AFNI\_NoteBook User’s Guide

**Overview:**

This MATLAB program is written to improve user’s experience of using AFNI in fMRI analysis. This toolbox is straightforward and intuitive, but before implementing it in your own research, at least you should have some basic knowledge about fMRI analysis and AFNI software, thus the target users would be intermediate AFNI users. The main features of this program are as follows:

1. It is a GUI-based program, once you load (manually) AFNI scripts to the program, you don’t need to remember those tedious scripts anymore, instead, just several clicks you will be able to get the scripts you loaded previously.
2. It is very simple to extend this program, you can add custom features to this program quickly and easily, and adapt it to your personal needs.

**Contact information:**

For more information about this toolbox, contact [liujin13@msu.edu](mailto:liujin13@msu.edu). Hope you enjoy this toolbox!

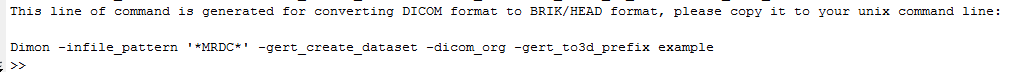
**Step-by-Step Guide:**

This guide is demonstrated on the sample data which has been uploaded to the department server, to replicate these processing steps you might want to run on this dataset.

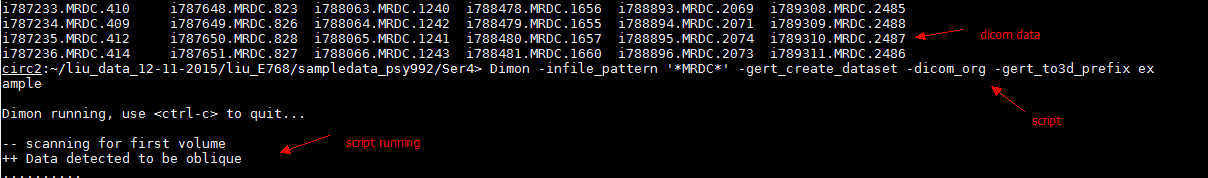
**1/ DICOM to Nii/AFNI**

This panel will be the first part you want to have a look at every time you get the raw data from MRI scanners. The raw data is usually stored in DICOM format, to read the data into common fMRI data processing packages you usually need to convert the DICOM format data into NIFTI (nii) format. AFNI also supports this nii format, but it also has its own compatible format: BRIK/HEAD. In this example, I will just illustrate how you can convert DICOM data to AFNI format data.

Since unix terminal supports regular expression and this toolbox is just used to write scripts for terminal, thus, it also supports regular expression. To convert single run data, you need to provide raw data pattern name to left input cell under the panel of “Local” (in the demo data: \*MRDC\*), to type output data name in right cell (in the demo: outputfile), then click “Generate” (Figure 1), the output script will be sent to your MATLAB terminal:



You can copy the line of script and paste it to your unix terminal:



You are expected to get the converted data (in this example, it will be example+orig.BRIK/HEAD) under your current directory.

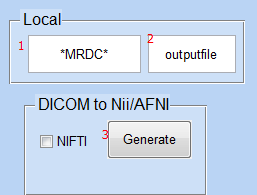


Figure 1.

This one line script can only deal with one run, but as we know, in reality each participant usually has multiple runs. To convert multiple runs’ data at a time, you need to check the “Loop” box in the “Global” panel, and give the folder pattern name. For example, in Figure 2, the “/Ser\*” in the given example represents the folders where raw datasets were stored. After checking “Loop” box and clicking “Generate” button, a file called “Dicom2Afni.txt” will be generated under your current MATLAB directory. You can copy its scripts and execute them in unix terminal, and ideally, a new folder named “outputfile” will be created and all converted data will be saved to that folder.

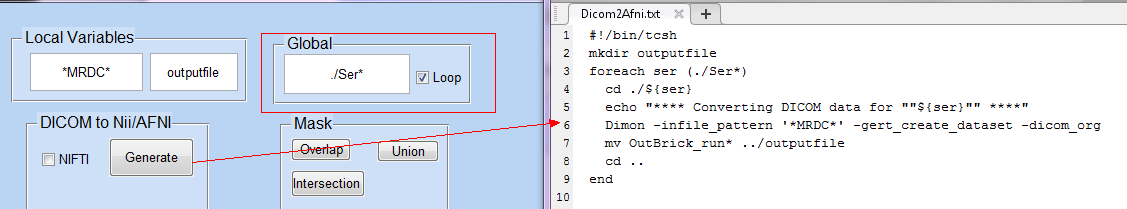


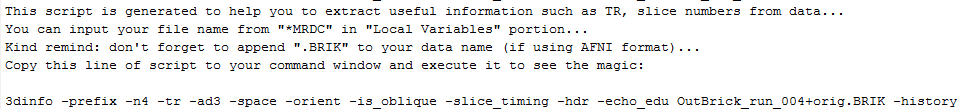
Figure 2.

**2/ Talk with AFNI**

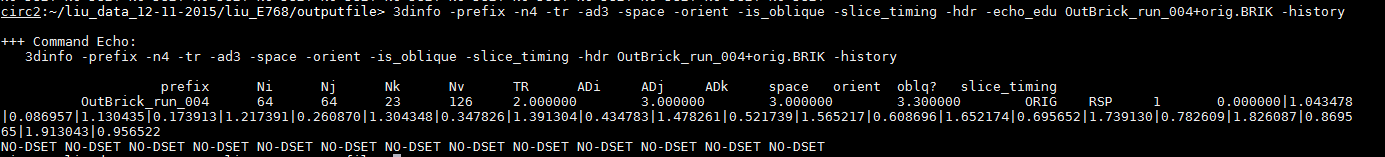
This is a simple function to help you to call AFNI from MATLAB, you might want to use AFNI’s image viewer to visually check the quality of data(e.g. ghosting, spiking) before running any specific analyses. Then the core of this panel is to call the uber\_subject.py (see Figure 3), a python-based gui written by AFNI team to generate standard processing (pre and post-processing) scripts. This gui is good enough for basic processing needs, so I just borrowed it and embedded it into my toolbox. For more information about the use of this gui program, go to: <http://afni.nimh.nih.gov/pub/dist/doc/program_help/uber_subject.py.html>

**3/ Data Information**

To extract basic informaiton, such as TR, voxel size, from data, you can click “Data Info” from this portion, but firstly, you need to input the name of data you want to check to left cell in “Local”. The output in MATLAB terminal will be:



Run this script in unix terminal, you will get output like this:



For more information about the meanings of these parameters, go to: <http://afni.nimh.nih.gov/pub/dist/doc/program_help/3dinfo.html>

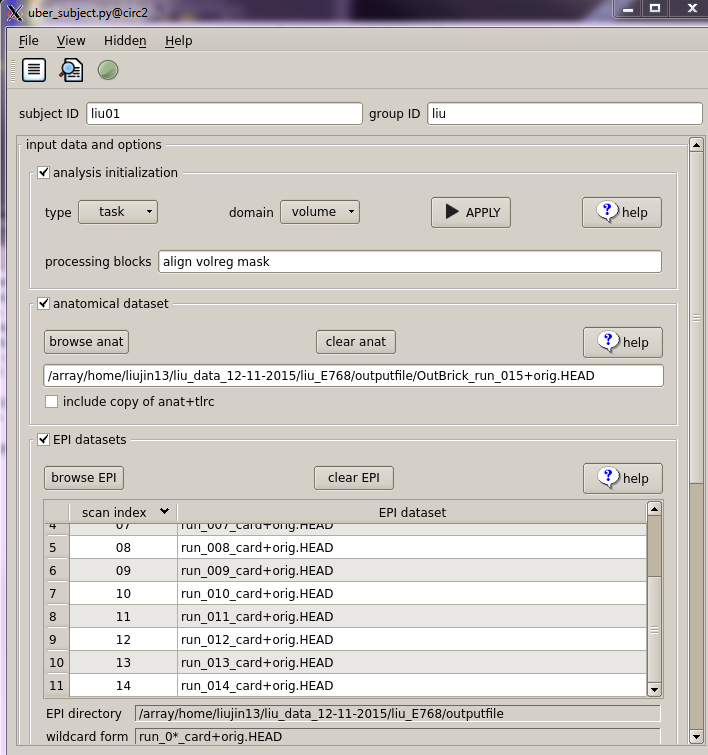
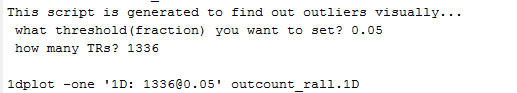
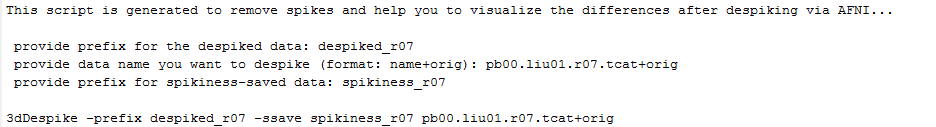


Figure 3.

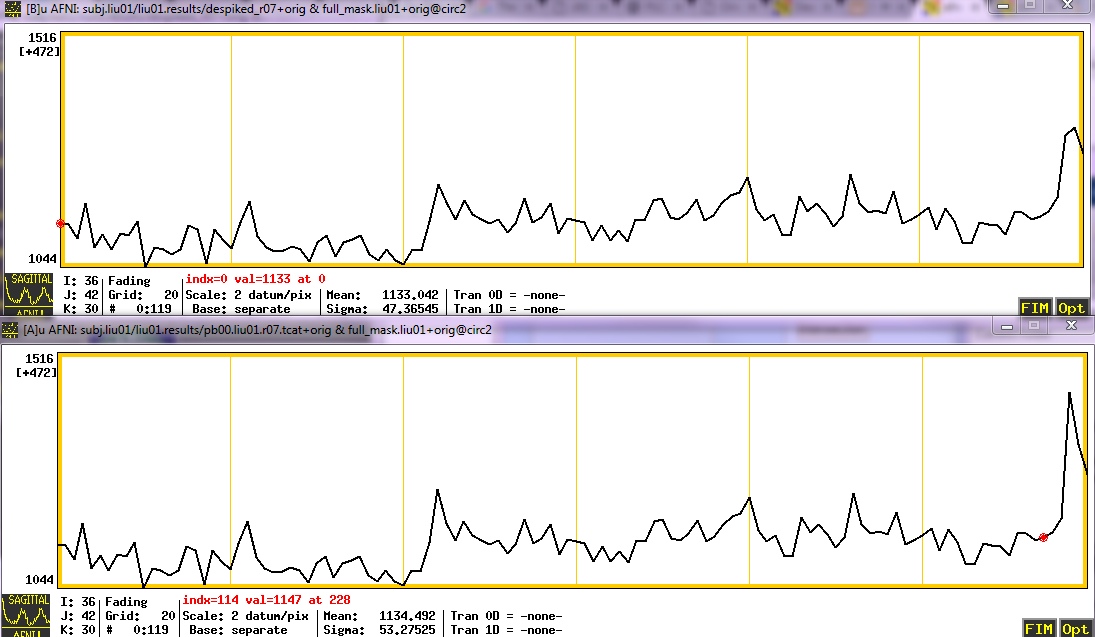
You might also want to check if there are any outliers in the dataset, to do this step, you need to firstly run the uber\_subject.py and get the “outcount\_rall.1D” file in your directory, this file is a single time series of outlier fraction (outlier voxel count divided by the number of all voxels) per TR, if the outcount fraction is large, that signals something was wrong with that TR (perhaps a big head motion), then you might want to scrutinize that TR. For example, after clicking “Visualize Outliers”, some questions will be popped up to ask you to input essential parameters, just as follows:



After running the generated script in unix terminal, you will get a plot (Figure 4) showing the time series of outlier fraction. Here we know the TR 837,TR 838 and TR 839 have serious outlier problem, since we know these TRs are in run 7, so if you load run7 EPI data (in the demo: pb00.liu01.r07.tcat+orig), move crosshair to (13.5,39,11.22), and plot time series, using movie model (press “v”) you will notice there was a small move at the end of this run. Thus this function is very useful to find outliers. After detecting outliers, we can use “Despike” function to despike outliers:



If we directly compare pre-despike data and despiked data, you will find despiking did its job (top is despiked data, bottom is original data, compare the ending parts of two time courses):



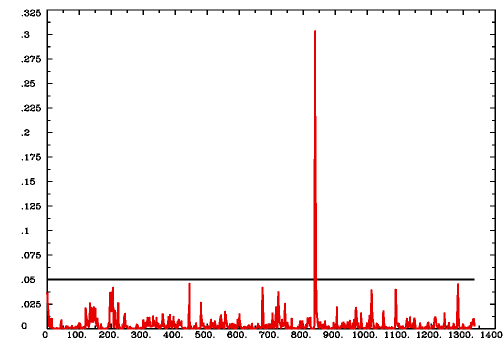


Figure 4.

“Review Table” function will be useful if users want to compile many out.ss\_review.SUBJECT.txt files spread in different folders together, and make a tab-delimited table of output fields, one infile/subject per line. The out.ss\_review.SUBJECT.txt file is generated from @ss\_review\_basic, a script which is automatically created after running uber\_subject.py.

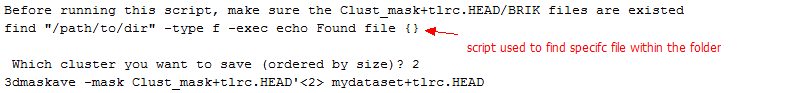
**4/ Normalization**

This portion is written to help with several normalization processing steps. When you click “Check Nor” button, you may feel you are fooled, because it actually has nothing to do with normalization, instead, it will generate the script for converting AFNI format data to NIFTI data. I have a reason for that before you get angry, I found checking normalization is a little complicated in AFNI, but in SPM, the job will be easily done, and SPM even has a specific button for checking registration, so I would suggest using SPM to do it. Because SPM can’t read AFNI format data, but NIFTI data can be loaded, so this function helps users to convert data format. Hope you are satisfied with the explanation, dear user!

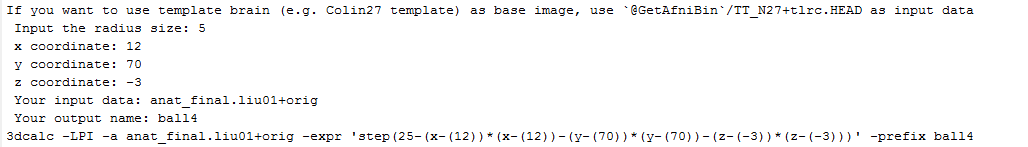
Sometimes, you probably want to transform data from one standard space to another standard space, if you want to transform from TLRC space to MNI space, use “tta2mni”, from MNI to TLRC space, use “mni2tta”. You might also want to go back to individual space when you need to localize individual brain, in this case, use “tta2native”.

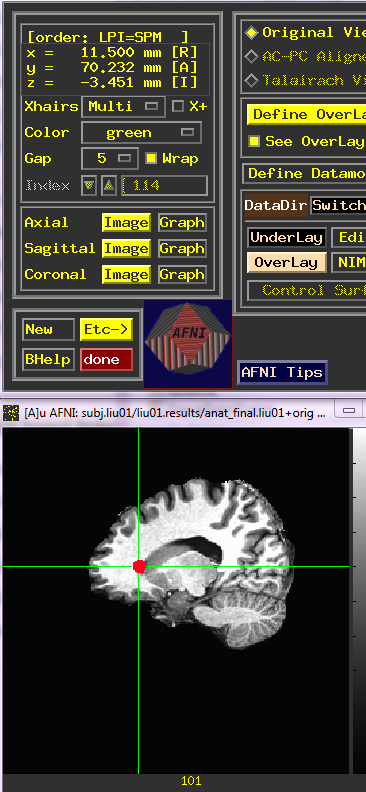
**5/ ROIs**

There are several ways of making ROIs, one of the most common ways is to save activations as ROIs. To do that, you first need to save multiple activation clusters as a file by pressing “SaveMsk” in AFNI’s cluster results, AFNI will automatically create “Clust\_mask+tlrc.HEAD/BRIK”, each cluster will be saved with a separate value in HEAD/BRIK files, with the HEAD/BRIK files in your directory, you are able to use “Activations” function under the MATLAB toolbox, just input the cluster index you would love to use as mask at the prompt, then you will get the script:

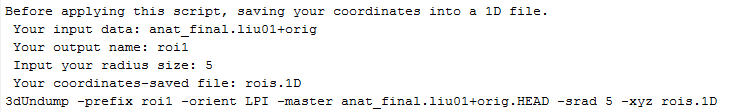


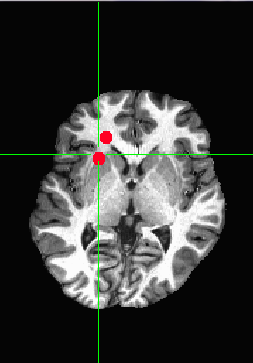
If you want to create your own ROI, you can seek help from “Single” function, you need to provide the radius size, x-y-z coordinates, base image name, and prefix for output data. Currently, it only supports to create sphere ROIs, maybe in the future, I will add more interesting ROI shape choices. I should also note that by default this sphere ROI is in LPI/SPM order. LPI and RAI is another very confusing topic in AFNI, if you have interests, refer to this post: <http://afni.nimh.nih.gov/afni/community/board/read.php?1,51292>





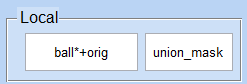
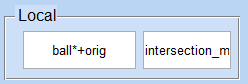
To save multiple ROIs at a time, use “Multiple” function:

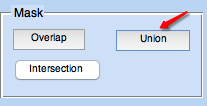
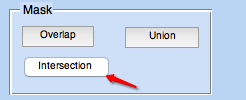




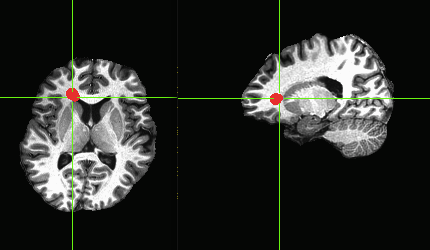
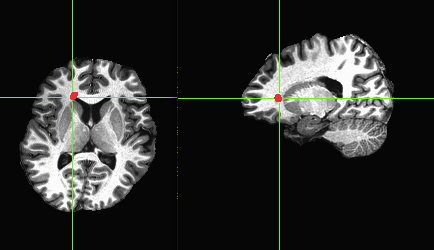
**6/ Mask**

After making individual ROIs, we may want to unite them or get the intersection of these ROIs. These works can be easily done in AFNI. In this given example, I have created two ROIs (ball4+orig and ball5+orig, ball4 has a radius size of 5 mm, centered on coordinates (12 70 -3), ball5 has a radius size of 6, centered on coordinates (15 73 -4)), you can play around to see how it works.

Union mask Intersection mask