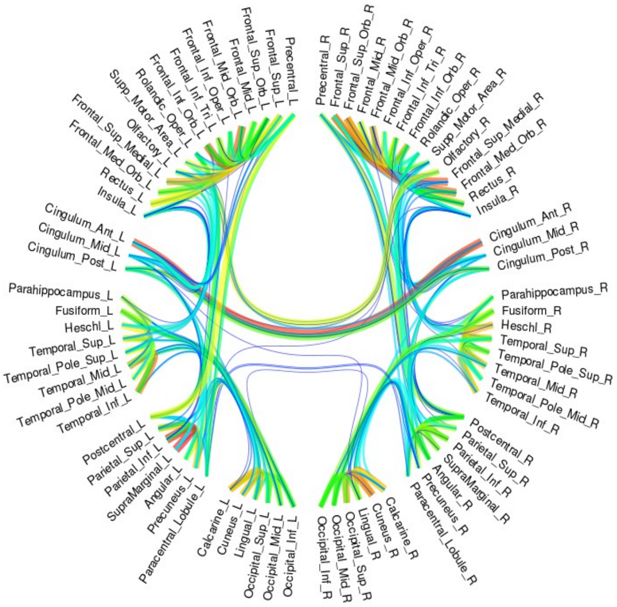
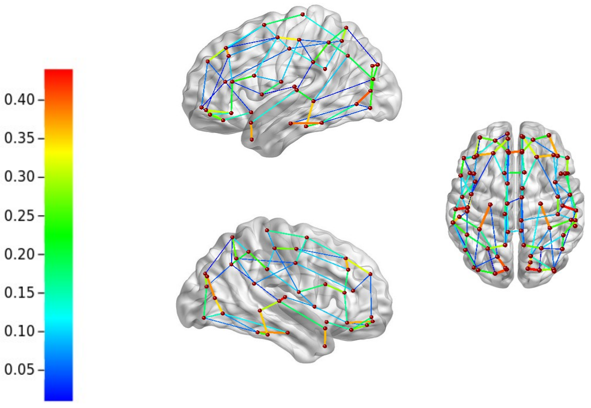
CIVILITY :
cloud based Interactive Visualization of Tractography Brain Connectome

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

Introduction

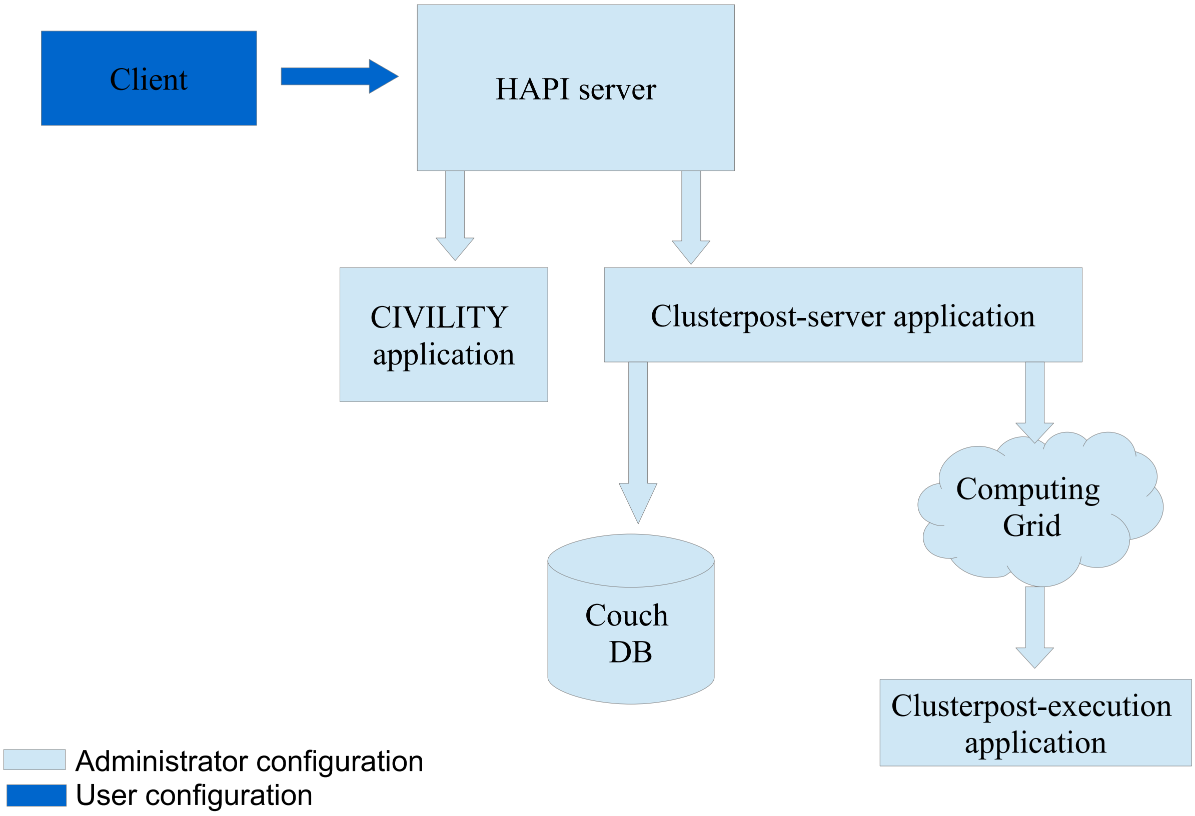
CIVILITY (cloud based Interractive Visualization of Tractography Brain Connectome) is a HAPI plugins used to visualized easily brain connectomes within an interactive web interface. This plugin configured with the HAPI plugin *execution-server* launches tractography scripts on a computing grid and uses a database to store 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

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.


Client
HAPI server
CIVILITY application
Clusterpost-server application
Computing Grid
Couch DB
Clusterpost-execution application
Administrator configuration
User configuration

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. The application CIVILITY requires an other plugin cluterpost-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 cluterpost-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 puglins 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 tractograpy 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*

Once the application is deployed and started by an administrator, each user needs to be connect to the HAPI server.
To configure :

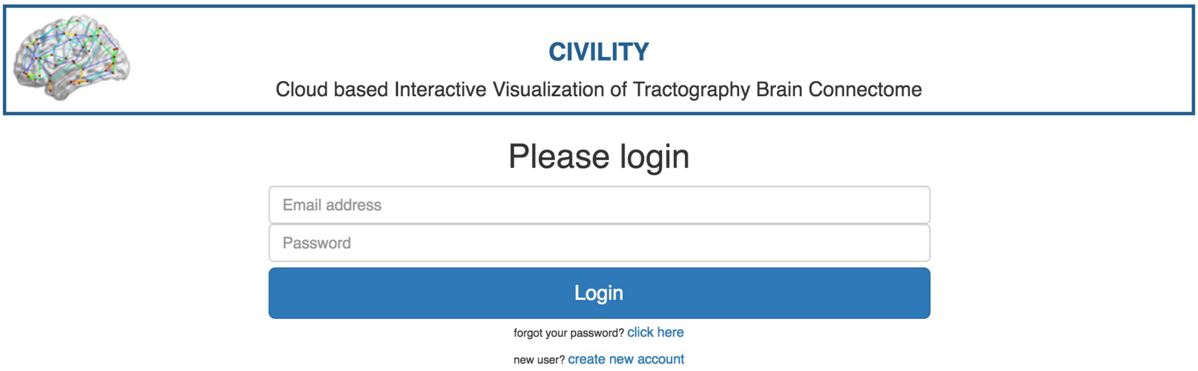
1. Type on terminal :

```
ssh ares.ia.unc.edu -L 8180:localhost:8180
```

2. Open an internet browser and connect to the website
   [***https://localhost:8180***](https://localhost:8180)

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.

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 contains a color label for each regions.
  If the surface doesn't contain label information, you must provide an other surface with label information which must be the same mesh as the Inner surface – in DWI space. (example : you can used the middle surface)

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

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

Pipeline :

- Creation of the diffusion data for using bedpostx
  → create Diffusion directory
  → convert

extract label surfaces documentation : https://github.com/danaele/ExtractLabelSurfaces

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 are available : https://github.com/NIRALUser/ProbtrackBrainConnectivity/tree/master/ScriptsApp/

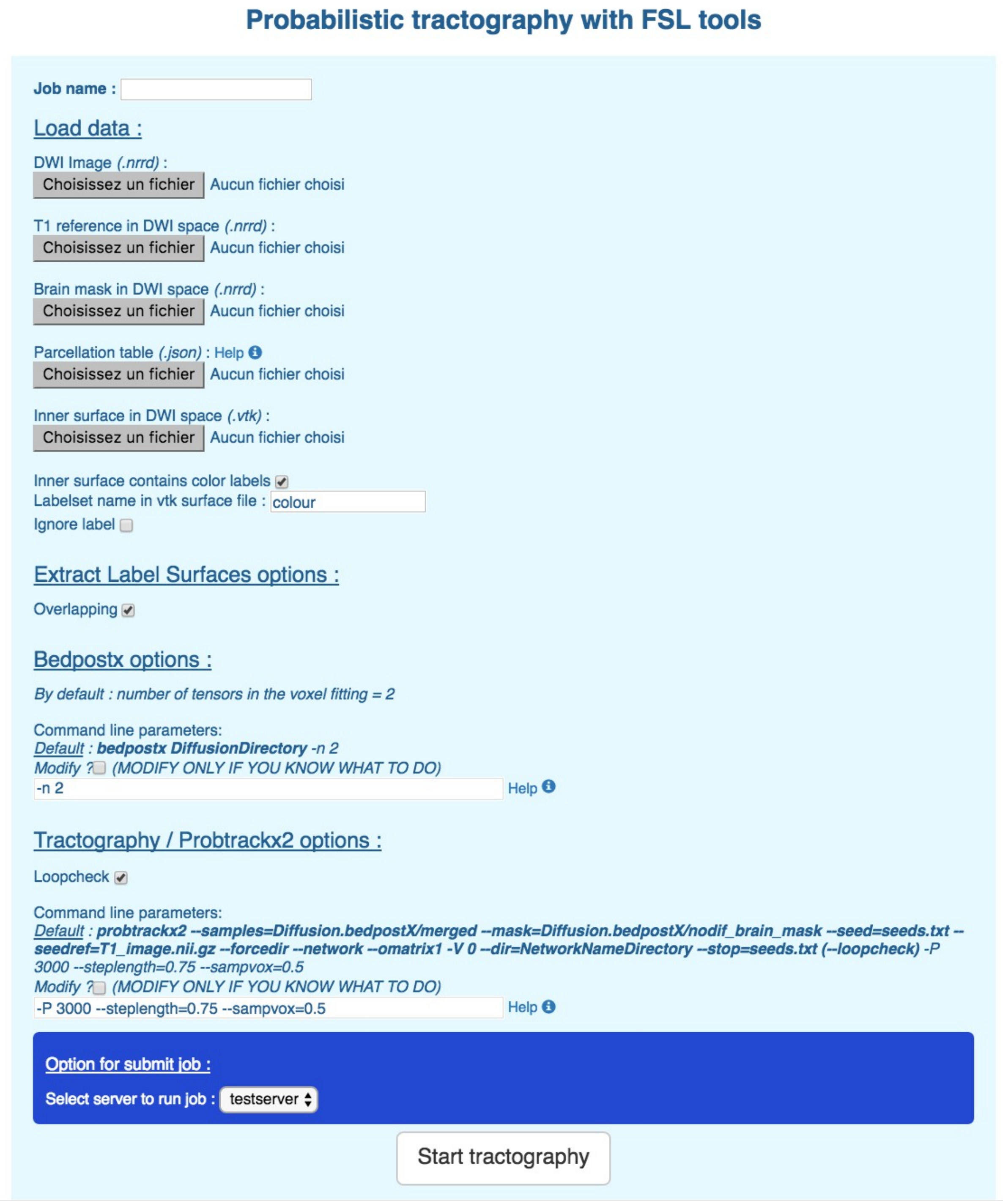
2.2. Run jobs

2.3. Results

3. Brain connectome visualization method

APPENDICES

1. ***Tractography form***



Probabilistic tractography with FSL tools

Job name :

Load data :

DWI Image *(.nrrd)* :
Choisissez un fichier Aucun fichier choisi

T1 reference in DWI space *(.nrrd)* :
Choisissez un fichier Aucun fichier choisi

Brain mask in DWI space *(.nrrd)* :
Choisissez un fichier Aucun fichier choisi

Parcellation table *(.json)* : Help ℹ
Choisissez un fichier Aucun fichier choisi

Inner surface in DWI space *(.vtk)* :
Choisissez un fichier Aucun fichier choisi

Inner surface contains color labels ☑
Labelset name in vtk surface file : colour
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)
-n 2 Help ℹ

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)
-P 3000 --steplength=0.75 --sampvox=0.5 Help ℹ

Option for submit job :

Select server to run job : testserver

Start tractography

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
 }
```