Paper 57-2018

Use of SAS® Graph Template Language (GTL) and HIGHLOWPLOT to Create a Next-Generation Sequencing Multi-Gene Panel Grid

John P. Bennett, Genomic Health, Inc., Redwood City, CA.

ABSTRACT

A graphical display of the results of a multi-gene panel is an effective means for viewing and understanding the number, types, and clustering of detected genomic alterations. One potential method for creating this display in SAS involves use of the HIGHLOWPLOT statement. This statement can be used to create a color-coded grid to display gene panel results for a patient cohort. When this technique is used in combination with other graphical statements, attributes, and definitions, it can create a data-rich display that is both eye-catching and useful as a multi-dimensional summary of gene panel results.

This paper contains a stepwise set of instructions for using the HIGHLOWPLOT statement to create a grid, using as an example the results of a next-generation sequencing (NGS) gene panel assay for patients with non-small cell lung cancer (Eberhard 2018). The steps involve use of the SAS Graph Template Language (GTL) and the SGRENDER procedure using SAS 9.4. The HIGHLOWPLOT statement does the heavy lifting in this example, but additional techniques are used to increase the utility of the display, such as a layout lattice, attribute maps, additional scatterplots, and discrete legends.

INTRODUCTION

As use of NGS increases, there is a need to summarize multi-gene panel results in a single graphic that is both intuitive and informative. One method has been to use combinations of colors in a grid to create a heat map, as demonstrated in prior analyses (Broom 2017; Lee 2017; Spina 2016). This paper describes the method used to create a multi-gene panel grid with SAS GTL.

The main advantage of displaying a multi-colored grid is to find patterns in the data that might not be evident in a tabular display. For example, one could find that most or all variants of a certain type are mutually exclusive of another type. There might also be unexpected combinations of data that would prompt further investigation. To ensure the data are presented in a way that help with finding patterns, it is important to sort the data according to the groupings we would like to see. In order to visualize the difference between grouped and ungrouped data, this paper contains graphs that display both.

The HIGHLOWPLOT statement creates bars or lines that connect a minimum and maximum value by the values of a categorical variable. To create a grid as rows of bars, this example uses the HIGHLOWPLOT statement with TYPE = BAR and no caps specified. The subject's identification (ID) number is used to place the bar on the x-axis, and the same relative minimum and maximum values are applied to every bar, which displays each bar as a rectangle of the exact same size. The categorical variable in this example is the clinical diagnosis subtype and list of genes, which places the bars into separate rows on the y axis.

METHOD

DESCRIPTION OF THE DATASET

The dataset contains a total of 72 patients with stage IV non-small cell lung cancer (NSCLC) who had at least one reported variant in an NGS panel. The NGS panel was performed using circulating tumor DNA isolated from blood plasma. The first set of results from this panel were presented in an abstract and poster by Eberhard et al. at the European Lung Cancer Congress in April 2018 (Eberhard 2018). One of the graphs on this poster was a representation of all the clinically relevant genetic variants reported by the NGS panel. A modified version of this graph is presented in this paper.

There are 5 variables used to create this graph:

1. Patient: An ID number for each patient included in the graph. IDs are numbered 1 through 72, which will be used to place data in 72 columns in the graph.
2. rownum: The row number in which the observation will appear in the graph. The first row (rownum=0) contains information about the NSCLC subtype, followed by 12 rows that display the genes in which each variant is found (1=*EGFR*, 2=*KRAS*, 3=*PIK3CA*, etc.).
3. VariantType1: Description of the variant type, which will be color-coded in the final graph. Variants are respectively numbered 1 through 4 as single nucleotide variants (SNVs), copy number variants (CNVs), translocations, and insertions/deletions/substitutions (indels). Variants 50 and 51 are the NSCLC clinical diagnosis subtype (squamous/nonsquamous). Variant 99 is used when no variant was reported for that gene.
4. VariantType2: Allows for instances where a second variant type is reported for a single gene in the same patient. This field uses the same codes as VariantType1, but will only include SNVs, CNVs, translocations, and indels. A null value means that a second variant was not found for this gene.
5. germline: Variants that have an allelic fraction ≥ 40% are considered germline (i.e., inherited). Germline variants will be denoted in the graph using a special symbol. If a variant is germline, it will have the same value as rownum. Otherwise, it will be null.

Patient	rownum	VariantType1	VariantType2	germline
4	0	50	.	.
4	1	1	4	.
4	2	99	.	.
4	3	99	.	.
4	4	4	.	.
4	5	99	.	.
4	6	99	.	.
4	7	99	.	.
4	8	4	.	.
4	9	99	.	.
4	10	99	.	.
4	11	99	.	.
4	12	99	.	.
48	0	50	.	.
48	1	99	.	.
48	2	99	.	.
48	3	99	.	.
48	4	1	.	4

Output 1. Output from a PROC PRINT statement for 18 rows of the dataset.

GTL, PROC TEMPLATE, AND THE HIGHLOWPLOT STATEMENT

The graph created for this paper was done using SAS GTL, which involves two steps: the TEMPLATE procedure, which defines the structure of the graph, followed by the SGRENDER procedure, which creates the graph (Matange 2013):

```
proc template;
  define statgraph variantgrid;
   begingraph;
     layout lattice / border=false;
      layout overlay /
        yaxisopts=(display=(tickvalues) reverse=true
         tickvalueattrs=(style=italic)
         linearopts=(tickvaluelist=(0 1 2 3 4 5 6 7 8 9 10 11 12)))
        xaxisopts=(label='Patients' display=(label)
         labelattrs=(size=9));
        highlowplot y=rownum high=eval(Patient+0.5)
         low=eval(Patient-0.5) / type=bar;
      endlayout;
     endlayout;
   endgraph;
  end;
run;

proc sgrender data=grid template=variantgrid;
  where VariantType1 < 99;
run;
```

In the source code above, a TEMPLATE procedure defines a STATGRAPH called `variantgrid`. In the subsequent SGRENDER procedure, the `variantgrid` graph is created.

The HIGHLOWPLOT statement uses the TYPE = BAR option, which produces a bar with a width that is specified with the LOW and HIGH options. In this case, the patient ID (a whole number between 1 and 72) is used, minus 0.5 for the LOW and plus 0.5 for the HIGH. This combination will create a consistent rectangular shape for each combination of patient and gene in the graph. The other elements in this code are documented in the SAS GTL User Guide (SAS Institute Inc. 2016). The result of the preceding source code is displayed in Figure 1. This example uses a WHERE statement to remove the results where a variant was not reported.

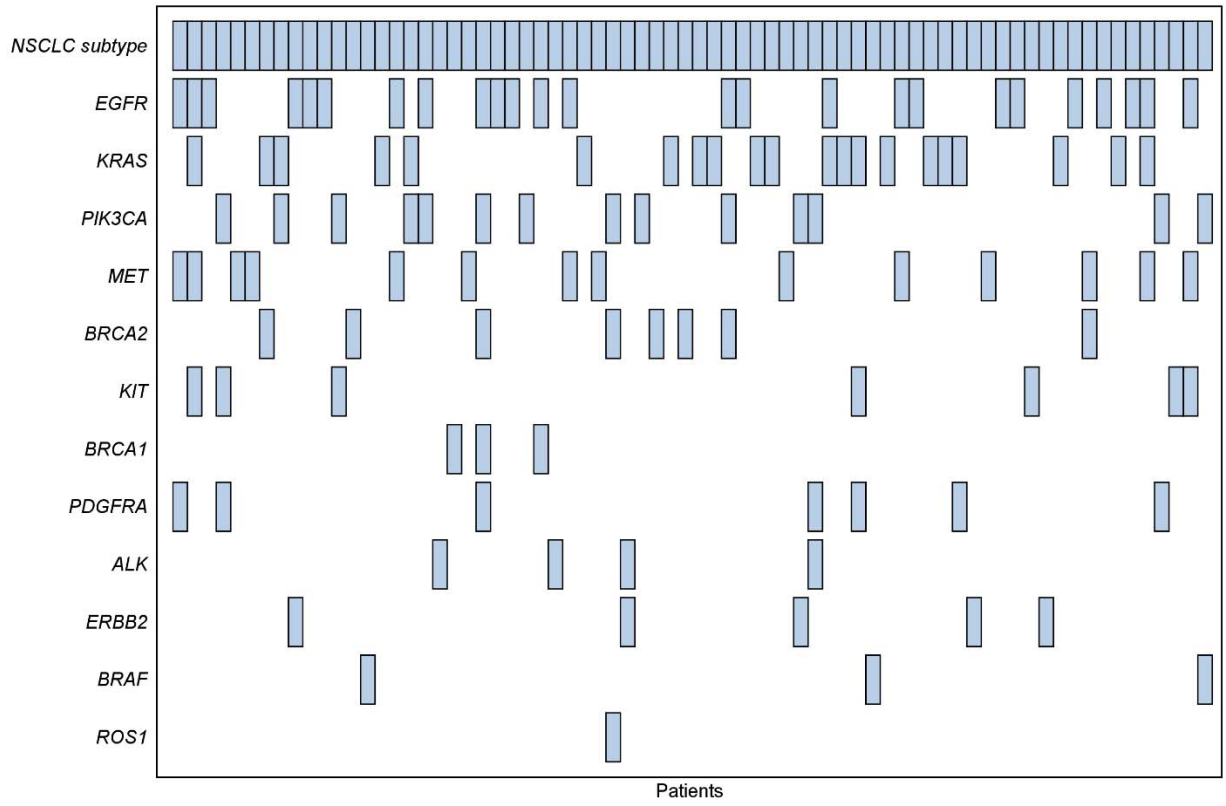

Figure 1. Basic HIGHLOWPLOT without additional elements.

ATTRIBUTE MAPS FOR ADDITIONAL COLOR

Color coding can be used to distinguish the different variant types that were reported for each gene. In a GTL STATGRAPH, the DISCRETEATTRMAP option can be used for this purpose:

```
proc template;
  define statgraph variantgrid;
   begingraph;
     discreteattrmap name='vt';
     value 'SNV' / fillattrs=(color=cx00B0F0) lineattrs=(color=cxFFFFFF);
     value 'Indel' / fillattrs=(color=cxFF0000) lineattrs=(color=cxFFFFFF);
     value 'Translocation' / fillattrs=(color=cxFFD700)
      lineattrs=(color=cxFFFFFF);
     value 'CNV' / fillattrs=(color=cx87BF62) lineattrs=(color=cxFFFFFF);
     value 'Squamous' / fillattrs=(color=cx000080)
      lineattrs=(color=cxFFFFFF);
     value 'Non-squamous' / fillattrs=(color=cx808080)
      lineattrs=(color=cxFFFFFF);
     value 'None' / fillattrs=(color=cxDEDDED) lineattrs=(color=cxFFFFFF);
     enddiscreteattrmap;
     discreteattrvar attrvar=vartyp1 var=VariantType1 attrmap='vt';
     layout lattice / border=false;
      layout overlay /
        yaxisopts=(display=(tickvalues) reverse=true
```

```
          tickvalueattrs=(style=italic)
          linearopts=(tickvaluelist=(0 1 2 3 4 5 6 7 8 9 10 11 12)))
        xaxisopts=(label='Patients' display=(label)
          labelattrs=(size=9));
        highlowplot y=rownum high=eval(Patient+.5)
          low=eval(Patient-.5) / type=bar group=vartyp1;
      endlayout;
    endlayout;
  endgraph;
 end;
run;

proc sgrender data=grid template=variantgrid;
run;
```

In the source code above, a discrete attribute map `vt` is defined with separate values for each variant type and NSCLC subtype. Each VALUE is defined by a text string in single quotes, which are the formatted values associated with `VariantType1`. The `vt` discrete attribute map is connected to the `VariantType1` variable in the DISCRETEATTRVAR statement. The string defined in ATTRVAR is used again in GROUP in the HIGHLOWPLOT statement, which indicates that the colors defined in the discrete attribute map `vt` should be used in the high low plot.

The colors for each gene variant and NSCLC subtype are defined with the FILLATTRS and LINEATTRS statements. Each variant and subtype are given a distinct color in the FILLATTRS statement. The line surrounding each variant is made invisible by matching its color in the LINEATTRS statement to the background color. Figure 2 is the output from the source code above.

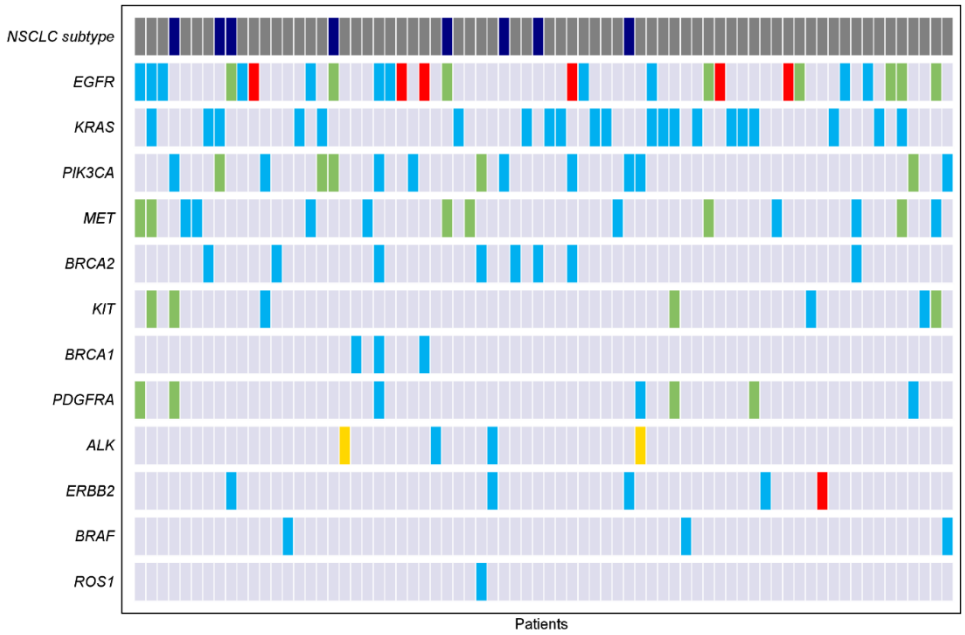

Figure 2. HIGHLOWPLOT with color added for NSCLC subtype and variant type.

SORTING

By assigning different colors to each element in the graph, Figure 2 clearly shows the distribution of the number and types of NSCLC subtypes, the genes, and their variants. However, it is difficult to determine whether there are any relationships between these genes and their variants, or between the NSCLC subtypes, the genes, and the variant types. In order to make it easier to see these relationships, the graph can be grouped by NSCLC subtype, genes, and variant types. This grouping is accomplished by sorting the dataset.

The dataset is sorted first by clinical diagnosis subtype, so that all patients with non-squamous NSCLC appear as a group on the left side of the graph. It is then sorted by each gene, then by the variant types, so they are grouped together as much as possible. After the sorting is complete, the patients are re-numbered in order using the patient ID variable `Patient`. SAS GTL then displays the patients in this new order by `Patient` from 1 to 72 using HIGHLOWPLOT.

ADDITIONAL GRAPHICAL ELEMENTS

Additional GTL statements are used to put the finishing touches on the graph. First, several LEGENDITEM statements are added to specify exactly what appears in the legend:

```
legenditem type=marker name="nonsq_legend" / markerattrs=(color=cx808080
  symbol=squarefilled size=11) label="Non-squamous";
legenditem type=marker name="sq_legend" / markerattrs=(color=cx000080
  symbol=squarefilled size=11) label="Squamous";
legenditem type=marker name="snv_legend" / markerattrs=(color=cx00B0F0
  symbol=squarefilled size=11) label="SNV";
legenditem type=marker name="indel_legend" / markerattrs=(color=cxFF0000
  symbol=squarefilled size=11) label="Indel";
legenditem type=marker name="trans_legend" / markerattrs=(color=cxFFD700
  symbol=squarefilled size=11) label="Translocation";
legenditem type=marker name="cnv_legend" / markerattrs=(color=cx87BF62
  symbol=squarefilled size=11) label="CNV";
legenditem type=marker name="gl_legend" / markerattrs=(symbol=asterisk
  size=8) label="Germline";
```

Note that a legend can be created automatically in GTL by using the NAME option with one of the graphing statements (in this example it would have been the HIGHLOWPLOT statement). However, specifying individual legend items allows the programmer the most flexibility to determine what is shown in the legend. In this case, the variant type ‘None’ is not to be included in the legend, so the individual legend items are specified.

Two additional graphical elements are included in the final graph: a second variant type for the genes that had them, and a symbol for germline variants. A new discrete attribute map `vt2` defines the second variant type, using the same color scheme as the original set:

```
discreteattrmap name='vt2';
value 'CNV' / markerattrs=(color=cx87BF62 size=7 symbol=squarefilled);
value 'Translocation' / markerattrs=(color=cxFFD700 size=7 symbol=squarefilled);
value '.' / markerattrs=(size=0);
enddiscreteattrmap;
```

The second variant types are placed on the graph using two SCATTERPLOT statements that each create two square markers, which are overlaid on the bottom half of the high low plot elements by adding 0.13 and 0.32 to the row numbers (the numbers added to `rownum` were determined via trial and error):

```
scatterplot y=eval(rownum+.13) x=Patient / group=vartyp2;
scatterplot y=eval(rownum+.32) x=Patient / group=vartyp2;
```

The germline variants are also added to the graph using a SCATTERPLOT statement, this time using an asterisk as the symbol:

```
scatterplot y=germline x=Patient / markerattrs=(symbol=asterisk size=4);
```

To place the legend to the right of the graph, update the original LAYOUT LATTICE statement to include 2 columns:

```
layout lattice / columns=2 columnweights=(0.8 0.2) border=false;
```

Next, add an additional LAYOUT LATTICE with 2 rows. Two rows are specified to create separation between the legend items for the NSCLC subtypes and the variant types:

```
layout lattice / rows=2 rowweights=(0.1 0.9) border=false;
  layout overlay;
   discretelegend 'nonsq_legend' 'sq_legend' / down=2
     border=false halign=left valign=center;
  endlayout;
  layout overlay;
   discretelegend 'snv_legend' 'indel_legend' 'trans_legend'
     'cnv_legend' 'gl_legend' / down=5 border=false halign=left
     valign=top;
  endlayout;
endlayout;
```

Finally, use the options available for BORDERATTRS and WALLDISPLAY in the LAYOUT OVERLAY statement to remove the border around the graph:

```
layout overlay / borderattrs=(color=white) walldisplay=none
```

The final graph is displayed in Figure 3. The complete source code that produces this graph is in the Appendix.

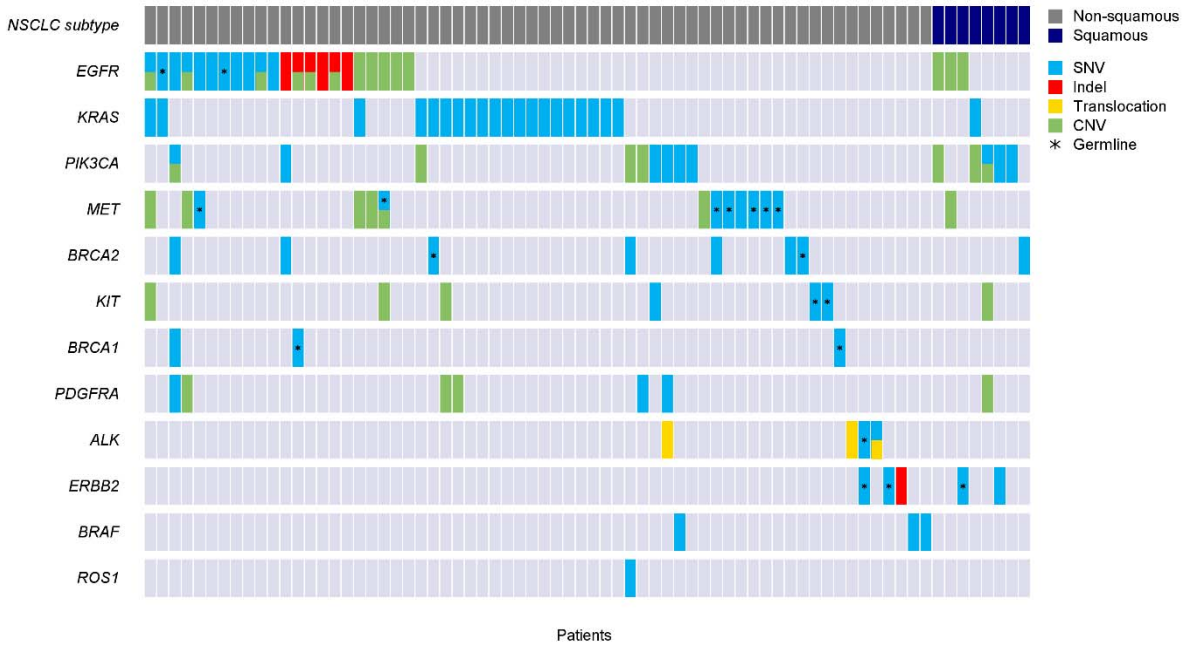

Figure 3. Final output with additional graphical elements, including scatter plots and legends.

Sorting the data has led to a grouped set of variants, especially for the genes in the first few rows. This makes it easier to find patterns or anomalies in the data. For example, the only variants present for patients with squamous NSCLC are SNVs and CNVs. Also, *EGFR* and *KRAS* variants appear to be mutually exclusive for most patients, which could prompt for additional investigation of the patients with variants in both genes.

CONCLUSION

GTL in SAS 9.4 provides several flexible methods for creating interesting visual displays of data. The HIGHLOWPLOT statement, in combination with other GTL graphical elements, is an effective means to create a grid of variants. HIGHLOWPLOT could potentially be used anytime a heat map or grid is required for a graph. Sorting the results so that the data are logically grouped on the graph can reveal interesting patterns that may lead to further investigation.

REFERENCES

Broom BM, Ryan MC, Brown RE, et al. A Galaxy implementation of next-generation clustered heatmaps for interactive exploration of molecular profiling data. *Cancer Res*. 2017 Nov 1; 77(21):e23-e26.

Eberhard DA, Bennett J, Davison D, et al. Clinical testing of ctDNA from NSCLC patients using a 17-gene liquid biopsy mutation panel. European Lung Cancer Congress 2018; April 11-14, 2018; Geneva, Switzerland.

Lee HY, Lee SH, Won, JK, et al. Analysis of fifty hotspot mutations of lung squamous cell carcinoma in never-smokers. *J Korean Med Sci*. 2017 Mar, 32(3): 415-420.

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.

SAS Institute Inc. 2016. *SAS® 9.4 Graph Template Language: User's Guide, Fifth Edition*. Cary, NC: SAS Institute Inc.

Spina V, Khiabanian H, Messina M, et al. The genetics of nodal marginal zone lymphoma. *Blood*. 2016 Sep 8; 128(10): 1362-73.

ACKNOWLEDGEMENTS

The author would like to thank Cindy Loman for her invaluable help with GTL coding and Rita Lopatin, Jim Whitmore, Anna Lau, Barry Grobman, Jennifer Duke, and Ruixiao Lu for their review of this paper and helpful suggestions.

CONTACT INFORMATION

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

John Bennett
Genomic Health, Inc.
1-650-569-5681
jbennett@genomichealth.com

APPENDIX

The complete set of statements that produced the final graph in Figure 3 follows:

```
proc template;
  define statgraph variantgrid;
   begingraph;
     discreteattrmap name='vt';
     value 'SNV' / fillattrs=(color=cx00B0F0) lineattrs=(color=cxFFFFFF);
     value 'Indel' / fillattrs=(color=cxFF0000)
      lineattrs=(color=cxFFFFFF);
     value 'Translocation' / fillattrs=(color=cxFFD700)
      lineattrs=(color=cxFFFFFF);
     value 'CNV' / fillattrs=(color=cx87BF62) lineattrs=(color=cxFFFFFF);
     value 'Squamous' / fillattrs=(color=cx000080)
      lineattrs=(color=cxFFFFFF);
     value 'Non-squamous' / fillattrs=(color=cx808080)
      lineattrs=(color=cxFFFFFF);
     value 'None' / fillattrs=(color=cxDEDDED)
      lineattrs=(color=cxFFFFFF);
     enddiscreteattrmap;
     discreteattrmap name='vt2';
     value 'CNV' / markerattrs=(color=cx87BF62 size=7
      symbol=squarefilled);
     value 'Translocation' / markerattrs=(color=cxFFD700 size=7
      symbol=squarefilled);
     value '.' / markerattrs=(size=0);
     enddiscreteattrmap;
     discreteattrvar attrvar=vartyp1 var=VariantType1 attrmap='vt';
     discreteattrvar attrvar=vartyp2 var=VariantType2 attrmap='vt2';
     legenditem type=marker name="nonsq_legend" / markerattrs=
      (color=cx808080 symbol=squarefilled size=11)
      label="Non-squamous";
     legenditem type=marker name="sq_legend" / markerattrs=
      (color=cx000080 symbol=squarefilled size=11) label="Squamous";
     legenditem type=marker name="snv_legend" / markerattrs=
      (color=cx00B0F0 symbol=squarefilled size=11) label="SNV";
     legenditem type=marker name="indel_legend" / markerattrs=
      (color=cxFF0000 symbol=squarefilled size=11) label="Indel";
     legenditem type=marker name="trans_legend" / markerattrs=
      (color=cxFFD700 symbol=squarefilled size=11)
      label="Translocation";
     legenditem type=marker name="cnv_legend" / markerattrs=
      (color=cx87BF62 symbol=squarefilled size=11) label="CNV";
     legenditem type=marker name="gl_legend" / markerattrs=
      (symbol=asterisk size=8) label="Germline";
     layout lattice / columns=2 columnweights=(0.8 0.2) border=false;
      layout overlay /
        borderattrs=(color=white) walldisplay=none
        yaxisopts=(display=(tickvalues) reverse=true
```

```
        tickvalueattrs=(style=italic)
        linearopts=(tickvaluelist=(0 1 2 3 4 5 6 7 8 9 10 11 12)))
      xaxisopts=(label='Patients' display=(label)
        labelattrs=(size=9));
      highlowplot y=rownum high=eval(Patient+.5)
        low=eval(Patient-.5) /
        type=bar group=vartyp1;
      scatterplot y=eval(rownum+.13) x=Patient / group=vartyp2;
      scatterplot y=eval(rownum+.32) x=Patient / group=vartyp2;
      scatterplot y=germline x=Patient /
        markerattrs=(symbol=asterisk size=4);
    endlayout;
    layout lattice / rows=2 rowweights=(0.1 0.9) border=false;
      layout overlay;
       discretelegend 'nonsq_legend' 'sq_legend' / down=2
         border=false halign=left valign=center;
      endlayout;
      layout overlay;
       discretelegend 'snv_legend' 'indel_legend' 'trans_legend'
         'cnv_legend' 'gl_legend' /
         down=5 border=false halign=left valign=top;
      endlayout;
    endlayout;
   endlayout;
  endgraph;
 end;
run;

proc sgrender data=grid template=variantgrid;
run;
```