Paper SAS294-2014

Prescription for Visualization: Take One SAS® Graph Template Language Graph before Seeing the Patient

Radhikha Myneni, SAS Institute Inc.; Eric C. Brinsfield, SAS Institute Inc.

ABSTRACT

The steady expansion of electronic health record (EHR) systems over the past decade has increased the use of observational healthcare data for analysis. One of the challenges with EHR data is to combine information from different domains (diagnosis, procedures, drugs, adverse events, labs, quality of life scores, and so on) onto a single timeline to get a longitudinal view of the patient. This enables the physician or researcher to visualize a patient's health profile, thereby revealing anomalies, trends, and responses graphically, thus empowering them to treat more effectively. This paper attempts to provide a composite view of a patient using SAS® Graph Template Language (GTL) to create a profile graph using the following data elements: key event dates, drugs, adverse events, and Quality of Life (QoL) scores. For visualization, the GTL graph uses X and X2 axes for dates, vertical reference lines to represent key dates (for example, when the disease is first diagnosed), horizontal bar plot for duration of drugs taken and adverse events reported, and a series plot at the bottom to show the QoL score.

INTRODUCTION

Data from medical records comes from different domains (diagnosis, procedures, drugs, and so on) and most of the time in different formats. One of the biggest challenges is to combine and correlate this data for secondary use. In addition, this data can be huge, getting into the realms of "Big Data", which poses an additional set of challenges to solve.

The simplest, most basic, and yet powerful exploration for this data is the patient profile graph—one graph per patient, in which data from the most important domains is charted on a timeline in order to view and understand any relationships and dependencies. The key to this exercise is to have the end user, the physician or researcher, identify the list of key data elements, treatments, or events of interest.

Our work was initiated by a development partner who provided us with a set of requirements and patient data for those treated for cancer. The goal was to come up with a profile graph that plots all the necessary details the physician wants to compare over time. In this use case, a chart showing events and relationships is exactly what the physician needed in order to visualize a patient's response to treatment. The oncologist could easily see the treatment regimen with possible side effects, both in terms of adverse events and quality of life measures.

We use SAS® Graph Template Language (GTL) in SAS® 9.3 to create this composite picture, as it provides the capability of creating reusable templates. Base SAS® and SAS® macro capabilities are also used for data preparation and for automating the process of generating the profile graph iteratively for all the patients once the template is finalized.

DATA AND PREPARATION

To create the profile graph, the data elements listed below were chosen as the key players. These elements were selected to show how one graph can be used to understand a patient's outcome in terms of adverse events, QoL (Quality of Life) measures, after receiving a series of chemo drugs, and any possible correlations between them.

- Patient ID, unique patient identifier
- When patient died or his last known visit date
- Cancer diagnosis date, when the cancer was first diagnosed
- Cancer metastasis date, when the cancer metastasized (which is indicative of an advanced stage)
- Chemotherapy treatment
- Adverse events reported
- QoL score, a composite score of various quality of life measures like ability to perform daily activities, psychiatric state, and so on, using surveys. The higher the score, the better their quality of life.

To put it simply, the questions we want to answer are: Are the side-effects of the chemotherapy marginal or severe, and how is the patient coping with life in terms of performing daily activities, mental state, and so on. Combining this information into one graph makes the visualization a powerful tool in the hands of a clinician or researcher.

The first step when working with any data is cleaning and preparing. The dummy data from our partner was well formatted and clean, but we added a few sanity checks around dates. For example, we ensured that end dates, when specified, are the same or greater than start dates, and we dropped entries that did not comply. (Please note that all data used in this presentation is fictitious and therefore not regulated by HIPAA or any other privacy regulations. Patterns, however, are realistic.)

In addition to the actual dates, we calculate differential days as the number of days from an index date. In our case, the index date is the patient's cancer diagnosis date but it could be any point in time that is of interest, like first encounter date, date when chemotherapy was started, and so on. The differential days are computed to show the actual dates alongside the number of days from diagnosis, and they are very helpful when comparing different patients since they are date agnostic. It also provides the ability to hide actual dates when needed, still preserving the visualization and all of its relations.

The chemo drugs are named Drug_A, B, C, D, E, and F. A drug label is created by appending the start and stop dosage and schedule to its name. Care has to be taken such that the label conveys all the necessary details but is not too long, which can sometimes be a tricky task to balance.

NOTE: We highly recommend reading Sanjay Matange's SAS Global Forum 2013 Paper 160-2013 titled "Patient Profile Graphs Using SAS®", available at http://support.sas.com/resources/papers/proceedings13/160-2013.pdf. It provides a good introduction to profile graphs in general and how data needs to be prepped for different chart types. The paper uses SGPLOT, but we use GTL as it provides greater flexibility in customization.

Sample data is shown in Table 1 and Table 2 below; the tables are two parts of the same wide table showing all the columns and data. The data in these tables corresponds to the profile graph shown in Figure 1 in the Results section.

	stdt	endt	evt_lowlbl	ae_y	ae_grp	ae_highcap	rx_y	rx_highcap	qol_y
1	15JUN09	.							.
2	08JUL10	.							.
3	11MAR11	.							60.08
4	14AUG11	.							67.81
5	18SEP11	.							58.93
6	16OCT11	.							53.6
7	18NOV11	.							52.92
8	23DEC11	.							41.12666...
9	06JAN12	.							48.33
10	22JAN12	.							49.86
11	17FEB12	.							54.52
12	31JUL12	.							.
13	18SEP11	17FEB12	AE:FATIGUE/AS...	AE:FATIGUE/AS...	Moderate	FilledArrow			.
14	18SEP11	18SEP11	AE:HAND FOOT...	AE:HAND FOOT...	Moderate				.
15	18SEP11	16OCT11	AE:MUCOSITIS	AE:MUCOSITIS	Moderate				.
16	17FEB12	18MAY12		AE:MUCOSITIS	Mild				.
17	16OCT11	17FEB12	AE:OTHER-COU...	AE:OTHER-COU...	Moderate	FilledArrow			.
18	18MAY12	18MAY12	AE:OTHER-HAIR...	AE:OTHER-HAIR...	Mild				.
19	22JAN12	18MAY12	AE:OTHER-TAS...	AE:OTHER-TAST...	Mild	FilledArrow			.
20	17FEB12	13APR12	AE:OTHER-THR...	AE:OTHER-THR...	Mild				.
21	21AUG11	11DEC11	RX:Drug_B,10 m...				RX:Drug_B,10 m...		.
22	08JUL10	25FEB11	RX:Drug_A,25,25				RX:Drug_A,25,25		.
23	28DEC11	.	RX:Drug_D,600				RX:Drug_D,600	FilledArrow	.

Table 1. Input Data Used for Figure 1 (Part A)

	evt	stday	enday	initdx_stday	metdx_stday	death_stday	lastknwn_stday	ae_stday	ae_enday	rx_stday	rx_enday	qol_stday
1	initdx	0	.	0	.	.	.	.	.	.	.	.
2	metdx	388	.	.	388	.	.	.	.	.	.	.
3	qol	634	.	.	.	.	.	.	.	.	.	634
4	qol	790	.	.	.	.	.	.	.	.	.	790
5	qol	825	.	.	.	.	.	.	.	.	.	825
6	qol	853	.	.	.	.	.	.	.	.	.	853
7	qol	886	.	.	.	.	.	.	.	.	.	886
8	qol	921	.	.	.	.	.	.	.	.	.	921
9	qol	935	.	.	.	.	.	.	.	.	.	935
10	qol	951	.	.	.	.	.	.	.	.	.	951
11	qol	977	.	.	.	.	.	.	.	.	.	977
12	death	1142	.	.	.	1142	.	.	.	.	.	.
13	ae	825	977	.	.	.	.	825	977	.	.	.
14	ae	825	825	.	.	.	.	825	825	.	.	.
15	ae	825	853	.	.	.	.	825	853	.	.	.
16	ae	977	1068	.	.	.	.	977	1068	.	.	.
17	ae	853	977	.	.	.	.	853	977	.	.	.
18	ae	1068	1068	.	.	.	.	1068	1068	.	.	.
19	ae	951	1068	.	.	.	.	951	1068	.	.	.
20	ae	977	1033	.	.	.	.	977	1033	.	.	.
21	rx	797	909	.	.	.	.	.	.	797	909	.
22	rx	388	620	.	.	.	.	.	.	388	620	.
23	rx	926	1142	.	.	.	.	.	.	926	1142	.

Table 2. Input Data Used for Figure 1 (Part B)

As seen in the sample data above, there are common columns that all event-types (rx:drugs, ae:adverse events, qol:quality of life scores) share, like “evt”, which is the type, “stdt” and “endt” for start and end dates, and “evt_lowlbl” for label. Some columns are event-type specific. For example, adverse event has:

- ae_y: label.
- ae_grp: severity. This can be Mild, Moderate, Severe, or Other, which is used to color the bar as green, dark orange, red, or tan respectively.
- ae_highcap: indicator that event is ongoing. This is displayed with a filled arrow at the end of the bar.

Similarly, chemo drugs have rx_y and rx_highcap, and QoL score has qol_y, and so on. Finally, for each date, there is a corresponding column that ends with “day” to enable two types of timelines as described before—one with actual dates and other differential days. The event-type specific columns were created so that GTL can draw different charts for different event-types (HIGHLOW for drugs and adverse events, SERIES for QoL score, and REFLINE for important dates). The common columns aid in lining up all these onto the same timeline.

NOTE: HIGHLOW, SERIES, and REFLINE are the various plot statements supported by SAS Graph Template Language. Refer to SAS documentation about SAS Graph Template Language for additional details.

METHODOLOGY USING GTL

The advantage of using a template like GLT is the ability to generate multiple graphs using one template. Once the effort to design a profile template is complete, it can be reused for all the patients time and again. The ingredients needed to make this work is obviously the template itself, data in a format that the template needs (described in Data and Preparation section), and finally the rendering call to use the template repetitively for every patient. The work is performed using SAS 9.3 with Base SAS and SAS/GRAPH® package capabilities.

When you use GTL, you define the template using PROC TEMPLATE and then pass in the data and template to PROC SGRENDER to draw the chart. The template and the rendering procedure uses the DYNAMIC statement to enable parameter passing between them, so instead of hardcoding labels or column names in the template, you can specify them dynamically when the graph is rendered.

Here is a brief explanation of the template code that follows:

- Starts with variable declarations using the DYNAMIC statement.
- Define colors for adverse event severity using DISCRETEATTRMAP. Mild severity is represented with green, Moderate with dark orange, Severe with red, and Other with tan.
- Begin a lattice layout that divides the graph area into two sections—85% top and 15% bottom.

- Begin overlay layout for the top section and define four REFERENCELINE (for dates: diagnosis, metastasis, last known visit or death) and two HIGHLOW plots (adverse events and chemotherapy drugs). The four SCATTER plots that follow form the basis for creating the timeline on X and X2 axes.
- Begin overlay layout for the bottom section and define a SERIES plot for QoL scores.
- The rest of the statements use COLUMNAXES and SIDEBAR to create legends.

TEMPLATE CODE

```
proc template;
  define statgraph patprofile_template;
  dynamic
    RL1_X RL1_LBL
    RL2_X RL2_LBL
    RL3_X RL3_LBL
    RL4_X RL4_LBL
    HL1_Y HL1_LOW HL1_HIGH HL1_LOW_LBL HL1_GRP
    HL2_Y HL2_LOW HL2_HIGH HL2_LOW_LBL
    SR1_X SR1_Y;

  begingraph;

      discreteattrmap name='SevColorMap';
        value 'Mild'     / lineattrs=(color=green) markerattrs=(color=green)
                           fillattrs=(color=green);
        value 'Moderate' / lineattrs=(color=darkorange)
                           markerattrs=(color=darkorange)
                           fillattrs=(color=darkorange);
        value 'Severe'   / lineattrs=(color=red) markerattrs=(color=red)
                           fillattrs=(color=red);
        value 'Other'    / lineattrs=(color=tan) markerattrs=(color=tan)
                           fillattrs=(color=tan);
      enddiscreteattrmap;
      discreteattrvar attrvar=AttrSeverity var=HL1_GRP attrmap='SevColorMap';

      layout lattice /rows=2 columndatarange=union rowweights=(0.85 0.15);

      layout overlay / border=true yaxisopts=(type=discrete display=none)
                       xaxisopts=(type=linear display=(line ticks tickvalues)
                       offsetmax=0.02 linearopts=(thresholdmin=0 thresholdmax=0))
                       x2axisopts=(type=linear display=(line ticks tickvalues)
                       offsetmax=0.02 linearopts=(thresholdmin=0 thresholdmax=0));

     referenceline x=RL1_X / lineattrs=(color=royalblue pattern=solid thickness=2)
                             name="dxdt" legendlabel=RL1_LBL;
     referenceline x=RL2_X / lineattrs=(color=royalblue pattern=DashDashDot
                                        thickness=2)
                             name="mdxdt" legendlabel=RL2_LBL;
     referenceline x=RL3_X / lineattrs=(color=black pattern=solid thickness=2)
                             name="lastdt" legendlabel=RL3_LBL;
     referenceline x=RL4_X / lineattrs=(color=red pattern=solid thickness=2)
                             name="ddt" legendlabel=RL4_LBL;

     highlowplot y=HL1_Y low=HL1_LOW high=HL1_HIGH / group=AttrSeverity
                 includemissinggroup=false
                 type=bar barwidth=0.6 lowlabel=HL1_LOW_LBL open=HL1_LOW
                 lineattrs=(pattern=solid)
                 highcap=ae_highcap name="aesev";

     highlowplot y=HL2_Y low=HL2_LOW high=HL2_HIGH / type=bar barwidth=0.6
                 lowlabel=HL2_LOW_LBL open=HL2_LOW lineattrs=(pattern=solid)
```

```
                    fillattrs=(color=cxb572b5) highcap=rx_highcap
                    outlineattrs=(thickness=0) name="meds"
                    legendlabel='Drug-Dosage-Freq';

      scatterplot y=evt x=stdt /  xaxis=x2 markerattrs=(size=0);
      scatterplot y=evt x=endt /  xaxis=x2 markerattrs=(size=0);

      *This is needed to draw X axis when ONLY ReferenceLine variables are available;
       scatterplot y=evt x=stday / xaxis=x markerattrs=(size=0);
       scatterplot y=evt x=enday / xaxis=x markerattrs=(size=0);
    endlayout;

    layout overlay / border=true xaxisopts=(type=linear display=(line ticks
                                            tickvalues));
      seriesplot x=SR1_X y=SR1_Y;
    endlayout;

    columnaxes;
      columnaxis / griddisplay=on label='Number of days from Event Of Interest';
    endcolumnaxes;

    sidebar / align=top spacefill=false;
      discretelegend "dxdt" "mdxdt" "lastdt" "ddt" / title="Key Events" valign=top;
    endsidebar;

    sidebar /align=bottom spacefill=false;
      layout gridded / columns=2;
        discretelegend "aesev" / title="AE Severity" across=4;
        discretelegend "meds" / title="Medications";
      endlayout;
    endsidebar;

    endlayout; /* ends lattice layout */

  endgraph;

  end;
run;
```

RENDERING CODE

The rendering of the template is very simple: PROC SGRENDER takes the input data set (&input_data) and the template (patprofile_template) to produce the chart. Be sure to create the template before using it in SGRENDER procedure. The `&patid` (Patient ID) and `&idxdt` (Index date) macro variables are used in the title.

```
ods graphics / reset  width=8in height=6in imagename="PatientProfile_&patid" noscale;
proc sgrender data=&input_data template= patprofile_template;
  dynamic
    RL1_X='initdx_stday'   RL1_LBL="Diagnosis"
    RL2_X='metdx_stday'    RL2_LBL="Metastasis"
    RL3_X='lastknwn_stday' RL3_LBL="Last Visit"
    RL4_X='death_stday'    RL4_LBL="Death"

    HL1_Y='ae_y' HL1_LOW='ae_stday' HL1_HIGH='ae_enday'
    HL1_LOW_LBL='evt_lowlbl' HL1_GRP='ae_grp'
    HL2_Y='rx_y' HL2_LOW='rx_stday' HL2_HIGH='rx_enday'
    HL2_LOW_LBL='evt_lowlbl'

    SR1_X='qol_stday' SR1_Y='qol_y'
  ;
  title "Patient ID = &patid, Index Date = &idxdt";
run;
title;
```

RESULTS

To summarize, the patent profile graph uses various charts in GTL to visualize different types of medical events. The visualization uses vertical reference lines to mark key dates like cancer diagnosis (in blue), when cancer metastasized (dotted blue), and when the patient died (red) or his last known visit (black). A horizontal bar chart shows the duration of chemotherapy drugs and adverse events reported. Finally, a series plot displays the QoL scores. The QoL score is a composite score available in the data that indicates the patient's overall quality of life. The X and X2 axes are the timelines. The X axis has "number of days from cancer diagnosis" and X2 has the actual dates.

We present three samples in Figures 1, 2, and 3, showing patients with an increasing number of events. This shows how the graph is sized and adjusted as the data size increases. Note that the data presented in Tables 1 and 2 (in the Data and Preparation section) corresponds to Figure 1.

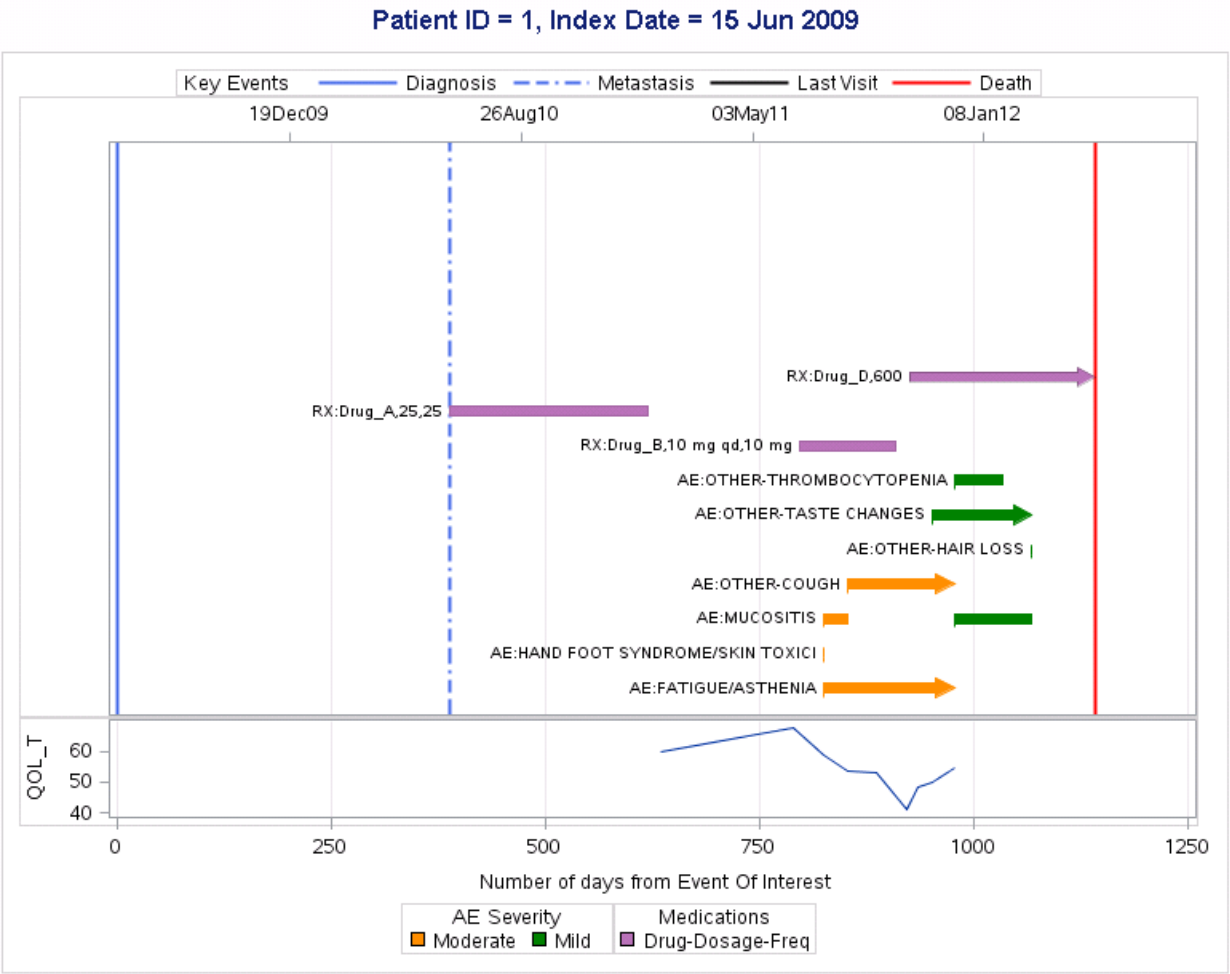

Figure 1. Patient Profile Example 1

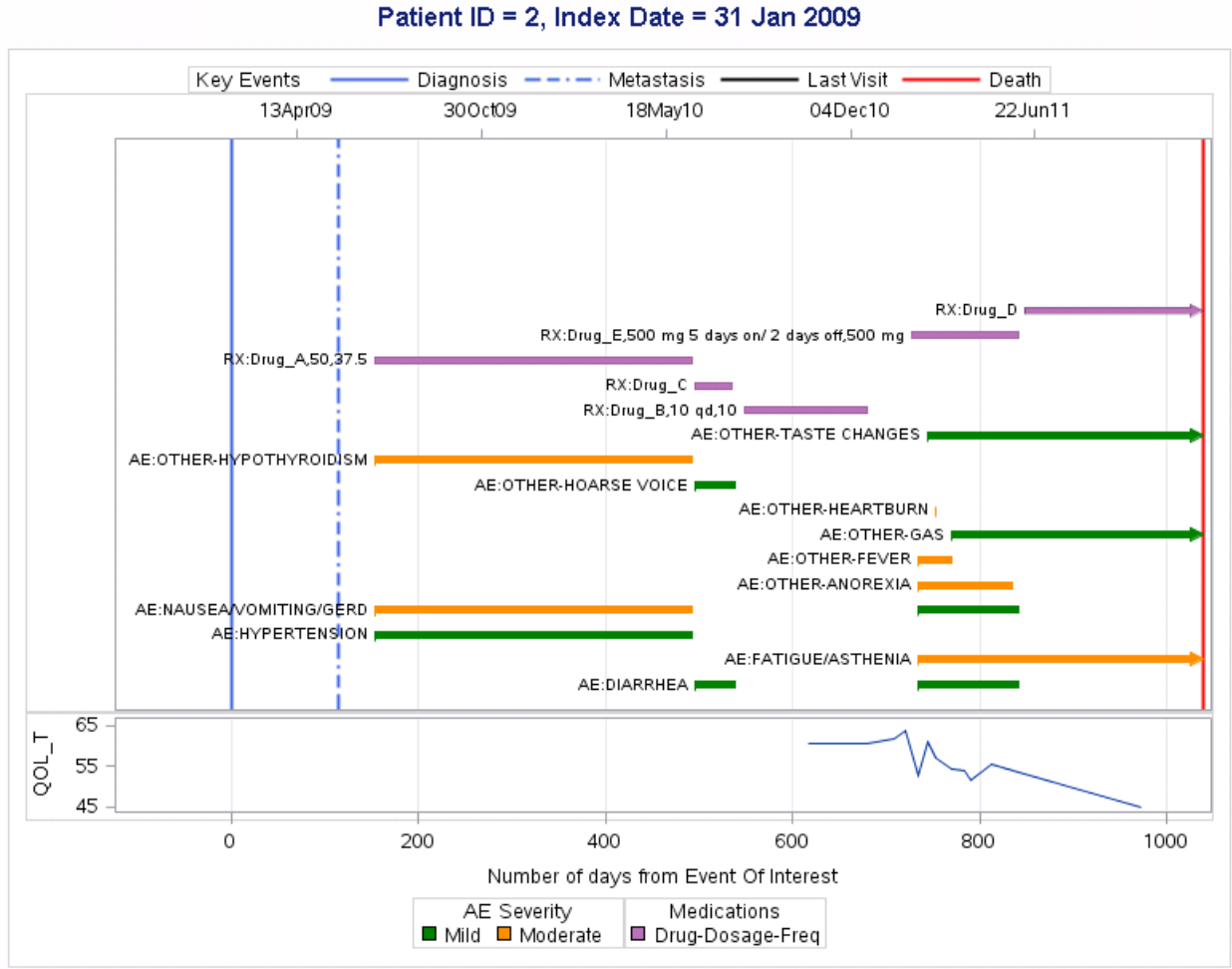

Figure 2. Patient Profile Example 2

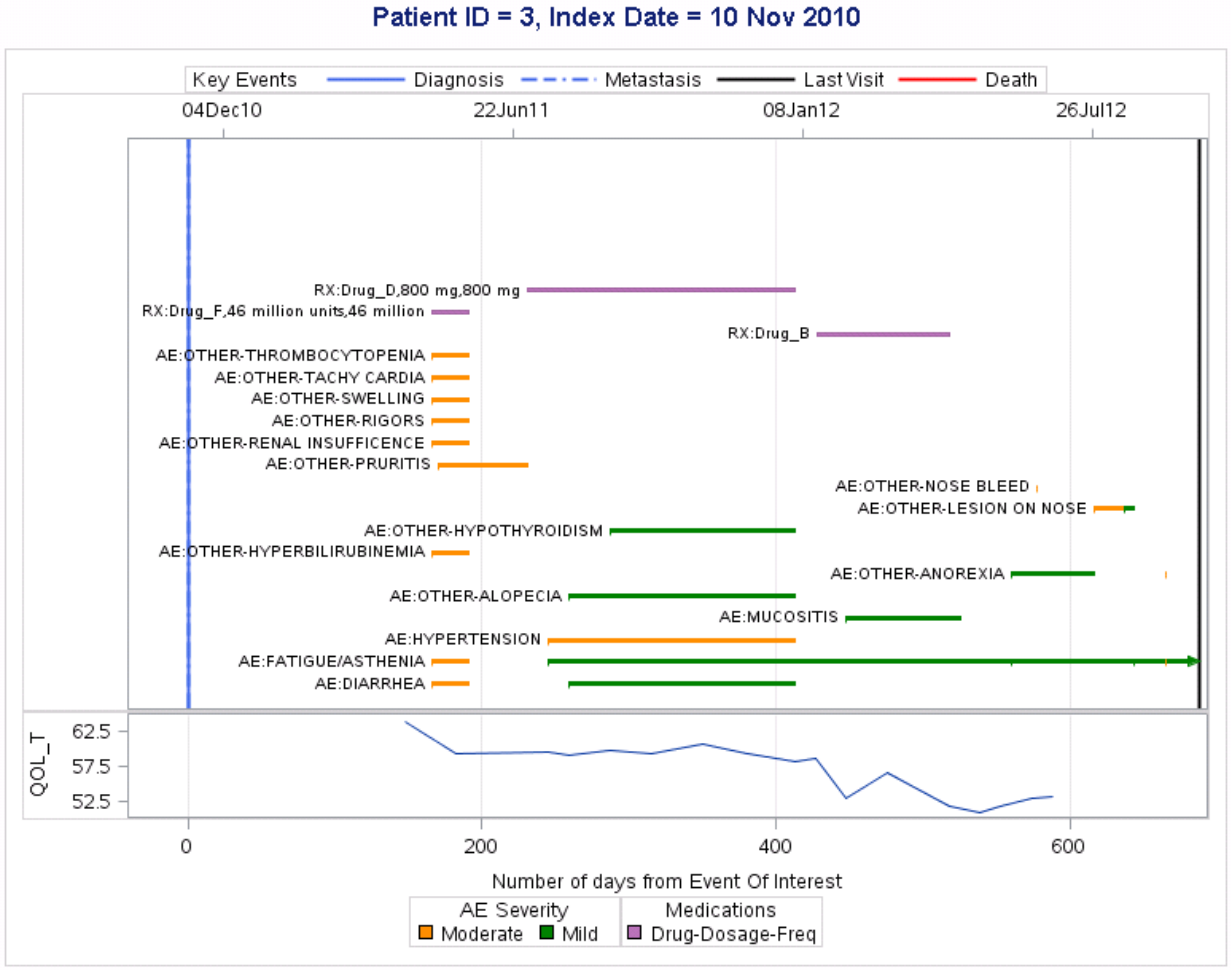

Figure 3. Patient Profile Example 3

The graph provides a very quick and overall picture of the patient's profile in terms of the events shown in it. Because the timeline lines up the events, it is very easy for a clinician to see any potential relationships between drugs, adverse events, and patient's quality of life.

In Figure 3, the patient's diagnosis and metastasis dates overlap. This might indicate that the cancer was diagnosed at a later stage or that the patient was referred to this physician or facility and the true diagnosis date is not known or available. There is a long list of adverse events reported when the patient was on Drug_F, but when he was subsequently moved to Drug_D, most of them are no longer reported, though some (AE:FATIGUE, AE:DIARRHEA) are milder and other new affects were added (AE:HYPERTENSION). The color of the adverse event represents severity reported by the patient; orange events are more severe than green ones. The value of QoL score in and by itself is not useful, but the direction of its trend over time is.

Graphs of this sort could be useful to clinicians at the bedside as well as to researchers. Using SAS macros, the process of generating this profile graph for all patients was automated by prepping the data and calling SGRENDER for each patient iteratively and writing the result using SAS Output Delivery System (ODS) into a PDF.

Lastly, since observational data can be overwhelming even at the patient level and due to the limitations in chart area, we highly recommend adding a calling interface that allows physicians (or users) to filter and select a limited list of events to be included in the graph. This prevents the graph from getting overcrowded with unnecessary details and allows the clinician to target specific treatments and concerns.

CONCLUSION

Having the capability to build and customize a visualization using templates is a powerful feature available in SAS/GRAPH software. The paper utilizes SAS Graph Template Language to design the template, and SAS macros to programmatically render the template for all patients. Since this process is automated, it can be re-run at a later point in time to get an updated picture of the patient profile. Or, the template can be updated to add a new chart or to modify a label or coloring scheme with minimal effort.

Though the visualizations are shown for cancer data, there is no reason why they cannot be reused or extended to observational data in EHR or other sources, as long as the data is prepped accordingly. We believe this methodology of designing visualizations using GTL in conjunction with other Base SAS capabilities to automate the process is a simple yet powerful tool to have in your toolbox for observational data analysis.

REFERENCES

- Matange, Sanjay. 2013. "Patient Profile Graphs Using SAS®". Proceedings of the SAS Global Forum 2012 Conference. Cary, NC: SAS Institute Inc. Available at http://support.sas.com/resources/papers/proceedings13/160-2013.pdf.
- Kuhfeld, Warren F. 2010. Statistical Graphics in SAS®: An Introduction to the Graph Template Language and the Statistical Graphics Procedures. Cary, NC: SAS Institute Inc.
- SAS Institute Inc. 2011. SAS® 9.3 Graph Template Language: User's Guide. Cary, NC: SAS Institute Inc.

ACKNOWLEDGMENTS

We would like to thank Amy Abernethy, M.D., Brad Hirsch, M.D., and their team at the Duke Center for Learning Healthcare for sharing their data and use cases on this project. Their feedback was very valuable in understanding how physicians can use the patient profile graph and how its capabilities can be extended in aiding them treat patients effectively, thus enabling us to iteratively design and deliver the best solution.

CONTACT INFORMATION

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

Radhikha Myneni
SAS Institute Inc.
100 SAS Campus Drive
Cary, NC 27513
Work Phone: (919) 531 3736
Fax: (919) 677 4444
E-mail: radhikha.myneni@sas.com

Eric C. Brinsfield
SAS Institute Inc.
100 SAS Campus Drive
Cary, NC 27513
Work Phone: (919) 531 0213
Fax: (919) 677 4444
E-mail: eric.brinsfield@sas.com