

Brain Viewer Technical Overview

Pierpaolo Rametta

25 agosto 2014

1 Labels and surfaces

The freesurfer's surfaces are .asc files. These files are a mesh of vertexes where every vertex can be associated with a label. A label file is a text file capturing a list of vertices belonging to a region. To label the surface the tool does its associations with the Destrieux Atlas. Freesurfer's application denotes its segmentation volume with the suffix "a2009s". Finally the labels are generated from an annotation file with the command "`mri_annotation2label`" and they are already present in the brain viewer.

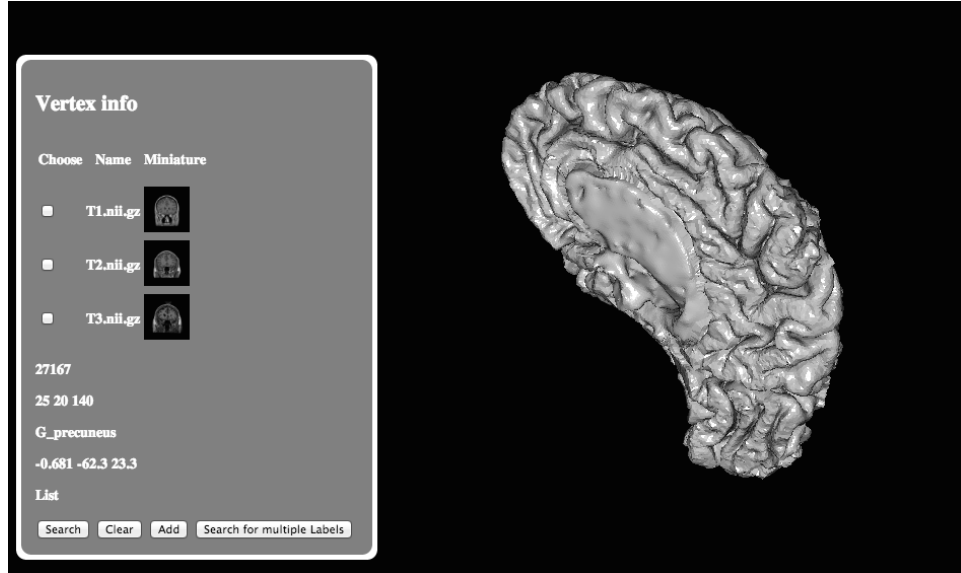
1.1 Segmentation with Destrieux Atlas

For every subject's folder we have inside a mri folder in which we can find a segmentation file named "`aparc.a2009s+aseg.mgz`". This file map the Destrieux's regions with mri labels. We need to convert this into a nii file using the command line tool "`mri_convert`". If in the mri folder we do not have this file, we can generate it using "`mri_aparc2aseg --annot aparc.a2009s`".

2 Representing the brain

To display the brain we use the javascript library brainbrowser. It gives the possibility to load a surface in ASCII format. The loading function is inside of "`pages/brainbrowser/surface-viewer-settings.js`" and we display the `rh.pial.asc` file.

3 Viewer' s Input



Clicking on the brain we can select a vertex and the tool will retrieve its label. The information about it and the relative region are shown in the box on the left. Now using the "add" button we can append the color value to the query and then using the "search for multiple labels" button we can perform the query. If we want to select only one region we can click on a vertex and then we can use the button "search". Using the "clear" button all the information about the vertex is removed.

4 Querying

The searching mechanism works this way:

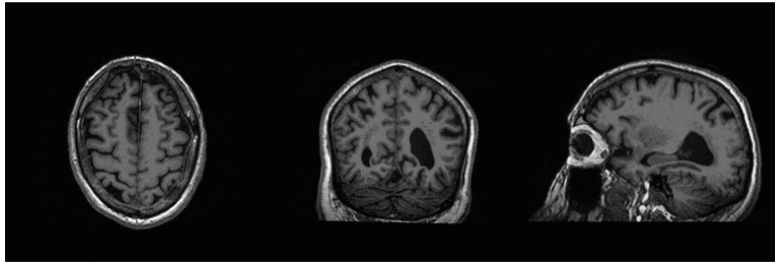
1. When a vertex is clicked it finds its label
2. It searches the color code for that label
3. It sends the color code to the server and finds the ids linked with it in the FreeSurferColorLUT.txt¹
4. It finds which slices contains these ids and saves them as volumes

¹FreeSurferColorLUT.txt is a standard lookup table. You can find this file in the freesurfer's directory and it is used to link colors and regions with an id.

4.1 Searching in the volumes

Once we have a list of ids we can search in which slice these ids are present. To do this we call a jar file called "niftiLabelsFinder.jar" that given a list of ids, a segmented and an original volume produces three volumes in gif format. The java program used to create the jar use imageJ library to load and to work on the nifti files. You can launch the jar as follows "java -jar niftiLabelsFinder.jar label_label_label imagesDir segmentedVol volumeName savingDir" i.e. "java -jar niftiLabelsFinder.jar 26_255 /Users/pierpaolo/git/brain_viewer/images Segmentation/T2.nii.gz T2.nii.gz /Users/pierpaolo/Desktop/tmp/gif/img". To make things simpler the .nii files are stored in a folder named images/ and in a subfolder named Segmentation there are the segmented volumes.

4.2 Output



If a slice contains one of the ids we put it in a volume that will be shown. The volumes to be displayed are 3 for the different planes (sagittal, coronal, axial) used also in Slicer.

5 Talking to the server

The server is made using the "express" web application framework for node. We have a dispatcher that handles all the requests. The web page makes ajax calls using jQuery to get the information about the clicked vertex and to get the list of nifti files. Then when we want to perform a search it creates dynamically a form that makes the request. All the call to the server are stored in "pages/brainbrowser/surface-viewer-settings.js". On the ser-

ver side the tool uses its own libraries "imageCreator.js", "labelsReader.js", "surfaceLabelFinder" and "dispatcher.js". These libraries do the following:

ImageCreator uses the "niftiLabelsFinder.jar" and puts the output in a folder. When the output is shown it removes the folder.

LabelsReader finds the id for a given color.

SurfaceLabelFinder finds the information about a vertex.

Dispatcher handles the web requests.