This is the flowchart of all files used for the clone classification and cluster analysis. In is presented in the order in which the files must be executed. It lists the inputs, outputs, whether it can be run on the cluster and a short description of its role.

***1. ProcessImages.R***

Cluster: No

Input: PIC files of the unformated images

Output: PIC files of filtered and tubed images

Description: This script first takes the PIC files, converts them to 8 bit and downsamples them to 386 by 386. It then runs the anisotropic filtering program Neura (Broser, J Biomed Opt, 2004). Finally, it runs the tubeness ImageJ plugin.

***2. segment\_remaining\_images.m***

Cluster: No

Input: *XXX-tubed.PIC* files of filtered and tubed images

Output: *XXX\_filtered2\_tubed.mat* files of thresholded and segmented images

Description: This thresholds the tubed image to produce a binary output. It then segments the image into connected region using the matlab command “bwlabeln”.

***3. process\_remaining\_images\_for\_dimension\_reduction.m***

... which uses ***image\_dimension\_reduction.m***

Cluster: Yes

Input: *XXX\_filtered2\_tubed.mat* files of thresholded and segmented images

Output: *XXX\_dimensionReduced.mat* files containing list of dot coordinates and other parameters

Description: Performs the dimension reduction algorithm proposed by Bialek et al.

***4. reformat\_remaining\_images.m***

*…* which uses ***reformat\_coords.m***

Cluster: No, fast enough to run on desktop

Input: *XXX\_dimensionReduced.mat* files containing list of dot coordinates

Output: *XXX\_reformated.mat*files containing list of reformatted coordinates

Description: Transforms coordinates of image onto IS2 template. Requires access to the registration directories (ie. IS2\_SAAG7-1\_01\_warp\_m0g80c8e1e-1x26r4.list). Also requires access to the tubed PIC files in order to retrieve their dimensions.

***5. calculate\_properties\_remaining\_images.m***

… which uses ***extract\_properties.m***

Cluster: Yes

Input: *XXX\_reformated.mat* files containing list of reformatted dot coordinates

Output: *XXX\_properties.mat*files containing list of reformatted coordinates, along with tangent vector and measure of how one-dimensional the structure is.

Description: Calculates the moment of inertia from the 20 nearest neighbors for each dot in the image. The principal eigenvector is the tangent vector, alpha is (eigenvalue 1 –eigenvalue 2)/sum(eigenvalues). Points outside of a mask (IS2\_nym\_mask.tif) which covers the central brain and interior tracts, in addition to points with alpha<.25, are removed. The remaining points and tangent vectors are saved as p.gamma2 and p.vect2, respectively.

***6. find\_matched\_dots\_remaining\_images.m***

… which uses ***compare\_images.m***

Cluster: Yes

Input: *XXX\_properties.mat* files with reformatted coordinates and tangent vectors

Output: *XXX\_matchedPoints.mat*files with a includes binary matrix indicating the dots that meet matched points cutoff. If a matchedPoints file already exists for an image but you want to add comparisons to additional images, this script will add the new images automatically.

***7. create\_image\_classifier.m***

… which uses ***collect\_clone\_information.m***

… and ***build\_MI\_structure.m***

… and ***classify\_image.m***

Cluster: No

Input: The directory where the XXX\_matchedPoints.mat files are stored. Also, a .*tab* file which contains the current list of brains and the clones they contain. This file can be exported from FileMaker. Images should appear in the first column and clones in the second. It should look something like below:

% ImageA CloneA

% ImageB

% ImageC

% ImageD CloneB

% ImageE

%

% Here, Images A-C contain clone A and images D and E

% contain clone B.

Output: a *.mat* file containing the mutual information for each dot and each clone, the ROC score, plus the optimal mutual information threshold.

Description: For each clone

***8. compare\_image\_to\_all\_clones.m***

… which uses ***classify\_image.m***

… or ***classify\_image\_not\_in\_database.m***

Cluster: No

Input: The name of the brain to be tested (ie SAKB19). Output: A matrix containing the score each for clone across different conditions.

Description: For each clone, the function uses the images containing the clone as a template set. For all dots in the template set, determine which dots match dots in the test image. The score is then based on how many dots over a minimum mutual information were a match, or a sum of the matching dots weighted by their mutual information. Must specify the directory containing the XXXmatchedPoints.mat files.