Large Channel-Image Data Visualizations

Focus: MRI Scans Visualizations

Exercises: nibabel, ByteIO, scipy.io python libraries

Images are the operational data types for certain application domains, like radiology, satellite imagery, genetics and many biomedical domains (genomic, imaging and clinical). It comes in several formats DICOM, NIfTI, TIFF, EDF, HDF5 , CSV, JSON and many more. Out of these 3D images are quite popular due to their large information capacity and detailed sequence capturing capability. Here are some of the application domains for the 3D (Multi Channel Images):

Domain	Best Tools
Clinical Imaging & Diagnostics	3D Slicer, ITK-SNAP, RadiAnt, OsiriX
Microscopy & Cell Biology	Imaris, Fiji (3D Viewer), Napari
Molecular & Protein Visualization	PyMOL, ChimeraX
Geospatial Health & Environment	QGIS 3D, ArcGIS Pro, CesiumJS, Kepler.gl
AI & Computational Modeling	MONAI Label, ParaView, VTK
Digital Twin / Simulation	CesiumJS, Unity3D, Unreal Engine

Most of these images have high channel and large memory size few MBs to several hundred MBs, so these files are kept in specific formats and have particular ways to be read and analysed.

Exercises

To start with, we will practice on some of the exercises from MRI scan dataset given in [1] and available at link [2] (Larger Version), this dataset is also available as a downloadable zip file, "MRI-Dataset.zip" in.

This dataset has three parts as, structural and functional directories and some additional files as :

-functional/snffM00640_*.{hdr,img} : This directory has scans to analyze brain's activity over time through BOLD signals (Blood-Oxygen-Level Dependent). These scans can be used for understanding functional connectivity, observing motor activity and visual cortex.

-structural/nsM00587_002.{hdr,img}} : this directory has scans about structural information of brain such as shape, size and physical properties. These files can be used detect tumors , analyzing bain development and measuring the gray/white matter volumes.

Both types of images are needed for the proper localization of activations /tissues.

-factors.mat : The file 'factors.mat' contains a labelling of the scans in terms of the

experimental 'factors', photic stimulation, motion and attention. They

are specified in terms on block onsets.

-block_regressors.mat: **this file has information about the timing of the tasks and when stimulus was on and off. The primary purpose of this file is to design the experiment to check the bain's activation.**

multi_condition.mat: It has experimental multiple condition for fMRI experiment , used for Statistical Parametric Mapping and format conversions. It defines the tasks condition and their timing information etc.

multi_block_regressors.mat: this file is used in block-design for fMRI experiments for multiple tasks such as motor activation, visual stimulations and audio .

README_DATA.txt : Information about the dataset.

Step-1: unzip the files from the folder " MRI-Dataset.zip " as follws:

```
from  zipfile  import  ZipFile  as  zf
with  zf('MRI-Dataset.zip','r')  as  fp:
   fp.extractall( )
```

As an output there a folder " attention " containing two foloders and five other files as explained above.

Step 2: Let's check the content of files and folders , extracted from the previous step

```
import os
print(os.listdir('attention/structural'))
print(os.listdir('attention/functional))
```

Output will have two types of files as header files(.hdr) and image files (.img) . Header files contains meta data about the images corresponding to each type 'structural' and 'functional'. Image files contains the image data and other information in context with other image files based on the experiment and observation style.

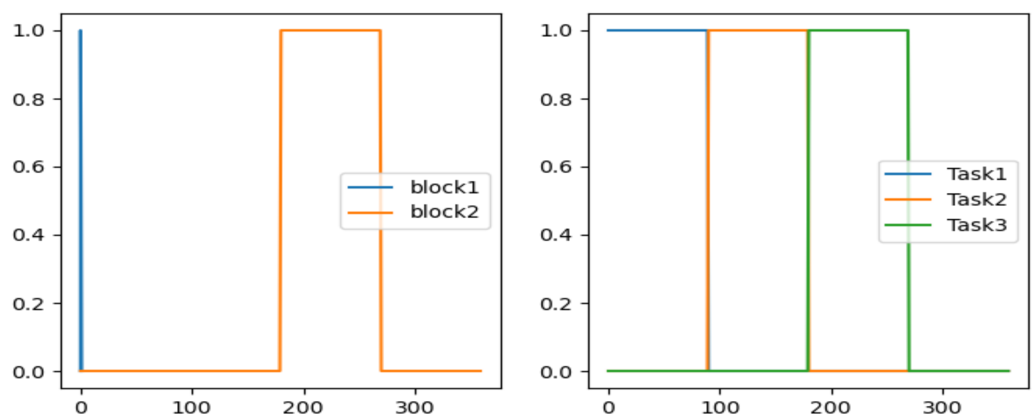
Step 3: Analyzing (.mat) files , 'multi_block_regressors.mat', 'block_regressors.mat', 'factors.mat

```
Import scipy.io as sio
mat = sio.loadmat('attention/block_regressor.mat')
mat2 = sio.loadmat('attention/multi_block_regressor.mat')
mat2.keys( )
mat2[ 'R' ]
```

Plotting these regressors:

```
import matplotlib.pyplot as plt
fig, ax =plt.subplots(1,2, figsize=(8,4))
ax[0].plot(mat['block1'],mat['block2'],mat['block3'])
ax[0].legend(['block1' , 'block2'])
ax[1].plot(mat2['R'])
ax[1].legend(['Task1' ,'Task2', 'Task3'])
plt.show()
fig.savefig('1.png')
```

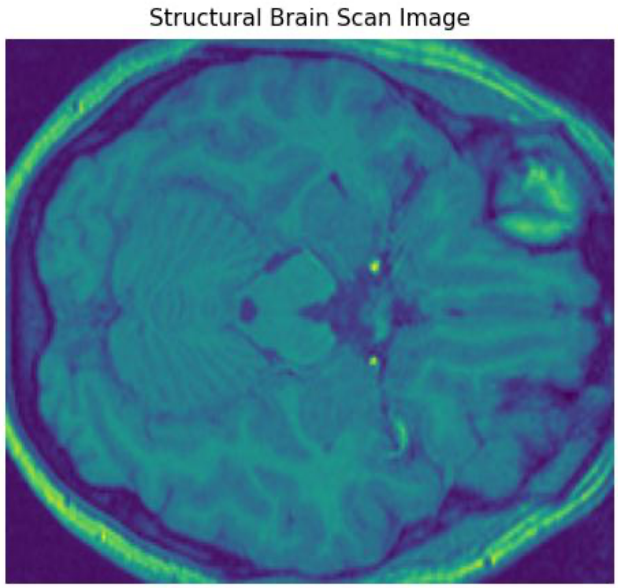
Output:



Step 4: Reading Analyzing and Visulaizing files from structural file folder (“ attention/functional ”) 'multi_block_regressors.mat', 'block_regressors.mat', 'factors.mat

```
import io

import nibabel as nib

from io import BytesIO

header1= 'attention/structural/nsM00587_0002.hdr'

image1 ='attention/structural/nsM00587_0002.img'

with open(header1, "rb") as fp:

        hdr_bytes=fp.read()

with open(image1, "rb") as fp1:

        img_bytes=fp1.read()


file_map = {

   'header': nib.FileHolder(fileobj=io.BytesIO(hdr_bytes)),

   'image':  nib.FileHolder(fileobj=io.BytesIO(img_bytes))

}


img = nib.analyze.AnalyzeImage.from_file_map(file_map)

data = img.get_fdata()

fig,ax = plt.subplots()

print (data.shape)

plt.imshow(data[:,:,15])

ax.axis('off')

fig.savefig('2.jpg')

fig.suptitle('Structural Brain Scan silce -15')
```

output:



Reference and useful links:

[1] Buchel, C and K. Friston (1997). Modulation of Connectivity in Visual Pathways by Attention: Cortical Interactions Evaluated with Structural Equation Modelling and fMRI, Cerebral Cortex, December, 7, pp 768-778.

[2] https://www.fil.ion.ucl.ac.uk/spm/data/attention/

[3] https://nipy.org/nibabel/api.html

[4] https://realpython.com/ref/stdlib/io/