1 CLARITY Brain Processing and Algorithms

1.1 Introduction:

In order to analyze CLARITY brains, ClarityViz utilizes a number of algorithms. Here we provide a simple and succinct explanation of each algorithm applied, the location of the algorithm implementation inside the ClarityViz package, and a general rationale behind why we used the algorithm.

1.2 Histogram Equalization (Image Contrast):

Use Case: Decomposes 3D image data into a histogram such that we can increase the contrast in the image. Used in

Code Location: claritybase ”applyLocalEq” function.

Methodology: Let n_i be the number of voxels with a specific gray scale color i (where i is bounded from 0 to 255). Let n be the total number of voxels. We can thus define the normalized histogram p as: $p_x(i) = p(x = i) = \frac{n_i}{n}$, for $i = 0, 1, 2, ...255$. We can then define a cumulative density function for p_x as the sum of all $p_x(i)$. Effectively, we want the new transformed cumulative density function to be $i \times K$, where K is some arbitrary pre-selected constant (this thus generates our ”equalized” histogram). Specifically, we applied the CLAHE (Contrast Limited Adaptive Histogram Equalization) technique. While the methodology I described above describes the approach for normalizing across an entire image, CLAHE focuses on normalizing across a more ”local” region. The contrast limiting part effectively sets a threshold value for clipping the histogram, thereby preventing the over-amplification of noise. Thus, the CLAHE approach increases the contrast for neighbors to increase the discernability of the voxels relative to some set of their neighbors.

Reasoning: Histogram equalization makes individual bright points that are surrounded by a cluster of bright points even brighter, while making the backgrounds (dark points in the image) darker. CLAHE makes similarly-bright regions have greater inter-voxel brightness values.

1.3 Thresholding (Bright Point Selection):

Use Case: Select random sample of voxels in data set after histogram equalization that are a specific grey-scale value (typically 255).

Code Location: claritybase ”brightPoints” function.

Methodology: Preselect a number of samples (ex: 10,000 samples). If the number of samples is less than the number of points with brightness 255, then randomly sample from the set of all 255 gray scale bright points. If not, then select all the 255 points and then randomly sample from the remaining.

Reasoning: We want to analyze the activity of the most active regions in the brain. Thus, we use thresholding and random samplying to get some subset of the brightest points in the brain (eg: some random set of 10,000 255 gray-scale value voxels).

1.4 Affine/Reverse Affine Transformation and LDDMM Registration (Atlas Registration Process):

Use Case: Transform the coordinate plane of the brain data to the coordinate plane for the atlas, such that we can apply the atlas registration process. The reverse transformation allows us to tag each point in the original brain with their corresponding atlas location. The LDDMM registration is the process by which the atlas coordinates are fitted to the brain coordinates.

Code Location: brainalign.py, atlasregiongraph ”plotting” function.

Mathematics: Apply linear affine transformation to voxel image data to fit the image data with the existing orientation from the atlas. This allows us to work in the same coordinate space/reverse the transformation to apply the atlas back on the existing data. LDDMM registration is the Large Deformation Diffeomorphic Metric Mapping (LDDMM) tool; it's an application which aims to assign metric distances on the space of anatomical images in Computational Anatomy, thereby allowing for the direct comparison and quantization of morphometric changes in shapes. The software package can be grabbed here: http://www.cis.jhu.edu/software/lddmm-volume/about.php.

Reasoning: In order to fit the atlas to the brain data, we must find out the relative positions of each voxel relative to their location in the atlas. However, brain data is often collected in a different orientation from the atlas, and so we must transform the coordinate plane from the brain data to fit the coordinate plane of the atlas. We then use the LDDMM to fit the brain data to the atlas data. After fitting, we reverse the original transformation to return the brain back to its original orientation, but we've now tagged each region with its corresponding atlas location.

1.5 Epsilon Ball Radius (Density Plotting):

Use Case: Given a specific node (as calculated from the bright points script, one particular bright point), give the node edges such that all edges are less than or equal to a predetermined radius.

Code Location: claritybase ”plot3d” function (radius defined there).

Mathematics: We can find the Euclidean distance ($x^2 + y^2 + z^2 = r^2$, where r is the radius) between any two points in x, y, z space. Given the distance, we set an upper threshold (eg: 25); if any two nodes have a distance less than or equal to the threshold, an edge is inserted between the nodes.

Reasoning: In order to find the relative number of bright points in a given region/the relative connectivity within a given region, we use the Epsilon ball to generate connections based on proximity.