USING SAS/GRAPH® SOFTWARE TO CREATE TREES THAT MODEL DEVELOPMENTAL BIOLOGICAL PHENOMENA

Perry Watts, Fox Chase Cancer Center
Samuel Litwin, Fox Chase Cancer Center

ABSTRACT:
Interpretation of clinical or biological data often requires an underlying mathematical or statistical model. Such models can increase the power of experiments to confirm or refute precisely stated hypotheses, but they often require visual displays to explain their construction. Tree structures are especially appropriate for displaying developmental models.

Three different types of trees created with SAS/GRAPH® are presented along with supporting code and intermediate graphics output to illustrate selected models and to show how the trees are actually constructed. Alternative visual displays that the trees supplement or replace are also presented.

The first tree, a dendrogram, uses PROC CLUSTER to create a data set that ranks distances and locates hierarchical clusters. The dendrogram's construction is illustrated by using a distance matrix of mileages between 10 American cities taken from the SAS® SAMPLES library. The dendrogram that models biological phenomena is an evolutionary tree of 21 B·Cells obtained as hybridomas and sequenced.

The second tree is an incomplete binary tree with variable length branches that illustrates left branch tree synthesis of randomly dividing cells alive at a fixed time horizon. A template data set containing fixed values for the X-coordinates merged with a second data set containing randomly generated values for the Y-coordinates is used in the construction of this tree. The third tree is called a sampled binary tree, because a random sample of mutational information on 21 lineages is taken from a complete tree containing up to 3 million cells. The resulting tree actually is not binary, since a parent may have one or two offspring. The trees are plotted with PROC G S L I D E, PROC G A N N O and the ANNOTATE facility.

DENDROGRAMS:
Lower triangular distance matrices measuring a variety of phenomena are submitted to PROC CLUSTER with a stated method of analysis. PROC TREE then takes the data set created by CLUSTER and produces a line printer dendrogram that is easier to interpret with the addition of some lines. Figure 1 groups 10 American cities in roughly East to West order. The line printer graphic unfortunately cannot be upgraded to a publication quality product, so the output data set created by CLUSTER is again used as input to a series of DATA steps that involve two-way array processing that will be described below. Figure 2 shows the output data set from PROC CLUSTER. The line printer graphic uses the more obvious variable _HEIGHT_ for plotting purposes whereas the graphic developed for this presentation is based upon _FREQ_ which is the number of leaves (cities) defined by the node listed under _NAME_. The use of _FREQ_ enables the dendrogram to accommodate a nonuniform
distribution of response data and to display data where parent clusters can have zero distances or distances that are even less than those of their progeny. However, care must be taken to label each distance so that the graphic is informative. The labelling can limit the size of the dendrogram that is displayed. Figures 3 and 4 are the completed dendrograms.

Figure 3. City distances with cluster method: AVG NONORM NOSQ.

Figure 4. Evolutionary Tree of 21 B-Cells Obtained as Hybridomas and Sequenced.

The key to programming the dendrograms is to make sure that lines never cross. This task is made especially difficult by the fact that the trees are not balanced. For example, the right branch of Cluster #1 in the lineage dendrogram contains two clusters whereas the left branch contains six. Furthermore, there is no relationship between the order of cities under NAME in the CLUSTER data set and their appearance at the base of the graph. However, the data is sorted in reverse _PARENT_ order, and a variable LOC for location, is created in a first pass of the data where the goal is to rank order the leaves and use the ranks later on as X-coordinates. How LOC works is described in the next paragraph and illustrated in figure 5.

LOC is initialized to 0 and then defined as -1/2 for clusters #2 and #5. This means that all progeny of cluster #2 will have LOC values of less than zero and all progeny of cluster #5 will have LOC values of greater than zero. No matter how many offspring cluster #2 has, they will never interfere with the offspring of cluster #5, because each time a new parent cluster is encountered, LOC is redefined in successively smaller increments. LOC, in this instance, is conforming to Zeno's paradox where the addition of smaller and smaller increments to -1/2 never brings one back to 0. The same process is going on in the opposite direction with cluster #5 and with all the subtrees in the dendrogram. In figure 5, for example, LOC for CL1 is set to 0. CL2 has a value of -1/2, and CL4 really is -1/2+1/4 = -3/4 with HSTON coming in at -1/4+1/16 for -5/16 which is its final value for LOC. This value ranks 7 among the cities, and 7 becomes the X-coordinate.

A second pass is made in the opposite direction to fix the X-coordinates of the parent clusters midway between their offspring. This is a simple matter of using information already known. For example the X-coordinate of CL7 is 1.5, since ATLANT equals 1 and CHIGO equals 2. Y-coordinates are set equal to the value of _FREQ_ as previously mentioned. Both X and Y coordinates are converted to percentages using a conversion formula that maps output onto a restricted portion of the screen. The conversion formula is:

\[
x_{\text{pot}} = \frac{100(x-x_{\text{min}})}{(x_{\text{max}}-x_{\text{min}})}
\]

By changing values for \(a\) or \(b\) mapping areas can easily be redefined to accommodate different text labelling needs. Figure 5, for example, had a reset from 0.05 to 0.15 to accommodate the additional text at the base of the graph. The dendrograms are plotted with PROC GSLIDE and the ANNOTATE facility.

INCOMPLETE BINARY TREES:

The model that motivated this graphics display attempts to determine the effects of somatic cell mutation and selection on immune cells that are collected many cell divisions after the onset of infection. The simulation starts with an unmutated ancestral cell. Each cell is succeeded by a left and a right daughter and each lineage is continued for a predetermined length of time. Cells can mutate or remain the same. Mutations either cause death or change the generation time of a mutant's successors. In a real life situation a complete tree of cells dividing every 18 hours would contain nearly 300 million cells at 21 days; a computational nightmare. This is where the left branch algorithm comes into play. With this algorithm, each lineage is examined by following a complete line of left descendants to the cell that crosses the time horizon. Once information is collected, the
lineage of cells can be discarded, and processing continues by next examining the lineage of the right daughter of the cell’s immediate ancestor. If a cell dies, no information is collected. By systematically examining and discarding information storage is limited to the length of the current lineage from ancestor to horizon.

While the flow chart in figure 6 is inadequate to explain the left branch algorithm, it is needed for programming the incomplete binary tree in figure 7. In that tree, only the highlighted lineage is in memory, and the information collected at the time horizon is denoted by an asterisk. Cell generation times clearly vary, and there is a lethal mutation in path #5. Notice that no information is collected in the fifth frame.

As mentioned above, two SAS® data sets are required for this display. One is the template, and it contains X-coordinates by cell number of a complete binary tree of n levels. Each cell is assigned a level by the left branch algorithm. The template in figure 8 shows how LEVEL is used for identifying a cell’s right and left daughter. Unlike the dendrogram, calculating the ranks of the leaves is a simple process that involves identifying those nodes having LEVEL equal to the number of levels in the tree and then incrementing their rank by one to fix X-coordinates. For example, cell #4 in figure 8 has rank 1, cell #5 has rank 2, and cell #15 has rank 8.

The second data set uses the SAS® function UNIFORM to generate random numbers that determine a cell’s status of dead or alive, and if alive, the length of time to division. If the template describes a fixed tree as in figure 8, then the Y-coordinates are equal to LEVEL. In the random data set, the Y-coordinates are equal to LEVEL with the addition of variability created by a call to the random number generator. Random tree cells are initialized to dead, because if a parent cell is declared dead by a call to UNIFORM, then the progeny must also be dead. Next, path numbers defining lineages are calculated, and the random data set is duplicated by copying all previous path numbers up to the current path until the total number of paths have been processed. The template data set is duplicated 2\(^n-1\) times. The two data sets are merged, and their values are converted to percentages using the dendrogram conversion formula prior to being plotted with PROC GANNO that uses the ANNOTATE data set.

Serial plots are created with a call to PROC GREPLAY. Note that frame #7 in figure 7 is exactly twice as high and twice as wide as the smaller frames. This is done so that the shape of the nodes or cells is uniform for the entire graphic. Otherwise the distortion that is present would be evident. To have no distortion, the height and width of a plot would need to be equal in length. Cells in such a plot would be perfectly circular instead of elliptical as they are in figure 7.

SAMPLED BINARY TREES:
In a Monte Carlo simulation, the statistician wants to create N trees for analysis. In that instance, a random sample...
Figure 9. The distribution of cells showing how a sampled binary tree is created.

Figure 10. A sampled binary tree showing mutation rates for approximately 40 generations of cells.
If more than 40 generations are in a sample, an alternate tree is produced that defines the length of a branch as the number of ancestral cell divisions involving just one offspring. Lines are drawn to connect parents of a given branch to their binary offspring. The resulting tree is now binary but no longer balanced. This type of tree is pictured in Figure 11. The tree displays summary information on shr only. If Figure 10 were put into Figure 11 format, only the bold entries in Figure 10 would be shown. Both graphs again rely on PROC GSLIDE and ANNOTATE for their actual display.

CONCLUSIONS:

Trees are a graphic of choice for displaying intermediate results of developmental phenomena. Three different trees are presented in this paper. The first, a dendrogram is binary, but not balanced since the two main subbranched do not have the same number of levels. This feature creates a challenge when it comes to calculating the tree’s X-coordinates. The incomplete binary tree, on the other hand, is balanced, even when selected nodes are not displayed. However, a value for the variable LEVEL needs to be associated with each cell so that both right and left daughter cells can be identified. The sampled binary tree presents unique problems in that the tree is not actually binary. In this instance, a variable NPATH records the number of paths that pass through a given cell. This information is used in calculating a cell’s X-coordinates and drawing lines that connect tree branches. Both the incomplete binary tree and the sampled binary tree present graphic representations of the left branch tree algorithm.

SAS and SAS/GRAPH are registered trademarks of SAS Institute Inc., Cary, NC, USA.

REFERENCES:


SAS® Sample Library. SAS®ROOT/SAMPLES.STAT

CLUSSTD, SAS


