spine-generic Documentation

Julien Cohen-Adad

Sep 18, 2023

Table of contents

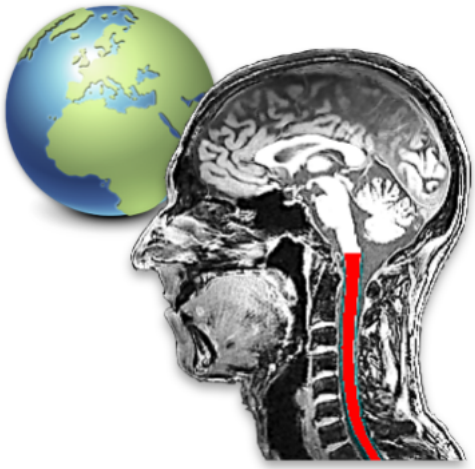
1 Data collection and organization **3**
- 1.1 Imaging Protocol . . . 3
- 1.2 Multi-center data . . . 3
- 1.3 Data conversion: DICOM to BIDS . . . 3
- 1.4 Checking acquisition parameters . . . 6
- 1.5 Ethics and anonymization . . . 6
- 1.6 Example of datasets . . . 7

2 Analysis pipeline **9**
- 2.1 Dependencies . . . 9
- 2.2 Getting started . . . 9
- 2.3 Quality Control . . . 10
- 2.4 Generate figures . . . 12
- 2.5 Results . . . 13

3 Body size pipeline **15**
- 3.1 Dependencies . . . 15
- 3.2 Spinal cord image analysis . . . 15
- 3.3 Cerebral image analysis . . . 16
- 3.4 Generate figures . . . 16
- 3.5 Results . . . 16

4 Contributors **17**

5 License **19**



Quantitative spinal cord MRI is fraught with a number of challenges, among which is the lack of standardized imaging protocols. Here we present quantitative MRI protocols (which, collectively, we refer to as the "spine generic protocol") for the three main MRI vendors (GE, Philips and Siemens), that provide valuable metrics for assessing spinal cord macrostructural and microstructural integrity: T1w and T2w imaging for SC cross-sectional area (CSA) computation, multi-echo gradient echo for gray matter CSA, as well as magnetization transfer and diffusion weighted imaging for assessing white matter microstructure. The spine generic protocol was used in a single-subject reproducibility study across 19+ centers (spanning all three vendors) and in multiple subjects across 42+ centers.

The protocols and datasets are open-access and the data analysis code is open-source. Full documentation, including a description of the processing pipeline with detailed text and video instructions can be found in this website.

The main publications for this project are:

- Cohen-Adad et al. *Generic acquisition protocol for quantitative MRI of the spinal cord*. **Nature Protocols** 2021 (doi: 10.1038/s41596-021-00588-0). PDF
- Cohen-Adad et al. *Open-access quantitative MRI data of the spinal cord and reproducibility across participants, sites and manufacturers*. **Scientific Data** 2021 (doi: 10.1038/s41597-021-01026-2). PDF

CHAPTER 1

Data collection and organization

1.1 Imaging Protocol

The "Spine Generic" MRI acquisition protocol is available at this link. Each site was instructed to scan six healthy subjects (3 men, 3 women), aged between 20 and 40 y.o. Note: there was some flexibility in terms of number of participants and age range.

If your site is interested in contributing to the publicly-available database, please coordinate with Julien Cohen-Adad.

1.2 Multi-center data

In the context of this project, the following dataset have been acquired and are available as open-access:

- Multi-center, single-subject
- Multi-center, multi-subject

1.3 Data conversion: DICOM to BIDS

To facilitate the collection, sharing and processing of data, we use the BIDS standard. An example of the data structure for one center is shown below:

```
data-multi-subject
│
├── dataset_description.json
├── participants.json
├── participants.tsv
├── sub-ubc01
├── sub-ubc02
├── sub-ubc03
├── sub-ubc04
```

(continues on next page)

(continued from previous page)

```
├── sub-ubc05
├── sub-ubc06
│   │
│   ├── anat
│   │   ├── sub-ubc06_T1w.json
│   │   ├── sub-ubc06_T1w.nii.gz
│   │   ├── sub-ubc06_T2star.json
│   │   ├── sub-ubc06_T2star.nii.gz
│   │   ├── sub-ubc06_T2w.json
│   │   ├── sub-ubc06_T2w.nii.gz
│   │   ├── sub-ubc06_flip-1_mt-off_MTS.json
│   │   ├── sub-ubc06_flip-1_mt-off_MTS.nii.gz
│   │   ├── sub-ubc06_flip-1_mt-on_MTS.json
│   │   ├── sub-ubc06_flip-1_mt-on_MTS.nii.gz
│   │   ├── sub-ubc06_flip-2_mt-off_MTS.json
│   │   └── sub-ubc06_flip-2_mt-off_MTS.nii.gz
│   │
│   └── dwi
│       ├── sub-ubc06_dwi.bval
│       ├── sub-ubc06_dwi.bvec
│       ├── sub-ubc06_dwi.json
│       ├── sub-ubc06_dwi.nii.gz
│       ├── (sub-ubc06_acq-b0_dwi.json)
│       └── (sub-ubc06_acq-b0_dwi.nii.gz)
│
└── derivatives
    │
    └── labels
        └── sub-ubc06
            │
            ├── anat
            │   ├── sub-ubc06_T1w_RPI_r_seg-manual.nii.gz  <---------- manually-
↪corrected spinal cord segmentation
            │   ├── sub-ubc06_T1w_RPI_r_seg-manual.json  <------------ information␣
↪about origin of segmentation (see below)
            │   ├── sub-ubc06_T1w_RPI_r_labels-manual.nii.gz  <------- manual␣
↪vertebral labels
            │   ├── sub-ubc06_T1w_RPI_r_labels-manual.json
            │   ├── sub-ubc06_T2w_RPI_r_seg-manual.nii.gz  <---------- manually-
↪corrected spinal cord segmentation
            │   ├── sub-ubc06_T2w_RPI_r_seg-manual.json
            │   ├── sub-ubc06_flip-2_mt-off_MTS_seg-manual.nii.gz  <-------- manually-
↪corrected spinal cord segmentation
            │   ├── sub-ubc06_flip-2_mt-off_MTS_seg-manual.json
            │   ├── sub-ubc06_T2star_rms_gmseg-manual.nii.gz  <------- manually-
↪corrected gray matter segmentation
            │   └── sub-ubc06_T2star_rms_gmseg-manual.json
            │
            └── dwi
                ├── sub-ubc06_dwi_moco_dwi_mean_seg-manual.nii.gz  <-- manually-
↪corrected spinal cord segmentation
                └── sub-ubc06_dwi_moco_dwi_mean_seg-manual.json
```

To convert your DICOM data folder to the compatible BIDS structure, you need to install dcm2bids. Once installed, download this config file (click File>Save to save the file), then convert your Dicom folder using the following command (replace xx with your center and subject number):

```
dcm2bids -d PATH_TO_DICOM -p sub-ID_SITE -c config_spine.txt -o SITE_spineGeneric
```

For example:

```
dcm2bids -d /Users/julien/Desktop/DICOM_subj3 -p sub-milan03 -c ~/Desktop/config_
↪spine.txt -o milan_spineGeneric
```

A log file is generated under `tmp_dcm2bids/log/`. If you encounter any problem while running the script, please open an issue and upload the log file. We will offer support.

Once you have converted all subjects for the study, create the following files and add them to the data structure:

dataset_description.json (Pick the correct values depending on your system and environment)

```
{
    "Name": "Spinal Cord MRI Public Database",
    "BIDSVersion": "1.2.0",
    "InstitutionName": "Name of the institution",
    "Manufacturer": "YOUR_VENDOR",
    "ManufacturersModelName": "YOUR_MODEL",
    "ReceiveCoilName": "YOUR_COIL",
    "SoftwareVersion": "YOUR_SOFTWARE",
    "Researcher": "J. Doe, S. Wonder, J. Pass",
    "Notes": "Particular notes you might have. E.g.: We don't have the ZOOMit license,
↪ unf-prisma/sub-01 and unf-skyra/sub-03 is the same subject.
}
```

Example of possible values:

- **Manufacturer**: "Siemens", "GE", "Philips"
- **ManufacturersModelName**: "Prisma", "Prisma-fit", "Skyra", "750w", "Achieva"
- **ReceiveCoilName**: "64ch+spine", "12ch+4ch neck", "neurovascular"
- **SoftwareVersion**: "VE11C", "DV26.0", "R5.3", . . .

participants.json (This file is generic, you don't need to change anything there. Just create a new file with this content)

```
{
    "participant_id": {
        "LongName": "Participant ID",
        "Description": "Unique ID"
    },
    "sex": {
        "LongName": "Participant gender",
        "Description": "M or F"
    },
    "age": {
        "LongName": "Participant age",
        "Description": "yy"
    },
    "date_of_scan": {
        "LongName": "Date of scan",
        "Description": "yyyy-mm-dd"
    }
}
```

participants.tsv (Tab-separated values)

```
participant_id  sex age date_of_scan    institution_id  institution manufacturer
↪manufacturers_model_name    receive_coil_name   software_versions   researcher
sub-unf01   F   24  2018-12-07  unf Neuroimaging Functional Unit (UNF), CRIUGM,
↪Polytechnique Montreal  Siemens Prisma-fit  HeadNeck_64 syngo_MR_E11    J. Cohen-
↪Adad, A. Foias
sub-unf02   M   29  2018-12-07  unf Neuroimaging Functional Unit (UNF), CRIUGM,
↪Polytechnique Montreal  Siemens Prisma-fit  HeadNeck_64 syngo_MR_E11    J. Cohen-
↪Adad, A. Foias
sub-unf03   M   22  2018-12-07  unf Neuroimaging Functional Unit (UNF), CRIUGM,
↪Polytechnique Montreal  Siemens Prisma-fit  HeadNeck_64 syngo_MR_E11    J. Cohen-
↪Adad, A. Foias
sub-unf04   M   31  2018-12-07  unf Neuroimaging Functional Unit (UNF), CRIUGM,
↪Polytechnique Montreal  Siemens Prisma-fit  HeadNeck_64 syngo_MR_E11    J. Cohen-
↪Adad, A. Foias
sub-unf05   F   23  2019-01-11  unf Neuroimaging Functional Unit (UNF), CRIUGM,
↪Polytechnique Montreal  Siemens Prisma-fit  HeadNeck_64 syngo_MR_E11    J. Cohen-
↪Adad, A. Foias
sub-unf06   F   27  2019-01-11  unf Neuroimaging Functional Unit (UNF), CRIUGM,
↪Polytechnique Montreal  Siemens Prisma-fit  HeadNeck_64 syngo_MR_E11    J. Cohen-
↪Adad, A. Foias
```

Once you've created the BIDS dataset, remove any temp folders (e.g., `tmp_dcm2bids/`) and zip the entire folder. It is now ready for sharing! You could send it to Julien Cohen-Adad via any cloud-based method (Gdrive, Dropbox, etc.).

1.4 Checking acquisition parameters

To ensure the acquisition protocol was properly followed by each site, we implemented a parameter validator that verifies if the pulse sequence parameters match the required ones from the generic protocol (within a tolerance range). Basic parameters are checked, including: repetition time, echo time, flip angle. These parameters are read from the json sidecar file (generated by the DICOM to BIDS conversion). Note that BIDS file naming convention is also checked by the validator. If a parameter does not match, a warning message is triggered.

This validator is exposed in this command line interface (CLI) function: **sg_params_checker**. This function is run during continuous integration (CI), for each dataset, ensuring valid dataset throughout the life cycle of the project.

The json file containing the recommended acquisition parameters is located under */spinegeneric/cli/specs.json*.

Example usage and expected output:

```
sg_params_checker -path-in ~/data-single-subject/
WARNING: Incorrect FlipAngle: sub-douglas_T2w.nii.gz; FA=120 instead of 180
WARNING: Incorrect RepetitionTime: sub-mgh_T2w.nii.gz; TR=2 instead of 1.5
WARNING: Incorrect FlipAngle: sub-tokyoSigna1_T2star.nii.gz; FA=20 instead of 30
WARNING: Incorrect FlipAngle: sub-tokyoSigna2_T2star.nii.gz; FA=20 instead of 30
WARNING:sub-ucl_T2star.nii.gz Missing Manufacturer in json sidecar; Cannot check
↪parameters.
```

1.5 Ethics and anonymization

Each subject consented to be scanned and to have their anonymized data put in a publicly-available repository. To prove it, an email from each participant should be sent to the manager of the database (Julien Cohen-Adad). The email should state the following: "I am the subject who volunteered and I give you permission to release the scan freely to the public domain."

Anatomical scans where facial features are visible (T1w) could be "defaced" before being collected (at the discretion of the subject).

This can be done automatically using R or manually, in case the automatic defacing fails.

1.5.1 Automatic defacing with R

1. Install R, then open R (type "r" in the Terminal) and install the following dependencies:

```
install.packages("sessioninfo")
install.packages("remotes")
remotes::install_github("muschellij2/oro.nifti")  # answer "Yes" to "install from␣
↪source?"
install.packages("fslr")
install.packages("argparser")
install.packages("devtools")
remotes::install_github("muschellij2/extrantsr")  # choose "1" when prompted
```

2. Download this repository and install Python's dependencies as instructed in *Getting started*.
3. Run:

```
sg_deface_using_r -i PATH_TO_BIDS_DATASET -o PATH_TO_DEFACED_BIDS_DATASET -f
sg_deface_using_r -i PATH_TO_BIDS_DATASET -o PATH_TO_DEFACED_BIDS_DATASET
```

4. To launch the QC report of the defacing across multiple subjects, run:

```
sg_qc_bids_deface
```

1.5.2 Manual Defacing

Automatic defacing might fail in some subjects, so this section explains how to deface manually. This procedure takes less than a minute per subject. Here we use FSLeyes but you can use any other NIfTI image editor.

Open FSLeyes and load the T1w scan. Go to **Tools > Edit mode**, Select the pencil with size 100, deface, then save.

Below is an example of a defaced subject:

1.6 Example of datasets

T1w - sub-vuiisAchieva02

T2w - sub-milan01

T2star - sub-brnoCeitec01

flip-1_mt-on_MTS - sub-barcelona04

flip-1_mt-off_MTS - sub-barcelona04

flip-2_mt-off_MTS - sub-barcelona04

DWI - sub-barcelona04

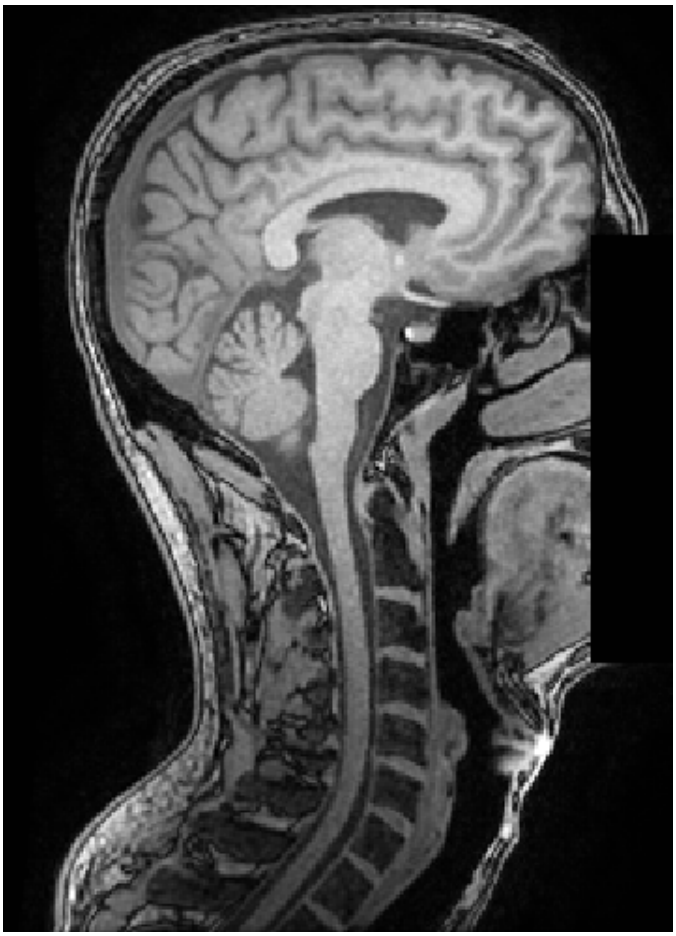


Fig. 1: Example of manual defacing.

CHAPTER 2

Analysis pipeline

This repository includes a collection of scripts to analyse BIDS-structured MRI data and output the following metrics for each contrast:

- **T1**: Spinal cord CSA averaged between C2 and C3.
- **T2**: Spinal cord CSA averaged between C2 and C3.
- **T2s**: Gray matter CSA averaged between C3 and C4.
- **DWI**: FA in WM averaged between C2 and C5.
- **MTS**: MTR in WM averaged between C2 and C5. Uses flip-1_mt-on_MTS and flip-1_mt-off_MTS.
- **MTS**: MTSat & T1 map in WM averaged between C2 and C5. Uses flip-1_mt-on_MTS, flip-1_mt-off_MTS and flip-2_mt-off_MTS.

2.1 Dependencies

MANDATORY:

- For processing: SCT 5.0.1.
- For generating figures: Python >= 3.7

OPTIONAL:

- FSLeyes for correcting segmentations.

2.2 Getting started

Download (or `git clone`) this repository:

```
git clone https://github.com/spine-generic/spine-generic.git
```

Install Python dependencies:

Note: If you prefer to preserve your default Python's libraries, you could first create a virtual environment, and then run the commands below.

```
cd spine-generic
pip install -e .
```

Create a folder where results will be generated (feel free to modify the destination).

```
mkdir ~/spineGeneric_results
```

Launch processing:

```
sct_run_batch -jobs -1 -path-data <PATH_DATA> -path-output ~/spineGeneric_results/ -
↪script process_data.sh
```

Note: `<PATH_DATA>` points to a BIDS-compatible dataset. E.g., you could use one of the dataset listed in *Multi-center data*

2.3 Quality Control

After running the analysis, check your Quality Control (QC) report by opening the file `~/spineGeneric_results/qc/index.html`. Use the "Search" feature of the QC report to quickly jump to segmentations or labeling results.

2.3.1 Segmentation and vertebral labeling

If you spot segmentation or labeling issues, manually fix them using the procedure described below. Also see the video tutorial below.

- Create a .yml file that lists the data awating manual correction. You can create this file using any text editor (vim, nano, atom, etc.).
- In the QC report, enter the string "deepseg" to only display segmentation results or the string "vertebrae" to only display vertebral labeling.
- Review all spinal cord and gray matter segmentations and vertebral labeling. Use the keyboard shortcuts up/down arrow to switch between subjects and the left arrow to toggle overlay.
- If you spot *major* issues with the segmentation (e.g. noticeable leaking or under-segmentation that extends over several slices) or wrong labeling, add the image name into the yml file as in the example below:

```
FILES_SEG:
- sub-amu01_T1w_RPI_r.nii.gz
- sub-amu01_T2w_RPI_r.nii.gz
- sub-cardiff02_dwi_moco_dwi_mean.nii.gz
FILES_GMSEG:
- sub-amu01_T2star_rms.nii.gz
FILES_LABEL:
- sub-amu01_T1w_RPI_r.nii.gz
- sub-amu02_T1w_RPI_r.nii.gz
```

Some explanations about this yml file:

- `FILES_SEG`: Images associated with spinal cord segmentation
- `FILES_GMSEG`: Images associated with gray matter segmentation
- `FILES_LABEL`: Images associated with vertebral labeling

Hint: For the interest of time, you don't need to fix *all* slices of the segmentation but only the ones listed in the "Relevant levels" column of the table below.

Segmentation	Associated image	Relevant levels	Used for
sub-XX_T1w_RPI_r_seg.nii.gz	sub-XX_T1w_RPI_r.nii.gz	C2-C3	CSA
sub-XX_T2w_RPI_r_seg.nii.gz	sub-XX_T2w_RPI_r.nii.gz	C2-C3	CSA
sub-XX_T2star_rms_gmseg.nii.gz	sub-XX_T2star_rms.nii.gz	C3-C4	CSA
sub-XX_flip-2_mt-off_MTS_seg.nii.gz	sub-XX_flip-2_mt-off_MTS.nii.gz	C2-C5	Template registration
sub-XX_dwi_concat_crop_moco_dwi_mean_seg.nii.gz	sub-XX_dwi_concat_crop_moco_dwi_mean.nii.gz	C2-C5	Template registration

- After you finished the QC, run `sg_manual_correction` as in the example below:

```
sg_manual_correction -config files.yml -path-in ~/spineGeneric_results/results/data -
↪path-out <PATH_DATA>
```

This script will loop through all the files that need correction (as per the .yml file that you created earlier), and open an interactive window for you to either correct the segmentation, or perform manual labels. Each manually-corrected label is saved under the `derivatives/labels/` folder at the root of `<PATH_DATA>`, according to the BIDS convention. The manually-corrected label files have the suffix `-manual`.

Your name will be asked at the beginning, and will be recorded in the .json files that accompany the corrected labels.

2.3.2 Upload the manually-corrected files

A QC report of all the manual corrections will be created locally and archived as a zip file. To update the database with the manual corrections, follow this procedure:

- Commit and push the manually-corrected files, which should be placed in the appropriate folders under `derivatives/labels/`
- Create a pull request
- In the pull request body, briefly explain the purpose of these changes, and upload the zipped QC report so the admin team can easily review the proposed changes.
- If the team accepts the pull request, a new release of the dataset will be created and the zipped QC report will be uploaded as a release object.

Note: In case processing is ran on a remote cluster, it it convenient to generate a package of the files that need correction to be able to only copy these files locally, instead of copying the ~20GB of total processed files. If you are in this situation, use the script `package_for_correction`.

2.3.3 Re-run the analysis

After you have corrected all the necessary segmentations/labels, you can re-run the analysis (the `sct_run_batch` command above). If a manually-corrected file exists, it will be used instead of re-creating a new one automatically.

> **Warning:** If you re-run the analysis, make sure to output results in another folder (flag `-path-output`), otherwise the previous analysis will be overwritten.

2.4 Generate figures

Generate figures based on the output csv files using `sg_generate_figures.py` script. Run this script in `/results` folder (folder containing csv files) or specify this folder using `-path-results` flag. The flag `-exclude` points to a yml file containing the subjects to be excluded from the statistics. To generate html that contrains interactive figures, add the flag `-output-html` :

```
sg_generate_figure -path-results ~/spineGeneric_results/results -exclude subject_to_
↪exclude.yml
```

To generate a mosaic of images, run:

```
# Sagittal views of 3D T1w data
sg_create_mosaic -i *T1w_RPI_r_flatten.nii.gz -ifolder ~/project/results_multi_
↪20200907/data_processed/ -p sag -col 20 -row 13 -o fig_mosaic_T1w.png
# Axial views of GRE-T1w data
sg_create_mosaic -i *flip-2_mt-off_MTS.nii.gz -ifolder ~/spineGeneric_results/data_
↪processed -s _seg -p ax -col 20 -row 13 -o fig_mosaic_GRE-T1w.png
```

2.5 Results

2.5.1 Cord CSA from T1w

2.5.2 Cord CSA from T2w

2.5.3 CSA agreement between T1w and T2w data

2.5.4 CSA agreement between T1w and T2w data per vendor

2.5.5 Gray matter CSA

2.5.6 Magnetization Transfer Ratio

2.5.7 Magnetization transfer saturation

2.5.8 T1

2.5.9 Fractional Anisotropy

2.5.10 Mean diffusivity

2.5.11 Radial diffusivity

CHAPTER 3

Body size pipeline

This pipeline extends the original Analysis pipeline and assesses the influence of body size on the structure of the central nervous system. All prerequisities and methods described for the Analysis pipeline applies even here with additional required dependencies and methods as described bellow.

This pipeline has been available since the source code release RXYZXXXX (to be edited) and presented in:

Labounek et al. (2022) Body size influences the structure of the central nervous system: a multi-center in vivo human neuroimaging study [Under Review]

3.1 Dependencies

MANDATORY:

- For processing: All Analysis pipeline dependencies.
- For generating figures: YAMLMatlab >= 0.4.3; Matlab >= R2017b (utilized version R2021b; This version automatically aligns axis labels. Lower version will work, but labels can be worse readable.)

OPTIONAL:

- FSLeyes for correcting segmentations.
- FreeSurfer >=7.2 for segmentation and assessmnet of cerebral morphology from the **T1** scan.

3.2 Spinal cord image analysis

The same methods as used in the Analysis pipeline, i.e sub-sections Getting started and Quality Control.

3.3 Cerebral image analysis

You need to proceed your own automated cerebral segmentation of **T1** scans through the FreeSurfer >=7.2 software or download values obtained by the UMN MILab, which visually inspected and corrected (if necessary) all segmentation outcomes prior to extraction of the quantitative measurements.

3.3.1 Where to download UMN MILab measurements?

```
git clone https://github.com/umn-milab/spine-generic-body-size-results.git
cd spine-generic-body-size-results
cp fs_measurements.xlsx sg.*.aparc.stats.*.csv <PATH_RESULTS>
```

3.3.2 Your own FreeSurfer segmentation and measurements

- Group `brain_t1` in the exclude.yml file defines list of scans whose cerebral volumes are not possible to be segmented correctly due to listed reasons.
- You need to build your own `fs_measurements.xlsx` table.
- The bash script extract_fs_measures.sh can help you to automatically export the sg.*.aparc.stats.*.csv files.
- Store all table files in the `<PATH_RESULTS>` folder.

3.4 Generate figures

- Make a copy and re-edit the spine-generic/matlab/sg_example_script.m regarding to your own HDD.
- Open MATLAB >= R2017b (>= R2021b recomended due to automated axis label orientations)
- Execute re-edited script or execute following commands with a variable setting fitting your HDD

```
% path to the YAMLMatlab 0.4.3 toolbox
path_yamltoolbox = '/home/user/toolbox/matlab/YAMLMatlab_0.4.3';
% path to the spine-generic source code
path_spinegeneric = '/home/user/git/spine-generic';
% path to the <PATH_RESULTS> folder
path_results = '/home/user/spine-generic/data-multi-subject_results';
% path to the <PATH_DATA> folder
path_data = '/home/user/spine-generic/data-multi-subject';
% Execute
addpath(path_yamltoolbox);
addpath(fullfile(path_spinegeneric,'matlab'))
stat = sg_structure_versus_demography(path_results,path_data);
% Figures exported in fig_*.png format and stat_labounek2022.mat file
% will appear in the <PATH_RESULTS> folder
```

3.5 Results

(to be extended once the whole dataset is analyzed)

CHAPTER 4

Contributors

A list of contributors for the analysis pipeline is available here. If you would like to contribute to this project, please see contribution guidelines.

CHAPTER 5

License

MIT License

Copyright (c) 2019 Julien Cohen-Adad and contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.