

aa

aa provides a powerful batching system with lots of useful features, for example it is restartable (i.e. remembers where it was up to if it gets interrupted), provides multi-level, interlinked diagnostics, supports 4D NIFTI and can handle parallel processing of multiple subjects/sessions.

Accessing the cluster

1. Login the cluster login node
Launch putty
 - Host Name: <user name>@psyclogin.rhul.ac.uk
 - Post: 22
 - Password: *****
 2. Launch a VNC server (if not running)
`vncserver -geometry 1920x1010 :<session number>`
- N.B.:
Setting geometry is optional and usually matches with the screen resolution (- considering menu bars).
3. Close putty
`exit`
 4. Launch vncviewer
 - VNC server: psyclogin.rhul.ac.uk:<session number>
 - Password: *****
 5. Open terminal
Applications > System Tools > Terminal
 6. Access working directory in /MRIWork
`cd <path to working directory>`

Starting SPM/aa

1. Launch MATLAB
`matlab`
- N.B.
MATLAB R2015b is the default
2. Setup (only once)
`addpath /usr/local/apps/psycapps/spm/spm12-r7219`
`spm_jobman initcfg`
`addpath /usr/local/apps/psycapps/automaticanalysis/automaticanalysis`
`aa_ver5`
`savepath <path to working directory>/pathdef.m`
-
- ```
parallel.importProfile('/usr/local/apps/psycapps/cluster/PsychoSGE_2015b.settings')
parallel.defaultClusterProfile('PsychoSGE')
```

SPM and aa are essentially large collections of Matlab scripts, and you can integrate these into your own custom analyses as needed.

## Prepare User Master Script

All the pre-processing steps described below are included in a batch script you can find in your imaging space at:

```
/imaging/train0Xlinux/Workshop/Material/4_aa
```

If you cannot find them there, you can always grab them from:

```
/MRIWork/Training/Material/tibor_auer/4_aa
```

If you've not already done so, make a directory in your imaging space for workshop related stuff. In either matlab or a terminal window type:

```
mkdir <path to working directory>/4_aa
```

Then copy the content of the workshop folder to your local directory. In a terminal window, type:

```
cp /MRIWork/Training/Material/tibor_auer/4_aa * <path to working directory>/4_aa
```

(N.B.: You can also issue system command in matlab with a prefix "!" )

Once you've done this, open the script.

The data set we're going to be working on comes from an auditory task where people were listening a lullaby. The raw data are located in:

```
/usr/local/apps/psycapps/automaticanalysis/aa_demo/aa_demo/2014_03_29_9001
```

The script is set up to create sub directories within *aap.acq\_details.root/aap.directory\_conventions.analysisid* for each module and subject

As the pre-processing progresses, you'll notice that various prefixes are added to the images. When the functional data is first imported, the images will have the prefix 'f'. Realignment (+unwarping) will change it to 'uf'. If you do Slice timing adds 'a', normalisation 'w', and smoothing 's'. At the end of the pre-processing pipeline, you end up with a series of images with prefixes like 'swauf' (=smoothed, normalised, slice time corrected, unwarped), 'wauaf', 'sauaf' etc depending on exactly which pre-processing steps you carry out.

But you do not have to worry about them because the aa stream-concept will deal with them.

As you analyse more and more data, the temptation becomes just to run the preprocessing automatically and not really check the results. Whilst the typical preprocessing pipeline is fairly robust, it's always worth spending some time **checking your data** (report, diagnostics in the module directory). At the very least it's worth having a look at the mean and variance images for each session, the results from *tsdiffana*, the plot generated for the raw data right after conversion and after preprocessing.

You can get (not always, though 😊) extra information about each of the spm routines/aa modules by typing *help <function name>* in the Matlab Command Window.

### *If you are running the fully written script (aa\_demo\_script\_full.m)*

To get a better idea of what's going on at each line, you can run the script one line at a time. Put a breakpoint on the first line (click the "-" sign on the left of the Matlab editor window next to line 18. A red dot should appear). Press F5 to run the script, then press F10 to run each line in turn. To see the contents of variables at each step, hover the mouse cursor over the variable name. As you step through the script, read the comments (in green) to get a better idea of what's happening on each line.

You can also run it section-by-section (%%).

The first few lines of the script define several variables that will be used, for example the paths of various directories etc.

Before running anything, change

- line 16 to your tasklist:  
`aap=aarecipe('aap_parameters_defaults_RHUL.xml','<path to working directory>/4_aa/aa_demo_tasklist.xml');`
- line 20 to your e-mail address: `aap.options.email = '<your e-mail address>';`
- line 40 to top analysis folder: `aap.acq_details.root = '<path to working directory>/4_aa';`

Let's run the script and then I show you around.

**Navigate analysis folder.**

**Navigate .aa folder (qsub).**