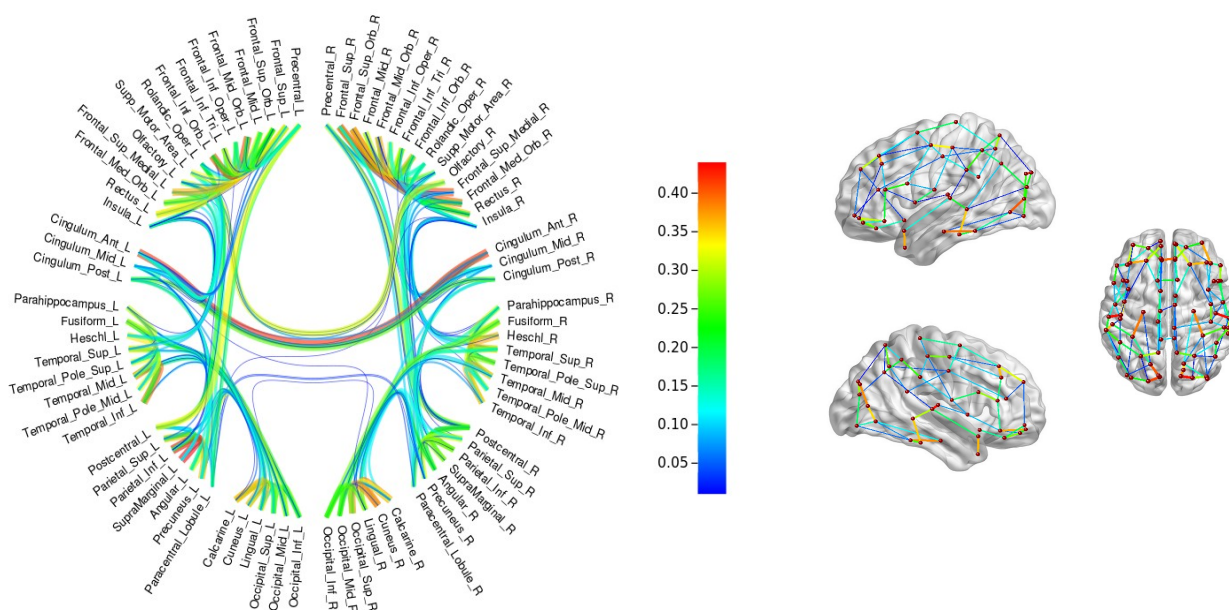


CIVILITY : **cloud based Interactive Visualization** **of Tractography Brain Connectome**

DOCUMENTATION

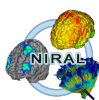


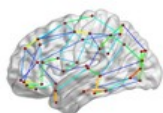
Circle visualization

Brain template visualization

CONTENTS

1. Application architecture and installation
2. Jobs and Tractography pipeline
3. Run jobs and results
4. Brain connectome visualization
5. Appendices





Introduction

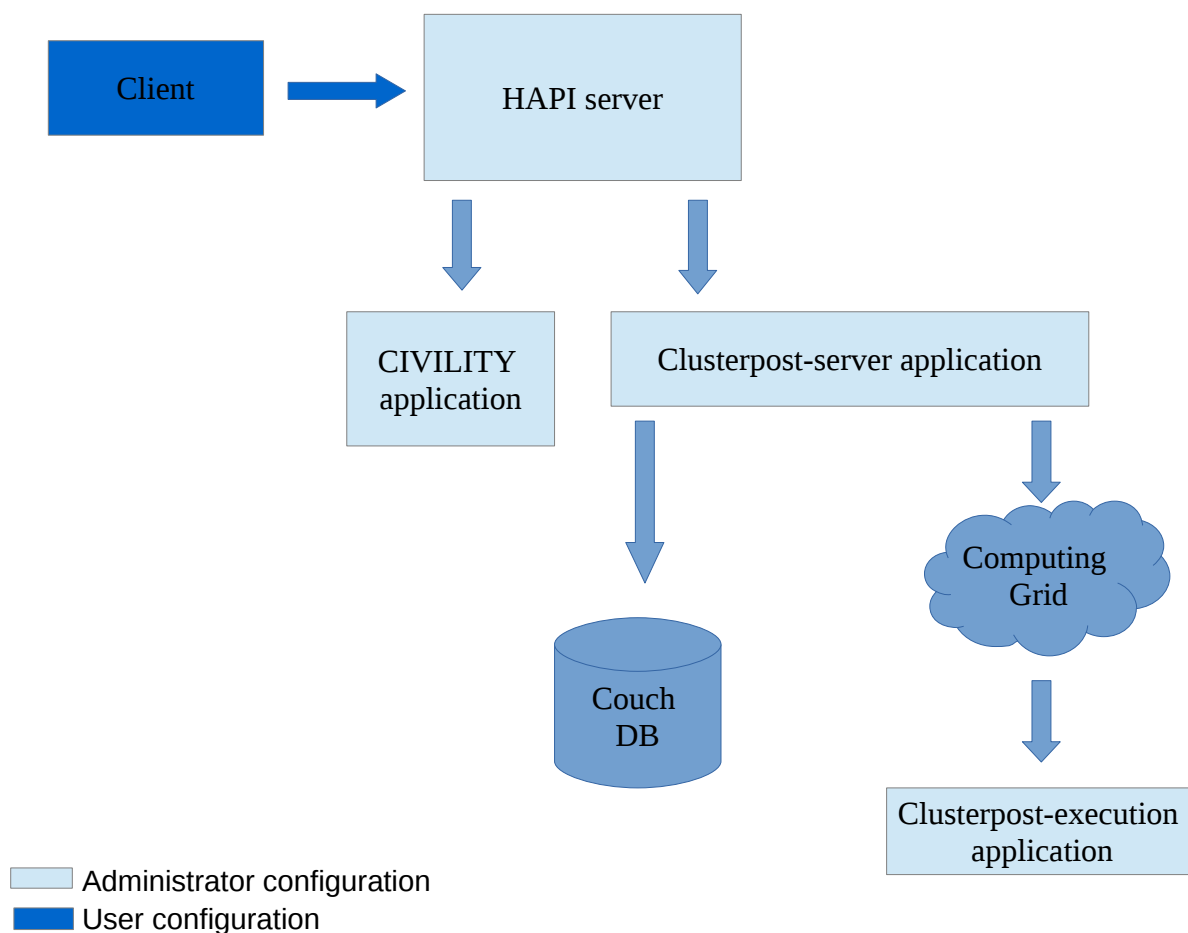
CIVILITY (cloud based Interactive Visualization of Tractography Brain Connectome) is a HAPI plugins used to visualized easily brain connectomes within an interactive web interface. This plugin, configured with the HAPI plugin *execution-server*, launches tractography scripts on a computing grid and uses a database to store tractography data. The brain connectome is computed with a probabilistic tractography method (FSL tools) using surfaces as seeds. In this documentation, the configuration of CIVILITY is first described, then the tractograpy pipeline is explained in details, and finally the interactive visualization methods are present.

1. Application architecture and installation

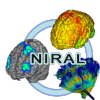
This section describes the architecture of the application, and how to configure it.

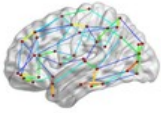
1.1. Architecture

Here is the architecture of the application CIVILITY and its dependencies.



Architecture of CIVILITY application





CIVILITY application is running on a HAPI server – this server must have an IP public to be accessible by everyone in the institution. CIVILITY users had privileges associated. By default, the user can only access to the Visualization tab (inputs are system files), the scope “clusterpost” needs to be added by an administrator to have access to the computing grid and launches jobs. Also, an “admin” scope is required to have access to the whole database, set users privileges etc.

To be able to launch tractography jobs on a computing grid server, the application CIVILITY requires an other plugin clusterpost-server running on the same HAPI server which must be configured properly by an administrator.

Clusterpost-server plugin is a server side application using REST api. It creates/posts 'job' document describing task and input data, run tasks on remote computing grids and retrieves outputs throw a database. All process is done automatically.

The documentation of the plugin clusterpost-server and its configuration are available here :

<https://www.npmjs.com/package/clusterpost-server>.

An other plugin called clusterpost-execution must be installed in the computing grid server. This plugin downloads data from the data provider to the local disk, submits tasks to the kob manager. It also checks if the task has finished and upload the results back to the dataprovider (database).

The documentation of the plugin clusterpost-execution and its configuration are available here :

<https://www.npmjs.com/package/clusterpost-execution>

The next paragraph will explains how to deployed the plugin CIVILITY.

1.2. Application deployment

To deployed the application :

git clone : <https://github.com/NIRALUser/ProbtrackBrainConnectivity>

Then, in this cloned directory, on the terminal do :

```
npm install
```

In the subdirectory static/public/ on the terminal do :

```
bower install
```

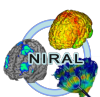
Finally to start the application, the administrator must update the file : conf.production.json according to the clusterpost-server documentation :

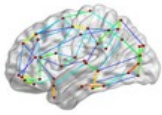
<https://www.npmjs.com/package/clusterpost-server> and then on the terminal in the application directory do :

```
NODE_ENV=production node index.js
```

The application is now running.

Administrator must also install all tools required by the tractography pipeline in the Computing Grid server.





Tools required :

- FSL : bedpostx and probtrackx2 tools
<http://fsl.fmrib.ox.ac.uk/fsl/fslwiki/FslInstallation>
- DWIConvert : <https://github.com/BRAINSia/BRAINSTools/tree/master/DWIConvert>
→ using ITK : <https://itk.org/ITK/resources/software.html>
SlicerExecutionModel : <https://github.com/Slicer/SlicerExecutionModel>
- ExtractLabelSurfaces : <https://github.com/danaele/ExtractLabelSurfaces>
→ using RapidJSON : <https://github.com/miloyip/rapidjson>
SlicerExecutionModel (using ITK)
VTK : <https://gitlab.kitware.com/vtk/vtk.git>

Python must also be installed.

1.3. User configuration

1.3.1. *Run application*

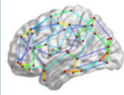
Open an internet browser and connect to the website

<https://152.19.9.68:8180>

If the browser warning you about certificates permissions – you must accept the exceptions and get the certificate. Go in advanced options in your browser and Add exception to get the certificate and then confirm the security exceptions.

1.3.2. *User account*

First, each user needs to be authenticate to use CIVILITY application ; a user account is required.



CIVILITY
Cloud based Interactive Visualization of Tractography Brain Connectome

Please login

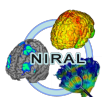
Email address

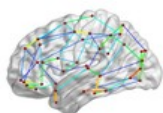
Password

Login

[forgot your password? click here](#)
[new user? create new account](#)

Home page of CIVILITY





Create account

On first use, the user must create an account. Click on the link : “*create new account*”. You are redirected to a new page* with a form to complete.

Set *user name*, *email address* and *password* (the password must contains 6 characters including at least one uppercase letter, one lowercase letter, one number) and click on the button : “*Create and Login*”. The user account is created and automatically login to the application.

An alert window is appear at the creation of the account specifying to the user to send an email to the Application administrator if he wants to be able to launch tractography scripts.

**On this page page, user still can return to the login page by clicking on “Login with your account”.*

Login

If the user already have an account, fill the form on the home page by providing *email address* and *password* (the password must contains 6 characters including at least one uppercase letter, one lowercase letter, one number) and click on the button : “*Login*”.

The user is login into the application.

Forgot password

If you don't remember your password, fill the field Email Address in the Home page and then click on the link : “*Forgot your password ? click here*”.

An email will be sent to your email address containing a link. This link will redirect you to a reset page. In the page you need to type 2 times your new password and then click on “*Reset and Login*” button to login into the Application.

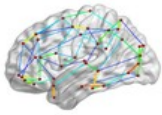
Once the user is connected to the application, three mains tabs are available.

The first tab is a form where users upload files and set some parameters for the tractography. The second tab is a summary of all jobs launched by the user. And the tab “visualization” allow user to print Brain Connectome by uploading files from user system.



Header of CIVILITY Application – View of all tabs

The next section explains how to launch jobs and describes the tractography pipeline running.



2. Jobs and Tractography pipeline

2.1. Tractography pipeline

CIVILITY computes the brain connectome with a probabilistic method of tractography using surfaces as seeds. The tractography is computed by using FSL tools : bedpostx and probtrackx2.

First, all images and surfaces of the subjects required for the tractography are :

- DWI (diffusion weighted imaging) image (.nrrd)
- T1 image – **must be in the DWI space** (.nrrd)
- Brain mask – **must be in the DWI space** (.nrrd)
- Inner surface – **must be in the DWI space** (.vtk)

This surface must contain a color label for each region.

If the surface doesn't contain label information, you must provide another surface with label information which must be the same mesh as the Inner surface – in DWI space. (example : you can use the middle surface)

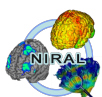
All images and surfaces must be in the same space which is the DWI space (also called Diffusion space).

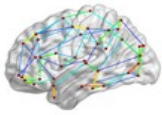
Another file required is the parcellation table which is a json file describing the brain atlas. The structure of this json file is described in the **appendix 2**.

Pipeline :

- Creation of the diffusion data for using bedpostx
 - create Diffusion directory
 - convert DWI et brain mask images to Nifti format (.nii.gz)
- BedpostX (FSL tool) , fitting of the probabilistic diffusion model on data. The voxel fitting uses 2 tensors by default.
- Creation of seeds surfaces (ExtractLabelSurfaces tool) : each seed is created as an ascii file (.asc) describing the mesh
- Write the seeds list, run a python script writeSeedList.py which writes all paths to the seeds surfaces (.asc file) according to the description table (.json file)
- Convert T1 image to nifti format with DWIConvert tool
- Run probtrackx2 (FSL tool), computation of the tractography according to the seeds list created.

default command line : `bedpostX DiffusionDir -n 2`
`default command line` : `probtrackx2 --samples=${SUBJECT}/Diffusion.bedpostX/merged --mask=${SUBJECT}/Diffusion.bedpostX/nodif_brain_mask --seed=${SUBJECT}/seeds.txt --seedref=${T1_nifti} --forcedir --network --omatrix1 -V 0 --dir=${NETWORK_DIR} --stop=${SUBJECT}/seeds.txt ${probtrackParam} ${loopcheckFlag} -P 3000 --steplength=0.75 --sampvox=0.5`





- Normalization of the connectivity matrix
- Save the matrix plotting (color grid) as a pdf file – run a python script : `plotMatrix.py`

The shell script excecuting this pipeline is available here :

<https://github.com/NIRALUser/ProbtrackBrainConnectivity/blob/master/ScriptsApp/tractographyScriptApp.sh>

Python scripts which create the seeds list and the figure of the matrix as pfd are available :
<https://github.com/NIRALUser/ProbtrackBrainConnectivity/tree/master/ScriptsApp/>

2.2. Run jobs

To run a user must complete the form (see Appendix 1) in the Tractography application tab do :

- Set a job name ; to identify the job.

This id/name gonna be the name of the directory containing all outputs of the pipeline

- Upload DWI image. The file format is NRRD.

- Upload T1 image. This brain mask must be in the same space as the DWI space, so it must be in the diffusion space. The file format is NRRD.

- Upload brain mask. This image must also be in the diffusion space. The file format is NRRD.

- Upload the parcellation table. This file is a description of the parcellation table (ROIs). This file also describe the way how to plot the brain connectome in circle plot and also contains information about matrix rows. See appendix 2 to have more details about the file architecture. The file format is JSON.

- Upload the Inner surface. This is a VTK file which represent the white matter surface. This surface must be in the diffusion space.

- This surface contains labels for each regions ?

In the vtk file is ROIs are identify by labels, check the box. By default is checked.

If there is no labels to identify ROIs in the VTK surface, you must upload an other surface (for example grey matter surface called Middle Surface) with the same mesh and containing labels information. This second surface must also be on the diffusion space.

- In the VTK file containing labels, the labelset is identify by an ID. You must specify this labelset name in the text box associated.

- If you want to ignore a label in the vtk file, check to box "Ignore label" and enter the value/ID of the label to ignore (example : "0 0 0").



- There is an additional option in the surfaces extraction : overlapping. If the checkbox corresponding to “Overlapping” is selected that mean each region overlaps the neighbors regions. (by default : selected).

*For FSL tools, it's possible to modify their parameters. Parameters write in **bold** can't be modify.*

- Bedpostx parameters

By default the number of tensors in the voxel fitting is equal to 2. You can modify bedpostx parameters if you are familiar with this tool by clicking on the checkbox “Modify”.

- Probtrackx2 parameters

Select loopcheck or not for the tractography computation.

Probtrackx2 help → Loopcheck: By default, we terminate pathways that loop back on themselves -i.e paths that travel to a point where they have already been.

You can modify probtrackx2 parameters if you are familiar with this tool by clicking on the checkbox “Modify”.

- Select server where the tractography is computed – by default the first server found is selected
- Finally click on “Start tractography” button to submit the job to server selected before.

Tractography can take +/- 1 week of computation.

2.3. Jobs results

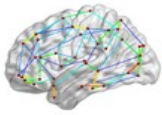
Once a job is submit, a summary of all jobs is available in the tab : Jobs.

Each jobs get a current status, which indicates in which step of the

-Status list :

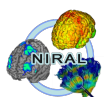
- CREATE : job document is created but it is not running on the computing grid.
- DOWNLOADING : inputs files are downloading to the database.
- RUN : job is runnin on the computing grid
- KILL : job is kill
- FAIL : error in the pipeline, job fail
- UPLOADING : output data are ulploading to the database
- DONE : the job is finish, all data are store in the DB.
- EXIT : if the job failed, the job ended by the status EXIT

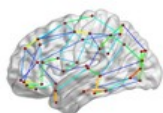
For each jobs launched, some options are available according to the job status (see Appendix 4).



CIVILITY : cloub based Interactive Visualization of Tractography Brain Connectome

3. Brain connectome visualization method





APPENDICES

1. Tractography form

Probabilistic tractography with FSL tools

Job name :

Load data :

DWI Image (.nrrd) :

Aucun fichier choisi

T1 reference in DWI space (.nrrd) :

Aucun fichier choisi

Brain mask in DWI space (.nrrd) :

Aucun fichier choisi

Parcellation table (.json) : [Help](#) ⓘ

Aucun fichier choisi

Inner surface in DWI space (.vtk) :

Aucun fichier choisi

Inner surface contains color labels ☒

Labelset name in vtk surface file :

Ignore label ☐

Extract Label Surfaces options :

Overlapping ☒

Bedpostx options :

By default : number of tensors in the voxel fitting = 2

Command line parameters:

Default : **bedpostx DiffusionDirectory -n 2**

Modify ? ☐ (MODIFY ONLY IF YOU KNOW WHAT TO DO)

[Help](#) ⓘ

Tractography / Probtrackx2 options :

Loopcheck ☒

Command line parameters:

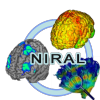
Default : **probtrackx2 --samples=Diffusion.bedpostX/merged --mask=Diffusion.bedpostX/nodif_brain_mask --seed=seeds.txt --seedref=T1_image.nii.gz --forcedir --network --omatrix1 -V 0 --dir=NetworkNameDirectory --stop=seeds.txt (--loopcheck) -P 3000 --steplength=0.75 --sampvox=0.5**

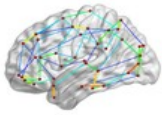
Modify ? ☐ (MODIFY ONLY IF YOU KNOW WHAT TO DO)

[Help](#) ⓘ

Option for submit job :

Select server to run job :





2. Parcellation table

This file is a description of the parcellation table (brain atlas). This is a json object is an array of objects defined like bellow :

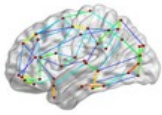
Structure :

```
[
  {
    Object 1 / Region 1
  },
  {
    Object 2 / Region 2
  },
  ...
  {
    Object N / Region N
  }
]
```

Object X :

//Object corresponding to one region – this example is based on the AAL90 parcellation table

```
{
  "VisuOrder": 78,           //This is the rank in the circle plotting
  "MatrixRow": 1,           // Rank in the connectivity matrix - first row = 1 = first column ( if = -1
                             // not in the matrix )
  "name": "Precentral_L",    //Name of the region/seed
  "VisuHierarchy": "seed.left.frontal.", //Hierachy of the seed (for circle plotting)
  "coord": [                 //Coordinates of the seed
    -38.65, // X
    -5.68, // Y
    50.94 // Z
  ],
  "labelValue": "131 44 78", //Value of the label in the vtk file
  "AAL_ID": 1                // label/seed ID in the AAL90 table
}
```



3. Jobs informations examples

1) Unfinished job :

Job name : neo-0332-1-1y

Job id : 213fdaf320ce8bc1aa0ef1828c000314

Start time : Thu Jun 09 2016 13:04:35 GMT-0400 (EDT)



Status : RUN

Kill Job

Print job info

Delete Job in database

2) Finished job :

Job name : 0087-2-1y

Job id : 22521e39a2ded6c2681f896ff400309b

Start time : Thu Jun 02 2016 10:19:16 GMT-0400 (EDT)



Status : DONE

Connectivity visualisation

Download output directory

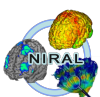
Print job info

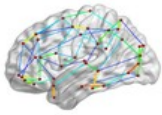
Restart

View log output

View log error output

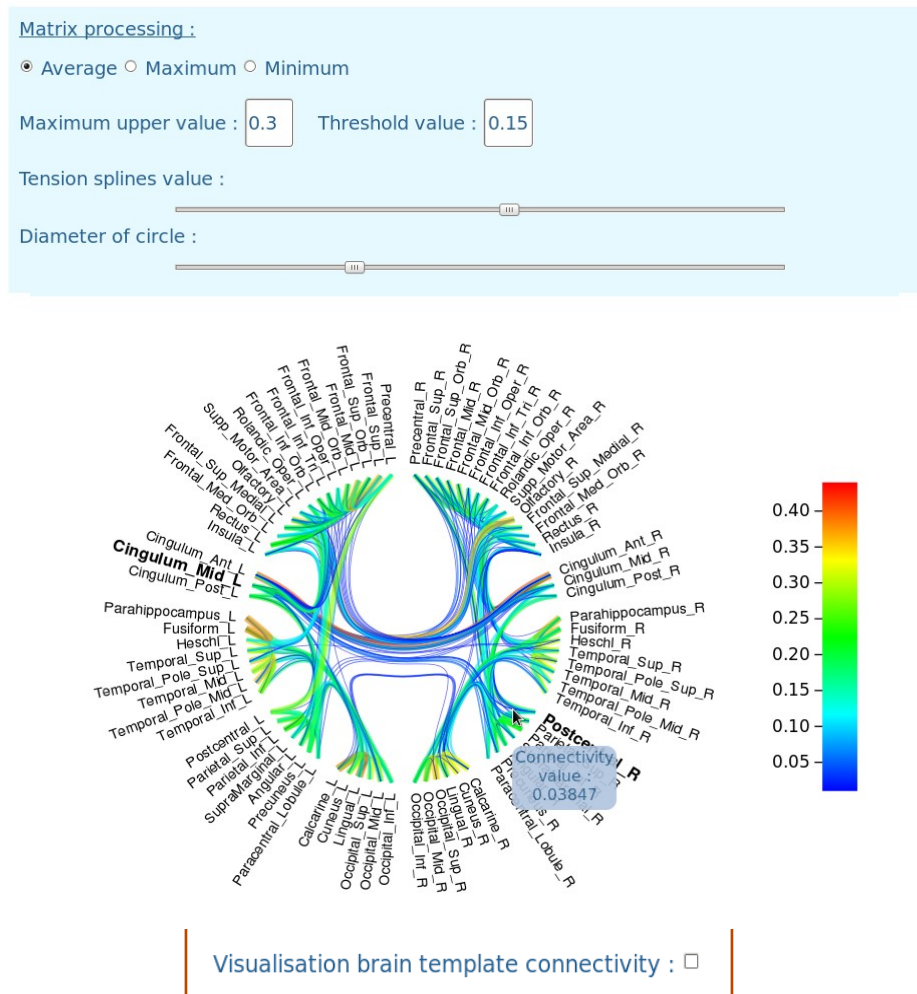
Delete Job in database



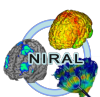
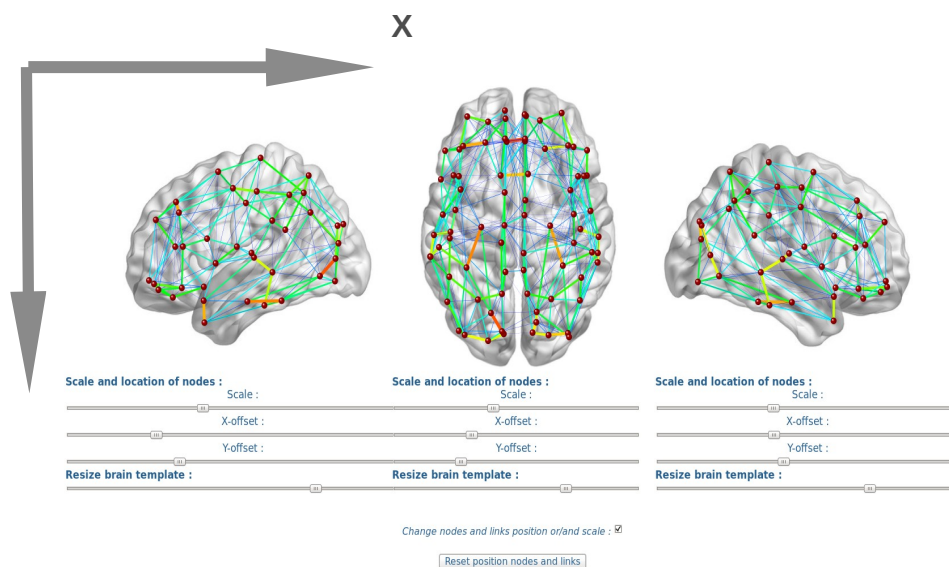


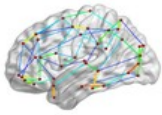
4. Visualization

1) Brain connectome circle plotting



2) Brain connectome Visualization on brain templates





Acknowledgment :

This work is part of the National Alliance for Medical Image Computing (NAMIC), funded by the National Institutes of Health through the NIH Roadmap for Medical Research, Grants U01 MH070890 – U54 HD079124 - R01 MH091351.

UNC : Danaele Puechmaille, Juan Carlos Prieto, Martin Styner

