ENIGMA HALFpipe INSTRUCTION MANUAL RESTING STATE

Updated April 2025

Written by Clara El Khantour, Lea Waller, Seann Wang, Pierre Lune Bellec, Frank Hillary, Ilya Veer, Clara Moreau

Adapted from the **MDD ENIGMA resting-state fMRI manual** and the **OCD ENIGMA task-based fMRI manual.**

1. Preparation

Required Resources:

- 8GB of RAM.
- Disk Space :
    - 0.1GB for Atlas and Seeds Folder
    - 5.32GB for HALFpipe container (23GB if running with Docker)
    - ~500 MB for a working directory for one subject, one task, and one session.

Required Licensing:

Preprocessing requires a FreeSurfer license file which you can get for free when you submit the FreeSurfer Registration form. You will receive the license in the following minutes via email.

Required Downloads

Please ensure these downloaded items are accessible in your environment before proceeding. This means either placed inside your base directory or another directory available to your environment (for example, if using a High-Performing cluster, you need to put the atlas folder on your cluster). We recommend that you write down the path where these files are located, you will need them later.

- Atlas-Based Connectivity (≈80 MB) and Seed-Based Connectivity (≈21 MB) Files: here.
- **Optional** Trial Dataset: For a small and clean dataset (≈8 GB) to trial this pipeline, use the open-source dataset ds004101 found on OpenNeuro here.

Estimated Processing Time: *Estimated with a sample size of n=155 and n=261.*

On a local machine:

- Few hours (varies by dataset).

On High-Performance Cluster:

- Working directory creation: >1 hour (varies by dataset).
- Full pipeline run: Approximately 2–3 hours.

Data requirements

- **Raw (non-skull-stripped) 3D T1-weighted (anatomical) image** with .nii or .nii.gz extension. De-faced images will work out of the box.
- **Raw 4D functional (task) image** with .nii or .nii.gz extension.
- **Field map images (if available).** Three types are supported if the required metadata is available (e.g., echo spacing, echo time, etc.):
    - Siemens field maps, as phase and magnitude image pairs.
    - Philips field maps, as one field map image and one or two magnitude images.
    - Blip-up/blip-down field maps, as a pair of EPI images with opposite phase encoding directions (mostly AP and PA).

Recommended: Processing using multiple cores or on a high-performance compute cluster (HPC) will be faster and more resource efficient. The manual is built for these conditions although single core processing is possible.

Troubleshooting: If you encounter issues running the pipeline, please refer to the GitHub page here. For further assistance, submit a GitHub issue using the form here (recommended) or contact the support email at halfpipe@fmri.science. You can also join the HALFpipe Mattermost (similar to Slack) using this link.

---

2. Installing Container Platform

Installation of a container platform is only required once.

1. **Select an option based on operating system:**
    a. **Mac OS X/Windows:** Use Docker from the install link below.
    b. **Linux:** Use Apptainer on newer systems or otherwise Singularity (via Neurodebian repositories).
    c. **High-Performance Clusters (HPC):** Use Singularity or Apptainer (depending on which one is available on your cluster)

2. **Run the installation based on option selected:**

Singularity (3.x)	Follow install instructions link
Singularity (2.x)	`sudo apt install singularity-container`
Apptainer	Follow install instructions link or `sudo apt install -y apptainer`
Docker Desktop	Follow install instructions here: Docker Desktop

3. **Check to see if container platform is installed:**
    a. If **Docker Desktop:**
        i. You should now have an application named Docker Desktop on your computer. Look at your applications folder.
        ii. Open it to finish the installation.
    b. If **Singularity or Apptainer:**
        i. Open a terminal session. ***Only if using a HPC***: connect to your server.
        ii. Type `apptainer` or `singularity` and press enter.
        iii. If you see instructions on using Singularity and do not get "`command not found`", then you can proceed to the next section.

Note: If using Apptainer or Singularity, the command syntax is identical. Replace the tool name accordingly in the commands below.

3. Working Environment Setup

Here are the steps of setting up your base and working directory where HALFpipe will be run.

1. Open a terminal session. ***Only if using a HPC***: connect to your server.
2. Type `pwd` (or `cd` on Windows) and verify that this is the location where you want to begin creation of your environment.
    a. If not, please move to the folder where you want to run the pipeline and save the outputs using the command `cd path/to/your_working_space/`
3. To create a new base directory (output folder), type `mkdir your_enigma_directory`.
    a. Verify that the file has been created using the command `ls`
4. Then do `cd your_enigma_directory` to enter the folder just created.
5. Then type `mkdir your_working_dir` to create the working directory where all the outputs will be stored.

4. Downloading HALFpipe

Download the container image file:

1. Open a terminal session.
    a. If **Docker desktop**:
        i. Open the terminal in Docker desktop. Refer to this documentation: Explore Docker Desktop
    b. If **Singularity or Apptainer**:
        i. Open a terminal session. ***Only if using a HPC***: connect to your server.
        ii. Navigate to your base directory with `cd path/to/your_working_space/your_enigma_directory`
2. Use one of the following commands.

Singularity (3.x) or Apptainer	`curl -O https://download.fmri.science/singularity/halfpipe-1.2.3.sif` Run `find $(pwd) -maxdepth 1` to show all files in the current directory with their full path. You should see a file named halfpipe-1.2.3.sif, or similar, listed among the files. This file is the container image that includes all software and code needed for processing. Please note down the path to this file, you will need this later.
Singularity (2.x)	`curl -O https://download.fmri.science/singularity/halfpipe-1.2.3.simg`

Docker Desktop	`docker pull --platform linux/amd64 halfpipe/halfpipe:1.2.3` *Note: The Docker image for HALFpipe occupies about 23 GB of disk space upon installation.*

Note:
We recommend installing version 1.2.3 of HALFpipe, which includes fMRIPrep LTS. Compatibility testing with the newer fMRIPrep release is ongoing (as of April 2025, already available in the development version).

Additional Information:
You will need the paths to both your data and the atlases when running HALFpipe. You will need to specify their locations during the settings configuration. These files do not need to be in the same location as the pipeline image file—just be sure you know where they are stored. To do this, please use the command `cd` to navigate to your atlas or data folder, then run the command `pwd` (or `cd` on Windows).

---

5. Working Environment Check

Here we ensure that the working environment is correctly set up before beginning to run HALFpipe.

1. Open a terminal session.
    a. If **Docker desktop**:
        i. Open the terminal in Docker desktop. Refer to this documentation: Explore Docker Desktop
        ii. Note that you need to launch HALFpipe via Docker Desktop (**Section 6.1**).
    b. If **Singularity or Apptainer**:
        i. Open a terminal session. ***Only if using a HPC***: connect to your server.
2. Navigate to your working directory using this command:
   `cd path/to/your_working_space/your_enigma_directory/your_working_dir`
3. Then use the command `ls`
4. Ensure that these items are directly inside or moved into your working directory before proceeding:
    a. Freesurfer license file (license.txt).
    b. You can move the license.txt file using the command `mv path/to/license.txt path/to/your_working_space/your_enigma_directory/your_working_dir` (on Windows, use `move`)
5. Navigate back to your base directory (i.e. `your_enigma_directory)` using this command: `cd ..`
6. Use the command `ls`. Ensure that these items are directly inside your base directory before proceeding:
    a. Your working directory (i.e. `your_working_dir)`
    b. ***If Singularity or Apptainer***: HALFpipe container file (.sif or .simg)
7. Ensure that these items are accessible in your environment before proceeding:
    a. Atlas and Seed files (downloaded in section 1.)
    b. Your unprocessed, resting-state fMRI dataset.

---

6. Launching the Pipeline

If you are running it on a local computer: **run 6.1**
If you are running it on a high-performance cluster: **run 6.2**

Note: if you have a large dataset (>500 participants), we recommend that you look at point 5 on the Troubleshooting page (section 11) or contact halfpipe@fmri.science. You can also join the HALFpipe team Mattermost (similar to Slack) using this link.

6.1 On a Local Machine

1. Open a terminal session (if using Docker open the terminal via Docker Desktop).
2. Run the appropriate command for your container platform.
    a. Note: to access your files more easily, you can bind in the path where the data, working directory and atlases are located. For example: `--bind /home/user/scratch:/home/user/scratch`

Singularity or Apptainer (3.x)	`singularity run --contain --cleanenv --bind /:/ext halfpipe_1.2.3.sif --subject-chunks --nipype-n-procs 1 --keep none`
Singularity (2.x)	`singularity run --contain --cleanenv --bind /:/ext halfpipe_1.2.3.simg --subject-chunks --nipype-n-procs 1 --keep none`

Docker Desktop	***On MacOS***
	`docker run --interactive --tty --volume /:/ext halfpipe/halfpipe:1.2.3 --subject-chunks --nipype-n-procs 1 --keep none`
	
	On Windows:
	● *Use forward slashes (/) instead of backslashes (\) to specify file paths.*
	● *Be sure to replace `C:` with the appropriate drive letter for your system.*
	`docker run --interactive --tty --volume C:/:/ext halfpipe/halfpipe:1.2.3 --subject-chunks --nipype-n-procs 1 --keep none`

Note 1: The flags `--subject-chunks` and `--nipype-n-procs 1` are optional. While they may slow down the pipeline, they help ensure you don't exceed your computer's processing capacity. Use them if you're concerned about system limitations.

Note 2: On some systems, Docker will restrict the number of CPU cores available to HALFpipe by default. If your computer can handle it, you can allocate more CPUs using the Docker flag `--cpus`. Refer to your system's documentation to learn your CPU limitations.

By adding more CPUs, you can add the HALFpipe flag `--nipype-n-procs` to accelerate the process.
Example using Docker: `docker run --interactive --tty --cpus=3 --volume /:/ext halfpipe/halfpipe:1.2.3 --nipype-n-procs --subject-chunks --keep none`

3. Go to **Section 7.**

6.2 On a High-Performance Cluster (HPC)

1. Open a terminal session and connect to your HPC.
2. Running an **Interactive Job** on the cluster**.**
    - On HPC, start an interactive job to run the Halfpipe container. To do this, please refer to your HPC's documentation or your IT support. Here are a few examples of how to do this.

SLURM	`srun --time 0-6 --ntasks=1 --pty bash`
SGE	`qlogin -now no`
Torque/PBS	`qsub -I -l walltime=12:00:00`

    - Load Singularity/Apptainer packages: `module load singularity`
      Note: Re-run this command every time you log out and back in again, and every time you start working on a node.

3. Run the **Execution Command** for your container software**.**

Singularity (3.x)	`singularity run --contain --cleanenv --bind /:/ext halfpipe_1.2.3.sif --use-cluster`
Singularity (2.x)	`singularity run --contain --cleanenv --bind /:/ext halfpipe_1.2.3.simg --use-cluster`
Apptainer	`apptainer run --contain --cleanenv --bind /:/ext halfpipe_1.2.3.sif --use-cluster`

4. Go to **Section 7**.

Additional Information:

- *File Location: Ensure the HALFpipe_1.2.3.sif file is in your current working directory or specify its full path.*
- *Directory Binding: The --bind (or -B) option maps directories from your host system into the container. For example, /home/user/Documents on the host becomes /ext/home/user/Documents inside the container.*
- *Cluster Flag: Use the* `--use-cluster` *flag when running the execution command on an HPC.*

7. Pipeline Settings

You have now opened the user interface, which should look like this:

The prompts that will be displayed by HALFpipe are shown below in `blue`, examples are in yellow, answers that need to be input/selected are `underscored and red.` We will first specify the Data (see **section 7.1**) and then the Features (see **section 7.2**).

Note: If you want to go back/deselect an option, you can use Ctrl+C

7.1 Data

1. `Specify Working directory:` Your output folder.
    - You can navigate using the **Up/Down arrow keys** ↓ to go to a folder and the **right arrow key**→ to select this folder.
    - Do this until you get to `path/to/your_working_directory`
2. `Is the data available in BIDS format?`

    If your data is already in BIDS format, follow the instructions of **point 2.1**. If not, follow the instructions of **point 2.2**.

    Note: When running on a local computer, be aware that the BIDS format is very strict. If you're unsure whether your data is properly formatted, we recommend answering "No" to this question and specifying the file name manually, as shown in Point 2.2. This can help prevent errors during execution.

    2.1. If your data is in BIDS:

    `Is the data available in BIDS format?` `Yes`

    `Specify the path of the BIDS directory`

- Here, simply indicate the path to your BIDS data as `path/to/bids/dataset/` using the arrow keys.
- It will now print how many T1-weighted files were found, for example: `Found 9 files for 9 subjects and 1 session`
- And how many BOLD files were found, for example: `Found 9 files for 9 subjects, 2 sessions, and 1 task`

```
Is the data available in BIDS format?
[Yes] [No]

Specify the path of the BIDS directory
[scratch/bids_dataset/ds004101/]

Found 9 T1-weighted image files for 9 subjects and 1 session

Found 18 BOLD image files for 9 subjects, 2 sessions, and 1 task
```

- You can now directly go to **point 3.**

2.2. If your data is not in BIDS:

`Is the data available in BIDS format?` `No`

2.2.1. `Specify anatomical/structural data`

`Specify the path of the T1-weighted image files`

- If your data is organized as `path/to/T1/subj01/T1.nii.gz` - where subj01 can be any ID you have to enter `path/to/T1/{subject}/T1.nii.gz`.
- It will then find T1 files for all subjects: `path/to/T1/subj01/T1.nii.gz`, `path/to/T1/subj02/T1.nii.gz` etc...
- It will print how many files are found, for example: `Found 9 files for 9 subjects and 1 session`

For example

```
Is the data available in BIDS format?
[Yes] [No]

Specify anatomical/structural data

Specify the path of the T1-weighted image files
Put {subject} in place of the subject names
You can also use {session}, {run}, and {acquisition}
[scratch/bids_dataset/ds004101/{subject}/{session}/anat/{subject}_{session}_T1w.nii.gz]
```

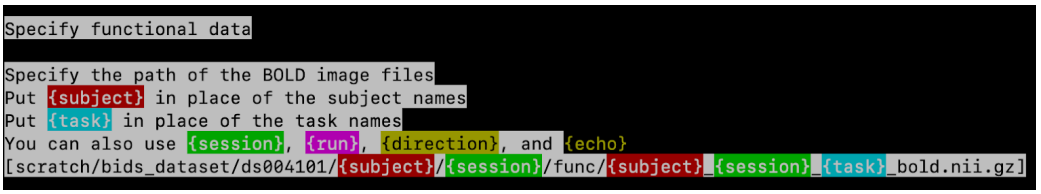
2.2.2. `Specify functional data`

`Specify the path of the BOLD image files`

- Follow the same logic as for structural images. Put *{subject}* in place of the subject names (just as you did with the T1).

Note: {task} can be used to tag task names (e.g., auditory), as well as a resting-state paradigm.

- It will now tell you how many files have been found: Found 9 files for 9 subjects, 2 sessions, and 1 task.For example: `Found 9 files for 9 subjects and 1 task`



2.2.3. `Check repetition time values`
Check if the repetition time values are correct.
For example: `18 images - 2.4 seconds`
`Proceed with these values?` Yes/No (unless the TR is incorrect)

2.2.4. `Add more BOLD image files?` Yes/No (unless you have more)

3. `Do slice timing?`

3.1. Yes if you know the slice timing details during the acquisition for your data. If unsure, check with the MR physicist/technician or any dataset information.
In the following steps you will check the slice order and slice timing values.
- Confirm with Yes if they are correct.
- Otherwise, select No and then enter the correct values.

3.2. No otherwise

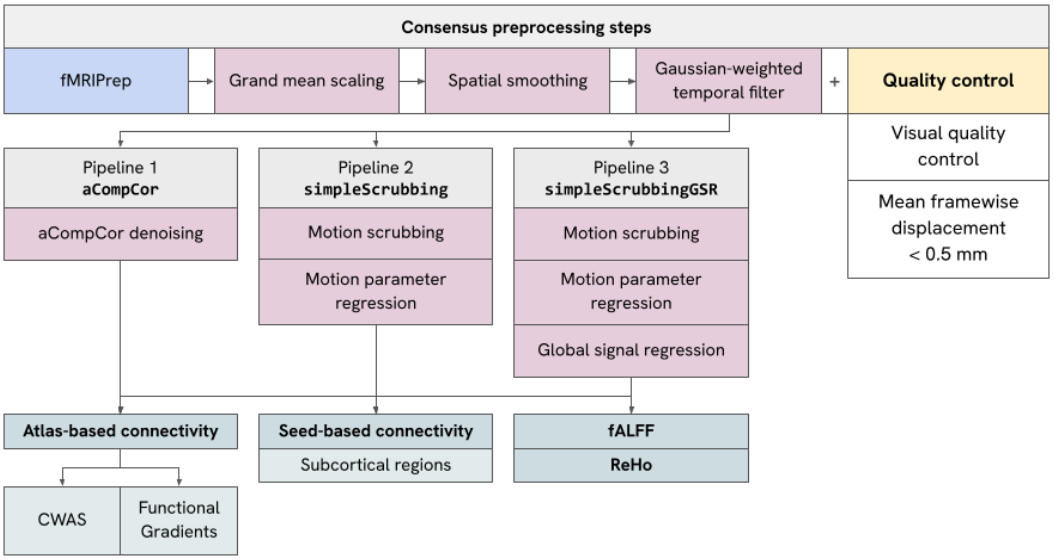
4. `Remove initial volumes from scans?` : Enter 0, which should be pre-filled.
If you believe that your data might require removal of initial volumes, for example due to non-equilibration, you can also enter a different value.

5. `Specify field maps?`
Note: If the data was imported from a BIDS directory, this step will be omitted.
5.1. Yes if you acquired field map images for your sample.
In the following steps you will specify the type of field map, their file names and the acquisition parameters.
5.2. No otherwise

6. `Specify first-level features?` Select Yes

7.2 Features

We will now specify which features we'd like to extract and specify their settings.



The features will be extracted with **three** types of confound removal using:

1) **Pipeline 1**: aCompCor
2) **Pipeline 2**: Motion parameters with scrubbing
3) **Pipeline 3**: Pipeline 2 + Global Signal (GSR)

Here is a summary of the confound parameters to apply for each pipeline when running HALFpipe:

Pipeline 1
aCompCor

```
Remove confounds?
[ ] ICA-AROMA
[ ] Motion parameters
[ ] Derivatives of motion parameters
[ ] Motion parameters squared
[ ] Derivatives of motion parameters squared
[ ] Motion scrubbing
[*] aCompCor (top five components)
[ ] White matter signal
[ ] CSF signal
[ ] Global signal
```

Pipeline 2
Motion params

```
Remove confounds?
[ ] ICA-AROMA
[*] Motion parameters
[*] Derivatives of motion parameters
[*] Motion parameters squared
[*] Derivatives of motion parameters squared
[*] Motion scrubbing
[ ] aCompCor (top five components)
[ ] White matter signal
[ ] CSF signal
[ ] Global signal
```

Pipeline 3
Motion params + Global signal

```
Remove confounds?
[ ] ICA-AROMA
[*] Motion parameters
[*] Derivatives of motion parameters
[*] Motion parameters squared
[*] Derivatives of motion parameters squared
[*] Motion scrubbing
[ ] aCompCor (top five components)
[ ] White matter signal
[ ] CSF signal
[*] Global signal
```

***Optional**: If you are planning to run a multiverse analysis, we recommend running two additional pipelines: Pipeline 2 and Pipeline 3, each without scrubbing.*

You can also follow the manual as it is and add these optional pipelines later by using point 6 in the Troubleshooting section (Section 11), which explains how to add features after the initial setup.

Seed-based connectivity

1. `Specify the feature type`: Use the ↓ **(down arrow key)** until you reach Seed-based connectivity, then **press enter.**

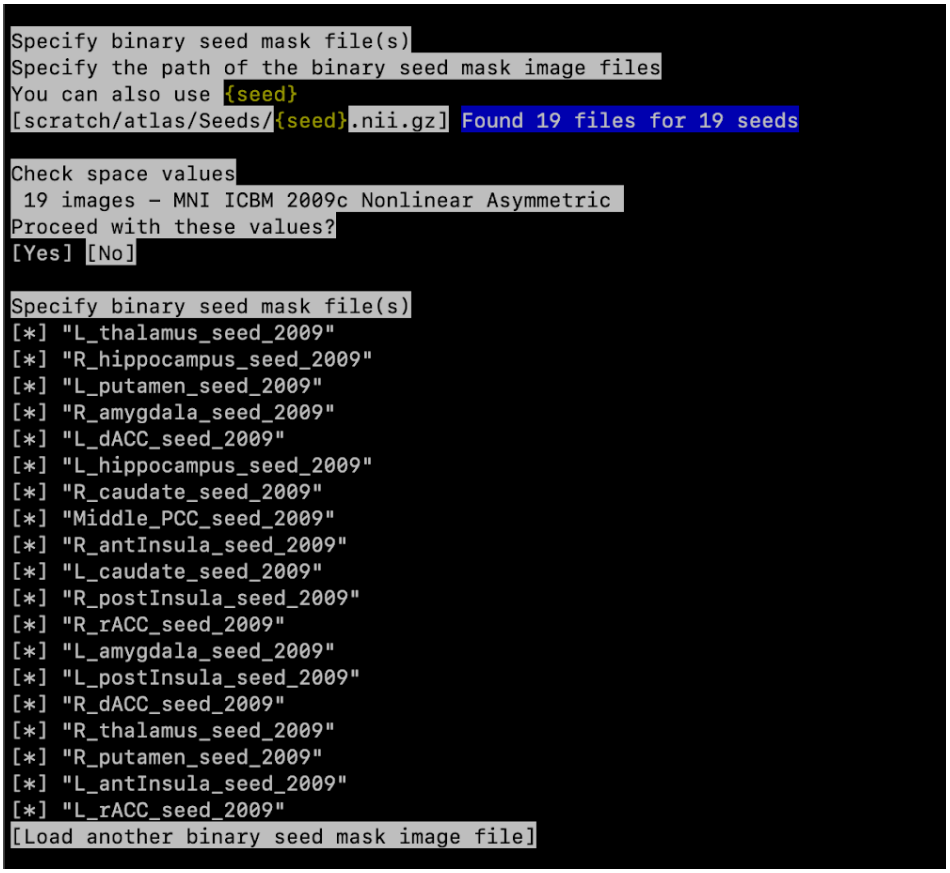
   ```
   [Task-based]
   [Seed-based connectivity]
   [Dual regression]
   [Atlas-based connectivity matrix]
   [ReHo]
   [fALFF]
   ```

2. `Specify feature name`: Type `seedCorr1` (and then **press enter**)

3. `Specify images to use:` if you have multiple tasks or sessions, select the session(s) or task(s) corresponding to resting-state data. To do this, **press space to untick** a session, **press enter** to validate the selection.

4. `Specify path to seed` : give the path to the folder that contains seed files.
   4.1. To do this, add your path followed by {seed}.nii.gz : `path/to/enigma/{seed}.nii.gz`.
   It will now tell you, for example : `Found 19 files for 19 seeds`
   4.2. `Proceed with these values`: `Yes`
   4.3. `Specify the seed masks`, just leave all seeds selected (**press enter**)

Example seed mask selection:

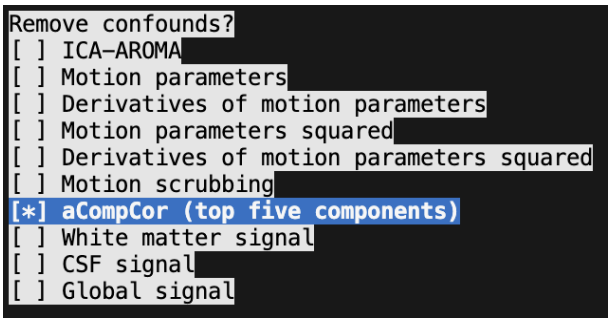


5. `Specify minimum seed coverage by individual brain mask: 0.5`
6. `Apply smoothing Yes`
7. `Specify smoothing FWHM in mm: 6.0` (This should be by default 6.0, simply **press enter**)
8. `Specify the filter width in seconds`
   - 8.1. `Low-pass width [0] [Skip]`
   - 8.2. `High-pass width [125.0] [Skip]`

For pipeline 1

We will apply pipeline 1, i.e. aCompCor:

9. `Remove confounds?`
   - Use your **down arrow key** ↓ until `aCompCor (top five components)`
   - Then **press space** to select it. Make sure no other confounds are selected, and **press enter**



`Add another first-level feature? Yes`

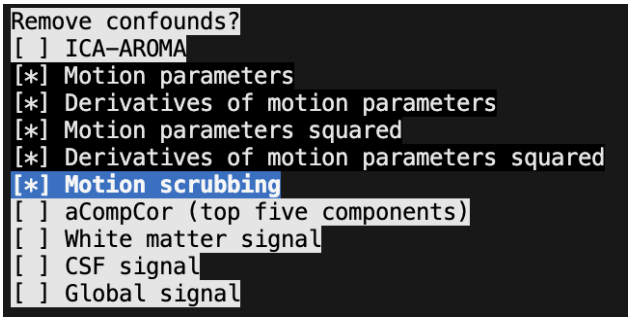
We will now add another seed-based connectivity (repeat all the steps, **from 1 to 8**), but in the last step we will select all motion parameters, their derivatives, and motion scrubbing for **pipeline 2** instead of aCompCor.

In short, you will now choose:

1. `Specify the feature type: Seed-based connectivity`,
2. `Specify feature name:` Type `seedCorr2`,
3. `Specify images to use:` select session(s) or task(s) with **space**, then **press enter**
4. `Specify the seed masks:` **press enter**
5. `Specify minimum seed coverage`: **press enter**
6. `Apply smoothing: Yes`
7. `Specify smoothing FWHM: 6.0`,
8. `Specify the filter`: `Skip` and `125.0`.

For pipeline 2

9. `Remove confounds`: Use your **down arrow key** ↓ and **press space** for each parameter.
   `Motion parameters`,
   `Derivatives of motion parameters`,
   `Motion parameters squared`,
   `Derivatives of motion parameters squared`,
   `Motion scrubbing`



It will then ask:

`Do you really want to use inconsistent confounds across features?` `Yes`

`Add another first-level feature?` `Yes`

We will now add another seed-based connectivity (repeat all the steps), but in the last step (remove confounds) we will select all motion scrubbing, motion parameters and their derivatives as well as Global signal (**pipeline 3**) .

1. `Specify the feature type:` `Seed-based connectivity`,
2. `Specify feature name:` Type `seedCorr3`,
3. `Specify images to use:` select session(s) or task(s) with **space**, then **press enter**
4. `Specify the seed masks:` **press enter**
5. `Specify minimum seed coverage`: **press enter**
6. `Apply smoothing:` `Yes`
7. `Specify smoothing FWHM:` `6.0`,
8. `Specify the filter`: `skip` and `125.0`.

For pipeline 3

9. `Remove confounds`: Use your **down arrow key** ↓ and **press space** for each parameter. Once the selection is done, **press enter**:

   `Motion parameters`,

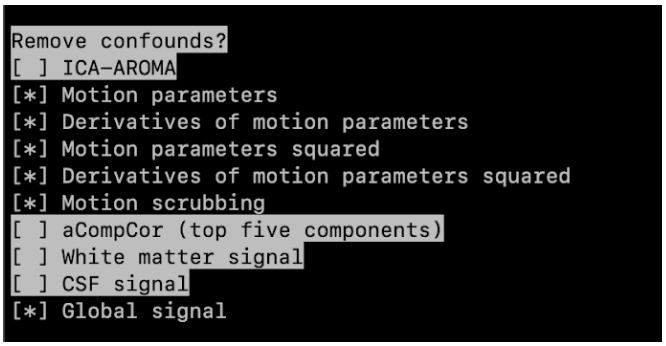
   `Derivatives of motion parameters`,

   `Motion parameters squared`,

   `Derivatives of motion parameters squared`,

   `Motion scrubbing`

   `Global signal`



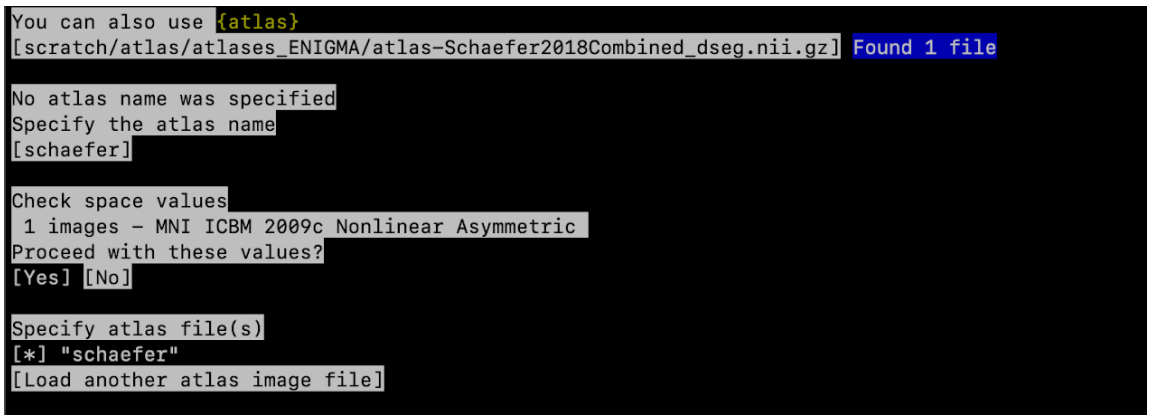
Atlas-based connectivity matrix

`Add another first-level feature?` `Yes`

We will now specify different settings:

1. `Specify the feature type`: `Atlas-based connectivity matrix`
2. `Specify feature name?` Type `corrMatrix1` and then **press enter**
3. `Specify images to use:` if you have multiple tasks or sessions, select the session(s) or task(s) corresponding to resting-state data. Press space **to untick** a session, **press enter** to validate the selection
4. `Specify atlas file(s)` give path to folder that includes the atlas.
   - 4.1. `Specify the path of the atlas image files`. Use your keys to navigate through the path until the schaefer atlas :
     `path/to/enigma/atlas-Schaefer2018Combined_dseg.nii.gz`
     It will now tell you how many atlases were found, for example: `Found 1 file`
   - 4.2. `Specify the atlas name`: Here write `Schaefer2018Combined`, then **press enter**
   - 4.3. `Check space values`
     `Proceed with these values?` `Yes`
   - 4.4. `Specify atlas file(s)`: **press enter**

   For example:



5. `Specify minimum atlas region coverage by individual brain mask` : `0.5`
6. `Apply a temporal filter?` `Yes`
7. `Specify the type of temporal filter`: `Gaussian-weighted`
8. `Specify the filter width in seconds`
   - 8.1. `Low-pass width [0] [Skip]`
   - 8.2. `High-pass width [125.0] [Skip]`

For example

```
Specify minimum atlas region coverage by individual brain mask
Atlas region signals that do not reach the requirement are set to n/a
[0.5]

Specify preprocessing setting

No smoothing will be applied

Grand mean scaling will be applied with a mean of 10000.000000

Apply a temporal filter?
[Yes] [No]

Specify the type of temporal filter
[Gaussian-weighted]
[Frequency-based]

Specify the filter width in seconds
Low-pass width  [0] [Skip]
High-pass width [125.0] [Skip]
```

As we performed for Seed-based connectivity, we will apply the 3 pipelines for atlas-based connectivity matrix. You can look at the figures for Seed-based if you need visual support.

For pipeline 1

We will first apply pipeline 1, i.e. aCompCor

9. `Remove confounds?`
   - Use your **down arrow key** ↓ until `aCompCor (top five components)`
   - **Press space** to select and then **press enter**

`Add another first-level feature?` Yes /No

We will now add another atlas-based connectivity (repeat all the steps, **from step 1 to 8**), but in the last step (step 9 : remove confounds) we will select all parameters for pipeline 2.

In short, you will now choose:

1. `Specify the feature type`: `Atlas-based connectivity matrix`
2. `Specify feature name?` Type `corrMatrix2`,
3. `Specify images to use:` select session(s) or task(s) with **space**, then **press enter**
4. `Specify atlas file(s)`: **press enter**
5. `Specify minimum atlas region coverage`: **press enter**
6. `Apply a temporal filter?` `Yes`
7. `Specify the type of temporal filter` `Gaussian-weighted`
8. `Specify the filter`: `Skip` and `125.0`.

For pipeline 2

9. `Remove confounds`: Use your **down arrow key** ↓ and **press space** for each parameter:
   `Motion parameters`,
   `Derivatives of motion parameters`,
   `Motion parameters squared`,
   `Derivatives of motion parameters squared`,
   `Motion scrubbing`

`Do you really want to use inconsistent confounds across features?` `Yes`

`Add another first-level feature?` Yes

We will now add another atlas-based connectivity (repeat all the steps 1-8), but in the last step (remove confounds) we will select all parameters for **pipeline 3**.

1. `Specify the feature type`: `Atlas-based connectivity matrix`
2. `Specify feature name?` Type `corrMatrix3`,
3. `Specify images to use:` select session(s) or task(s) with **space**, then **press enter**
4. `Specify atlas file(s)`: **press enter**
5. `Specify minimum atlas region coverage`: **press enter**
6. `Apply a temporal filter?` `Yes`
7. `Specify the type of temporal filter` `Gaussian-weighted`
8. `Specify the filter`: `Skip` and `125.0`.

For pipeline 3

9. `Remove confounds`: Use your **down arrow key** ↓ and **press space** for each parameter:
   `Motion parameters`,
   `Derivatives of motion parameters`,
   `Motion parameters squared`,
   `Derivatives of motion parameters squared`,
   `Motion scrubbing`
   `Global signal`

`Add another first-level feature?` `Yes`

ReHo

1. `Specify the feature type` : Use your keyboard to select `ReHo` and **press enter**
2. `Specify feature name?` Type `reHo1` then **press enter**

3. `Specify images to use:` if you have multiple tasks or sessions, select the session(s) or task(s) corresponding to resting-state data. Press space **to untick** a session, **press enter** to validate the selection

4. `Apply smoothing:` `Yes`

5. `Specify smoothing FWHM in mm`: `6.0`

For example

```
Specify feature name
[reHo]

Specify images to use
Session [*] "1pre" [*] "2post"

Apply smoothing?
[Yes] [No]

Specify smoothing FWHM in mm
[6.0]

Grand mean scaling will be applied with a mean of 10000.000000

Temporal filtering will be applied using a frequency-based filter
with a low cutoff of 0.01 Hz and a high cutoff of 0.1 Hz
```

For pipeline 1

We will first apply pipeline 1, i.e. aCompCor

6. `Remove confounds?`

- Use your **down arrow key** ↓ until `aCompCor (top five components)`
- **Press space** to select, then **press enter**

`Add another first-level feature?` `Yes`

We will now add another ReHo (repeat all the steps, **from 1 to 5**), select confounds for **pipeline 2**

In short, you will now choose:

1. `Specify the feature type` : `ReHo`
2. `Specify feature name?` Type `reHo2`
3. `Specify images to use:` select session(s) or task(s) with **space**, then **press enter**
4. `Apply smoothing:` `Yes`
5. `Specify smoothing FWHM in mm`: `6.0`

For pipeline 2

6. `Remove confounds`: Use your **down arrow key** ↓ and **press space** for each parameter:

   `Motion parameters`,

   `Derivatives of motion parameters`,

Motion parameters squared,

Derivatives of motion parameters squared,

Motion scrubbing

Do you really want to use inconsistent confounds across features? Yes

---

Add another first-level feature? Yes

We will now select all motion scrubbing, motion parameters and their derivatives as well as Global signal (**pipeline 3**) .

1. Specify the feature type : ReHo
2. Specify feature name? Type reHo3
3. Specify images to use: select session(s) or task(s) with **space**, then **press enter**
4. Apply smoothing: Yes
5. Specify smoothing FWHM in mm: 6.0

For pipeline 3

6. Remove confounds: Use your **down arrow key** ↓ and **press space** for each parameter:

   Motion parameters,

   Derivatives of motion parameters,

   Motion parameters squared,

   Derivatives of motion parameters squared,

   Motion scrubbing

   Global signal

---

fALFF

1. Specify the feature type : use your keyboard to select fALFF, then **press enter**
2. Specify feature name? Type fALFF1
3. Specify images to use: if you have multiple tasks or sessions, select the session(s) or task(s) corresponding to resting-state data. Press space **to untick** a session, **press enter** to validate the selection
4. Apply smoothing: Yes
5. Specify smoothing FWHM in mm: 6.0

For example

```
Specify feature name
[fALFF]

Specify images to use
Session [*] "1pre" [*] "2post"

Apply smoothing?
[Yes] [No]

Specify smoothing FWHM in mm
[6.0]

Grand mean scaling will be applied with a mean of 10000.000000

Temporal filtering will be applied using a frequency-based filter
with a low cutoff of 0.01 Hz and a high cutoff of 0.1 Hz
```

Again, we will run the 3 pipelines

For pipeline 1

We will first apply pipeline 1, i.e. aCompCor

6. `Remove confounds?`

- Use your **down arrow key** ↓ until `aCompCor (top five components)`
- then **press space** and **enter**

`Add another first-level feature?` `Yes`

We will now add another fALFF (repeat all the steps, **from step 1 to 5**), but in the last step (step 6: remove confounds) we will select all parameters for **pipeline 2** instead of aCompCor.

In short, you will now choose:

1. `Specify the feature type` : `fALFF`
2. `Specify feature name?` Type `fALFF2`
3. `Specify images to use:` select session(s) or task(s) with **space**, then **press enter**
4. `Apply smoothing:` `Yes`
5. `Specify smoothing FWHM in mm`: `6.0`

For pipeline 2

6. `Remove confounds`: Use your **down arrow key** ↓ and **press space** for each parameter:

   `Motion parameters`,

   `Derivatives of motion parameters`,

   `Motion parameters squared`,

   `Derivatives of motion parameters squared`,

   `Motion scrubbing`

`Do you really want to use inconsistent confounds across features?` `Yes`

`Add another first-level feature?` `Yes`

We will now select all motion scrubbing, motion parameters and their derivatives as well as Global signal (**pipeline 3**) .

1. `Specify the feature type` : `fALFF`
2. `Specify feature name?` Type `fALFF3`
3. `Specify images to use:` select session(s) or task(s) with **space**, then **press enter**
4. `Apply smoothing:` `Yes`
5. `Specify smoothing FWHM in mm`: `6.0`

For pipeline 3

6. `Remove confounds`: Use your **down arrow key** ↓ and **press space** for each parameter:
   `Motion parameters`,
   `Derivatives of motion parameters`,
   `Motion parameters squared`,
   `Derivatives of motion parameters squared`,
   `Motion scrubbing`
   `Global signal`

We have completed specifying all pipelines for the available features, and **no additional first-level features** will be added.

`Add another first-level feature?` **`No`**

`Output a preprocessed image?` `No` For ENIGMA the preprocessed image is **not** needed.

`Specify group-level model?` `No` For now, you can choose `No`, however, if you would like to have group statistics, you can select `Yes`, load in a file with group and covariate information and choose your preferred settings

You have now finished specifying the settings! 🎉

8. Run the pipeline

8.1 On a Local Machine

If you are running this on a local computer, the pipeline will have started. Once it has finished, continue to **section 9** to check pre-processing.

8.2 On a High-performing cluster (HPC)

1. Navigate to your working directory in terminal: `cd path/to/your_working_space/your_enigma_directory/your_working_dir`
2. Let's visualize the files that have been created with `ls`
    a. Check for the files and folders: **derivatives, nipype, rawdata, reports, submit.slurm.sh, work, workflow***.pickle.xz, execgraph***.pickle.xz.**
    b. If these files are created, continue to step 3, if not, please have a look at the error file and the troubleshooting page.
3. Open the automatically generated `submit.slurm` file. Please compare your output to the example provided in the next page:
    a. Make sure that the path associated with the flag `--bind` is the same as the one you used to launch the pipeline (**see section 6.2**).
    b. Verify all the paths, make sure that they are all correct.
    c. If necessary
        i. You can edit the `#bin bash` command (e.g. add `#SBATCH --account=def-account` in **submit.slurm.sh**).
        ii. If you are unsure, please refer to your HPC's documentation or contact your IT support.

Note: Depending on your cluster's configuration (e.g., SGE or Torque), these modifications may need to be applied to a different submit file than the SLURM example provided. Please consult your cluster's documentation or contact your IT support team to confirm.

Here is an example of a submit.slurm.sh file generated by HALFpipe:

```
#!/bin/bash

#SBATCH --job-name=halfpipe
#SBATCH --output=halfpipe.log.txt
#SBATCH --time=05:00:00

#SBATCH --ntasks=1
#SBATCH --cpus-per-task=2
#SBATCH --mem=10752M

#SBATCH --array=1-9

if ! [ -x "$(command -v singularity)" ]; then
module load singularity
fi

singularity run \
--contain --cleanenv \
--bind /home/username \
/home/username/projects/containers/halfpipe-1.2.3.sif \
--workdir /home/username/projects/working_directory \
--only-run \
--uuid 3309d606 \
--subject-list /home/username/projects/working_directory/subject-list.txt \
--subject-chunks \
--only-chunk-index ${SLURM_ARRAY_TASK_ID} \
--nipype-n-procs 2 \
--keep some
```

4. To reduce the number of files and the amount of disk space used, please update the submit.slurm file as shown below.
    - Look for lines marked with **modification ====>**, which indicates where you need to add a line and update it with your own path.
    - Lines marked with **copy-paste ====>** can be added as-is, without any changes.
    - Ensure that the unchanged lines match those in your original document (e.g., uuid, path to access your HalfPipe container, etc.).

    You can copy and paste the following lines into your script, making sure to adjust the paths accordingly:

Example of a submit.slurm.sh file modified to reduce number of file produced and disk usage:

```
#!/bin/bash

#SBATCH --job-name=halfpipe
#SBATCH --output=halfpipe.log.txt
#SBATCH --time=05:00:00

#SBATCH --ntasks=1
#SBATCH --cpus-per-task=2
#SBATCH --mem=10752M

#SBATCH --array=1-9

if ! [ -x "$(command -v singularity)" ]; then
module load singularity
fi

working_directory="/home/username/projects/working_directory"
nipype_directory="$(mktemp -d)"

singularity run \
--contain --cleanenv \
--bind /home/username \
--bind "${nipype_directory}:${working_directory}/nipype" \
/home/username/projects/containers/halfpipe-1.2.3.sif \
--workdir "${working_directory}" \
--only-run \
--uuid 3309d606 \
--subject-list "${working_directory}/subject-list.txt" \
--subject-chunks \
--only-chunk-index ${SLURM_ARRAY_TASK_ID} \
--nipype-n-procs 2 \
--keep none
```

modification ====>
copy-paste ====>
copy-paste ====>
copy-paste ====>
copy-paste ====>
modification ====>

5. Now run the pipeline by submitting the submit.slurm script that has been created in the pipeline folder.

SLURM	`sbatch submit.slurm.sh`
SGE	`qsub submit.sge.sh`
Torque/PBS	`qsub submit.torque.sh`

The pipeline will now preprocess all subjects. When they are finished, the quality has to be checked (**see section 9 and 10**).

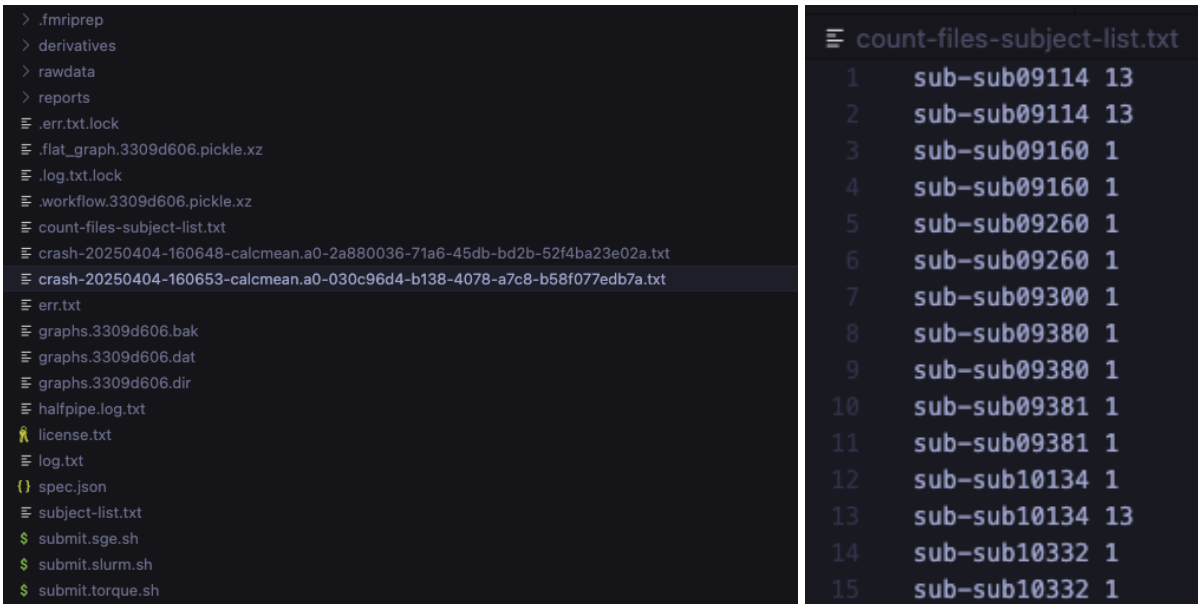
9. Check pre-processing and re-run if necessary

Below are instructions to validate a run of the HALFpipe pipeline and re-run any incomplete subjects.

1. Open a terminal session.
2. Navigate to your working directory: `cd path/to/your_working_dir`
3. Copy and paste the whole block of code into your terminal session. Then hit enter.

```
> incomplete-subject-list.txt
find "derivatives/halfpipe" -type d -path "*/func" | while read task_dir; do
 sub_id=$(echo "$task_dir" | grep -o "sub-[^/]*")
 file_count=$(find "$task_dir" -type f | wc -l)
 echo "$sub_id $file_count"
done | sort > count-files-subject-list.txt
echo "Done. See count-files-subject-list.txt for subject file counts."
```

4. There will be 2 generated files, an empty **incomplete-subject-list.txt** and a **count-files-subject-list.txt** file where Subject IDs and their output file counts are stored.
5. Run `ls` and look at the files in the working directory.
    a. If you find any crash files, open the **err.txt** file and inspect the error messages,
        i. Please check out the Troubleshooting section to understand the error messages and solutions, (**Section 11**), then proceed to **Step 6.**
    b. Otherwise you can go to **Section 10 (Quality control)**.
6. Connect **count-files-subject-list.txt** with HALFpipe code:
    a. Please manually inspect **count-files-subject-list.txt** and the `derivatives/halfpipe` subjects folders together to verify the list of subjects and their counts is accurate.
    b. If any subjects have less than the expected amount of file (or differ from others) in **count-files-subject-list.txt** they will likely need to be re-run.
        i. Open the file err.txt and inspect the error messages, if any.
        ii. It is also possible subjects have less runs, sessions of tasks, and will hence have less file numbers. Please verify this information in your original dataset to correctly identify only incomplete subjects.
        iii. Please check out the Troubleshooting section to understand the error messages and solutions (**Section 11**) before proceeding to point c.
    c. Please open the empty **incomplete-subject-list.txt** file and manually add only the Subject IDs that you have identified that have not been processed correctly.
    d. If all subjects have been successfully processed, proceed to **Section 10**. Otherwise, follow the instructions in **Point 7** to rerun the pipeline for any incomplete subjects.



Left: Example of crash files corresponding to processing errors.

Right: Count-files-subject-list.txt output showing inconsistent file counts per subject;

7. Rerun HALFpipe
    a. Run HALFpipe again with the same command as in **Section 6**, but add the option `--subject-list <path/to/incomplete-subject-list.txt>` to the end of the command. Then select `Run without modification`.

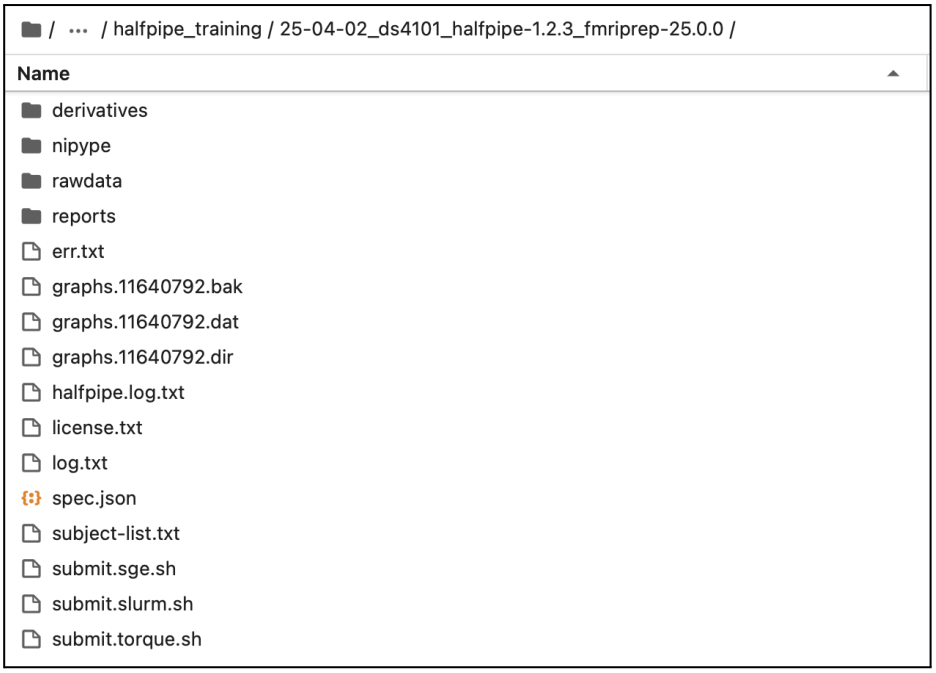
For example:

Apptainer	`apptainer run --contain --cleanenv --bind /:/ext halfpipe-1.2.3.sif --subject-chunks --nipype-n-procs 1 --keep none --subject-list <path/to/incomplete-subject-list.txt>`
Docker	*Note: Do not forget to adapt your number of CPUs* `docker run --interactive --tty --cpus=3 =-volume /:/ext halfpipe/halfpipe:1.2.3 --subject-chunks --nipype-n-procs --keep none --subject-list <path/to/incomplete-subject-list.txt>`

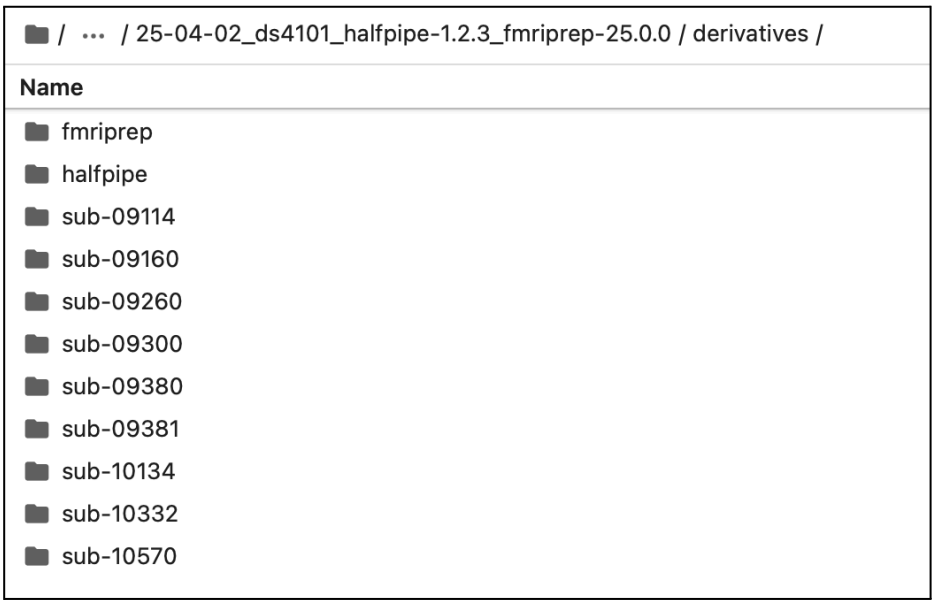
    b. ***Only if running on a HPC***:
        - Submit again the job following instruction in **Section 8.2.**
        - After the jobs have finished, rerun the steps in the section until no incomplete subjects remain.

Content of the output

Example of the working directory generated by HALFpipe after completing **Section 7**:



Example of the derivative folder generated by HALFpipe after completing **Section 8**:



10. Quality control

- Complete the quality assessment manual that is linked here.
- Please contact us at halfpipe@fmri.science if you are unsure about a particular image and have not been able to solve the issue with a knowledgeable person (scanner technician/MR physicist/other expert).
- Please also e-mail us if you discover a particularly interesting image that we may want to add to the quality assessment manual.
- **On a local machine**
    1. Open your file manager (finder on mac, or file explorer on Windows)
    2. Go in your working directory `path/to/your_working_directory`
    3. Go to the folder **reports**
    4. Open **index.html,** this will open a webpage where you can perform the QC as mentioned in the manual
- **Only if running on HPC:**
    1. Once processing completes, copy the **reports folder** from your working directory to a computer where you can comfortably do quality assessment.
    2. Once your QC is done, copy-paste the **exclude.json** file in your cluster in the `path/to/your_working_dir/reports.` You will need it later to exclude participants when performing statistical analysis

11. Troubleshooting

1. Freesurfer license error

If you get the following error:

```
RuntimeError: fMRIPrep needs to use FreeSurfer commands, but a valid
license file for FreeSurfer could not be found.
│ HALFpipe looked for an existing license file at several paths, in
this order:
│ 1) a "license.txt" file in your HALFpipe working directory
│ 2) command line argument "--fs-license-file"
└─Get it (for free) by registering at
https://surfer.nmr.mgh.harvard.edu/registration.html
```

Please request the license at https://surfer.nmr.mgh.harvard.edu/registration.html. You will receive an email with the license.txt file that you can copy and put directly in your current working directory.

2. Memory error

Example of an error in log.txt or in err.txt:

```
slurmstepd: error: Detected 1 oom-kill event(s) in step 13601774.batch cgroup.
```

Some of your processes may have been killed by the cgroup out- of-memory handler.

Solution: Request more memory when submitting the job by modifying the following parameters depending on your HPC's documentation.

SLURM	`#SBATCH --mem=10752M`
SGE	`#$ -l h_vmem=10752M`
Torque/PBS	`#PBS -l mem=10752mb`

3. **Singularity error**

Example of an error in log.txt: `singularity: command not found`
Solution: load singularity before submitting the job (depends per cluster, for example: `module load singularity`)

4. **Environment Errors in HPC**

- Ensure consistent `--bind` option used when building the container and the one in slurm.submit file as well as version of Singularity or Apptainer.
- If you have to load an environment in your submit.slurm (e.g. `module load StdEnv/2020`), make sure to load the environment before loading Singularity or Apptainer (i.e. `module load Singularity`)
- Make sure that all paths in the submit.slurm file are either **all relative or all absolute**—do not mix both types. **Consistency** is important to ensure the job runs correctly.

5. **Large dataset (for advanced user)**

On HPC cluster only: if you have a large sample, you can submit **step 6** as a job (instead of running it via interactive session). To do this, take the following steps:

- To specify the settings run: `singularity run --no-home --cleanenv --bind /:/ext halfpipe_latest.sif --only-spec-ui`. This will open the user interface as described above.
- Refer to **step 6** for the settings to enter. Once you have entered all the settings, a spec.json file will be created.
- Submit the following job to create the workflow and execgraph files: `singularity run --no-home --cleanenv --bind /:/ext <path/to/the/pipeline>/halfpipe_latest.sif --use- cluster --skip-spec-ui`

This step cannot be parallelized. In a small sample, this step will only take a few minutes, however, for hundreds of subjects, this may take up to a few hours. This step creates the following files and folders: **derivatives, nipype, rawdata, reports, submit.slurm.sh, work, workflow***.pickle.xz, execgraph***.pickle.xz**

6. **Resuming a HALFpipe container run from a crash, adding more features to an existing run, reselecting features for an existing run, starting a run from scratch.**

If at all you need to perform any of the above on an existing working directory follow this paragraph. Redo **section 6** to launch an interactive session and run the container launch command. Begin with the instructions in **section 7.1** and select your existing working directory instead of a new one. You will be prompted with the options above. Please select the one which you need.

7. **Avoiding interruption of HALFpipe execution due to HPC disconnection using tmux.**

For HPC Users: if your connection to the cluster drops during section 6/7, the execution will fail and you will have to restart the process. To avoid this, you can **use TMUX to run your HALFpipe execution persistently in the background (if available on your HPC)**. This way, even if your SSH connection drops, you are able to disconnect and later reattach to the session without interrupting HALFpipe's execution**.** Please refer to your cluster documentation or IT support.

Follow the steps below to run a TMUX session:

1. Open a terminal session and connect to your cluster,
2. Start a new TMUX session: `tmux new -s halfpipe`
3. Run your HALFpipe launch command from **Section 6**.
4. Once the process is running and after specifying the settings in **Section 7**, you can detach from TMUX: `Ctrl + b` then `d`
5. You can reattach at any time with: `tmux attach -t halfpipe`
    a. TIp: Type `tmux ls` to list active sessions.
6. When done, type `exit` inside the session to close it.

12. How to cite

HALFpipe software:

Waller, L., Erk, S., Pozzi, E., Toenders, Y. J., Haswell, C. C., Büttner, M., Thompson, P. M., Schmaal, L., Morey, R. A., Walter, H., & Veer, I. M. (2022). ENIGMA HALFpipe: Interactive, reproducible, and efficient analysis for resting-state and task-based fMRI data. Human brain mapping, 43(9), 2727–2742. https://doi.org/10.1002/hbm.25829

Seed Atlas used in this manual:

Pozzi, E., & Toenders, Y. (2024, November 8). ENIGMA HALFpipe ROIs for seed-based connectivity. Retrieved from osf.io/u786p

Atlases used for atlas-based connectivity:

Buckner, R. L., Krienen, F. M., Castellanos, A., Diaz, J. C., & Yeo, B. T. (2011). The organization of the human cerebellum estimated by intrinsic functional connectivity. Journal of neurophysiology, 106(5), 2322–2345. https://doi.org/10.1152/jn.00339.2011

Fischl, B., Salat, D. H., Busa, E., Albert, M., Dieterich, M., Haselgrove, C., van der Kouwe, A., Killiany, R., Kennedy, D., Klaveness, S., Montillo, A., Makris, N., Rosen, B., & Dale, A. M. (2002). Whole brain segmentation: automated labeling of neuroanatomical structures in the human brain. Neuron, 33(3), 341–355. https://doi.org/10.1016/s0896-6273(02)00569-x

Schaefer, A., Kong, R., Gordon, E. M., Laumann, T. O., Zuo, X. N., Holmes, A. J., Eickhoff, S. B., & Yeo, B. T. T. (2018). Local-Global Parcellation of the Human Cerebral Cortex from Intrinsic Functional Connectivity MRI. Cerebral cortex (New York, N.Y. : 1991), 28(9), 3095–3114. https://doi.org/10.1093/cercor/bhx179