The Pipeline of Processing fMRI data with Python Based on the Ecosystem NeuroDebian

Qiang Li[1,2], Rong Xue[1,2,3]*

[1]Sino-Danish College, University of Chinese Academy of Sciences, Beijing, 100049, China

[2]State Key Laboratory of Brain and Cognitive Science, Beijing MRI Center for Brain Research, Institute of Biophysics, Chinese Academy of Sciences, Beijing, 100101, China

[3]Beijing Institute for Brain Disorders, Beijing 100053, China

* Correspondence to: rxue@bcslab.ibp.ac.cn (Rong Xue）

doi:10.20944/preprints201904.0027.v1

Abstract

In the neuroscience research field, specific for medical imaging analysis, how to mining more latent medical information from big medical data is significant for us to find the solution of diseases. In this review, we focus on neuroimaging data that is functional Magnetic Resonance (fMRI) which non-invasive techniques, it already becomes popular tools in the clinical neuroscience and functional cognitive science research. After we get fMRI data, we actually have various software and computer programming that including open source and commercial, it's very hard to choose the best software to analyze data. What's worse, it would cause final result imbalance and unstable when we combine more than software together, so that's why we want to make a pipeline to analyze data. On the other hand, with the growing of machine learning, Python has already become one of very hot and popular computer programming. In addition, it is an open source and dynamic computer programming, the communities, libraries and contributors fast increase in the recent year. Through this review, we hope that can make neuroimaging data analysis more easy, stable and uniform base the one platform system.

Keywords: neuroscience; big data; functional Magnetic Resonance (fMRI); pipeline; one platform system

Introduction:

High-Field structural and functional MRI (Magnetic Resonance Imaging), this technique can non-invasively detect brain signal and has substantially high spatial resolution compared with EEG(Electroencephalography, MEG(magnetoencephalogram), etc. However, neuroimaging is a complex field that explores inter-disciplinary studies including physics, engineering, biological science, clinical medicine, physiology, statistics, and much more. The pure (f)MRI data only provide limited information and feedback about the brain, so data mining is a necessary and significant step for us to get more quantitative and broad functional information. When I started to analysis my first batch of (f)MRI data, there existed a lot of software and scripts on the Internet. Unfortunately, First, the source code was extremely messed. Second, the script was written by a variety of programming language, and it hardly connects all code together. Third, according to prior research, if you set up different parameters in the same software, it would affect the final results and make them unstable. Even worse, it's hard for us to combine all kind of software or scripts together, then batch processing (f)MRI data (Eklund et al. 2015; Pauli et al. 2016; Della-Maggiore et al. 2002). Fourth, the main computer programming is Matlab for the (f)MRI data analysis, but with the boost of machine learning and deep learning, Python is gradually beyond Matlab. In addition, Python is a totally open-source computer programming language, so compared to Matlab, it can accommodate the huge community contributes to the Python. What's more, Python has already formed a complete software and powerful community in the (f)MRI data analysis. Finally, compared to other computer programming languages, Python script is an interpretable and dynamic programming language, the source code is more simple and understandable (**Fig. 1**).

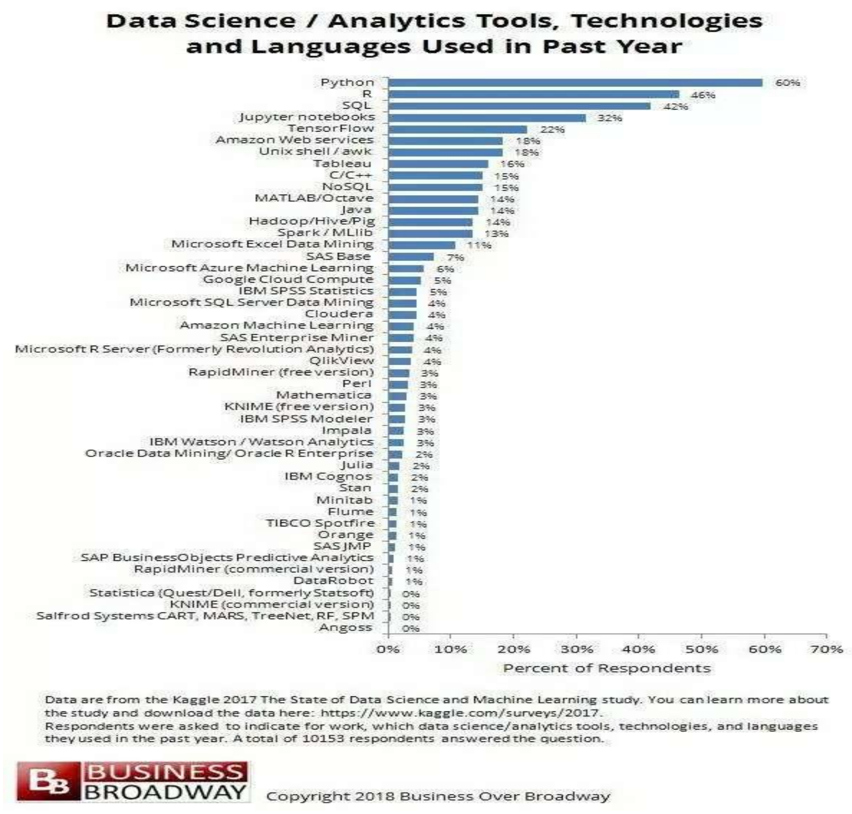


Fig. 1 Python compared with other main popular computer language used in data science. The main analytic computer languages for data analysis in the neuroscience and neuroimaging field also fit this trend. The image is adapted from *Business Broadway.*

With the advent of online neuroimaging databases, people can mine more features from these big data. This is also especially important for students at smaller colleges and universities without an MRI scanner. My goal is to make neuroimaging analysis understandable and accessible for anyone who is interested in neuroscience and neuroimaging. Above all, I aim to make fMRI analysis more precise and integrative.

1. The Ultimate Neuroscience Software Platform-Ecosystem

NeuroDebian (http://neuro.debian.net/) embrace a lot of popular neuroscience research software for the Debian operating system as well as Ubuntu and other derivatives. Popular packages include AFNI (https://afni.nimh.nih.gov/), FSL (https://fsl.fmrib.ox.ac.uk/fsl/fslwiki/), PyMVPA (http://www.pymvpa.org/), SPM (https://www.fil.ion.ucl.ac.uk/spm/) and many others (Halchenko and Hanke 2012).

1.1 Install NeuroDebian

1. Windows(64 bit), first choose a download server or imaging file, before you install it, you should install VirtualBox (more information in the VirtualBox download page https://www.virtualbox.org/wiki/Downloads). After installing VirtualBox, then finish configure it.
2. Mac OS X, the same way with Windows(64 bit).
3. Linux (Ubuntu, Debian etc), it has two conditions.

- **Only** install software by user chosen：

```
wget -O- http://neuro.debian.net/lists/bionic.cn-bj1.libre | sudo tee /etc/apt/sources.list.d/
neurodebian.sources.list

sudo apt-key adv --recv-keys --keyserver hkp://pool.sks -keyservers.net:80 0xA5D32F012
649A5A9

sudo apt-get update
```

- **All** software install once：

```
wget -O- http://neuro.debian.net/lists/bionic.cn-bj1.full | sudo tee /etc/apt/sources.list.d/n
eurodebian.sources.list

sudo apt-key adv --recv-keys --keyserver hkp://pool.sks-keyservers.net:80 0xA5D32F012
649A5A9

sudo apt-get update
```

Based on the NeuroDebian system, you can download software and data that you want by one simple command line. e.g.

sudo apt-get install ###pkgname###

1.2 Manage VirtualBox

HashiCorp Vagrant (https://www.vagrantup.com/) provides the easy, friendly workflow regardless of your role as a new user, designer or professional programmer. It use configuration file which describes all your program or software condition requirements or libraries, operating system configuration, dependency etc.

1. Install Vagrant. More details and info see here (https://www.vagrantup.com/docs/installation/).
2. Configure the system environment. The mechanism of Vagrant is very similar to Github. It can help you manage your system(e.g. VirtualBox).

2. The Data Management Tools—Data Portal

DataLad (https://www.datalad.org/) providing a data portal and a versioning system for data download or transmit, DataLad lets you can control and share your data more flexible.

As I have mentioned above, with the advent of online neuroimaging databases, some famous datasets are listed as follows:

- OpenfMRI - free and open sharing of raw magnetic resonance imaging (MRI) datasets. (The number of currently available datasets: **95**. The number of subjects across all datasets: **3372.** ***Until 5. Nov 2018***)
- Neurodata.io - the database of large-scale connectomics data.
- Core Nets - the substantial database of macaque tract-tracing data.
- Preprocessed Connectomes Project - preprocessed human fMRI data.
- NITRC - another portal that provides the ability to search across many different data sets and databases.

When we face these big data, the size of one subject scanning data can be larger than 1GB. Assume we want to transmit multi-subjects scanning data for remote services or share with other universities or people, that will be a big challenge for researchers. DataLad can fix these problems. You can download, upload, consume these reproducibility data with other people.

2.1 Install DataLad

DataLad can easily be installed via pip.

pip install --user datalad

If your developing system is Linux(e.g. Ubuntu, Debian etc.)

sudo apt-get install datalad

More information about how to install DataLad, click here (https://www.datalad.org/get_datalad.html).

3. Basic Unix Command Line

It is a necessary step for us to study image analysis. You need to know some basic and frequently-used command lines in the Unix system. e.g. ***ls, cd, rm, grep etc.*** More tutorial about Unix commands line link here (http://fsl.fmrib.ox.ac.uk/fslcourse/unix_intro/).

4. Version Control System

Git (https://git-scm.com/) is a free, powerful and open source distributed version control system designed to operate and cooperate projects from small to very large with speed and efficiency.

First, you can choose Git version based on your system. More information on how to install Git link here (https://git-scm.com/downloads). Here I will show you a simple case of how to use Git to share your scripts or control your version system.

The first step: build an account on the Github.com and configure your environment. More details link here (https://git-scm.com/doc). Then build a new repository and initial it. You can clone or download this repository into the local directory.

git clone ####The Link of Repository####

Now, you can modify, add, change and remove files or folders in your local directory. What has been changed in your repository, you can check it.

```
#Check current condition
git status
```

Before you submit to your remote repository, you need to add comments for what kind of information is updated. Then you can push your local repository into remote services.

```
git commit -m "add info for what has been changed"
git push -u origin master

%For sectuity, you need login your name and password%
%Then you can go back your Github to check your updated repository%
```

5. Docker

Docker (https://www.docker.com/) is a lightweight container software, it provides container software that is ideal and easy for users who are looking to occupy small computer memory and handle data with container-based applications, It has totally different principle with Virtual machine, because docker share the computer hardware resources and operating system with host machine, it can easy to make dynamic distribution of resources.

The docker can make each software or app or imaging run separately in the container. You also can build you imaging according to your needs in your work. More basic tutorials about Docker link here (https://neurohackweek.github.io/docker-for-scientists/). Docker is also very popularly used in the neuroimaging analysis recently.

6. Brain Imaging Data Structure(BIDS)

BIDS (http://bids.neuroimaging.io/), a format for organizing and describing neuroscience data structure of neuroimaging experiments (Gorgolewski et al. 2016).

One of the big problems for (f)MRI data analysis is data organization in the directory. In the fMRI experiment, it consists of BOLD data, anatomy data, behavioral data (**Fig.2**). The researchers can use BIDS app to make their data in a more standard format and order. The benefit can be listed as follows:

1. It will be easier for sharing your data.

2. It can make your data more integrity and order.

3. People will read and understand easier.

4. Many data analysis software all support BIDS formats.

PyBIDS is one of BIDS library that can centralize interactions with datasets. For more information about BIDS visit here (http://bids.neuroimaging.io). More command lines with Python link here (https://bids-standard.github.io/pybids/).

Here is an example for explaining how to use PyBIDS to reorganize (f)MRI data. More detailed information and tutorials you can check here (https://github.com/bids-standard/pybids/blob/master/examples/pybids%20tutorial.ipynb).

```
import bids.layout
import bids.tests
import os
```

The last data organization after BIDS is shown as follows:

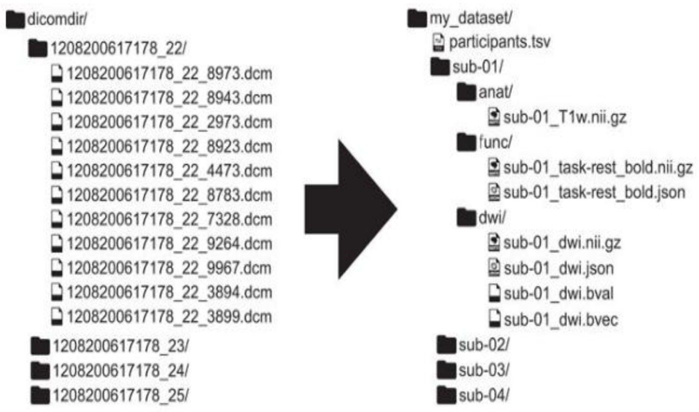

dicomdir/
1208200617178_22/
1208200617178_22_8973.dcm
1208200617178_22_8943.dcm
1208200617178_22_2973.dcm
1208200617178_22_8923.dcm
1208200617178_22_4473.dcm
1208200617178_22_8783.dcm
1208200617178_22_7328.dcm
1208200617178_22_9264.dcm
1208200617178_22_9967.dcm
1208200617178_22_3894.dcm
1208200617178_22_3899.dcm
1208200617178_23/
1208200617178_24/
1208200617178_25/
my_dataset/
participants.tsv
sub-01/
anat/
sub-01_T1w.nii.gz
func/
sub-01_task-rest_bold.nii.gz
sub-01_task-rest_bold.json
dwi/
sub-01_dwi.nii.gz
sub-01_dwi.json
sub-01_dwi.bval
sub-01_dwi.bvec
sub-02/
sub-03/
sub-04/

Fig.2 The BIDS processing result of (f)MRI data. The image is adapted from http://bids.neuroimaging.io/.

7. The Quality Control of (f)MRI data

This step is also very important before you start processing your raw data, In the Python, mriqc is a very powerful tool for checking your data quality and it also can provide a user-friendly and nice report for your data. More example for mriqc, you can check here (https://mriqc.readthedocs.io/en/stable/).

8. The Tools of Preprocessing of (f)MRI data Based on The NeuroDebian

In this section, we highly recommend two tools for preprocessing raw (f)MRI data with Python.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 2 April 2019 doi:10.20944/preprints201904.0027.v1

8.1 fmriprep: A Robust Preprocessing Pipeline for fMRI Data

Fmriprep (https://fmriprep.readthedocs.io/en/stable/) is a functional magnetic resonance imaging (fMRI) data preprocessing pipeline based Python, it provides very easy, friendly interface for users, it also allowed uses to input minimum parameters to drive the total pipeline works, but it providing more details output reporting (Esteban et al. 2018). Compared to SPM (Statistical Parametric Mapping, https://www.fil.ion.ucl.ac.uk/spm/) software, it would be more easy interface the next analysis with machine learning and other approaches.

The document of fmriprep link here (https://fmriprep.readthedocs.io/en/stable/). The total fmriprep workflow can be shown as follows (**Fig. 3**):

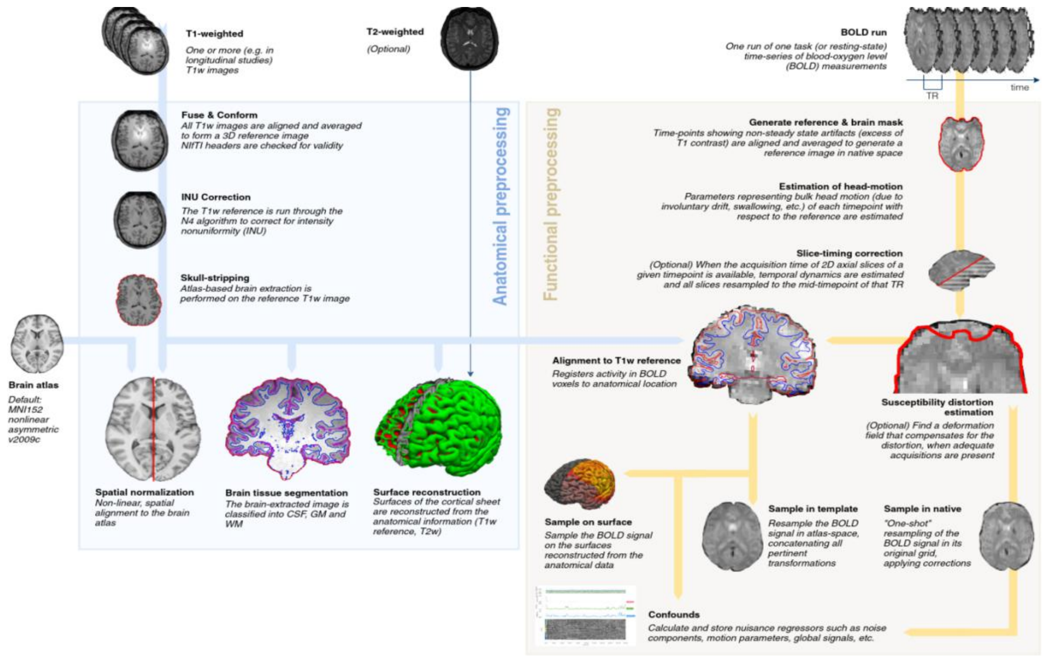

T1-weighted
One or more (e.g. in longitudinal studies) T1w images
T2-weighted
(Optional)
BOLD run
One run of one task (or resting-state) time-series of blood-oxygen level (BOLD) measurements
TR
time
Anatomical preprocessing
Functional preprocessing
Fuse & Conform
All T1w images are aligned and averaged to form a 3D reference image
NIfTI headers are checked for validity
INU Correction
The T1w reference is run through the N4 algorithm to correct for intensity nonuniformity (INU)
Skull-stripping
Atlas-based brain extraction is performed on the reference T1w image
Generate reference & brain mask
Time-points showing non-steady state artifacts (excess of T1 contrast) are aligned and averaged to generate a reference image in native space
Estimation of head-motion
Parameters representing bulk head motion (due to involuntary drift, swallowing, etc.) of each timepoint with respect to the reference are estimated
Slice-timing correction
(Optional) When the acquisition time of 2D axial slices of a given timepoint is available, temporal dynamics are estimated and all slices resampled to the mid-timepoint of that TR
Brain atlas
Default: MNI152 nonlinear asymmetric v2009c
Alignment to T1w reference
Registers activity in BOLD voxels to anatomical location
Susceptibility distortion estimation
(Optional) Find a deformation field that compensates for the distortion, when adequate acquisitions are present
Spatial normalization
Non-linear, spatial alignment to the brain atlas
Brain tissue segmentation
The brain-extracted image is classified into CSF, GM and WM
Surface reconstruction
Surfaces of the cortical sheet are reconstructed from the anatomical information (T1w reference, T2w)
Sample on surface
Sample the BOLD signal on the surfaces reconstructed from the anatomical data
Sample in template
Resample the BOLD signal in atlas-space, concatenating all pertinent transformations
Sample in native
"One-shot" resampling of the BOLD signal in its original grid, applying corrections
Confounds
Calculate and store nuisance regressors such as noise components, motion parameters, global signals, etc.

Fig.3 The fmriprep workflow chart. In the left side, it is used for preprocessing anatomic MRI data, e.g. head motion correction, tissue segmentation, surface reconstruction etc. In the right side, it is used for preprocessing functional MRI data, e.g. slicing correction, smooth and statistics etc.

For more tutorials on how to practice it, you can link here (https://github.com/poldracklab/fmriprep-notebooks). Here we supply a very nice example to show how it works with Python. The image is adapted from https://fmriprep.readthedocs.io/en/stable/.

8.2 Nipype: Neuroimaging in Python Pipelines and Interfaces

Nipype (https://nipype.readthedocs.io/en/latest/), an open source, community contribute based from NiPy, is a Python project that provides a uniform interface to existing neuroimaging software and facilitates interaction between these packages within a single workflow. (Gorgolewski et al. 2011)

The mechanism of Nipype is very similar to fmripype. It also combined different packages together (e.g. ANTS, FreeSurfer etc) (**Fig. 4**). More document of Nipype can be found here (https://nipype.readthedocs.io/en/latest/documentation.html).

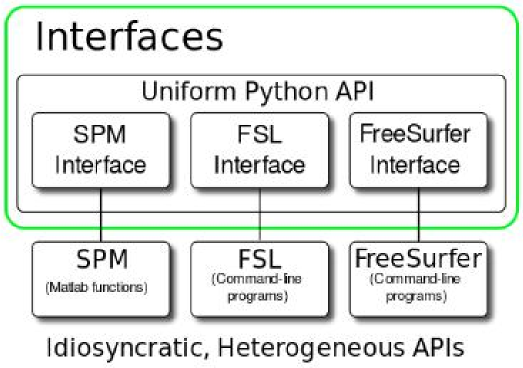


Fig.4 The framework of Nipype. It clusters SPM, FSL, FreeSurfer etc software togehter and supplies user-friendly interface with Python. The image is adapted from https://nipype.readthedocs.io/en/latest/index.html.

9. Machine learning

After preprocessing (f)MRI data, we can do more exploration for these data and mining more medical information. Machine learning is one of the perfect examples to explain it. Here I will introduce some libraries with Python to do machine learning for neuroimaging.

9.1 Nilearn

Nilearn (https://nilearn.github.io/) is a Python module for fast and easy statistical learning on NeuroImaging data. It based on the scikit-learn (https://scikit-learn.org/stable/index.html) toolbox (popular machine learning package with Python) for multivariate statistics with applications such as classification, decoding, or connectivity analysis etc, it still growing and adding more function by large community contributors, in the future, it will have more function and property into it.

Among the best libraries of machine learning for neuroimaging with Python, I think Nilearn is the best choice for us. In the next part, I will show you some simple examples to help you quickly get into Nilearn. Nilearn also supplies interface that helps you to automatically download data from OpenfMRI and NeuroVault (a new home for all brain statistical maps).

```
#Import module from nilearn
from nilearn import datasets

#Download the data from nilearn
motor_images = datasets.fetch_neurovault_motor_task()
stat_img = motor_images.images[0]

#Plotting the statistics images
from nilearn import plotting
plotting.plot_glass_brain(stat_img, threshold=3)

plotting.show()
```

The plotting result is shown as follows (**Fig. 5**). More examples with Nilearn, check here (http://nilearn.github.io/auto_examples/index.html).

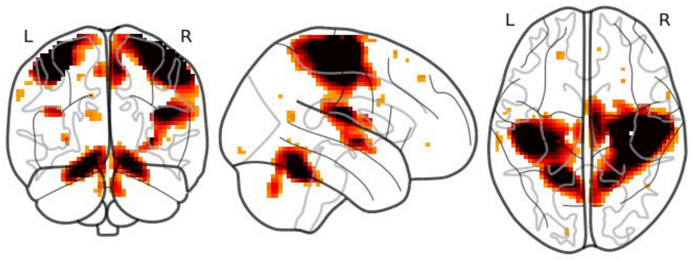
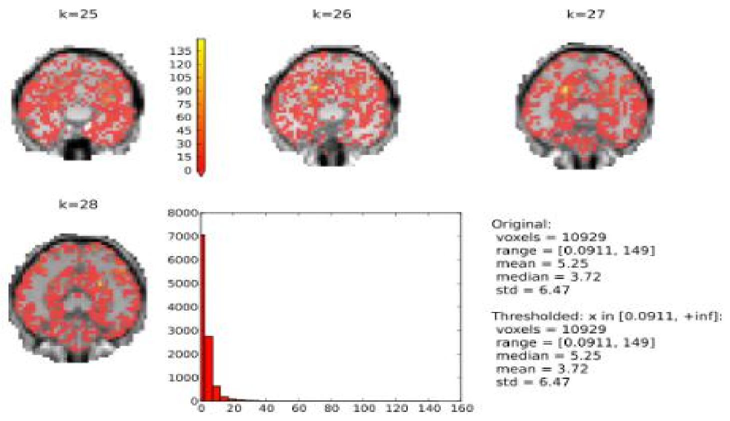


Fig.5 The statistical map of motor task is plotted with glass brain format.

9.2 Pymvpa - MultiVariate Pattern Analysis (MVPA) in Python

PyMVPA is a Python package intended to more statistical learning analyses of large datasets. It provides more enrich high-level interface to a broad range of algorithms for classification, regression, feature selection, result plot. It is designed to integrate well with related software packages, such as scikit-learn, shogun (http://www.shogun-toolbox.org/), etc. PyMVPA is free software and requires nothing but free-software to run (Hanke et al. 2009). The next code from PyMVPA official website.

The result image looks like as follows (**Fig. 6**). More source code link here (http://www.pymvpa.org/examples/mri_plot.html):



```
Original:
 voxels = 10929
 range = [0.0911, 149]
 mean = 5.25
 median = 3.72
 std = 6.47

Thresholded: x in [0.0911, +inf]:
 voxels = 10929
 range = [0.0911, 149]
 median = 5.25
 mean = 3.72
 std = 6.47
```

Fig.6 The basic f(MRI) plotting with Pymvpa.

The more examples, courses, tutorials, and manuals together link here (http://www.pymvpa.org/docoverview.html).

10. Deep learning

As mentioned before, with the development of supercomputer and machine learning, more and more data is produced from society. In the neuroscience research field, the traditional data analysis approaches face a big challenge and bottleneck when the big data fade into it, so that's why deep learning make very popular in the neuroscience research area that specific for fMRI data analysis, through deep learning model that we can train enough data to dig more medical information or cognitive mechanism. When we back to see the published paper in the recent year, the research topic related to deep learning already become a hot topic. So what kind of tools that we can use to neuroscience data analysis, in the next section, I will introduce some famous, classical and friendly model or library that we can use in our research.

10.1 Tensorflow - Very popular open source deep learning model framework

Tensorflow (https://www.tensorflow.org/) is an open source software library for numerical computation using data flow graphs. It provides stable Python API and C APIs as well as without API backwards compatibility guarantee like C++, Java, JavaScript and Swift.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 2 April 2019 doi:10.20944/preprints201904.0027.v1

Tensorflow is fully supporting CPU and GPU, it also supports friendly interface with Python, you can through easy install or drive CPU and GPU works (Pattanayak 2017; Gad 2018; Pattanayak 2017).

The core idea of tensorflow is based graphy drive model, more abstract and details that can be described as follows (**Fig. 7**):

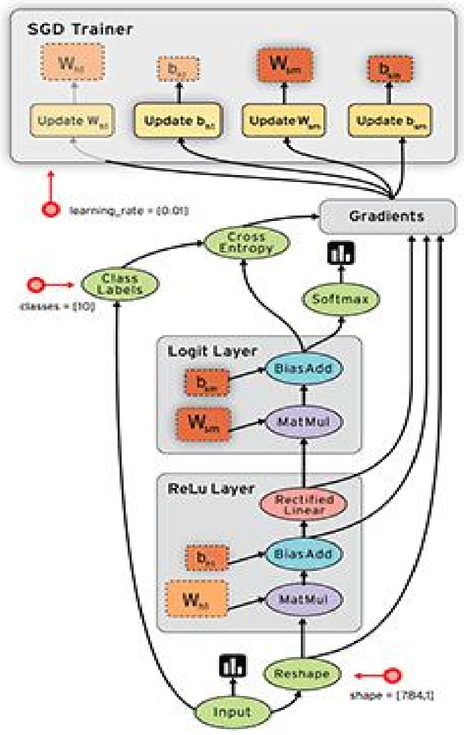


Fig.7 Tensorflow data flow chart. The graph nodes represent mathematical operations, while the graph edges represent the multidimensional data arrays (tensors) that flow between them. The image is adapted from https://www.tensorflow.org/guide/graphs.

More example or tutorials that link https://www.tensorflow.org/, you can find more API s or samples about tensorflow.

10.2 Others- The Deep Learning library

10.2.1 Keras

Keras (https://keras.io/) is a high-level neural networks API, written in Python and capable of running on top of TensorFlow, CNTK(Microsoft Inc. https://docs.microsoft.com/zh-cn/cognitive-toolkit/), or Theano (http://www.deeplearning.net/software/theano/). Keras is also a very popular deep learning framework, it offers a very easy interface with Python. More example and tutorials link https://keras.io/.

10.2.2 PyTorch

PyTorch (https://pytorch.org/), an open source deep learning platform that provides a seamless path from research prototyping to production deployment (Ketkar 2017; Chen et al. 2018).

PyTorch is an end-to-end deep learning framework, the user of PyTorch is already over tensorflow and keras in some period time, it also has complete and perfect documents and tutorials for getting started PyTorch from zero. More examples link https://pytorch.org/tutorials/.

10.2.3 Brain.js

brain.js (https://brain.js.org/) is a library of neural networks written in JavaScript. Front developing is the main and popular software method for display some context, actually, more programmer love JavaScript and use it to training some neural networks, what's more, JaveScript is also the very popular computer programming compared others. So the brain.js that is a deep learning library that based with JavaScript (Wang, Cai, and Wei 2016; van der Spuy 2012). In order to you can have a direct impression for brain.js, more examples link https://github.com/BrainJS/brain.js/tree/master/examples.

In this section, I only list some famous and frequently used deep learn frameworks, actually, more deep learning framework is developed according to each demand in various areas today, but for neuroscience data analysis, you can use above frameworks is already can solve your problem, of course, you also can contribute to these deep learning frameworks through Github, all these frameworks are open sources, welcome to contribute your idea and make it more perfect.

Summary

This review that we generalize all methods and software from the ultimate system platform to a high-level deep learning framework, we hope that this review that can give some new student or researchers can easily walk into neuroscience data analysis. In general, the basic approaches or methods in the neuroscience data analysis(specific for neuroimaging data, such as fMRI etc.) are all cover on this paper.

All software used in this review or more software in the neuroscience data analysis that we did a website that you can find by linking https://sinodanish.github.io/brainsoftware/.

Acknowledgments

This work was supported in part by the Ministry of Science and Technology of China grant (2015CB351701), the National Natural Science Foundation of China grant (31730039, 81871350), Beijing Natural Science Foundation grant（L182055） and the Chinese Academy of Sciences grant (XDBS01000000).

References

Eklund, Anders, Thomas Nichols, Mats Andersson, and Hans Knutsson. 2015. "Empirically Investigating the Statistical Validity of SPM FSL and AFNI for Single Subject FMRI Analysis". In *2015 IEEE 12th International Symposium on Biomedical Imaging (ISBI)*. IEEE. doi:10.1109/isbi.2015.7164132.

Pauli, Ruth, Alexander Bowring, Richard Reynolds, Gang Chen, Thomas E. Nichols, and Camille Maumet. 2016. "Exploring FMRI Results Space: 31 Variants of an FMRI Analysis in AFNI FSL and SPM". *Frontiers in Neuroinformatics* 10 (July). Frontiers Media SA. doi:10.3389/fninf.2016.00024.

Della-Maggiore, Valeria, Wilkin Chau, Pedro R. Peres-Neto, and Anthony R. McIntosh. 2002. "An Empirical Comparison of SPM Preprocessing Parameters to the Analysis of FMRI Data". *NeuroImage* 17 (1). Elsevier BV: 19–28. doi:10.1006/nimg.2002.1113.

Halchenko, Yaroslav O., and Michael Hanke. 2012. "Open Is Not Enough. Lets Take the Next Step: An Integrated Community-Driven Computing Platform for Neuroscience". *Frontiers in Neuroinformatics* 6. Frontiers Media SA. doi:10.3389/fninf.2012.00022.

Gorgolewski, Krzysztof J., Tibor Auer, Vince D. Calhoun, R. Cameron Craddock, Samir Das, Eugene P. Duff, Guillaume Flandin, et al. 2016. "The Brain Imaging Data Structure a Format for Organizing and Describing Outputs of Neuroimaging Experiments". *Scientific Data* 3 (June). Springer Nature: 160044. doi:10.1038/sdata.2016.44.

Esteban, Oscar, Christopher Markiewicz, Ross W Blair, Craig Moodie, Ayse Ilkay Isik, Asier Erramuzpe Aliaga, James Kent, et al. 2018. "FMRIPrep: a Robust Preprocessing Pipeline for Functional MRI", April. Cold Spring Harbor Laboratory. doi:10.1101/306951.

Gorgolewski, Krzysztof, Christopher D. Burns, Cindee Madison, Dav Clark, Yaroslav O. Halchenko, Michael L. Waskom, and Satrajit S. Ghosh. 2011. "Nipype: A Flexible Lightweight and Extensible Neuroimaging Data Processing Framework in Python". *Frontiers in Neuroinformatics* 5. Frontiers Media SA. doi:10.3389/fninf.2011.00013.

Hanke, Michael, Yaroslav O. Halchenko, Per B. Sederberg, Stephen José Hanson, James V. Haxby, and Stefan Pollmann. 2009. "PyMVPA: a Python Toolbox for Multivariate Pattern Analysis of FMRI Data". *Neuroinformatics* 7 (1). Springer Nature: 37–53. doi:10.1007/s12021-008-9041-y.

Pattanayak, Santanu. 2017. "Introduction to Deep-Learning Concepts and TensorFlow". In *Pro Deep Learning with TensorFlow*, 89–152. Apress. doi:10.1007/978-1-4842-3096-1_2.

Gad, Ahmed Fawzy. 2018. "TensorFlow Recognition Application". In *Practical Computer Vision Applications Using Deep Learning with CNNs*, 229–94. Apress. doi:10.1007/978-1-4842-4167-7_6.

Pattanayak, Santanu. 2017. *Pro Deep Learning with TensorFlow*. Apress. doi:10.1007/978-1-4842-3096-1.

Ketkar, Nikhil. 2017. “Introduction to PyTorch”. In *Deep Learning with Python*, 195–208. Apress. doi:10.1007/978-1-4842-2766-4_12.

Chen, Kathleen M, Evan M Cofer, Jian Zhou, and Olga G Troyanskaya. 2018. “Selene: a PyTorch-Based Deep Learning Library for Biological Sequence-Level Data”, October. Cold Spring Harbor Laboratory. doi:10.1101/438291.

Wang, Yao, Wan-dong Cai, and Peng-cheng Wei. 2016. “A Deep Learning Approach for Detecting Malicious JavaScript Code”. *Security and Communication Networks* 9 (11). Wiley: 1520–34. doi:10.1002/sec.1441.

van der Spuy, Rex. 2012. “Learning JavaScript”. In *Foundation Game Design with HTML5 and JavaScript*, 59–110. Apress. doi:10.1007/978-1-4302-4717-3_2.