Assignment 2: Tree of Life - how are things related

Due Thursday February 12, 5pm.

Trees are a mechanism for conveying hierarchical relationships among objects. Family trees, phylogenetic trees, and evolutionary relations are all examples of such trees. Mostly they are created by using some distance function to establish a hierarchical clustering of the data. Different graphical representations can then be used to convey not only the groupings but also the distances within and between groups.

For this project you will start with a set of named objects and a table of variables or a genetic sequence for each, from which you will compute distances between each pair of objects. For example (you don't have to use this dataset), at http://archive.ics.uci.edu/ml/datasets/Zoo you can find a table of 17 attributes of 101 animals. The distance calculation could just be the sum of the mismatches between the characteristics of 2 animals (for the Zoo dataset, you should ignore the first and last columns of the data in calculating the distances).

In the first phase you will implement an algorithm that performs hierarchical clustering of the data.

The easiest algorithm is called bottom-up agglomerative clustering. You start with a list of objects and their distances, and compute which pair of objects are closest. You then create a new object list by removing these two objects and put in a new object that is their parent. The new distance from each of the other objects to this node is the average distance to the nodes that were merged. You then repeat the process until all objects are merged. You'll need to keep track of all nodes that are merged into a cluster. This can be done with arrays or linked lists. The names of the cluster objects can be omitted or can be assigned default names, such as cluster1, cluster2, and so on. Note that this creates a binary tree.

In the second phase, you will draw the resulting tree.

For a basic view, we can focus just on the structure of the tree. For example, you can line up all the non-cluster objects along the bottom or side of the screen (this makes reading the text names easier) and then position the cluster objects offset up or across from other nodes based on which objects make up their cluster. There are only 3 possibilities - either the cluster is made up of 2 base objects, one base object and one cluster, or 2 clusters. The position of the cluster should be at a distance proportional to the iteration of the clustering algorithm in which it was formed; for the other coordinate you can use the midpoint between the two nodes that make up the cluster. This is just one possible approach - we will discuss others in class. These are the minimum requirements.

Remember that in the assignments part of your grade depends on going beyond the minimum requirements. There are many opportunities to move beyond the basics. For example:

- Write an algorithm to reorder the objects to minimize line crossings. One way to do this is to compute the tree and start rendering from the base of the tree (where all nodes are merged into one cluster) to the terminal nodes, minimizing crossings along the way.
- Use the actual distances to determine positioning, so rather than evenly spaced nodes you can have them convey the original or computed distances using positioning. You can also color modify the width of the connecting lines to convey distances.
- Use biological knowledge to shape the clustering and distance metrics.
- Add interactions and alternate views that align with some analytical goal relevant to biology.
- Relax the binary tree assumption: modify both the clustering algorithm and visualization technique to allow more than two child nodes per parent.

Don't restrict yourself to these examples. The goal is to develop and implement your own ideas.

Resources

The TreeVis.net site is a great resource if you want to explore alternative tree representations. Some are good, but some are absolutely terrible.

If you want to add things like sliders and text input, see the controlP5 library.

If you use Sublime Text for coding, here is a tool that will allow you to run Processing from within your editor.

Writeup Requirements

Your README.md should:

1. Describe how you created / adapted your algorithm that performs hierarchical clustering of the data.
2. Describe, at a high level, how your program draws the resulting tree based on the data.
3. If you are using outside code for the core of your project:
    - Identify the source(s) of the code.
    - Tell me how you modified it to make it your own work.
4. For the free-form (beyond the requirements) component:
    - Describe the biological significance of the additions you make.
    - Describe the technical significance of the additions you make.
5. Provide explicit instructions on running your program.

Turning in the project

Submit the following:

- All of the source code
- A README file containing the elements listed in the Requirements section

The source code must run on a Mac or Linux machine.

Submissions should be made using myWPI. Either:

1. (Recommended) Upload text or a .txt file with a link to your GitHub repo, or
2. Zip your files and README and upload to MyWPI

Grading

Each homework assignment is graded on a 100 point scale:

- 85 points will be based on whether the program fulfills the minimum requirements.
- 15 points will be based the additions you made:
  - 7.5 Bio-related additions
  - 7.5 Technical additions

Total – 100

(0 will be assigned if the code can't be compiled or run.)