataN. When using a GROUP option on a plot statement, SAS automatically assigns each group sequentially to style element for unique display. In order to add visual consistency within one study, different type of graphs should contain the same set of attribute. For example, if placebo group was in blue color and has circle dot in the box plot, then the MEAN/SE and rest of the graphs associated with treatment as a group classification should be assigned to the same set of attribute. This can be accomplished by adding a positive integer value corresponds to groups in the dataset to associate with GraphData1-GraphDataN style elements. The code here shows both MEAN/SE plot and Box plot use GRAPHDATA to keep element attributes consistent across different types of plots.

```
/* Code in MEAN/SE plot */
%do i=1 %to &inx_m;
/* &inx_m: number of treatment groups*/
/* &&offsd&i: offset value*/
scatterplot x=eval(&xval+&&offsd&i) y=meanval&i/
yerrorlower=eval(meanval&i-&sdv*stderr&i)
yerrorupper=eval(meanval&i+&sdv*stderr&i)
markerattrs=graphdata&i(size=9px weight=bold weight=normal)
errorbarattrs=graphdata&i(pattern=solid thickness=1) name="tr&i";
seriesplot x=eval(&xval+&&offsd&i) y=meanval&i/lineattrs=graphdata&i;
%end;

/* Code in BOX PLOT */
boxplot x=&xval. y=_&tsep&i /  discreteoffset=&&off&i
boxwidth=%SYSEVALF(1/%eval(&grp_m)*0.5) name="&&tn&i" legendlabel="&&t&i"
fillattrs=graphdata&i;
```

PLOT SCATTER PLOT OVER BOX PLOT

In GTL, different types of plots can be easily overlaid on each other. In this case, scatter plot and box plot are drawn in the same graph. Notice Box was plotted in category x axis, to make the dots scattered within each group, a separate numeric variable gr_&i was generated for each category. In this case normal distribution is used to generate random numbers to apply on the second x axis, so called X2 in the GTL template.

```
data final;
set gdata;
%do i=1 %to &grp_m;
```

```
/*xgrp: numeric treatment group value*/
if xgrp eq &i then gr_&i = xgrp + 0.1*rannor(0); %end;
run;

boxplot x=&grp y=&yval/display=(caps mean median connect) fillattrs=(color=white) ;
%do i=1 %to &grp_m;
scatterplot x=gr_&i y=&yval/index=xgrp xaxis=x2 MARKERATTRS=graphdata&i name="&&tn&i"
legendlabel="&&t&i";
%end;
```

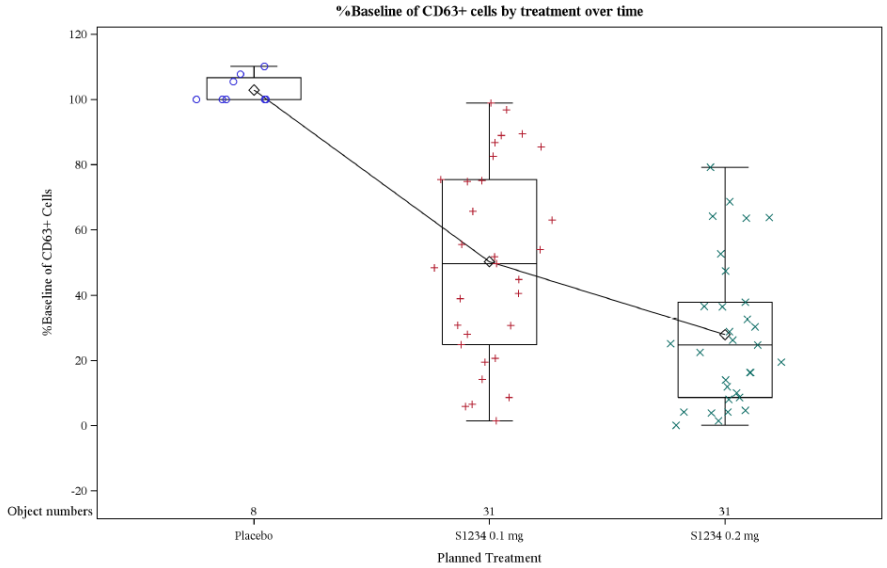

Figure 5. Scatter Plot over Box plot of %baseline in CD63 by time and treatment

CONCLUSION

This paper demonstrates how to utilize SAS Graphical Template Language to create flexible macro tool kit for commonly used graphs in biomarker research. It allows for a consistent production and provides a more efficient way in biomarker data visualization process.

REFERENCES

SAS Institute (2009), *SAS/GRAPH ® 9.2 Graph Template Language User's Guide,* SAS Institute, Inc., Cary, NC

SAS Institute (2008), *SAS/STAT® 9.2 User's Guide,* SAS Institute, Inc., Cary, NC

CONTACT INFORMATION

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

Xiaohui (Gianna) Huang
Gilead Sciences Inc.
333 Lakeside Drive
Foster City, CA, 94404
Work Phone: (650) 577- 6435
Gianna.huang@gilead.com

Jigar Patel
Gilead Sciences Inc.
333 Lakeside Drive
Foster City, CA, 94404
Work Phone: (609) 473-0631
Jigar.Patel@gilead.com