

# Manual: Atlas based Processing Pipeline for functional and structural MRI Data - AIDA

Niklas Pallast  
Department of Neurology  
University Hospital Cologne

2018  
October

## 1 Introduction

The **A**tlas based Processing Pipeline for functional and structural MRI **D**ata (AIDA) was developed for automated processing of preclinical high-field magnetic resonance imaging data of the mouse brain. AIDA is able to associate structural and functional datasets. That includes T2-weighted MRI (T2w), diffusion weighted MRI or diffusion tensor imaging (DTI) and functional MRI (fMRI). The Allen Brain Reference Atlas (ARA) is registered on each of these MRI datasets and is used to analyse regions of interest. Furthermore, the regions of the ARA are used as seed-points for the connectivity and activity matrices.

## 2 Installation

- Download the folders `/bin` and `/lib`  
<https://github.com/maswendt/AIDA>  
`/bin` and `/lib` should be located in the same directory
- Download example data to `testData.zip`  
[https://web.gin.g-node.org/pallastn/AIDA\\_dataset](https://web.gin.g-node.org/pallastn/AIDA_dataset)

- Download & Install FSL 5.0.1  
<http://fsl.fmrib.ox.ac.uk/fsldownloads>
- Download & Install Python 3.6 (it is recommended to use Anaconda)  
<http://docs.anaconda.com/anaconda/install/>
- Install NiftyReg by conducting the following steps:
  1. Generate an folder `.../NiftyReg/niftyreg_source`
  2. Download the latest version from the git repository on SourceForge, using:  
`git clone git://git.code.sf.net/p/niftyreg/git <path>/niftyreg_source`
  3. Change folder `cd <path>/niftyreg_source`
  4. Type `git reset --hard 83d8d1182ed4c227ce4764f1fdab3b1797eecd8d`
  5. Follow the steps of  
<https://cmiclab.cs.ucl.ac.uk/mmodat/niftyreg/wikis/install>
- Install DSI-Studio and copy the install path into  
`.../bin/3.2.DTICnectivity/dsi_studioPath.txt`  
<http://dsi-studio.labsolver.org/dsi-studio-download>
- Install nipy 1.1.2  
<https://nipy.readthedocs.io/en/0.11.0/users/install.html>
- Install lmfit 0.9.11  
<https://lmfit.github.io/lmfit-py/installation.html>
- Install progressbar2 3.38.0  
<https://pypi.org/project/progressbar2/>

### 3 Usage of AIDA

Attention: All program examples are only listed with the mandatory input parameters. For more details, call `python .../python <command> -h`. The command line examples are given with the identifier `testData<No.>.nii.gz`

and can be identically applied to other data. The test dataset is freely available and can be downloaded from <https://doi.org/10.12751/g-node.70e11f>. The user should not skip or resort the pipeline steps, since previous calculations provide results for subsequent steps.

### 3.1 Convert raw data

Convert bruker raw data to NIfTI files by specifying the folder containing all raw folders of each scan. A file with exactly the same name is created in the given input folder. It contains all sorted NIfTI files. The raw data should have the same orientation as the example dataset.

```
python pv_conv2Nifti.py -i .../testData
```

### 3.2 Processing of T2w & T2map data

Apply the reorientation, bias field correction and brain extraction to the T2w data set. The automatically attached endings of the processed filenames indicate which steps have been performed. Brain extraction should be of good quality and must be manually checked or corrected by adapting the default parameter.

```
python preProcessing.T2.py -i .../testData/T2w/testData.5.1.nii.gz
```

The registration will also work without the following step. The user can segment a region by taking the brain extracted dataset as reference (ends with ...BET.nii.gz). We recommend to conduct this step with **itk-SNAP**. The saved file should end with the extension ...Stroke\_mask.nii.gz

The next step includes the registration of the Allen Brain Reference Atlas with the brain extracted T2 dataset. The result is a variety of files. An impression of the registration can be obtained by superimposing the file the brain extracted file with the annotations of the Allen Brain (ends with ...Anno.nii.gz)

```
python registration.T2.py -i .../testData/T2w/testDataBiasBet.nii
```

If the user previously defined a region of interest, the region size, segmented parental regions and segmented original regions can be determined in that step. Here, the segmented region ...Stroke\_mask.nii.gz is overlaid with the Allen Brain Reference Atlas and saved in the file ...Anno\_mask.nii.gz. The user does not have to enter single files, but the path to the .../T2w folders

```
python getIncidenceSize_par.py -i .../testData/T2w
python getIncidenceSize.py -i .../testData/T2w
```

The results, such as affected regions and ROI volume are stored in the folder `.../T2w` in the following files

```
affectedRegions.txt
affectedRegions.nii.gz
affectedRegions_Parental.txt
affectedRegions_Parental.nii.gz
```

### 3.3 Processing of T2 data

From the stroke masks drawn on the T2 weighted images, it is possible to determine both the incidence map and the size of affective regions. For example, if a `day1` folder contains multiple `Mouse_1-Mouse_15` folders and the processed T2 data is in those folders, the command would be as follows

```
python getIncidenceMap.py -i .../day1 -s Mouse*
```

### 3.4 Processing of DTI data

The DTI processing procedure includes a dimension reduction, bias correction, a threshold application, and the subsequent brain extraction. The endings on the filenames indicate which steps have been performed. Brain extraction should be of good quality and must be manually checked or corrected by adapting the given parameters.

```
python preProcessing_DTI.py -i .../DTI/testData.7.1.nii.gz
```

The next step includes the registration of the Allen Brain Reference Atlas with the brain extracted DTI dataset. Here, two processing options are possible a) Registration of a mask that is related to an other dataset - append command `-p <filename of ref>` b) By omitting the command , the algorithm stroke mask from the same folder or no mask.

```
python registration_DTI.py -i .../DTI/testDataSmoothMicoBet.nii.gz
```

The connectivity is finally calculated via DSI-Studio. All connectivity matrices are based on the reference atlas.

```
python dsi_main.py -i .../DTI/testData.7.1.nii.gz
```

The connectivity matrices of the parental Atlas, the original Atlas and the related ROI are stored in the folder `.../DTI/connectivity` as `.txt` and `.mat`. DSI-Studio differentiates between matrices that count how many pass and end in

each region. The adjacency matrices can be visualised the related plot function.

```
python plotDTI_mat.py -i  
.../testData/fMRI/connectivity/testData*.connectivity.mat
```

## Processing of fMRI data

The fMRI processing is roughly comparable to the preprocessing of the DTI datasets. Brain extraction should be of good quality and must be manually checked or corrected by adapting the given parameters.

```
python preProcessing_fMRI.py -i .../fMRI/testData.6.1.nii.gz
```

The step includes the registration of the Allen Brain Reference Atlas with the brain extracted fMRI dataset. The result is a variety of files. An impression of the registration can be obtained by superimposing the file the brain extracted file with the annotations of the Allen Brain (ends with ...\_Anno.nii.gz)

```
python registration_fMRI.py -i .../testData/fMRI/testSmoothBet.nii
```

If the user do not have physiological data, the step will be conducted without the included regression. All activity matrices are based on the reference atlas.

```
python process_fMRI -i .../fMRI/testData.6.1.nii.gz
```

The activity matrices of the parental Atlas, the original Atlas are stored in the folder .../fMRI/regr as .txt and .mat with the prefix

MasksTCs. and MasksTCsSplit.. The adjacency matrices can be visualised the related plot function.

```
python plotfMRI_mat.py -i .../testData/fMRI/regr/MasksTCsSplit*.mat
```