Computing Methods for Experimental Physics and Data Analysis

Data Analysis in Medical Physics

Lecture 12: Reproducibility of results; uniform interfaces to existing software packages;

... and course sum up

Alessandra Retico
alessandra.retico@pi.infn.it
INFN - Pisa

Reproducibility studies

- Different analysis pipelines with the same purpose may rely both on different principles and on different algorithm implementations
- Extensive tests should be performed to assess the **reliability** of the features/measures they produce, before the output of SW packages for the analysis of medical data is used to infer results in the field of clinical research
- To estimate the precision of a SW package, a reference "gold standard" is necessary
- In most cases "gold standard" measures on large samples are not available, nevertheless, we can evaluate:
  - the robustness of a SW pipeline (**intra-method agreement**)
  - the reproducibility of the same measure across different SW pipelines (**inter-method agreement**)

- **The lack of intra-method and inter-method agreement can produce inconsistent results in medical imaging studies**

Reproducibility studies

- To test SW reliability in terms of both intra-method and inter-method agreement we can:
  - collect datasets containing test and retest images (i.e. images are acquired twice for each subject)
  - choose different SW packages developed to extract the same image measures
  - estimate the variability of measures on test-retest images (**intra-method agreement**)
  - estimate the variability of measures provided by different SW packages on the same images (**inter-method agreement**)

SPM

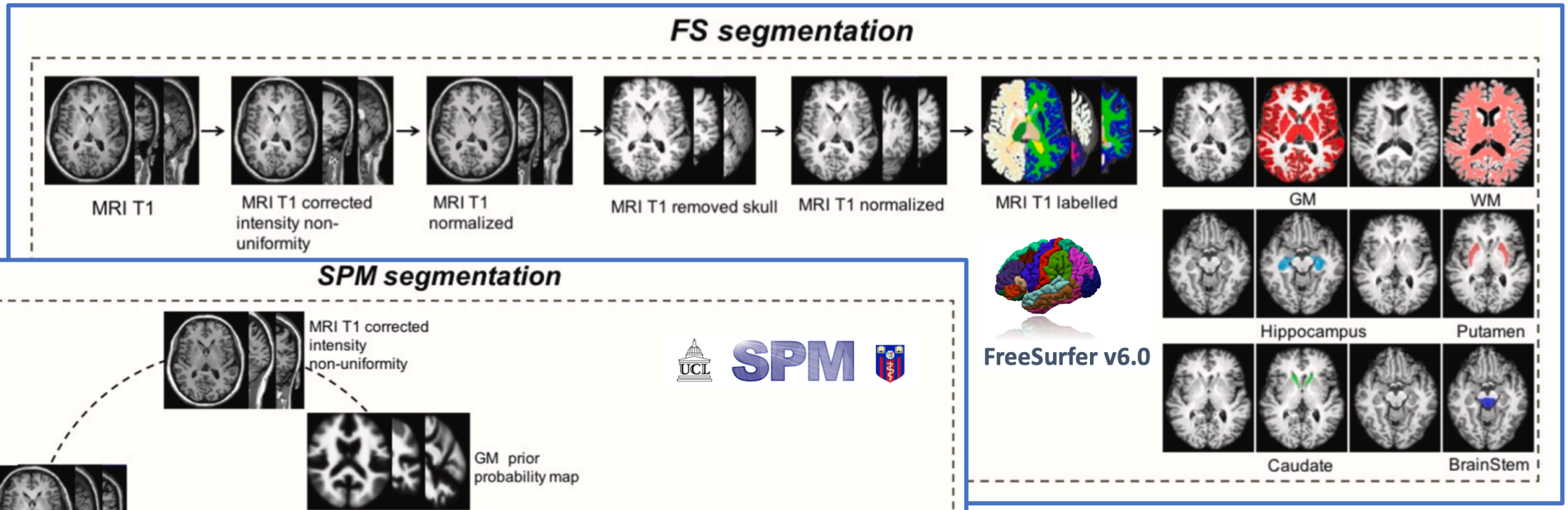
www.fil.ion.ucl.ac.uk/spm/

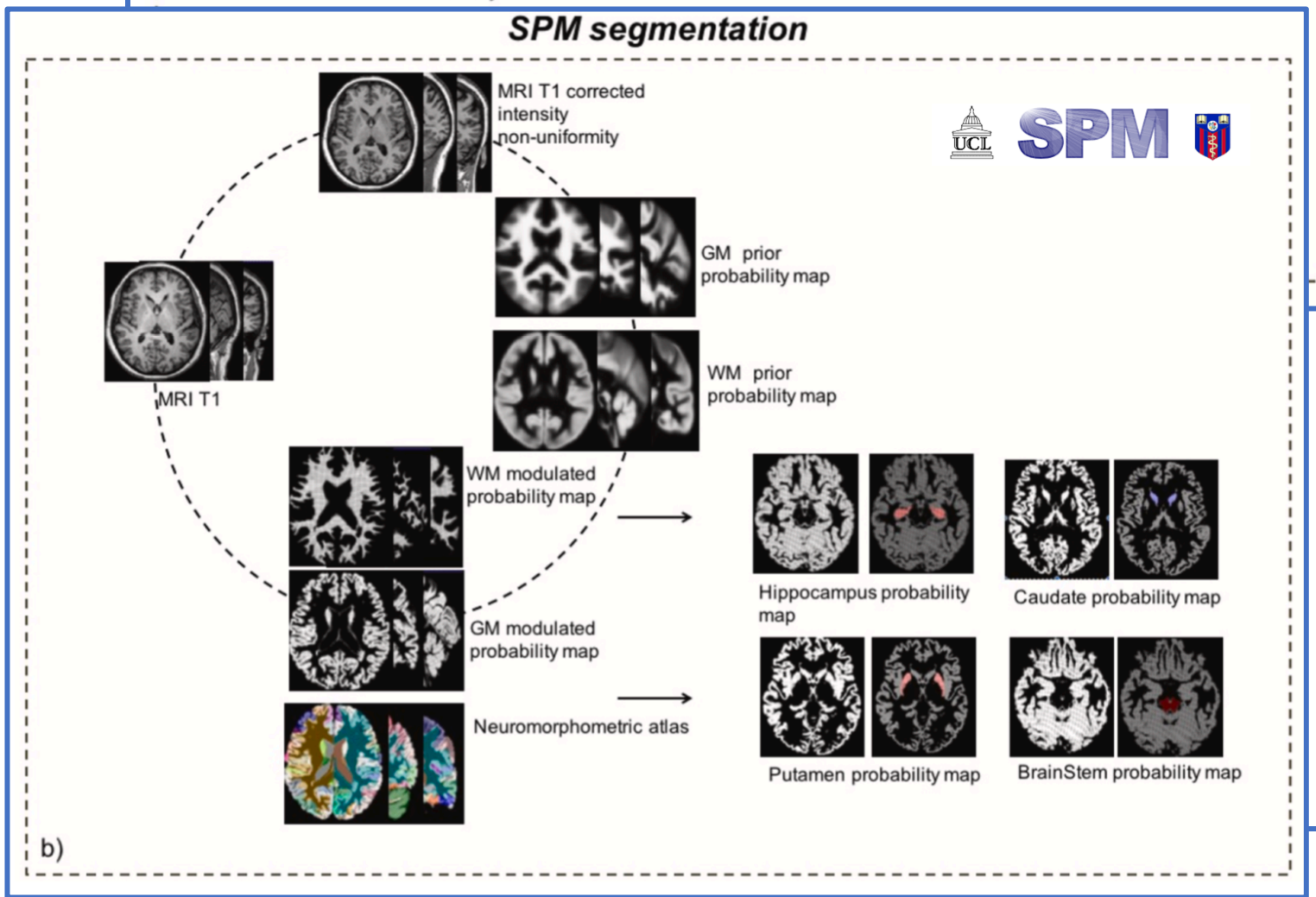
Evaluation of the intra- and inter-method agreement of brain MRI segmentation software packages: A comparison between SPM12 and FreeSurfer v6.0,
Palumbo L, Bosco P, Fantacci ME, Ferrari E, Oliva P, Spera G, Retico A,
Phys Med. 2019 Aug;64:261-272.

FreeSurfer v6.0
http://freesurfer.net/

DATASETS of 3D T1w brain structural MRI images:

- 21 healthy subjects of the Kirby-21 collection, https://doi.org/10.1016/j.neuroimage.2010.11.047
  - 3T MRI, scan and rescan in the same day
- 20 healthy subjects of the OASIS collection, https://www.nitrc.org/projects/oasis/
  - 1.5T MRI, scan and rescan with a mean delay of 21 days





Tasks implemented in both SPM and Freesurfer:

- Segmentation of GM, WM and sub-cortical brain structures
- Quantification of volumes of segmented regions

Reproducibility studies: **intra-method agreement**

Intra-method repeatability

- Bland-Altman plot detects possible systematic biases between test and retest scans.
  - The percentage differences between test and retest measures of volumes
    ($V_{test} - V_{retest}$)
    for each subject is reported as a function of their average
  - The 95% confidence interval limits on agreement are shown

- For Freesurfer, inconsistent results are obtained:
  - $V_{test} > V_{retest}$ for GM
    (5% decrease)
  - $V_{test} < V_{retest}$ for WM
    (3% increase!!!)

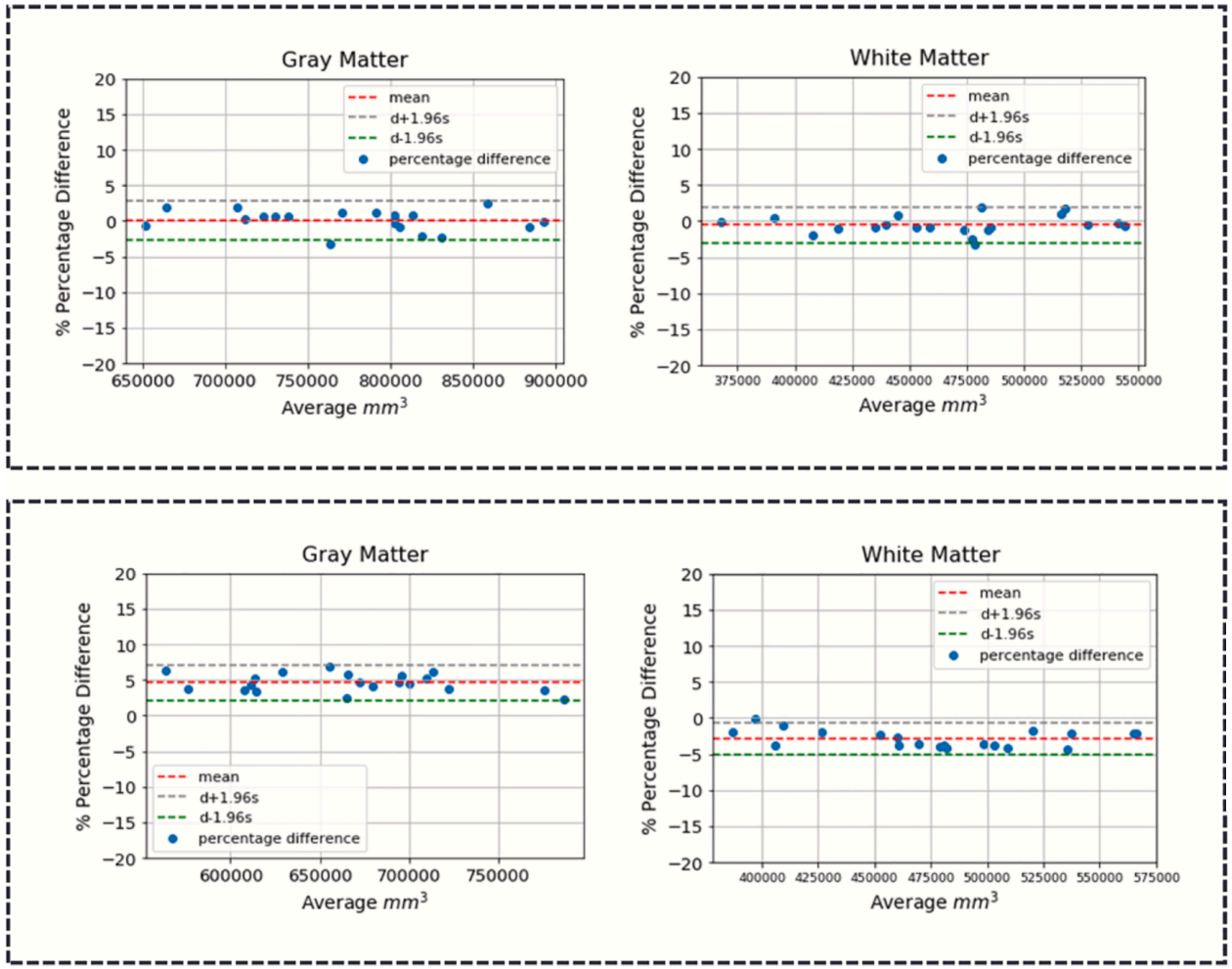


Fig. 4. Bland-Altman plots of segmented GM and WM by SPM (above) and by FS (below) in scan-rescan analysis on the OASIS dataset. Bland-Altman plots of the four segmented subcortical structures are shown in the Supplementary Material.

Reproducibility studies: **inter-method agreement**

Inter-method reproducibility

- SPM vs. Freesurfer are compared in terms of:
  - volume assessment (Pearson's correlation; Bland-Altman's plot parameters)
  - segmentation mask overlap (Dice indices)

- Results:
  - SPM and Freesurfer estimated volumes are highly correlated, except for hippocampus
  - SPM underestimates all volumes compared to Freesurfer, except for GM
  - The Dice index attests a systematic relationship between the SPM and Freesurfer segmentations: they are segmenting the same tissues/regions in different, but related ways.

$V_{SPM} - V_{Freesurfer}$

Kirby dataset

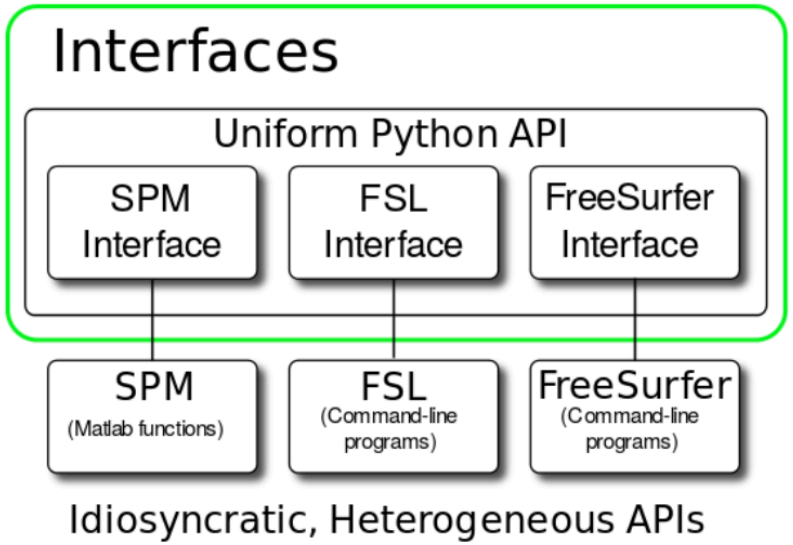
Brain region	Pearson's correlation	Percent difference ± standard deviation	D
GM	0.94	11 ± 3	0.80 ± 0.03
WM	0.98	−13 ± 2	0.81 ± 0.03
Hippocampus	0.72	−21 ± 6	0.78 ± 0.03
Putamen	0.86	−34 ± 5	0.76 ± 0.04
Caudate	0.80	−23 ± 8	0.80 ± 0.02
Brainstem	0.95	−26 ± 5	0.84 ± 0.02

OASIS dataset

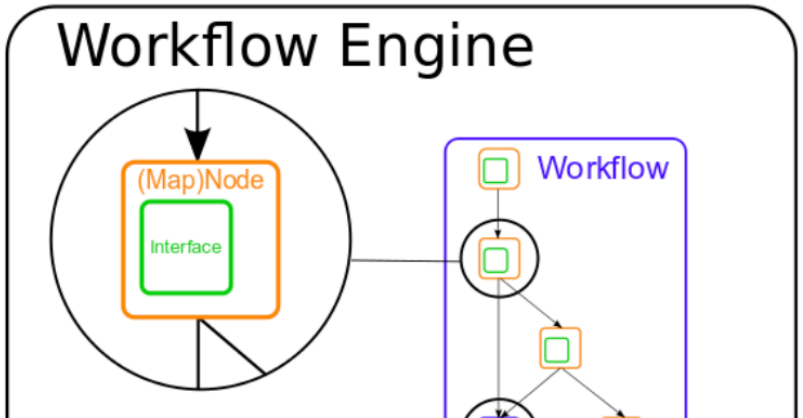
Brain region	Pearson's correlation	Percent difference ± standard deviation	D
GM	0.96	13 ± 3	0.83 ± 0.01
WM	0.95	−11 ± 3	0.83 ± 0.02
Hippocampus	0.78	−17 ± 5	0.78 ± 0.02
Putamen	0.84	−34 ± 6	0.77 ± 0.04
Caudate	0.93	−23 ± 6	0.79 ± 0.02
Brainstem	0.95	−34 ± 3	0.80 ± 0.02

https://nipype.readthedocs.io/en/latest/

Nipype: Neuroimaging in Python Pipelines and Interfaces



Idiosyncratic, Heterogeneous APIs



Current neuroimaging software offer users an incredible opportunity to analyze data using a variety of different algorithms. However, this has resulted in a heterogeneous collection of specialized applications without transparent interoperability or a uniform operating interface.

Nipype, an open-source, community-developed initiative under the umbrella of NiPy, is a Python project that provides a uniform interface to existing neuroimaging software and facilitates interaction between these packages within a single workflow. Nipype provides an environment that encourages interactive exploration of algorithms from different packages (e.g., ANTS, SPM, FSL, FreeSurfer, Camino, MRtrix, MNE, AFNI, Slicer, DIPY), eases the design of workflows within and between packages, and reduces the learning curve necessary to use different packages. Nipype is creating a collaborative platform for neuroimaging software development in a high-level language and addressing limitations of existing pipeline systems.

Nipype allows you to:

- easily interact with tools from different software

Google

Versions

Release	Devel
1.3.0-rc1	1.4.0-dev+gb5cb2aa7c
Download	Github

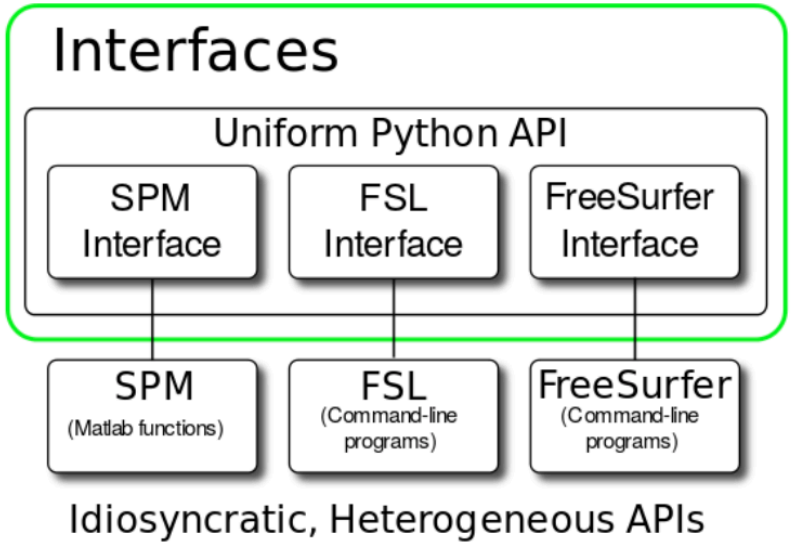
Links

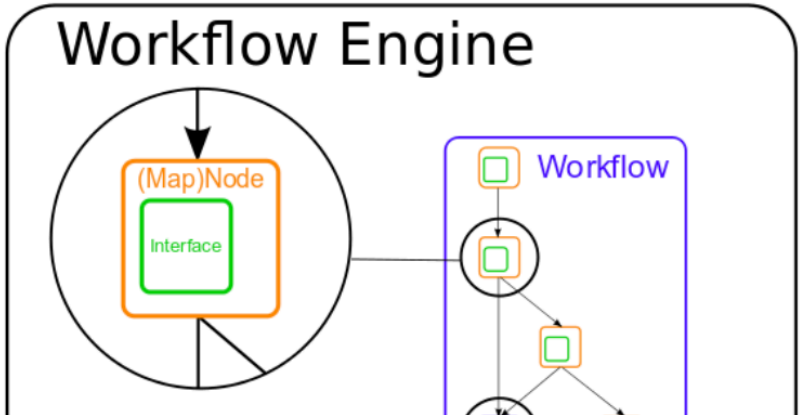
Docs: Stable · Dev
Code: Github · Bugs-Requests
Forum: User · Developer
Chat: Gitter · Slack
Funding ·

Nipype: Neuroimaging in Python Pipelines and Interfaces

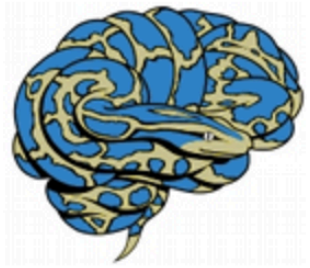
https://nipype.readthedocs.io/en/latest/

Home · Quickstart · Documentation · About · Nipy





- *Nipype* allows you to:
  - easily interact with tools from different software packages
  - combine processing steps from different software packages
  - develop new workflows faster by reusing common steps from old ones
  - process data faster by running it in parallel on many cores/machines
  - make your research easily reproducible
  - share your processing workflows with the community



Nipype: Neuroimaging in Python Pipelines and Interfaces

https://nipype.readthedocs.io/en/latest/

See live demo on Nipype
(by F. Laruina)

Home · Quickstart · Documentation · About · Nipy

Quickstart

Downloading and installing

- Download and install

Beginner's guide

Michael Notter's Nipype tutorial. Available here

How should I ask questions or report bugs?

- If you find a bug or a suggestion for improvement report it here: GitHub Issues
- If you have a conceptual or clarification question ask it here: Neurostars
- If you want to hangout with developers, you can use either Gitter or Slack

Nipype workshop materials

Self-assessment questionnaire with links to learning about each piece. Available here

Google

Table of Contents

Versions

Release	Devel
1.3.0-rc1	1.4.0-dev+gb5cb2

MATLAB and Python

- MATLAB provides a flexible, two-way integration with other programming languages, allowing you to reuse legacy code
- You can access Python functionality from MATLAB (you call Python functions and objects directly from MATLAB)

Install Supported Python Implementation

- MATLAB supports versions 2.7, 3.5, 3.6, and 3.7.
- To call a Python function, type py. in front of the module name and function name.
- Pass MATLAB data as arguments to Python function.
- MATLAB converts the data into types that best represent the data to the Python language.

```
>> pyversion                    % Change default version of Python interpreter → pyenv (since R2019b)

>> P = py.sys.path;             % Create a list of folders P using the Pythonsys.path variable.
>> methods(P)                   % Display the Python methods for a list type.
>> py.help('list.append')       % Shows the documentation for append.
```

Python and MATLAB

Get Started with MATLAB Engine API for Python

R2019b

The MATLAB® Engine API for Python® provides a Python package named `matlab` that enables you to call MATLAB functions from Python. You install the package once, and then you can call the engine in your current or future Python sessions. For help on installing or starting the engine, refer to:

- Install MATLAB Engine API for Python
- Start and Stop MATLAB Engine for Python

The `matlab` package contains the following:

- The MATLAB Engine API for Python
- A set of MATLAB array classes in Python (see MATLAB Arrays as Python Variables)

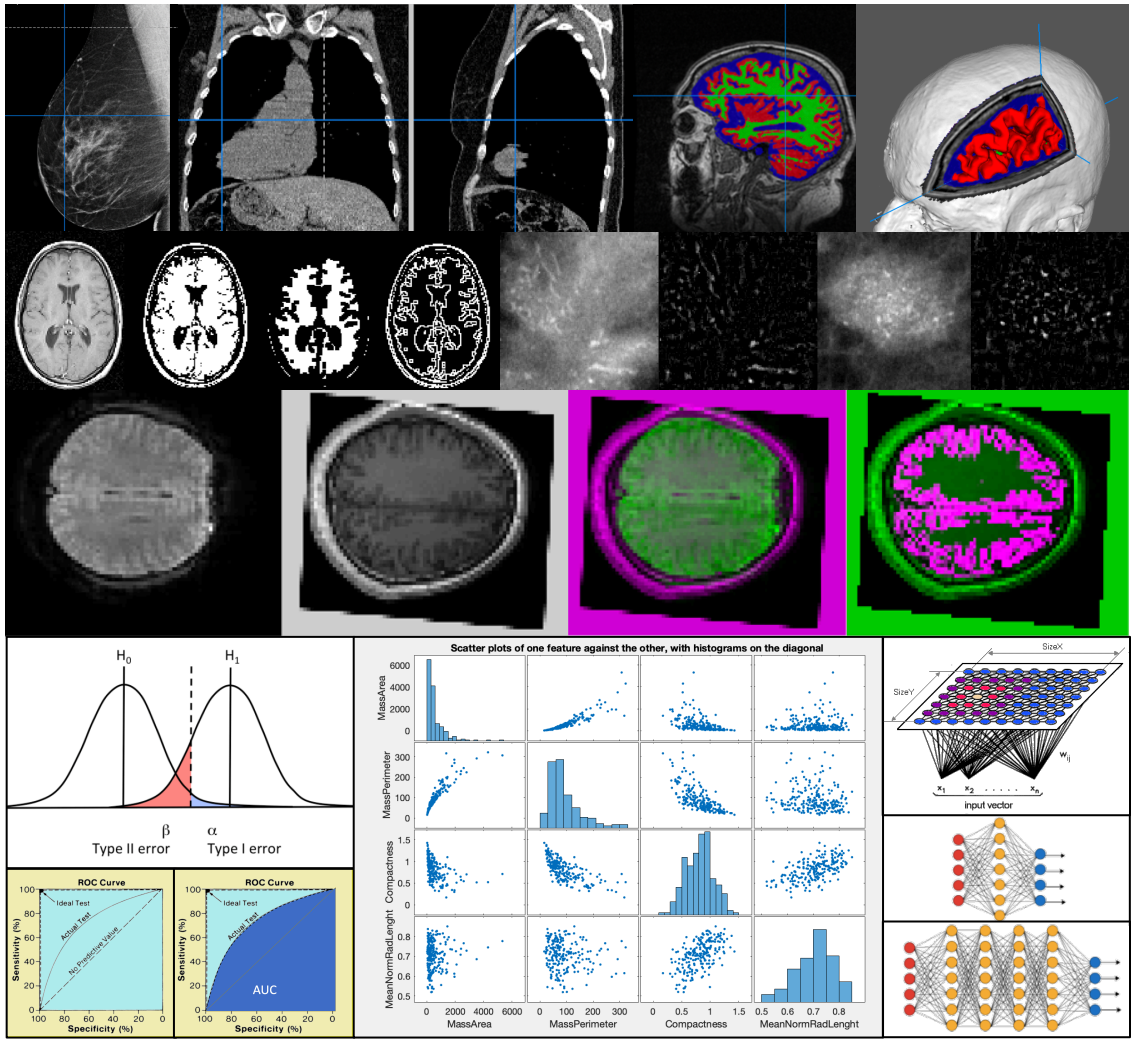
The engine provides functions to call MATLAB, and the array classes provide functions to create MATLAB arrays as Python objects. You can create an engine and call MATLAB functions with `matlab.engine`. You can create MATLAB arrays in Python by calling constructors of an array type (for example, `matlab.double` to create an array of doubles). MATLAB arrays can be input arguments to MATLAB functions called with the engine.

```
>>>  import matlab.engine
>>> eng = matlab.engine.start_matlab()
```

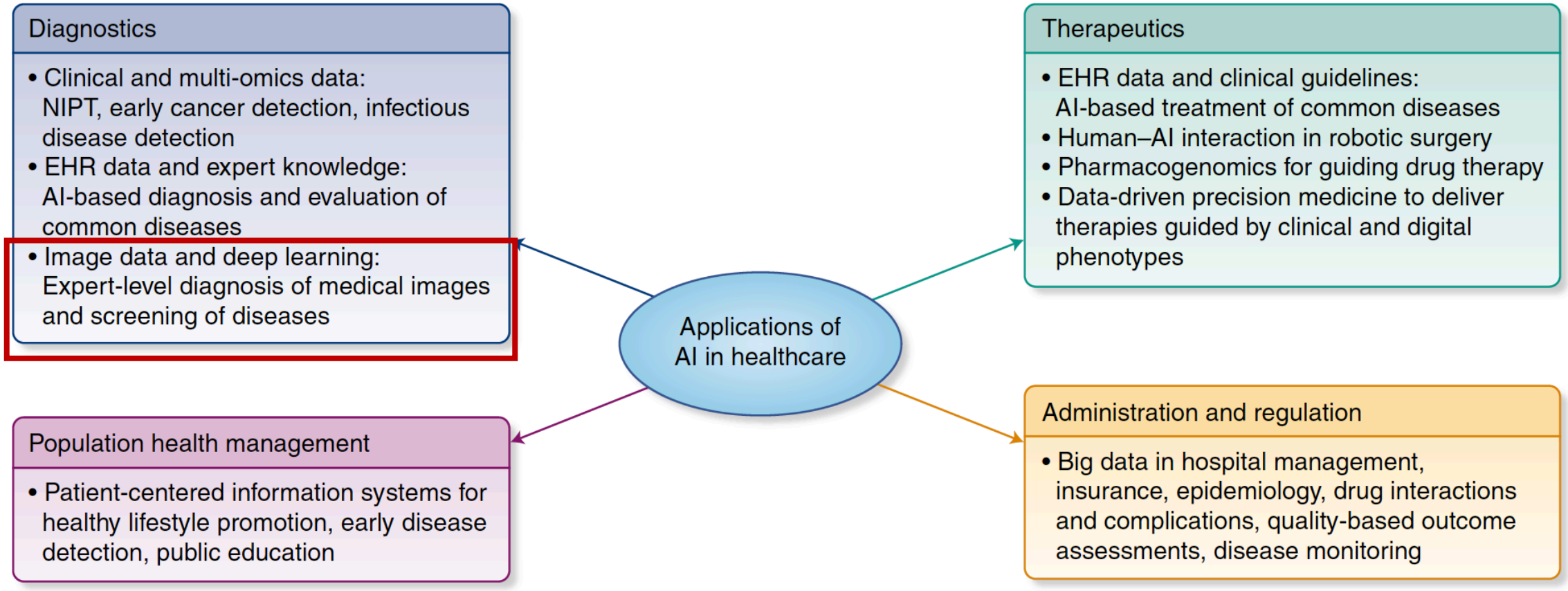
Course sum up

Medical data processing, feature extraction, feature/image classification:

- ✓ Handling standard-format medical data (DICOM), data anonymization, visualization
- ✓ Image filtering
- ✓ Deriving features form images, image segmentation
- ✓ Data quality control, outlier removal, dimensionality reduction
- ✓ Data analysis and classification
- ✓ Performance evaluation: figures of merit, cross-validation, permutation test
- ✓ Machine-learning and deep-learning tools for segmentation and classification



Artificial Intelligence applications in healthcare



Legend: HER, Electronic Health Records; NIPT, noninvasive prenatal test

*J He et al., The practical implementation of artificial intelligence technologies in medicine, Nature Medicine **25**, 30–36 (2019)*

The *gold standard (ground truth)* problem

- Expert radiologists :
  - They quite often do not agree on the presence of lesions or on lesion relevance
  - Their manual segmentations will not be highly reproducible inter and intra-readers
- The most common strategy to create a gold standard is to form a panel of experts to jointly come together to a solution in doubtful cases
- Problems when performing algorithm training and validation:
  - the algorithm performance may drastically change according to the gold standard
- A possible alternative approach is to rely on **synthetic datasets!**

Synthetic brain tumor database

GANs to build synthetic three-dimensional MRIs of brains with tumors

Brain tumor images from BRATS challenge
http://www.braintumorsegmentation.org

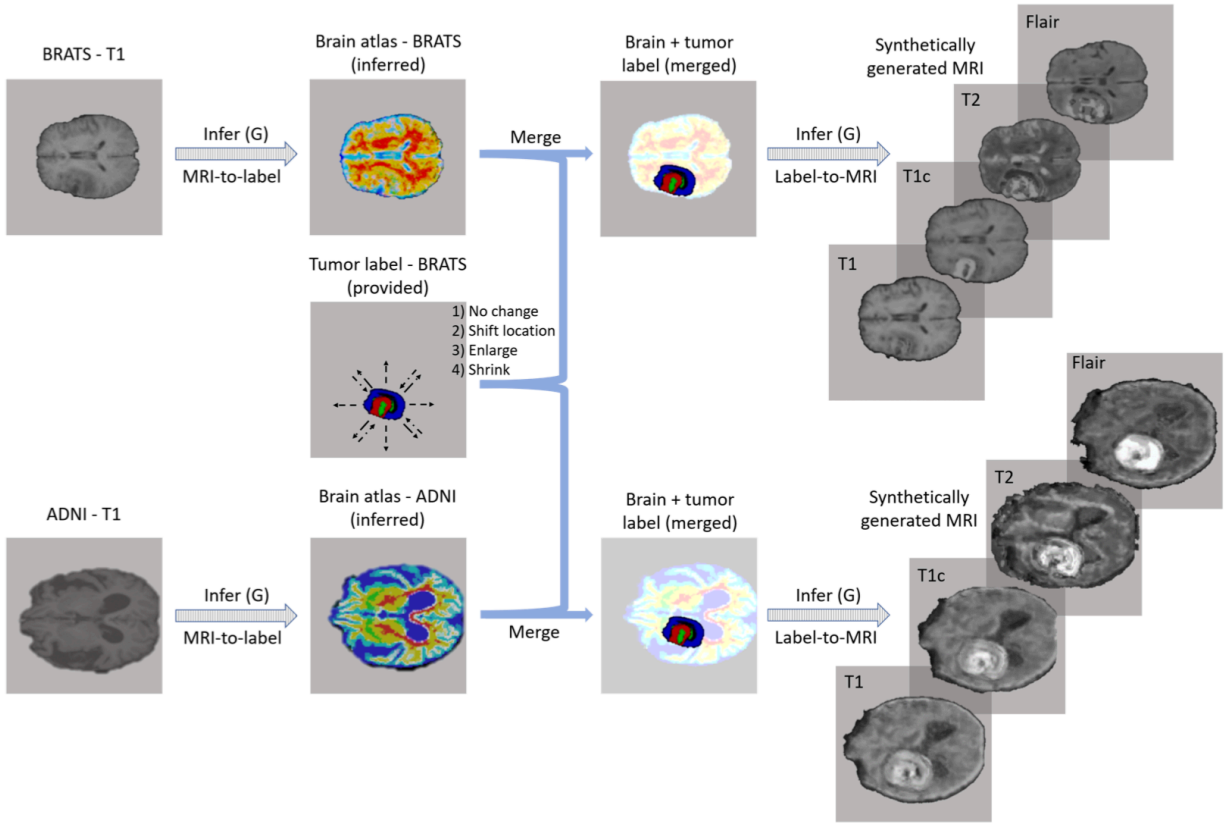
Medical Image Synthesis for Data Augmentation and Anonymization using Generative Adversarial Networks

Hoo-Chang Shin[1], Neil A Tenenholtz[2], Jameson K Rogers[2], Christopher G Schwarz[3], Matthew L Senjem[3], Jeffrey L Gunter[3], Katherine Andriole[2], and Mark Michalski[2]

[1] NVIDIA Corporation
[2] MGH & BWH Center for Clinical Data Science, Boston, MA, USA
[3] Mayo Clinic, Rochester, MN, USA
hshin@nvidia.com, jrogers24@partners.org, gunter.jeffrey@mayo.edu

Abstract. Data diversity is critical to success when training deep learning models. Medical imaging data sets are often imbalanced as pathologic findings are generally rare, which introduces significant challenges when training deep learning models. In this work, we propose a method to generate synthetic abnormal MRI images with brain tumors by training a generative adversarial network using two publicly available data sets of brain MRI. We demonstrate two unique benefits that the synthetic images provide. First, we illustrate improved performance on tumor segmentation by leveraging the synthetic images as a form of data augmentation. Second, we demonstrate the value of generative models as an anonymization tool, achieving comparable tumor segmentation results when trained on the synthetic data versus when trained on real subject data. Together,

Shin HC. et al. (2018) Medical Image Synthesis for Data Augmentation and Anonymization Using Generative Adversarial Networks. Lecture Notes in Computer Science, vol 11037. Springer, Cham



Workflow of getting synthetic images with variation. On BRATS data set, MRI-to-label image translation GAN is applied to T1-weighted images to get brain atlas. It is then merged with the tumor label given in the BRATS data set, possibly with alterations (shift tumor location; enlarge; shrink). The merged labels (with possibly alterations) are then used as an input to label- to-MRI GAN, to generate synthetic multi-parametric MRI with brain tumor.

Concluding remarks

- Medical diagnostic imaging daily produces to an incredible amount of digital information
  - ➢ not fully exploited neither for diagnosis nor for research!

- Clinicians need to be supported by reliable, effective and easy-to-use tools for diagnosing and monitoring a wide range of disease conditions

- Large Consortia are sharing multimodal and multicenter data in different medical fields

- The Medical Imaging community still lacks:
  - automated data quality pipelines
  - data harmonization strategies both for longitudinal and multicenter studies
  - new computational approaches to process and to mine multimodal data (imaging, genetics, clinical, demographic, etc.)
  - validated expert systems to support diagnosis and follow up of patients

Useful links

- https://it.mathworks.com/help/matlab/matlab-engine-for-python.html
- http://www.braintumorsegmentation.org

Docker

- https://www.docker.com
- https://miykael.github.io/nipype_tutorial/notebooks/introduction_docker.html

Nipype

- https://nipy.org/packages/nipype/index.html
- https://nipype.readthedocs.io/en/latest/
- https://miykael.github.io/nipype_tutorial/notebooks/introduction_nipype.html#3
- https://timvanmourik.github.io/Porcupine/examples/simple-example