PharmaSUG 2017 - Paper DV07

Using Animated Graphics to Show PKPD Relationships in SAS 9.4®

Andrew McCarthy, Eli Lilly and Co., Surrey, United Kingdom

ABSTRACT

The relationship between pharmacodynamics and pharmacokinetics is often complex and it can be difficult to conceptualize how drug induced changes relate to drug levels. Animated, multivariate visualisations can provide insight into these relationships. Using the features of SAS 9.4®, spectacular animated plots can be produced from SAS Graph Template Language (GTL) procedures simply with the use of a by statement. This paper will demonstrate how to produce a graphic that shows multivariate changes in electroencephalographic (EEG) spectral activity and the relationship these have with drug plasma exposure.

INTRODUCTION

Pharmacokinetics provides a measure of the amount of drug in a system over time whereas pharmacodynamics provides a measure of the impact of the drug on a system. The electroencephalogram measures brain activity as a change in voltage across two electrodes placed on the scalp. The voltage changes over time can be represented in the frequency domain by converting the signal with a Fourier transform. The EEG is a useful tool for assessing the pharmacodynamics of brain penetrant drugs. Understanding the PKPD relationship with multiple variables can be facilitated by animations. We will discuss the use of the graph template language (GTL) to create a dashboard that displays both the PK and PD data, which we then animate. Various challenges remain that need to be addressed for a good animated graphic. For example, measurements may be made on subjects at different intervals and rarely at a rate for nice animations. It is therefore useful to interpolate between data points over the course of the animation. In the present analysis the dataset preparation needed to use the expand procedure is covered. In summary, SAS 9.4 allows us to create effective visualisations of multivariate drug effects over time with minimal coding.

PHARMACOKINETICS

The time course of the drug level is best plotted using scatter and series statements to produce a graph that visually represents the clearance of the drug from the system following treatment (at time 0). From this graph the exponential decay pattern of clearance is evident. The code below was used to plot the PK data:

```
PROC SGPLOT data= dataForPkPlot;
        series x= time y =pkMean ;
        scatter x= time y =pkMean/
        yerrorlower=upper_pk yerrorupper=lower_pk;
run;
```

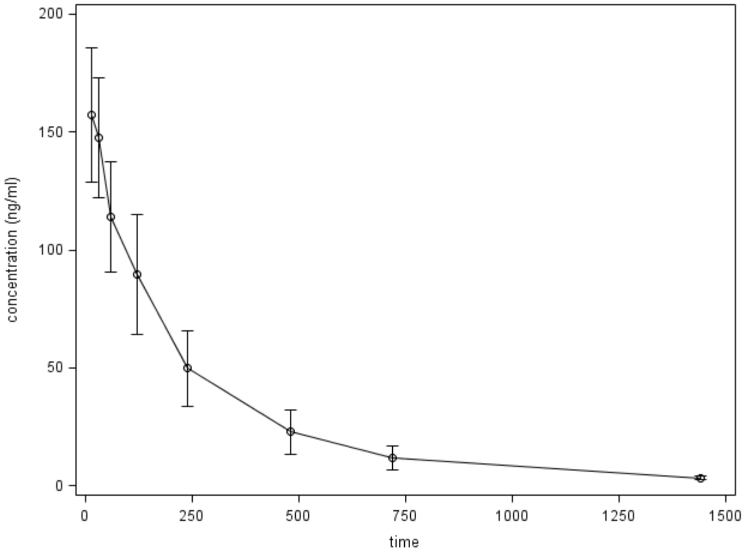

Figure 1: The pharmacokinetic profile of the drug plotted using series and scatter statements to show how mean ± standard error of the mean (SEM) drug plasma concentration (ng/ml) decreases over time (minutes) from treatment in 8 subjects.

SPECTRAL PROFILE OF THE ELECTROENCEPHALOGRAM

The spectral profile of the EEG represents the activity of the brain at different frequencies. Different frequency ranges are believed to be related to specific cognitive processes and can respond to pharmacological agents independently. It is quite possible that a drug may boost one frequency at a specific concentration, whilst another is inhibited. This leads to a complex multivariate pattern that varies with time. A change in the EEG relative to the period immediately prior to drug treatment can tell us which frequencies have been altered by the presence of the drug. The code below was used to plot the PD data:

```
PROC SGPLOT data = SpectralPlotData;
        series x=freq y=differenceMean/ group= treatment;
        band x= freq upper=upperlim lower=lowerlim /group= treatment
            transparency=0.5;
        yaxis label = "spectral power relative to baseline";
        refline 100 / lineattrs=(thickness=2 color=black);
        xaxis values = (0 to 200 by 20);
run;
```

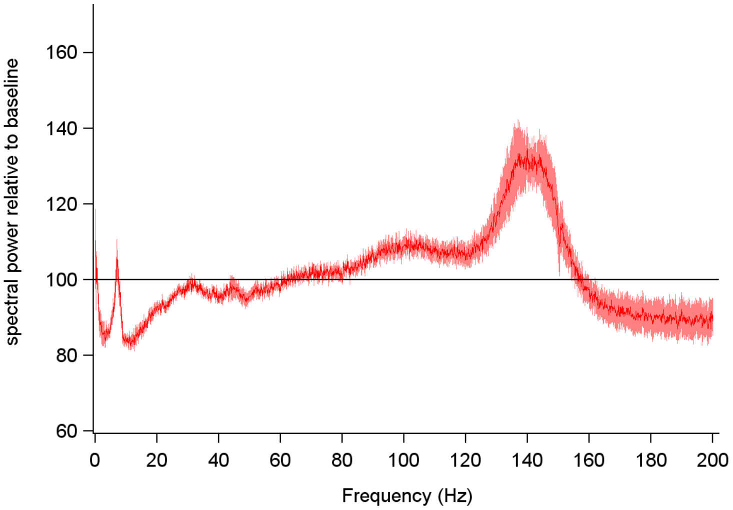

Figure 2: The EEG spectral effects of the drug 60 minutes after treatment plotted using the series and band statements to show mean ± standard error of the mean (SEM) spectral power changes relative to baseline across the frequency range.

A DASHBOARD USING THE GRAPH TEMPLATE LANGUAGE

Creating a dashboard that displays both EEG spectral data and the PK data together can be done using the Graph Template Language (GTL). Comparing the effects between time points requires the position in time to be clearly marked. This was achieved using a reference line on the PK plot and a textdisplay in the scatterplot of the EEG data. Importantly the positions of the axes and labels must remain fixed throughout the animation and these were determined empirically. The following code creates the composite plot containing both PK and PD data:

```
PROC TEMPLATE ;
define statgraph stdplot;
  begingraph / designwidth =1400 px designheight =1200 px;
    layout overlay / width =1200 px height =1000 px
          xaxisopts =( label ="Frequency (Hz)" offsetmin = 0.005 offsetmax = 0.005
            linearopts =( viewmin =0 viewmax =200
            tickvaluesequence =( start =0 end =220 increment =20) ))
          yaxisopts =( label ="spectral power ( amplitude )" type = linear
            linearopts =( viewmin =60 viewmax =200
            tickvaluesequence =( start =60 end =180 increment =20) ));
          entry "Spectral profile"/ valign =top;
          seriesplot x = frequency y = differenceMean /
            group = treatment;
          bandplot x = frequency limitupper = upperlim limitlower = lowerlim /
            group = treatment datatransparency =0.5;
          referenceline y =100 / lineattrs =( thickness =2 color = black );
          scatterplot x=x_labelpos y =y_labelpos /
            markercharacter = textdisplay markercharacterattrs =( size =15);

      layout gridded / border = false autoalign =( topleft ) width =400 px height
         =300 px;
        layout overlay /
              xaxisopts =( label = "time" linearopts =( viewmin =0 viewmax =1500
        tickvaluesequence =( start =0 end =1500 increment =250) )
        display =( label line ticks tickvalues ))
       yaxisopts =( label ="ng/ml" display =( label line ticks tickvalues
         ));
       entry "PK"/ valign =top;
       seriesplot x = time y = pk_mean;
       scatterplot x= time y =pk_mean /
        yerrorlower =pk_upper yerrorupper =pk_lower;
       referenceline X = vline_position / lineattrs =( thickness =2 color
         =red);
    endlayout;
   endlayout;

  endlayout;
 endgraph;
end ;
run;
PROC SGRENDER data = combined_plot template = stdplot;
   by minutes ;
run;
```

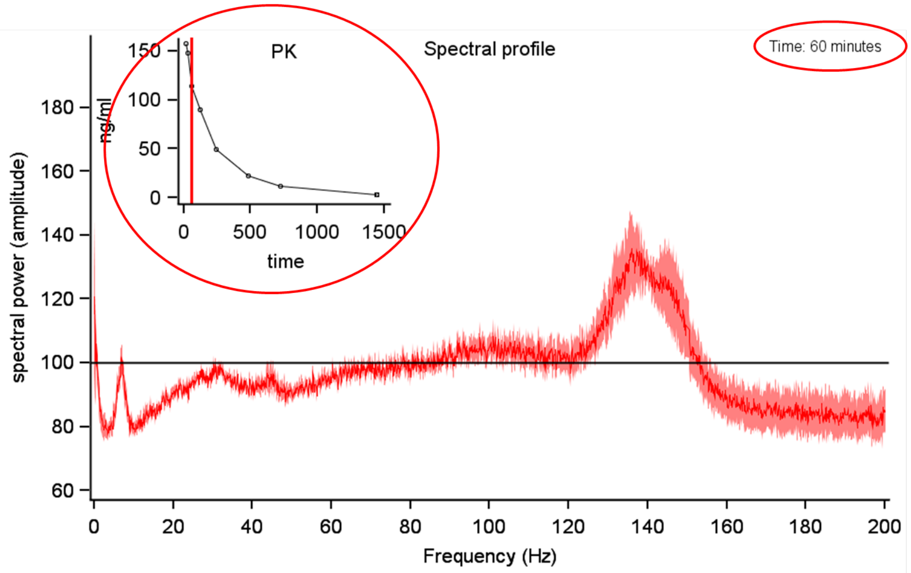

Figure 3: The EEG spectral effects of the drug 60 minutes after treatment plotted using the series and band statements to show mean ± standard error of the mean (SEM) spectral power changes relative to baseline across the frequency range.

INTERPOLATION OF THE TIME SERIES DATA

Interpolation is necessary in order to create enough frames to make a smooth animation. However, it is important that the interpolation method does not misrepresent the underlying data. Linear interpolation is the simplest option but splines make the animations smoother and more natural. The number of interpolated data points needed for the animated graphic depends on the length of the animation and the frame rate. In this case we have 8 hours of PD data and we would like a frame for each minute. In this case we will have 480 frames, which we will display at 60 frames per second giving an overall animation length of 8 seconds. The PK data is fixed but the position of the line needs to be moved in each of the 480 frames. The PD data however needs to be interpolated and to do this empty rows were created that will later be filled by the expand procedure. Additional data was generated using the following macro:

```
DATA expand_pk;
      set Doi_pk_ Data;
      hours = .;
      minutes = .;
run;
%macro do_pk_interpolation;
%do i=1 %to 479;
  %put &i;
  DATA pk_Data_Iterate;
        set work.Doi_pk_Data;
        minutes = 1* input(&i, best12 .);
        hours = round(minutes/60);
  run;
  PROC APPEND data =pk_Data_Iterate base = expand_pk;
  run;
%end;
%mend;
%do_pk_interpolation;
```

The PD interpolation loop creates empty rows for each minute between the hourly data points in the following way:

Table 1: Interpolation dataset structure for Proc Expand procedure

Timepoint	Frequency (Hz)	Power
60	1	100
61	1	.
62	1	.
...	...	...
120	1	160

Additional data was generated using the following macro:

```
%macro do_pd_interpolation;
%do i=1 %to 59;
  %put &i;
  DATA interpolate_Data_Iterate;
        set interpolate_Data;
        timepoint = timepoint + (1* input (&i, best12 .));
        differencevalue_mean = .;
        upperlim_pct = .;
        lowerlim_pct = .;
  run;
  PROC APPEND data = interpolate_Data_Iterate base = interpolated_Data_pd;
  run;
%end;
%mend;
%do_pd_interpolation;
```

Proc expand was then used to fill the missing values and create the dataset used for the animation. Interpolation was performed for the mean and standard errors. Individual subjects could be interpolated and error bars calculated from these but it is significantly more computationally intensive. The following code was used to interpolate using the expand procedure:

```
PROC EXPAND data = interpolated_Data_pd out= LinInterp_sigma;
by frequency;
      convert differencevalue_mean = differencevalue_mean2 / method = spline;
      id timepoint;
run;
```

ANIMATED OUTPUT

In SAS 9.4 an animated GIF can be created using the animation OPTIONS and the PRINTER statement as previously shown (http://www.pharmasug.org/proceedings/china2016/DV/PharmaSUG-China-2016-DV10.pdf). These statements can be wrapped around any sgplot or GTL with a by statement to produce an animated graphic. The following code creates an animation that loops through time showing the PKPD relationship:

```
options nobyline papersize =(’8in’, ’4.8in’) printerpath = gif animation = start
animduration =0.033 animloop =yes noanimoverlay;
ods printer file ="C:\AMcCarthy\Animated_plots\plot1.gif";
ods graphics / width =8 in height =4.8 in imagefmt =gif;
PROC SGRENDER data = combined_plot template = stdplot;
      by minutes;
run;
options printerpath =gif animation = stop;
ods printer close;
```

The end result is an animated dashboard showing how EEG changes occur in response to drug levels over time. From this we can conclude that the drug used strongly decreases EEG spectral power at lower frequencies (i.e. below 80 Hz) whilst increasing the EEG spectral power above 80Hz (and in particular between 80-120Hz). This EEG activity profile is common among psychoactive substances and may act as a biomarker of hallucinogenic compounds present within the blood plasma.

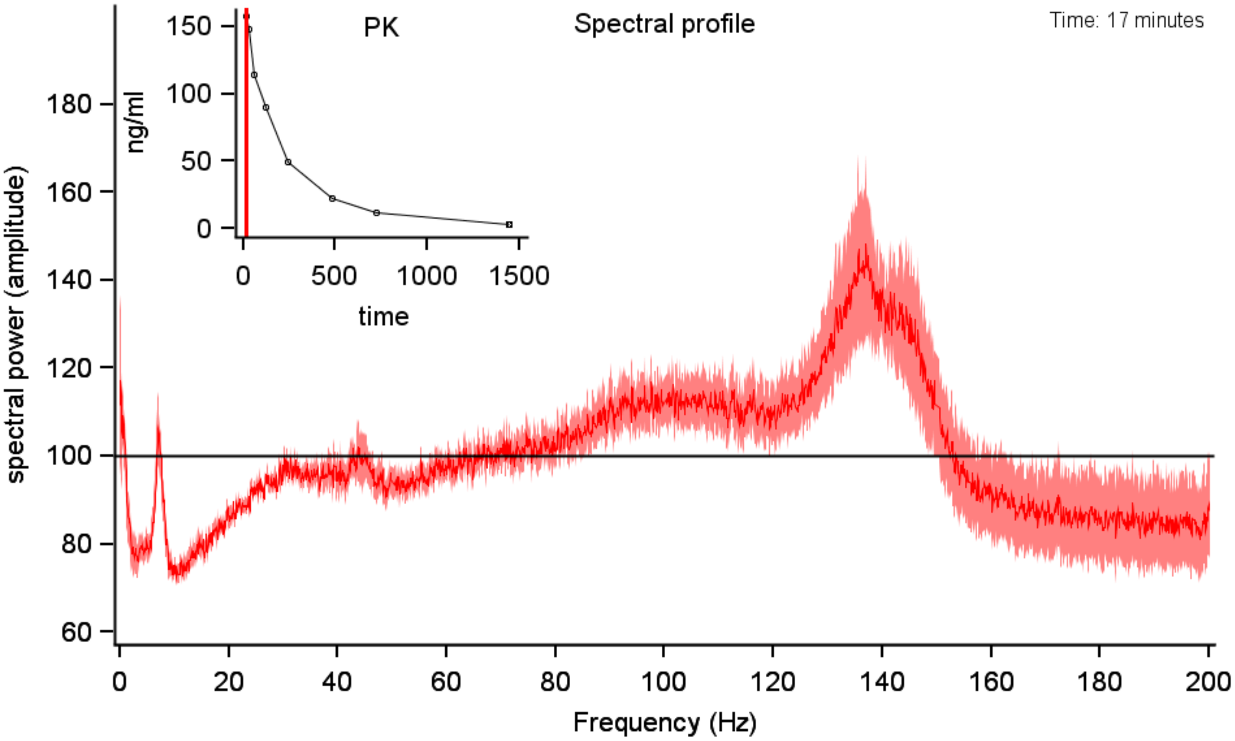

Figure 4: The EEG spectral effects of the drug relative to baseline can be animated over time using the by statement in the sgplot procedure.It is possible to animate the graph within this PDF using the buttons below the image.

CONCLUSIONS

Animated graphics can be made in SAS 9.4 simply using the animate options and the by statement in the plotting procedure. Time was interpolated between the data points to create a smooth animation using the expand procedure. Finally, multiple variables were displayed simultaneously by embedding two graphs with the GTL procedure. The result helps improve our understanding of the influence a compound has on different frequencies of brain activity.

REFERENCES

Matange, S. (2013). Getting Started with the Graph Template Language in SAS: Examples, Tips, and Techniques for Creating Custom Graphs. Cary, NC: SAS Institute Inc.

Harris, K. (2016). Animate Your Safety Data. PharmaSUG China – Paper 83.

ACKNOWLEDGEMENTS

I would like to thank Eli Lilly and Co., Kriss Harris, Claire Brittain and Keith Wafford.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:
Andrew McCarthy
McCarthy_Andrew_Peter@lilly.com