

A bottom-up parcellation of cortical surface using dMRI information

Brahim Belaoucha

November 16, 2015

1 How to use:

There is two ways to use this code. Either on a Cluster or import it as a module in your personal python code.

1.1 Input

input_path: It is a matlab file that contain the followings:

- Faces: The faces of the mesh (cortical surface) ($N \times 3$).
- Vertices: The coordinates of the vertices of the mesh (in anatomical or diffusion space) ($n \times 3$).
- Normal: The normal vector at each vertex ($n \times 3$).
- Connectivity: matrix that locate the edges between the vertices ($n \times n$).

They are used to save the parcellation at each iteration on a '.vtk' file. **output_path:** path to the folder where you want to save the data. Be sure you have the right to write in this path.

coordinate: path to the file containing the coordinates ($n \times 3$ of int elements) of the mesh in the diffusion space.

tracto_path: path to the tractograms. They must be in Nifti format ".nii.gz".

tracto_prefix: The beginning of the tractogram name. The tractogram obtained from FSL have the following name: tracto_prefix_x.y.z.nii.gz, where (x,y,z) is the coordinate of the tractogram.

Regions: is the list of the required number of regions. Remember that this algorithm does not give the exact number of regions as required but it uses this list to stop growing big regions.

Excluded_seeds: path to the file seeds that should be excluded from the parcellation (ex.Thalamus, or one hemisphere).

SM_method: the name of the similarity measure used in the parcellation. Fow now there is: "Correlation" (default), "Ruzicka", "Roberts", "Tanimoto", "Motyka", "Cosine"

Cvariance: The coefficient of variance threshold is used to stop merging regions that have high variance of the similarity measure values.

1.2 output

The parcellation at each iteration is saved in `output_path/RealLabel/` in '.vtk' format. At the end, the similarity measure between all pairs of the region is saved in `output_path/SM.all.txt` (a vector so the order is not preserved).

1.3 In a terminal (or Cluster)

It is easy and straight forward:

```
python run_parcellation -i input_path -o output_path -seed coordinate -t tracto_path  
-tb tracto_prefix -NR Regions -Ex Excluded_seeds -sm SM_method -cvth Cvariance
```

1.4 Import it in your code

You need to use it as follow:

```
import Region_preparation as RP  
from Cortical_surface_parcellation import Parcellation as CSP  
Parcel=CSP(tracto_path,tracto_prefix, output_path)  
Mesh_plot=RP.Mesh(Vertices,[],Faces,Normal)  
Excluded=numpy.loadtxt(Excluded_seeds)  
Parcel.Parcellation_agg(coordinate,Connectivity, Excluded,Regions,SM_method,Mesh_plot)
```

The only difference between the two ways of executing the parcellation is the input "Regions". In the terminal, the different number of regions should be separated by a comma ex. 100,200,300. But it must be an array of integers if you use the second way.

We advice to use to first way for the whole brain parcellation and the second for parcellating a region of interest.