The Functionnectome User-guide

Version 2.5.0

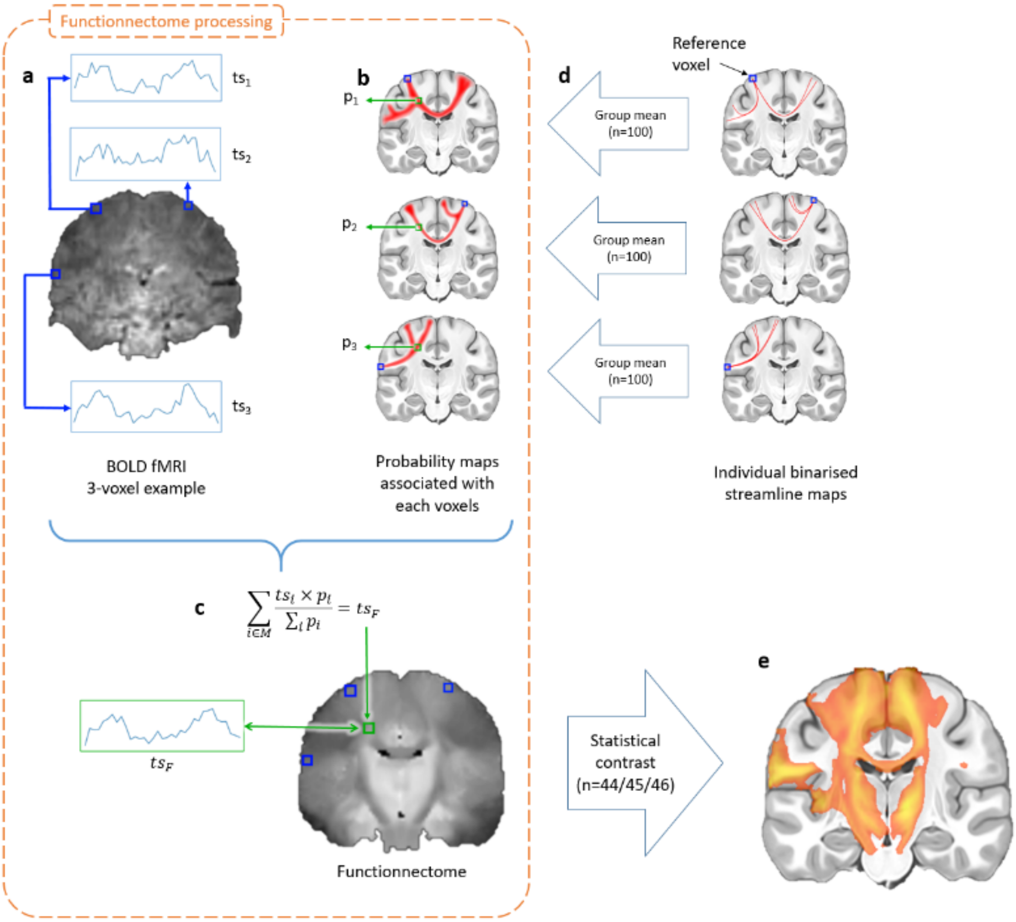
Content

Contact victor.nozais@gmail.com in case of problem with the Functionnectome

Introduction

The Functionnectome is a Python-based tool to allow projecting functional signals (or more generally grey matter information) onto the white matter. As input, the software accepts **NifTI 4D volumes (usually *f*MRI files)**, uses predefined (but modifiable if needed) white matter **priors** for the computation, and **outputs functionnectome 4D volumes**. The user can choose between navigation through commands or a GUI (Graphical User Interface).

The generated functionnectomes 4D volumes can then be processed with the same statistical tools used to analyze fMRI. For example, the functionnectome generated from a classic task fMRI can be analyzed with the usual tools (e.g. generalized linear model) to reveal the white-matter pathways involved with the task.



Disclaimer

No financial conflicts.

Not licensed for medical use.

The software and its codes are licenced under the GNU General Public License.

Definitions

- "Functionnectome" (with an uppercase "F") refers to the overall method and the program applying the method. "functionnectome" (with a lowercase "f") refers to the 4D volume resulting from the application of the Functionnectome.
- "Voxel-wise" analysis refers to the projection of functional signals onto the white matter as this is done at the voxel level. "Region-wise" analysis runs on predefined parcels of brain regions, grouping voxels together and reducing the number of steps to the order of hundreds at most, though at the cost of precision.

Requirements (Read this before starting!)

1. The Functionnectome runs on **Python 3.6** (or higher). In addition to the basic Python distributions, four python libraries are needed: **Numpy, Pandas, Nibabel** and **H5py**. If you install the Functionnectome toolbox using pip (see the "Installation" section), they will be installed automatically. Otherwise, you will have to install them manually on your python virtual environment.
2. Depending on the size of the input data, RAM usage can vary. An 8-12GB RAM is recommended, but for small input volumes, 4GB might suffice.
3. Expected input files are **NifTI (.nii or .nii.gz) 4D volumes in MNI152 space, with 2mm^3 isotropic voxels, with a 3D shape of 91 x 109 x 91 voxels**. Thus, the functional input will be of shape 91 x 109 x 91 x N (with N the number of time-point).
4. Input files should either be:
   - All in the **same folder,** e.g. :
     - *"/myHome/**dataFMRI**/subject01.nii"*
     - *"/myHome/**dataFMRI**/subject02.nii"*
   - Or in **different folders but with "similar paths,"** i.e., with the same number of subfolders and the folder with the data ID at the same depth, e.g. :
     - "*/myHome/**subject01**/data/mydata.nii*"
     - "*/myHome/**subject02**/data/mydata.nii*"
5. For a voxel-wise analysis, a brain (or rather gray-matter) **mask in MNI152 space, with 2mm^3 isotropic voxels,** selecting the voxels with "interesting" functional signal, is also necessary. The mask should be a **NifTI 3D volume**, with the same spatial dimensions as the main 4D volume input (**91 x 109 x 91**).
6. The **white matter anatomical priors** are available online (https://www.dropbox.com/s/22vix4krs2zgtnt/functionnectome_7TpriorsH5.zip?dl=0), but the toolbox gives you also the option to download them automatically.
7. The provided priors require input data to be in the **MNI152 space, with 2mm^3 isotropic voxels with a 3D shape of 91 x 109 x 91 voxels.**
8. We strongly recommend running the Functionnectome on a machine equipped with multiple CPUs (8 cores or more). At least for the voxel-wise analysis. The region-wise analysis is much less computationally demanding. Note that process parallelisation should not increase the RAM load.

Overview

In the GUI, the analysis parameters can be set and the input and the output can be defined. The Functionnectome analysis can be launched from the GUI once the parameters are set. The settings can be saved and loaded for future analysis.

- Input: 4D volume, NifTI, 91 x 109 x 91 x N voxels.
- (opt.) Input mask: 3D volume, NifTI. (required for voxel-wise analysis)
- Output1: the functionnectome, 4D volume, NifTI
  - The user defines the general output folder
  - A subfolder is generated for voxel-wise or region-wise analysis
  - A second subfolder is generated with the unique ID* of the input data
  - The output 4D data is saved there and named "functionnectome.nii.gz"
  - Example: "/myHome/myOutput/voxelwise_analysis/subject01/functionnectome.nii.gz"
- Output2: 3D volume, NifTI.
  - sum_ probaMaps_voxel.nii.gz or sum_ probaMaps_region.nii.gz
  - Stored in the same directory as Output1 (for voxel-wise analysis) or in the general output folder (for region-wise analysis)
  - Sum of all probability maps used for a given subject (depends on the mask)
  - Used during the processing, might be useful for alternative processing
  - Can be deleted after the Functionnectome finished running if you don’t need it

*Unique ID: If all input files are in the same folder, the program uses the filenames (without the extension). If the files are in different folders with similar paths (cf. bullet 4 in ***Requirements***), the ID is the name of the folder (e.g. it will be "subject01" *in the example given above with "/myHome/**subject01**/data/mydata.nii*")

A quick guide to run an analysis:

1. Launch the interface by entering “FunctionnectomeGUI” in a terminal (in the proper Python environment).
2. Select the input 4D files by clicking on either "Choose files" buttons (depending on your data storage organisation).
3. Select whether you want a region-wise or voxel-wise analysis
   - If you chose the voxel-wise approach, select the masks that will filter out unwanted voxels. You can either choose one mask per input file or choose one unique mask applied to all input files.
   - Region-wise is only available with V1 priors (so it’s mostly deprecated).
4. Select the number of parallel processes you want to launch concurrently. The more processors, the faster the analysis. Importantly, the priors should be saved on a storage device with fast reading speed. For an estimation of the time needed to run the Functionnectome on your data, check the FAQ at the end of the manual.
5. Choose the white matter priors. In the choice window, you will have the possibility to manually download any priors file. If the selected priors are not downloaded when you run the Functionnectome, you will be asked to.
6. Choose the output folder. If the provided path doesn't exist, it will be automatically generated.

7. Optionnally, you can provide a white matter mask, which will limit the computation of the Functionnectome to the voxels given by the mask. This will speed-up the process.
8. Launch the analysis. Or save the settings in order to launch it later or with a command line.

Comprehensive guide

Installation

First, you need a basic Python environment, as all the command line below will have to be entered in a terminal with a Pyhton environment set up. You can for example use Miniconda to set up one. Check our video tutorial for more details. You should be able to find it by searching "Functionnectome" on YouToube.

The software can be download through pip using the command line:

- pip install git+https://github.com/NotaCS/Functionnectome.git
  or
- python -m pip install git+https://github.com/NotaCS/Functionnectome.git

To update the Functionnectome to the last version (when already installed):

- pip install --upgrade git+https://github.com/NotaCS/Functionnectome.git
  or
- python -m pip install --upgrade git+https://github.com/NotaCS/Functionnectome.git

NB: For **Mac users**, you might need to install **Xcode**. If this is the case, the installation will fail at first, disaplying an error, but a popup window **should automatically open** offering to install the necessary tool. After installing it, re-run the intall command and it should work.

You can also manually download the project from GitHub:

https://github.com/NotaCS/Functionnectome

The priors will be automatically downloaded during the first launch of the Functionnectome, but you can also download them from our OSF page:

https://www.doi.org/10.17605/OSF.IO/FB9UC

If you download the priors manually, you will need to unzip the file. The path to the priors will be asked by the program during the first run (or can be manually set in the GUI's Priors selection window).

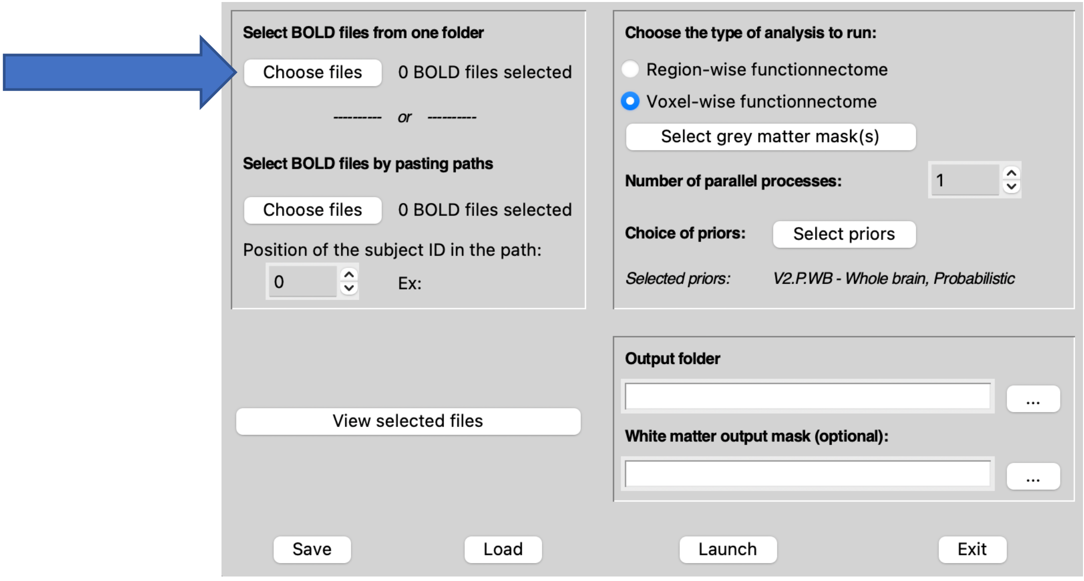
Loading the input 4D volume

Once you have downloaded the package, and have your data ready (**in MNI space, 2mm^3 isotropic voxels**, and a shape of **91**x**109**x**91**x**N**), launch the GUI. To do so, there are 3 possible ways:

- **If the toolbox was installed using pip**,
    1. simply call "FunctionnectomeGUI" (without the quotation marks) in a command window using the virtual environment where you installed the Functionnectome.
- **If you manually downloaded the codes**
    2. run the functionnectome_GUI.py script from a python interpreter
    3. or run "python functionnectome_GUI.py" in a command window.

In the GUI, you can specify the input data (assumed to be BOLD data, but any 4D nifty files are accepted). Two ways of selecting the files are possible depending on how they are stored on your computer:

1. ALL FILES IN 1 FOLDER: If they are all in the same folder, click on the "Choose files" button just under the "Select BOLD files from one folder" label.
   A generic interactive window will open to allow you to choose the files from a folder.



Select BOLD files from one folder

Choose files 0 BOLD files selected

Select BOLD files by pasting paths

Choose files 0 BOLD files selected

Position of the subject ID in the path: 0 Ex:

Choose the type of analysis to run:

○ Region-wise functionnectome

● Voxel-wise functionnectome

Number of parallel processes: 1

Choice of priors: Select priors

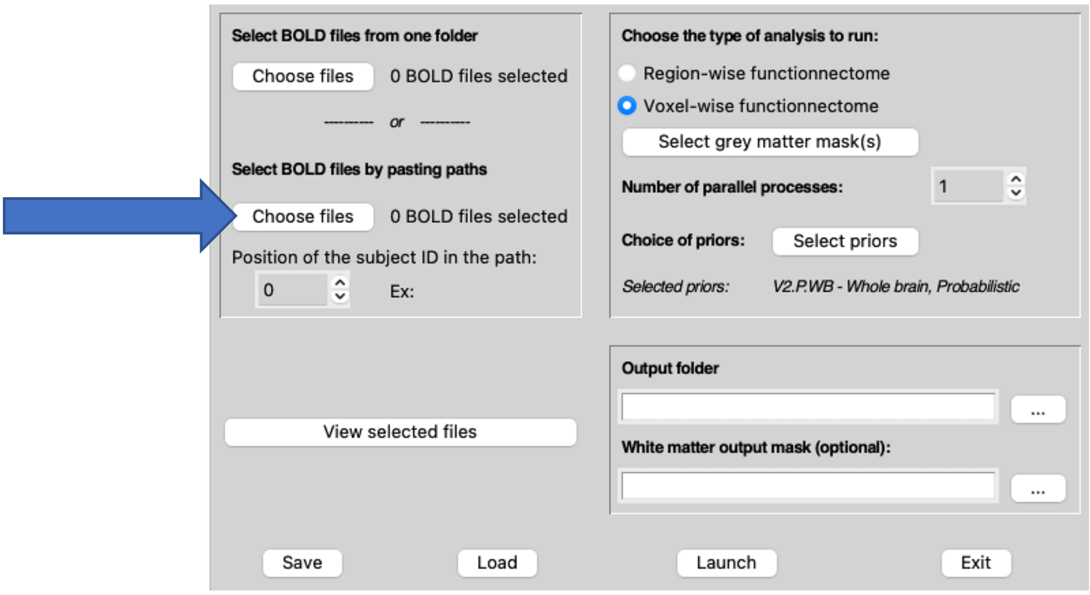
Selected priors: V2.P.WB - Whole brain, Probabilistic

Output folder

White matter output mask (optional):

   You can manually select the files you want to keep/discard. Only NifTI files are accepted (extension ".nii" or ".nii.gz").

2. SEPERATE FOLDERS FOR EACH PARTICIPANT: If the files are in different folders (but properly organised with folders and sub-folders following the same pattern), click on the "Choose files" button under the "Select BOLD files by pasting paths" label.



Select BOLD files from one folder
Choose files
---------- or ----------
Select BOLD files by pasting paths
Choose files
Position of the subject ID in the path:
Ex:
View selected files
Choose the type of analysis to run:
Select grey matter mask(s)
Choice of priors:
Select priors
Output folder
...
White matter output mask (optional):
...
Save
Load
Launch
Exit

0 BOLD files selected

0 BOLD files selected

Position of the subject ID in the path: 0

○ Region-wise functionnectome

● Voxel-wise functionnectome

Number of parallel processes: 1

Selected priors: *V2.P.WB - Whole brain, Probabilistic*

A new window will open. In the central box, you can paste the paths of each file. An easy way to obtain the paths of all the files requires the use of a command window (i.e. a terminal):

- **On MacOS/Unix/Linux**, use the "ls" command. For example, let's say that all my files are stored in paths as: "/myhome/current_project/subject????/functionalFiles/fMRI_acquisition.nii.gz"
  with "????" being a unique ID (e.g a number) that is different for every subject. Use as many "?" as there are digits in the ID numbers. To display all the file-paths, simply enter the command (in the terminal): "**ls -1** /myhome/current_project/subject????/funFiles/fMRI_acquisition.nii.gz"
  Here, "ls" will list all the paths, and the "-1" option will display one path per line.
- **On Windows**, use the "dir" command, similarly to the above-mentioned "ls". For example: "dir C:\myHome\project\subject????\fMRI_acquisition.nii.gz"

Copy the displayed result and paste it into the GUI.

If for some reason you enter or paste something wrong in the GUI text box, you can easily correct it manually or clear the whole box by clicking on "Clear". When you are done, click on "OK" to accept the file-paths. Or "Cancel" to return to the main window without keeping the paths or the changes made to existing paths.

Advanced tip: Once you have files selected, you can manually edit them (add/remove) by opening the "pasting" window and adding ore removing a line, even if you selected the all from one folder. Note that the added paths should still keep the same naming pattern as the other already there. Likewise, if you "Clear" the window and click on "OK", it will delete every path (not the files, of course, just the paths stored by the GUI). If you accidentally modify or clear the paths, simply click on "Cancel".

To differentiate the input files and to name the output folders, a unique ID is attributed to each of them. When selecting all inputs from the same folder (option 1), the unique ID is the name of the file (without the extension). When selecting from multiple similar folders (option 2), the program will automatically select the first divergence in the paths as the ID. The position of the ID in the path is displayed, and an example of ID (from the first selected path) is given. Subsequently, the GUI gives you the option to manually change the location in the paths where the program finds the IDs, using the little arrows near the position number: List of the paths given:

/myHome/workspace/myProject/myAcquisition/MRI/subject01/fMRI/readyData.nii.gz

/myHome/workspace/myProject/myAcquisition/MRI/subject02/fMRI/readyData.nii.gz

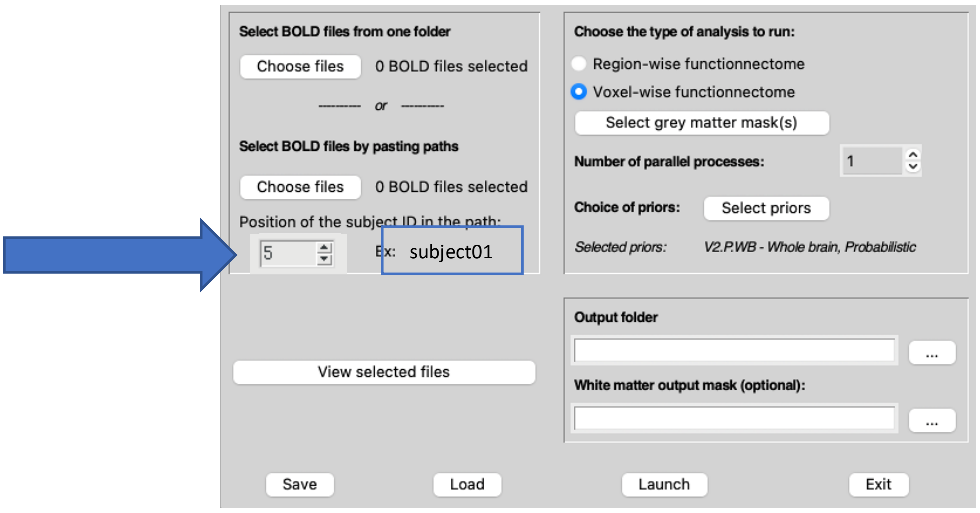
/myHome/workspace/myProject/myAcquisition/MRI/subject03/fMRI/readyData.nii.gz

Position of the subject ID in the path (index of the folder in the path, **starting at "0"**):

/myHome/workspace/myProject/myAcquisition/MRI/subject01/fMRI/readyData.nii.gz

0	1	2	3	4	5	6	7

Here, "subject01" has been automatically selected as it is the first (and only in the example) divergence between the paths.


Select BOLD files from one folder

0 BOLD files selected

Select BOLD files by pasting paths

0 BOLD files selected

Position of the subject ID in the path: 5 Ex: subject01

Choose the type of analysis to run:

○ Region-wise functionnectome

● Voxel-wise functionnectome

Number of parallel processes: 1

Choice of priors:

Selected priors: V2.P.WB - Whole brain, Probabilistic

Output folder

White matter output mask (optional):

Sometimes, the first divergence is not the one you want to keep. You can then manually modify the position. For example, if we have the paths:

/myHome/myProject/session1/subject01/readyData.nii.gz

/myHome/myProject/session1/subject02/readyData.nii.gz

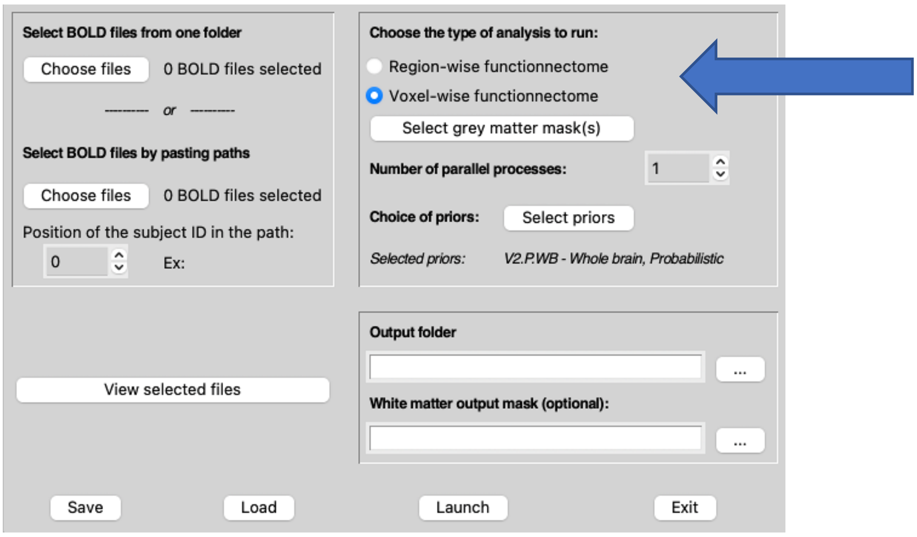
/myHome/myProject/session2/subject03/readyData.nii.gz

/myHome/myProject/session2/subject04/readyData.nii.gz

The number of the session is the first divergence between the paths, so the position will be automatically set at "2", and the example given will be "session1". The user will then have to change the position to "3" (which will change the example to "subject01").

Voxel-wise and region-wise analysis

Select whether you want to run a voxel-wise or a region-wise analysis. Region-wise analysis is much faster, but voxel-wise analysis is much more precise. If you have enough computational power, we recommend using the voxel-wise analysis.



Select BOLD files from one folder

0 BOLD files selected

Select BOLD files by pasting paths

0 BOLD files selected

Position of the subject ID in the path: 0 Ex:

Choose the type of analysis to run:

○ Region-wise functionnectome

● Voxel-wise functionnectome

Number of parallel processes: 1

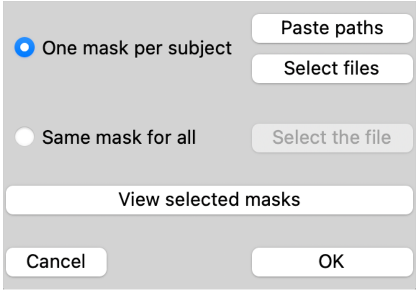
Choice of priors:

Selected priors: V2.P.WB - Whole brain, Probabilistic

Output folder

White matter output mask (optional):

If you select the **voxel-wise** analysis (i.e. "voxel-wise functionnectome"), you will need to specify the mask(s) that will filter out unwanted voxels. Click on "Select mask(s)". A new window will pop open:



● One mask per subject

○ Same mask for all

You can either attribute one mask per input file ("One mask per subject"), usually useful if the masks are created depending on the individual data; or select one unique mask that will be similarly applied to all inputs, for example, a mask delimiting the grey matter.

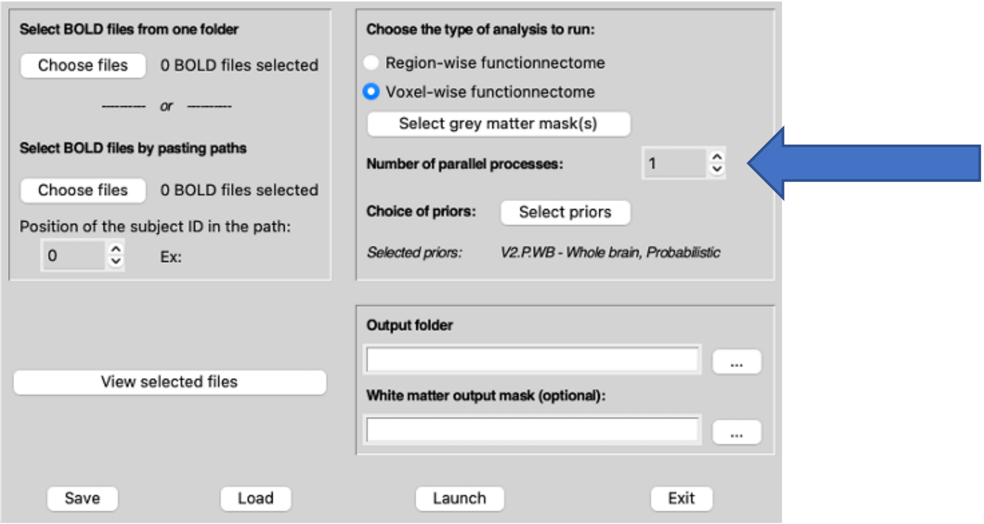
Selecting one mask per input uses the same procedure as the selection of the input 4D volumes (Manual selection in one folder, or pasting the paths).

The paths of **the 4D inputs and the masks** are automatically and systematically **sorted**. Consequently, these **inputs and the masks** must have the **same naming convention** because the pairing between a mask and an input 4D volume is done using the position of the paths in the corresponding list. Using the "View selected files" and "View selected masks" buttons, check the paths to verify that this is correct.

The **region-wise** analysis uses priors derived from a brain parcellation. It groups grey matter voxels in a given region/parcel together, and uses connectivity maps from the regions to the rest of the brain (instead of from each voxel). In the provided default V1 (old) priors, we used the HCP multi-modal parcellation (MMP, doi.org/10.1038/nature18933). You can easily create your own region-wise priors using any voxel-wise priors, an atlas, and the *MakeRegionwiseH5* command (see the FAQ *How can I use region-wise priors?*).

Number of processors to use

The Functionnectome processing time can be significantly cut down by parallelizing it. You can choose the number of processes (i.e. the number of CPU or core used at the same time). The maximum number of parallel processes is automatically detected and set as a top limit. The parallelization happens at the subject level, so only one subject is run at a time, which limits the RAM load.


Select BOLD files from one folder
Choose files
or
Select BOLD files by pasting paths
Choose files
Choose the type of analysis to run:
Select grey matter mask(s)
Select priors
Output folder
...
White matter output mask (optional):
...
View selected files
Save
Load
Launch
Exit

0 BOLD files selected

0 BOLD files selected

Position of the subject ID in the path: 0 Ex:

○ Region-wise functionnectome

● Voxel-wise functionnectome

Number of parallel processes: 1

Choice of priors:

Selected priors: V2.P.WB - Whole brain, Probabilistic

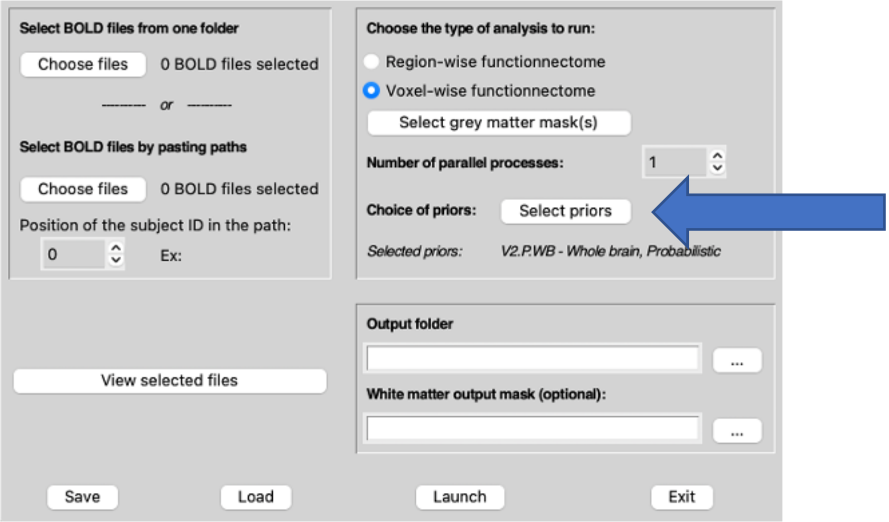
The Functionnectome parallel processing system does not support the spread of the computation across multiple machines/nodes. All the CPUs must be mounted on the same machine.

As an example, using 16 processes in parallel, the processing time of a typical *f*MRI scan with the voxel-wise analysis can be around 2 hours (but it depends a lot on the size of the input and how many voxels are kept with the mask).

White matter anatomical priors

As the Functionectome projects BOLD onto the brain’s connections, the white matter priors are necessary. They consist of the white matter probability maps and a brain template. The template is a T1 brain template and is used to define the dimensions of the images and to delimit the brain. We provide the priors, but they must be downloaded separatly from the Functionnectome. If you want to create your own priors Check the FAQ "*How do I specify different paths to alternate priors?*".

To Select the priors to use, click on “Select priors”:


Select BOLD files from one folder
Choose files
or
Select BOLD files by pasting paths
Choose files
View selected files
Choose the type of analysis to run:
Select grey matter mask(s)
Select priors
Output folder
...
White matter output mask (optional):
...
Save
Load
Launch
Exit

0 BOLD files selected

0 BOLD files selected

Position of the subject ID in the path: 0 Ex:

○ Region-wise functionnectome
● Voxel-wise functionnectome

Number of parallel processes: 1

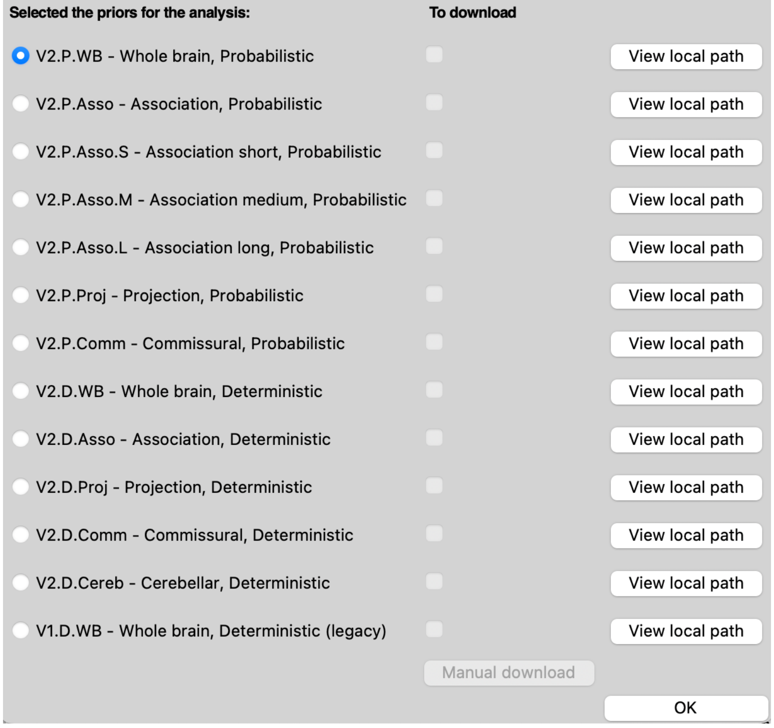
Choice of priors:

Selected priors: V2.P.WB - Whole brain, Probabilistic

A new window will open, where you can chose between different sets of priors.

There, you can select the priors you want. You can also download any priors by checking the check-box on the right of the priors' name then clicking on the "Manual download" button (or select the file by clicking on the corresponding Select button).

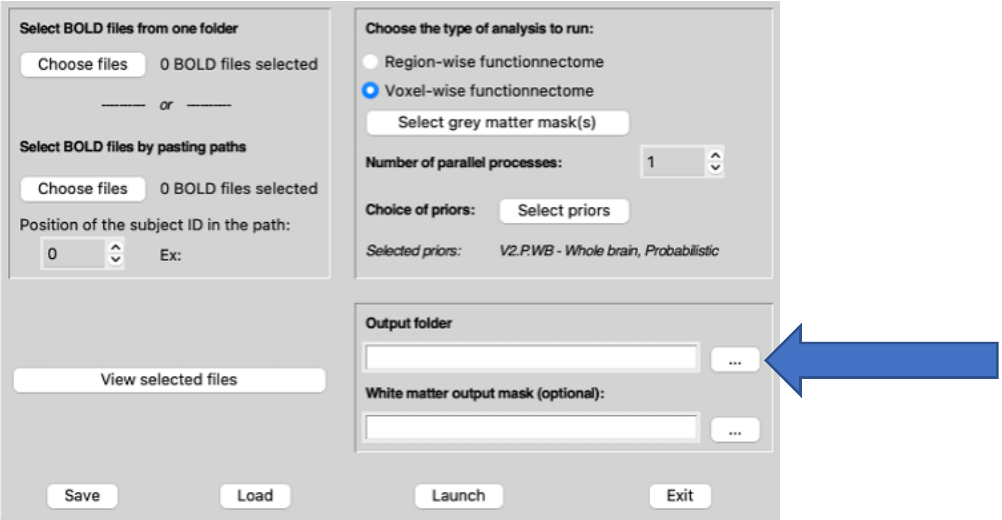
V1 or V2 denote the version (we recommend using V2 priors). **Probabilistic or Deterministic** denotes the type of algorithm used to do the tractography. We generally recommend Probabilistic. The last choice is whether you want to use priors derived from **whole brain** tractograms (the whole connectome of the brain), or tractograms filtered to only keep specific connections (**Association, Commissural, or Projection** fibers). We also provide sub-devided Association priors using fiber length. These "split" priors minimize fiber-crossing areas, which may be problematic when using the Functionnectome**. Ideally, you should run the Functionnectome with each type of split priors** to then reconstruct the whole brain functionnectome. However, this requiresto re-run the Functionnectome for each type of priors, which can be quite long.



Selected the priors for the analysis:	To download	
● V2.P.WB - Whole brain, Probabilistic	□	View local path
○ V2.P.Asso - Association, Probabilistic	□	View local path
○ V2.P.Asso.S - Association short, Probabilistic	□	View local path
○ V2.P.Asso.M - Association medium, Probabilistic	□	View local path
○ V2.P.Asso.L - Association long, Probabilistic	□	View local path
○ V2.P.Proj - Projection, Probabilistic	□	View local path
○ V2.P.Comm - Commissural, Probabilistic	□	View local path
○ V2.D.WB - Whole brain, Deterministic	□	View local path
○ V2.D.Asso - Association, Deterministic	□	View local path
○ V2.D.Proj - Projection, Deterministic	□	View local path
○ V2.D.Comm - Commissural, Deterministic	□	View local path
○ V2.D.Cereb - Cerebellar, Deterministic	□	View local path
○ V1.D.WB - Whole brain, Deterministic (legacy)	□	View local path

Output options

Choose the output folder by either clicking on the "..." button or by copy-and-pasting the path to the text-box.


Select BOLD files from one folder
Choose files
0 BOLD files selected
or
Select BOLD files by pasting paths
Choose files
0 BOLD files selected
Position of the subject ID in the path:
Ex:
View selected files
Choose the type of analysis to run:
Region-wise functionnectome
Voxel-wise functionnectome
Select grey matter mask(s)
Choice of priors:
Select priors
Output folder
...
White matter output mask (optional):
...
Save
Load
Launch
Exit

○ Region-wise functionnectome
● Voxel-wise functionnectome

Position of the subject ID in the path: 0

Number of parallel processes: 1

Selected priors: *V2.P.WB - Whole brain, Probabilistic*

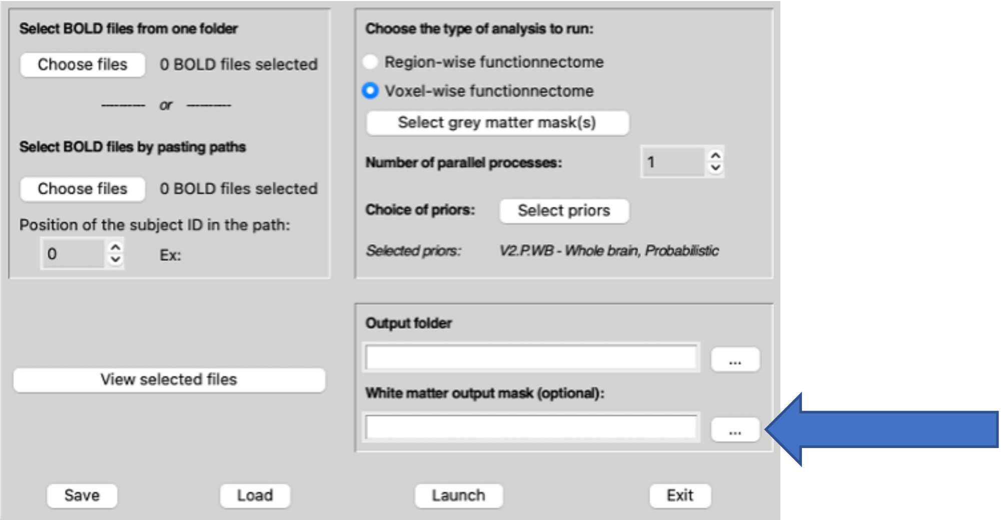
If the folder does not exist, it will be automatically created. It will serve as a "root" folder for the analysis, where the settings file (resulting from the parameters you entered in the GUI) and the additional sub-folders will be stored. Multiple analyses can use the same output folder, as long as the IDs of the input data are different.

Typically, if you select "/myHome/myProject/myOutputFolder/" as the path to the output folder, the results will be stored in this structure (assuming you are doing a voxel-wise analysis):

/myHome/myProject/myOutputFolder/

- settings.fcntm (text file containing the analysis parameters)
- voxelwise_analysis/
  - subject01/
    - functionnectome.nii.gz
    - sum_probaMaps_voxel.nii.gz
  - subject02/
    - functionnectome.nii.gz
    - sum_probaMaps_voxel.nii.gz
  - subject03/
    - functionnectome.nii.gz
    - sum_probaMaps_voxel.nii.gz
  - subject04/
    - functionnectome.nii.gz
    - sum_probaMaps_voxel.nii.gz

The last parameter available is the "White matter " option. There, you can give the path to a white matter mask. If filled, the Functionnectome will only compute the values of the voxels in the mask. It can drammatically speed-up the process.


Select BOLD files from one folder
Choose files
or
Select BOLD files by pasting paths
Choose files
View selected files
Choose the type of analysis to run:
Select grey matter mask(s)
Select priors
Output folder
...
White matter output mask (optional):
...
Save
Load
Launch
Exit

0 BOLD files selected

0 BOLD files selected

Position of the subject ID in the path: 0

Ex:

○ Region-wise functionnectome

● Voxel-wise functionnectome

Number of parallel processes: 1

Choice of priors:

Selected priors: V2.P.WB - Whole brain, Probabilistic

Save, Load, Launch & Exit

The buttons at the bottom of the GUI:

- Save: Saves the parameters entered on the GUI in a simple text file (with a "*.fcntm*" extension). It can be opened with any text editor. This settings file can be used to manually launch the Functionnectome from a terminal. All the parameters must be properly entered in the GUI before saving is possible.
- Load: Loads a settings file ("*.fcntm*") to fill the GUI parameters. Useful if you want to run a different analysis using the same input 4D files.
- Launch: Saves the parameters in the output folder, closes the GUI, and runs the analysis using the provided settings.
- Exit: Closes the GUI without doing anything.

Frequently Asked Questions and advanced functions

How long will the software take to process my data?

It really depends on a lot of factors. The size of the input data, but also the type of processors, the type of storing device on which the priors are written, and the number of processes all play a part.

Just to give you an example, the voxel-wise analysis of an fMRI with 284 time-points (so with dimensions of 91 x 109 x 91 x 284) on our machines, with 16 parallel processes, took about 10 minutes. Using fewer parallel processes or volumes with more time-points will increase that time, but not linearly. Using the ETA of the running analysis of one dataset can help you estimate how long it would take on your machine.

The Functionnectome is too slow, what can I do?

Here are a few things you can try:

- Increase the number of parallel processes.
- Provide a white matter mask to limit the number of computed voxels (option just below the output folder).
- Reduce the compression of the priors using the "DecompressH5" command (if available).
- Move the priors to a faster drive.
- Run the process on a more recent machine.
- Use region-wise priors (much much faster, but less precise)
- Contact me by opening an "Issue" on github.

How can I use the region-wise analysis?

First, you will need to create the region-wise priors. Don't worry, it's easy and relatively fast, you only need a priors HDF5 file (e.g. one of the default ones we provide) and an atlas in MNI space (with each region/parcel labeled by an integer.

Use the *MakeRegionwiseH5* command (you can display the help by calling "*MakeRegionwiseH5 -h*"). It transforms the input voxel-wise priors into the corresponding region-wise priors based on the atlas you provide. Where it saves it depends on the "mode" you chose ('create', 'append' or 'replace'):

- create : Creates a new HDF5 file with only the new region-wise priors inside.
- append : Add the region-wise priors to the input file (if it didn't have them). This is typically what you will use with the default priors to update them, as they don't have region-wise maps.
- replace : Equivalent to 'append', but overwrite any existing region-wise priors in the input file.

For example, to add (**append**) region-wise priors to the **whole-brain probabilistic priors** (labeled **V2.P.WP**, as shown in the GUI), using **/myProject/myAtlas.nii.gz** as the input atlas, with **4** parallel processes (to speed-up the computation), the command would be:

MakeRegionwiseH5 -pi V2.P.WP -a '/myProject/myAtlas.nii.gz' -m append -c 4

Once the file is created/updated, you can use it for a region-wise analysis, it can be selected directly on the GUI.

How do I use the Functionnectome with command lines only?

There are two ways.

1. You first need to create a settings file for your analysis (see below "Can I manually change/create a setting file?"). Let's assume it is called "settings.fcntm" and saved in /myPath/

   Then:

   - If you installed the Functionnectome package using pip:
     - You can directly call "Functionnectome" in a command window (with the correct virtual environment) followed by the path to the settings file, e.g. "Functionnectome */myPath/settings.fcntm*".
     - You could also create a simple script importing the package, then running the function. For example, something like that:

       ```
       from Functionnectome.functionnectome import
       run_functionnectome
       fpath="/myPath/settings.fcntm"
       run_functionnectome(fpath)
       ```
   - If you manually downloaded the scripts and saved it in, e.g. /myPath/, you can run the script by simply calling:

     python -u /myPath/Functionnectome/functionnectome.py /myPath/settings.fcntm

2. If you have installed the package, you can use the "*Funtome*" command line in a terminal (use "functome -h") to get the help text. This command takes as arguments all the input needed to run the Functionnectome. Beware, however, of bad arguments. There are fewer checks on the input on this way of calling the Functionnectome. Also, white spaces in file paths will break the argument, so if you have some, frame each path with quote marks (Funtome -of "/myHome/my folder").

Can I manually change/create a setting file?

Sure, it is a simple text file on which we changed the extension to ".fcntm". You can create your own file in a text editor (or copy a saved settings file from the GUI), you just need to make sure that the internal architecture is the same as in a GUI generated file.

Don't forget to have the "Number of subjects" and "Number of masks" values matching the respective numbers of paths you provide in the file.

The Functionnectome program possesses a simple system to check if the file is correctly organized, so if your custom settings file is completely off the mark, you'll get a notification.

Note that the "Position of the subjects ID in their path" starts at 0 (first folder). The last part of the path (i.e. the name of the file) can be designated with -1. This “-1” index is actually what is used when loading all the 4D input files from a single directory.

You can find examples of settings files in the project directory.

How do I specify different paths to alternate priors?

To give non-default priors or override the absence of default priors, you can use the “Custom priors” choice in the Priors selection window of the GUI. Alternatively, you can edit an existing settings file (with the ".fcntm" extension) and copy and paste the following lines at the end of the file (add an empty line before the first "###" in the file):

```
###
HDF5 path:
	/myHome/myProject/priors/functionnectome_priors.h5
Template path:
	/myHome/myProject/anat/brainT1template.nii.gz
Probability maps (voxel) path:
	/myHome/myProject/priors/probaMaps_voxels
Probability maps (region) path:
	/myHome/myProject/priors/probaMaps_atlas
Region masks path:
	/myHome/myProject/priors/decomposed_atlas
###
```

The labels, written in green above, must always be present. Conversely, you only need to fill in the paths (in blue above) that are useful. For example, if you are using an HDF5 priors file, you only need to enter the first path. The lines where the unnecessary path would have been written must be kept though, with empty lines:

```
###
HDF5 path:
	/myHome/myProject/priors/functionnectome_priors.h5
Template path:

Probability maps (voxel) path:

Probability maps (region) path:

Region masks path:

###
```

An example file using this option can be found in the project directory:

- settings_example2.fcntm

If you are using the “Funtome” command line, you can simply give the path of the custom priors where you would have given the priors label.

How do I make my own anatomical priors?

We originally used the "disconnectome" tool of the BCBtoolkit - http://toolkit.bcblab.com/ -, with one-voxel masks for the "lesions", for every brain voxel.

We have now a Python script for that purpose, using the scilpy library (a Python library developed by the Sherbrooke Connectivity Imaging Lab, based on DIPY). The script is available on our scilpy fork (https://github.com/NotaCS/scilpy), in the script folder. It is called *scil_make_priors_functionnectome.py*. Once you have installed the library, you can get some help on how to use the script by calling “scil_make_priors_functionnectome.py -h” in your terminal. Note that it is still a work in progress, but it is working fine on our end.

In essence, you are free to generate any kind of probability maps that reflect the probability of connection from each voxel to the rest of the brain, using the method of you choice. On that subject, we encourage you to build your own priors to better fit the type of analysis you are planning to do, or simply try and improve upon ours priors.

Note that the tractography used to generate the priors provided here used either a deterministic (V1.D.WB or V2.D.- priors) or a probabilistic (V2.P.- priors) approach.

How do I create my own HDF5 file for my priors?

We provide a script to export the priors in NIfTI format into a single HDF5 file.

If you installed the Functionnectome package through pip, you can directly call “MakeH5” from a terminal. Otherwise, you can find the script in the project/package under the name makeHDF5prior.py.

The command requires 5 arguments, each being the path to one of the input data. For example:

```
MakeH5 -o /myProject/myPriors.h5 -tpl /myData/brainT1template.nii.gz -vxl /myData/myVoxelwisePriors -reg /myData/myRegionwisePriors -rmas /myData/myParcellation
```

With:

- */myProject/myPriors.h5* the output path to the generated HDF5 file.
- */myData/brainT1template.nii.gz* the input path to the brain template file (used to mask the brain)
- */myData/myVoxelwisePriors* the input path to the folder containing the voxel-wise priors
- */myData/myRegionwisePriors* the input path to the folder containing the region-wise priors
- */myData/myParcellation* the input path to the folder containing the masks of the brain regions used to generate the region-wise priors

Region-wise priors can be ommitted, and can be added later using the *MakeRegionwiseH5* command.

The script will detect all the NifTI files in the given folders, and import them into the HDF5 file. The **voxel-wise probability maps** files **must** all be named with the following pattern: something_xx_yy_zz_vox.nii.gz
"something" can be any string of character, but ***must not contain a "_".*** "xx", "yy", and "zz" are the spatial coordinates of the specific voxel linked to the given probability map (e.g something_20_102_7.nii.gz). Be aware that these are the "raw" coordinates as imported by Nibabel, before applying the transformation matrix (affine) given in the NifTI header.

The **region-wise probability maps** and their associated **region masks** **must** have the same name.

For example, the two probability maps

*/myData/myRegionwisePriors/*region1.nii.gz
and
*/myData/myRegionwisePriors/*region2.nii.gz

must be associated with the region masks

*/myData/myParcellation/*region1.nii.gz
and
*/myData/myParcellation/*region2.nii.gz
respectively.

(The paths can be absolute or relative paths, both are accepted by the script)

How do I use the Functionnectome on a distant computational grid/server?

The simplest way would be to transfer the functionnectome.py script and the priors (in the HDF5 file) to the machine you plan to use. Prepare manually the settings file (see "*Can I manually change / create a setting file?*") and give the location of the HDF5 prior on the distant machine as an alternate priors file (see "*How do I specify different paths to alternate priors?*"). Then run the script with the settings file as the argument (see "*How do I use the Functionnectome with command lines only?").*

What is that sum_ probaMaps file that I get?

As described in the *Overview* paragraph, sum_probaMaps_voxel.nii.gz (or sum_probaMaps_regions.nii.gz) is a secondary output of the Functionnectome. It results from the sum of all probability maps corresponding to the voxels inside the grey matter mask given by the user as input. This file is used in a normalizing step, and might be of use in some other case, but you can delete it if you don't think you need it. It can easily be re-generated by launching the Functionnectome with the same mask used originally, as the creation of this file is the first step of the process (when you see that the file has been generated, you can cancel the process).

Will this work on my OS?

The scripts have been tested on a Linux machine and on a Mac. It hasn't been tested on Windows yet, but was coded with portability in mind, so it might work. Proper tests on different OS are planned (one day, maybe…).

Will parallelizing the analysis use more RAM?

No. Or at least at least it shouldn't. The script uses shared memory between processes to avoid RAM overload, and more processes should only split the data in smaller chunks, not copy it multiple times.

I have a problem / I think you are doing something wrong or poorly optimized…

Don’t hesitate to open an issue on the github page, or simply contact me at victor.nozais@gmail.com.