

MRI Neuroimaging Data Exploration and Masking with NiBabel and Nilearn

GitHub link: https://github.com/isadorasinha/python_assessment_2021

The current PhD rotation project involves exploring and analysing magnetic resonance imaging (MRI) neuroimaging data obtained from wildtype mice and mice with a paternal Grb10 knockout. As the MRI images of this data was not cropped nor many Python tools available for rodent neuroimaging, Python based computational tools for exploring and masking for human neuroimaging were explored.

There were two sets of example data used: firstly, one MRI brain image file and the corresponding mask. Secondly, later in the code the dataset used was obtained from Haxby *et al.* (2001) and contains full-brain fMRI data which were recorded with a volume repetition time of 2.5s. All files used in this data exploration were specialised formatted MRI output files called Neuroimaging Informatics Technology Initiative (NIfTI) files. NIfTI image files have three main elements: the header, the affine and the data itself. The header (as with other files) contains the metadata; the affine translates between data-space and world-space; and the data is a multidimensional array (usually 3D with X, Y, Z and the 4D files also contain T – time).

The code first installed the required NiBabel and Nilearn then loads any further code dependencies. The first example data files' images were loaded using NiBabel's `nib.load`. Image object (not actual data) loaded first to find out preliminary information. The NIfTI header itself is not easily accessible without using Matlab or Python. Thus, to determine the dimensions of the image data, the shape was determined, and it can be seen the image is 3-dimensional as the voxel dimensions are output. The datatype was also checked. At this point, the type and array dimensions were known so the data itself was uploaded (example_data1) using the `get_fdata()` function and the shape checked to ensure the voxel dimensions match up to that displayed previously. Both the isolated brain MRI data and the whole head MRI data are plotted and displayed along all three axes using the Nilearn `plot_epi()` function. The plot produced from the NIfTI file that was cropped to solely the brain is displayed in Figure 1.

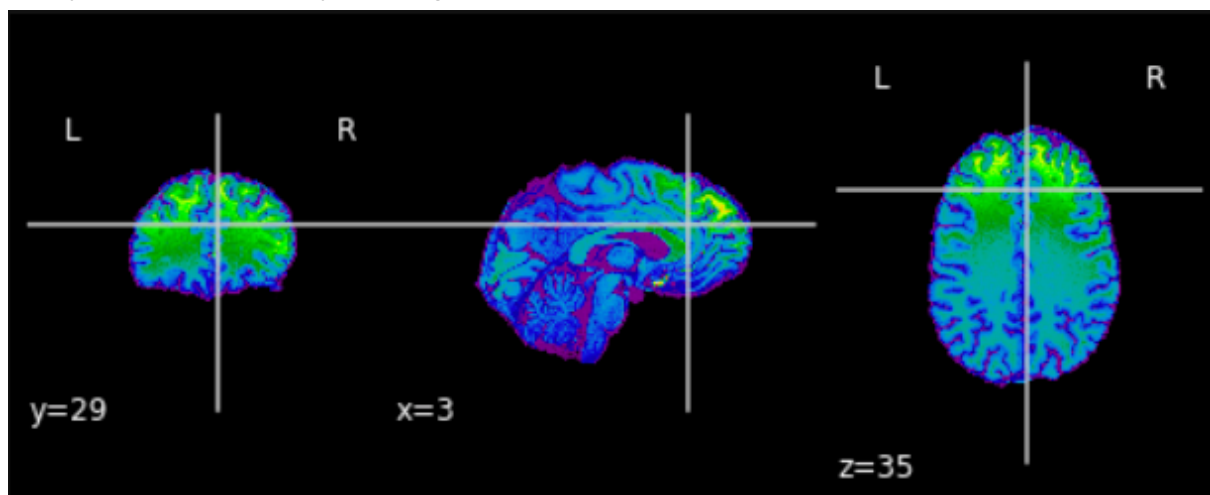


Figure 1 Three-dimensional human brain plotted using a NIfTI file and Nilearn's `plot_epi()` function.

Three-dimensional data can be harder to manipulate thus decreasing the dimensionality of the array to a new object enables greater flexibility for downstream data manipulation and analysis thus the original NIfTI data is reshaped (example_data1_2d). The new array's dimensionality is checked and confirmed as 2-dimensional.

Next the pre-made brain mask data is loaded and the shape checked (example_mask_data1). Here the voxel dimensions match up as the mask was created from the brain image itself. In order for masking to work and be applicable, the mask data must be Boolean thus the data type was converted and placed into a new object (example_mask_data_b). The Boolean mask had voxels with the value of either True or False was ready to be applied to the brain image data. Masking was done by simply indexing and this was confirmed by checking the output array dimensions. The array values were saved in a list for future used and plotted to view the array graphically.

The above code explored a single MRI image, the second part of the code used several MRI images downloaded directly using Nilearn's `datasets` function which automatically fetches the specified data. Information about where the dataset was located on my local drive was printed for convenience. An average brain image (haxby_mean) was compiled from the images in the dataset and was plotted as previous. The voxel dimensions checked showed that this MRI data has lower resolution than the example data used in the first part of the code. This can also be seen by comparing the plots produced from the previous data (Figure 1) and the average brain image plotted from the Haxby data (Figure 2).

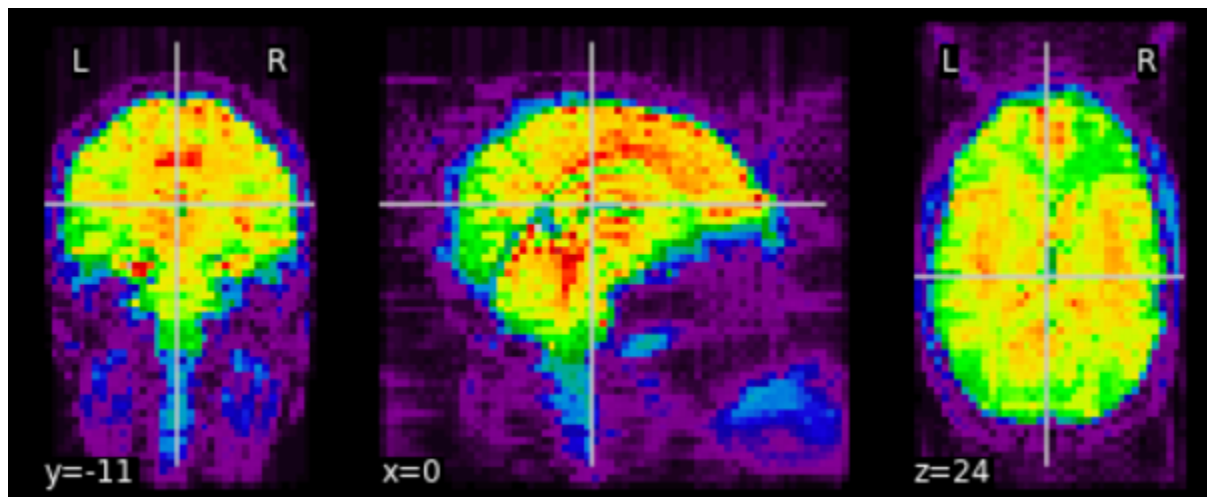


Figure 2 Three-dimensional average human brain plotted using from the Haxby data and Nilearn's `plot_epi()` function.

Nilearn can also generate whole brain masks from brain templates using the `compute_epi_mask()` function which essentially makes a 3D shape of the average brain (mask_haxby); this was viewed using the `plot_roi` function which displayed the mask overlapping the average (template) brain that was generated previously. The overlap shows the brain has been masked and successfully isolated as the region of interest (ROI) thus can be used for downstream analysis such as coregistration of all the MRI images with tools such as Advanced Normalization Tools (ANTs). Masking is an essential part of MRI neuroimaging analysis and can be manual and tedious, this automated mask generating function is beneficial as it decreases time and workload. The mask was applied to the image data using the `apply_mask` function and a 2D array was produced (masked_data) which contained the timepoints and the voxels. The first 500 timepoints from 5 voxels were plotted, the code can easily be adapted to change the number of timepoints and voxels plotted.

References:

1. Haxby, J., Gobbini, M., Furey, M., Ishai, A., Schouten, J., and Pietrini, P. (2001). Distributed and overlapping representations of faces and objects in ventral temporal cortex. *Science* 293, 2425-2430.
2. <https://github.com/ANTsX/ANTs>
3. <https://github.com/Brainhack-AMX/Brainhacking-101>
4. https://nilearn.github.io/modules/generated/nilearn.datasets.fetch_haxby.html
5. https://nipy.org/nibabel/nibabel_images.html
6. https://www.fmrib.ox.ac.uk/primers/intro_primer/ExBox14/IntroBox14.html