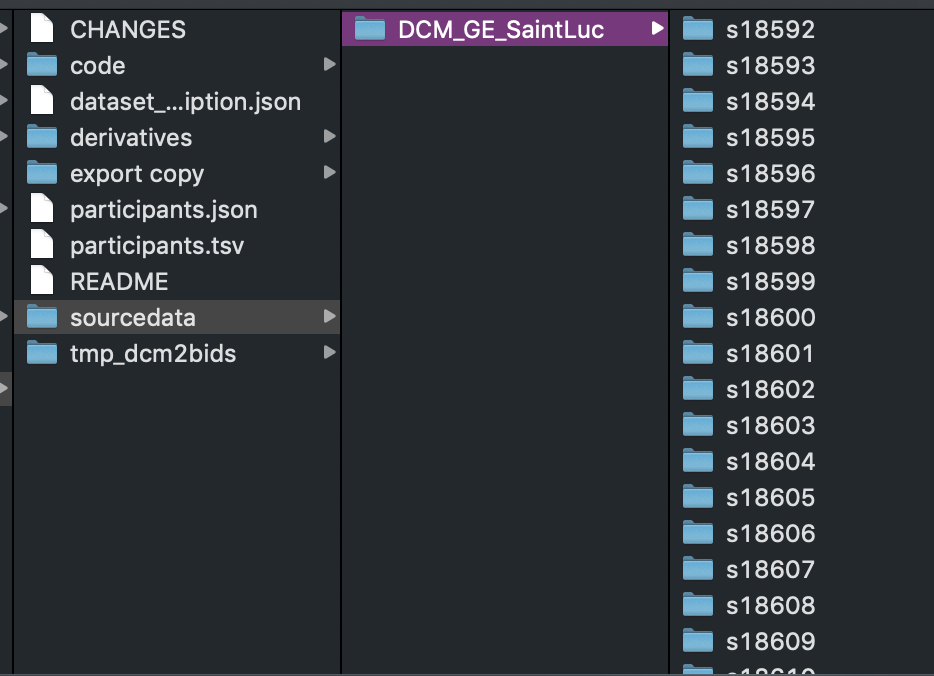
HOW TO CREATE A BIDS COMPATIBLE DATASET (using dcm2bids)

*A tutorial is offered on the dcm2bids github page:* [*https://unfmontreal.github.io/Dcm2Bids/docs/tutorial/first-steps/*](https://unfmontreal.github.io/Dcm2Bids/docs/tutorial/first-steps/)

At this link youfind all the instruction to install dcm2bids and its dependencies: <https://unfmontreal.github.io/Dcm2Bids/docs/get-started/install/>

Now let’s start to use dcm2bids to bidsify your data:

* Create a folder for your BIDS\_project (e.g.”VWFA\_BIDS\_project”);
* Open the terminal >> conda activate dcm2bids
* Cd in the folder e.g >> cd /Users/cerenbattal/Documents/data/VWFA\_signLanguage/VWFA\_BIDS\_project2
* (To check your current working directory type: pwd )
* Create the typical scaffold for the BIDS structure >> dcm2bids\_scaffold (this command is only creating a structure of empty folders, you need to do that only for your sub1).
* Copy into the “sourcedata” folder the source data acquired from the scanner: e.g.”DCM-GE\_SaintLuc”.
* 
* Convert the dicom files to nifti files and create the sidecar Jason file for each file: on the terminal type>> dcm2bids\_helper -d sourcedata/DCM\_GE\_SaintLuc (this command will create a temporary folder “temp\_dcm2bids” in which all the converted files + json files are stored).

Note that if you already ran the conversion on one or more subject and you did not empty the temporary folder “tmp\_dcm2bids”, you will get this error when running this step:

dcm2bids\_helper: error: Output directory /Documents/data/VWFA\_signLanguage/VWFA\_BIDS\_project2/tmp\_dcm2bids/helper isn't empty, so some files could be overwritten or deleted.

Rerun the command with --force option to overwrite existing output files.

What I do is to manually delete the temporary folder after the conversion of each subject. If you are not sure if you did it well and you would like to keep this folder, maybe change the folder name (e.g. “tmp\_dcm2bids\_sub01”) and move it in another folder. Then, create a new “tmp\_dcm2bids” empty folder in your project folder. Once you have this, you can re-run the command e.g. >>dcm2bids\_helper -d sourcedata/DCM\_GE\_SaintLuc

* Prepare the configuration file:
* This is a json file where you specify which nifti file you want to convert and which name to assign them.
* Here you find an [example of the configuration file](https://www.dropbox.com/s/jvbh4y6gtdqjr21/dcm2bids_config.json?dl=0) for my experiment, in which I wanted to select and convert: one anatomical file, 2 functional for a localizer task and 12 functional for an MVPA-sign language task.
* You can check that your json file is valid on an online validator: e.g  <https://jsonlint.com>.
* A good location for the configuration file is in the ‘code’ folder (e.g: *VWFA\_BIDS\_project/code/dcm2bids\_config.json*)

Graphical user interface, text

Description automatically generated

* Once your configuration file is ready you can run the dcm2bids on the terminal: dcm2bids -d path/to/source/data -p subject\_id -c path/to/config/file.json

The exact command I used for my project is: dcm2bids -d sourcedata/DCM\_GE\_SaintLuc/sub-HLS03 -p HLS03 -c code/dcm2bids\_config.json

* Running this command you should create a folder in your project folder with the subject ID name (in my case it will be a folder named: *sub-HNS02)* in which you should find the data you specified in your configuration file now converted in BIDS format.
* In addition, you will find also in the temporary folder a folder named as the subject ID in which you can find all the temporary files that have NOT been converted into BIDS.
* If you have functional data with a task (no resting-state) you should also include in the BIDS data folder a .tsv file for each run including the onset and duration of each events. Find [here](https://www.dropbox.com/s/qms65kuk3a3vau8/sub-HNS01_task-loc_run-01_events.tsv?dl=0) an example for one of my functional run (localizer task).
* Finally use an online [BIDS validator](https://bids-standard.github.io/bids-validator/) to check that your data structure is now BIDS compatible.