

Download all files and get ready to run:

- 1) This will download the python scripts and the Dockerfile needed to run the scripts in the self-contained Docker environment:
`git clone https://github.com/TIGRLab/FeenICS`
- 2) Install docker if you don't have it: `sudo apt-get install docker.io`
- 3) `cd FeenICS`
- 4) Copy the file Dockerfile to /usr/bin (this is where docker should be installed):
`sudo cp Dockerfile /usr/bin/.`
- 5) `cd /usr/bin`
- 6) Build the docker container from within /usr/bin: `sudo docker build -t feenics .`
NOTE: this may take a while - FSL needs to be installed in the docker environment, and some other commands run in the background after it is installed -WAIT-.
NOTE: If you have to update the Dockerfile, rebuild with the following command:
`sudo docker build -t feenics --no-cache .`
- 7) set up your directory structure with only the data (see 'initial directory structure in the table). Do not make the "Cleaned" directory until just before s3. Check that permissions are fully open to the world, eg: `chmod -R 777 <myworkingdirectory>`

Run the artifact detection scripts in the docker environment:

- 1) run docker: `sudo docker run -v /path/to/data:/input -i -t feenics`
In most cases, 'path/to/data' will be the same as '/input'. Example:
`sudo docker run -v /home/myworkingdirectory:/home/myworkingdirectory -i -t feenics`
Note: this command will put you at the root directory inside the docker environment.
Note: tab-complete works in docker, so you do not have to type the full command.
- 2) run the first step:
`s1_folder_setup.py /home/myworkingdirectory`
The options can be used if you are running on the cluster (-p) or do not want to use the suggested initial directory structure (-i, -s). Instructions for using the options are in the README.md file in FeenICS directory.
- 3) It will look like step1 has finished. It hasn't. WAIT... Do not run the next step until all instances of melodic have finished. To see if it has, run the "top" command (exit "top" with Ctrl-c or `q`).
- 4) run the second step (this is fast):
`s2_identify_components.py /home/myworkingdirectory`
Note: The output, fix4melview_Standard_thr20.txt, is a csv file and can be loaded into excel if you need to look at it and edit the final ICs for removal.
Note: you should not need to modify the midFactor and lowFactor.
Note: if you get a permissions error, go to the working directory and `chmod -R 777 ./` again. If that doesn't work, contact Nancy or Sofia.
- 5) **Check your data** to see if you agree with the automatic 'spiral artifact detection'. You can look at the results in fix4melview_Standard_thr20.txt. Otherwise, Erin Dickie's ICarus works well -
`cd /home/myworkingdirectory`
`sprls=`ls -ld */sprl*/``
`icarus-report ${sprls}`
ICarus output is put in the working directory and can be opened with your browser. ICs are sorted by signal/noise. If you don't agree, you can switch the label using the radio buttons under each IC. If you change any IC's labels, you will need to change the last line in fix4melview_Standard_thr20.txt for that spiral file. Probably a good idea to add a new line at the end of fix4melview_Standard_thr20.txt rather than overwriting:
...
[18,26,27,28,39,40,46,48,51,56,57,58,59,62,67,68,69,70,71,75,77,78,79,82,84,86,87,88,89,90,91,92,93,94,96,97,98,100,101,102,103,104,106,108]
[18, 26, 27, 28, 35, 36, 39, 40, 46, 48, 51, 53, 56, 57, 58, 59, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 73, 75, 77, 78, 79, 82, 84, 86, 87, 88, 89, 90, 91, 92, 93, 94, 96, 97, 98, 100, 101, 102, 103, 104, 106, 108]
6) Create the output directory for the denoised/cleaned data. In the docker:
`cd /home/myworkingdirectory` (if you aren't already there)
`mkdir my_cleaned_data`
- 7) Remove the artifact components from all sprlIN.nii and sprlOUT.nii and put output in cleaned data directory:
`s3_remove_flagged_components.py -c /home/myworkingdirectory/my_cleaned_data /home/myworkingdirectory`

If you see the following message, this means no noise components were found. Most often with sprlIN.

Removing components from <##directory##>, sprlIN

vector::_M_range_check

You will need to copy the sprlIN.nii file to the cleaned data directory and rename it to include the ".sprlIN.denoised" suffix to match the expected naming conventions, and then gzip it. (added 2018.03.06)

Exit the docker environment and recombine the corrected spirals.

- 1) Type 'exit' to close the docker
- 2) Run Sofia's recombine script to recombine the denoised sprlIN and sprlOUT files.
`cd /home/myworkingdirectory/my_cleaned_data`
`path_to_sofias_script/combinesprl` (should be in FeenICS/bin - make sure it is executable)

output will be in the 'my_cleaned_data' directory, with the same naming structure, eg:
 Subj01_ExamSeries01.sprlCOMBINED.denoised.nii
 There will also be a number of other files, which can be used for troubleshooting - see Nancy or Sofia if you don't understand the outputs and/or the script that made them.

NOTE: the version of FSL installed in the docker for melodic cleaning is a python distribution v5.0.10

NOTE: you can also run these python scripts from the command line, if all of the dependencies are correct. ICARUS has different dependencies than FeenICS. FSL v5.0.8 (linux) runs FeenICS python commands OK from the command line.

Initial Directory Structure	Final Directory Structure	Cleaned filenames conventions:
.. MyWorkingDirectory Subj01_ExamSeries01 sprlIN.nii sprlOUT.nii Subj01_ExamSeries02 sprlIN.nii sprlOUT.nii Subj01_ExamSeries03 sprlIN.nii sprlOUT.nii Subj02_ExamSeries01 sprlIN.nii sprlOUT.nii Subj02_ExamSeries02 sprlIN.nii sprlOUT.nii Subj02_ExamSeries03 sprlIN.nii sprlOUT.nii	.. MyWorkingDirectory my_cleaned_data Subj01_ExamSeries01 sprlIN sprlOUT sprlIN.nii sprlOUT.nii Subj01_ExamSeries02 sprlIN sprlOUT sprlIN.nii sprlOUT.nii Subj01_ExamSeries03 sprlIN sprlOUT sprlIN.nii sprlOUT.nii Subj02_ExamSeries01 ... etc	The "my_cleaned_data" directory will contain all denoised files, separated. These will be needed for the "recombine" step. Filenames will duplicate the directory structure: Subj01_ExamSeries01.sprlIN.denoised.nii.gz Subj01_ExamSeries01.sprlOUT.denoised.nii.gz Subj01_ExamSeries02.sprlIN.denoised.nii.gz Subj01_ExamSeries02.sprlOUT.denoised.nii.gz Subj01_ExamSeries03.sprlIN.denoised.nii.gz Subj01_ExamSeries03.sprlOUT.denoised.nii.gz Subj02_ExamSeries01.sprlIN.denoised.nii.gz Subj02_ExamSeries01.sprlOUT.denoised.nii.gz Subj02_ExamSeries02.sprlIN.denoised.nii.gz Subj02_ExamSeries02.sprlOUT.denoised.nii.gz Subj02_ExamSeries03.sprlIN.denoised.nii.gz Subj02_ExamSeries03.sprlOUT.denoised.nii.gz

Important note: your final output files will be motion-corrected and skull-stripped (betted) - those steps should not be repeated. The images have not been registered to template.

Once you are happy with the corrections, you can apply whichever pipeline you would prefer. This includes re-running ICA if you feel the need to remove residual motion artifact and/or pulsatility, etc.

NOTE: If you plan to include motion-correction parameters as regressors in your analysis, you may need to do some more work since sprlIN is motion-corrected separately from sprlOUT. The two sets of vectors do not necessarily correlate 100%, for each component of the transformation matrix. The first three parameters tend to correlate at $r > 0.90$. One option is to move the denoised images back to native non-motion-corrected space and then run the recombine step. You would then use whatever motion-correction scheme you think appropriate for the data. Data would still be skull stripped. 2018.03.06.