

Precision-medicine-toolbox development pipeline

Lisa Lavrova



Motivation

Medical
Imaging

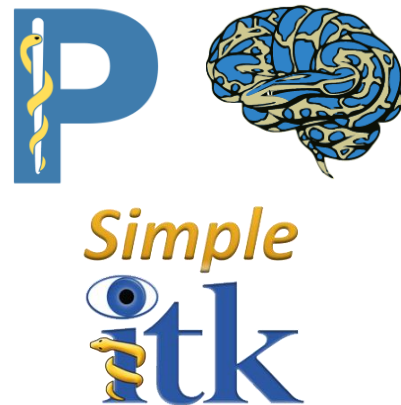
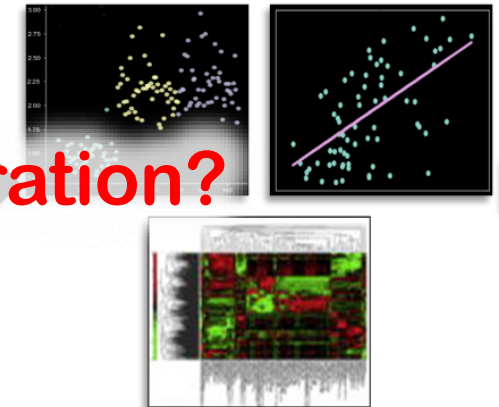
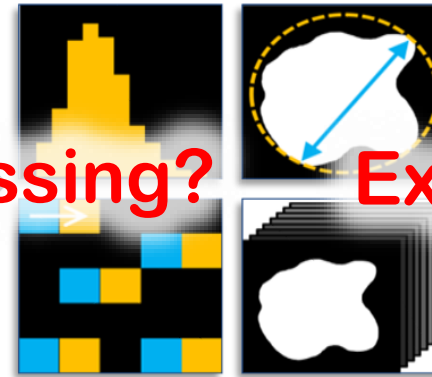
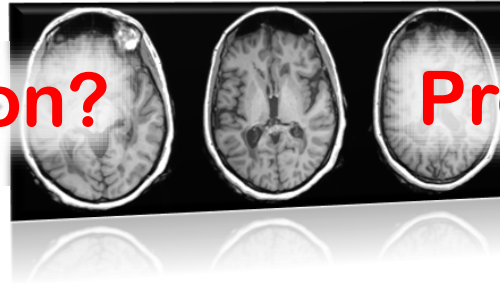
ROI Features
Extraction

Radiomic
Analysis

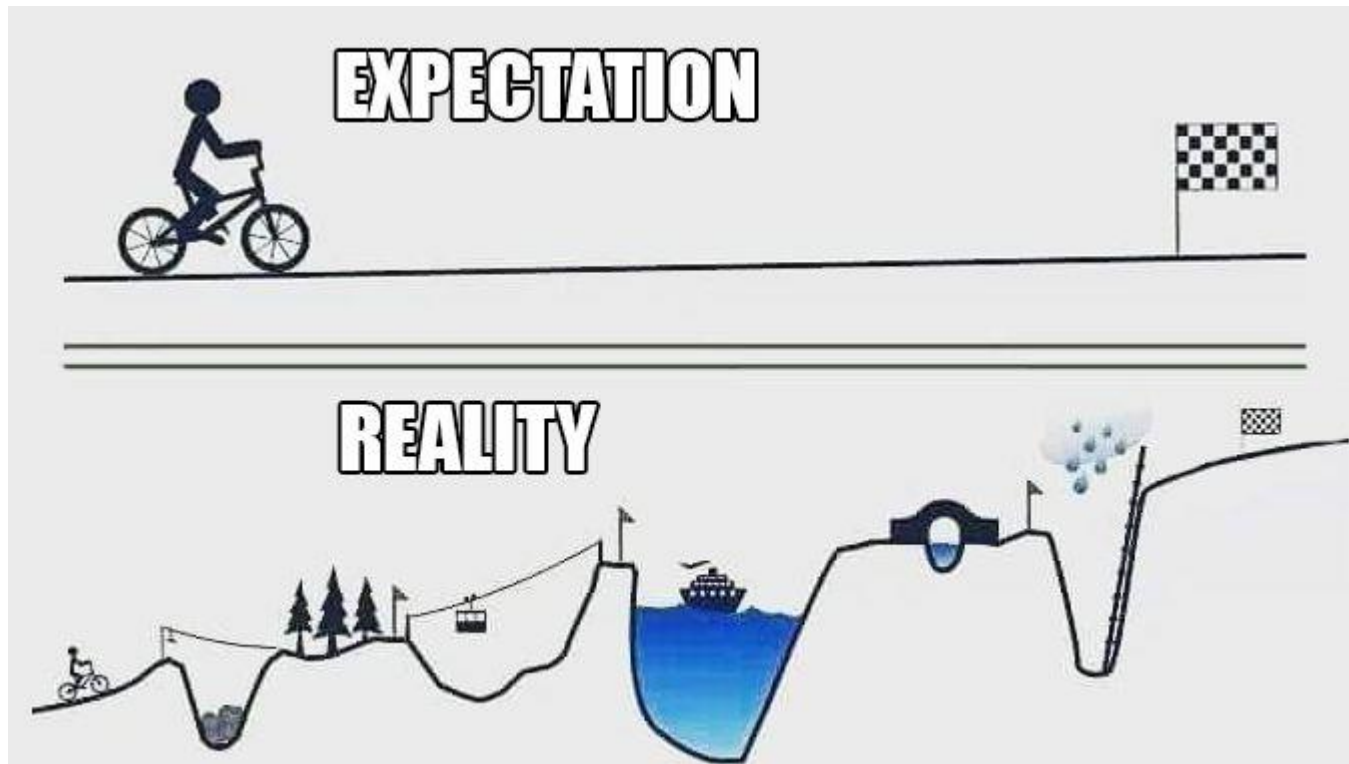
Curation?

Processing?

Exploration?



IDEA → CODE → PROJECT/PRODUCT?



Coding&testing

- Write the code for people, not computers
- Let computer do the work
- Don't repeat yourself
- Use libraries
- ...

AnalysisBox.py

DataSet.py

FeaturesSet.py

GenerateResultBox.py

ResultSet.py

ToolBox.py

```
def plot_distribution(self, features_to_plot: list=[], binary_classes_to_plot: list=[]):
    """Plot distribution of the feature values in classes into int"""

    Arguments:
        features_to_plot: List of specific features to be selected
        binary_classes_to_plot: List, containing 2 classes of interest
    """

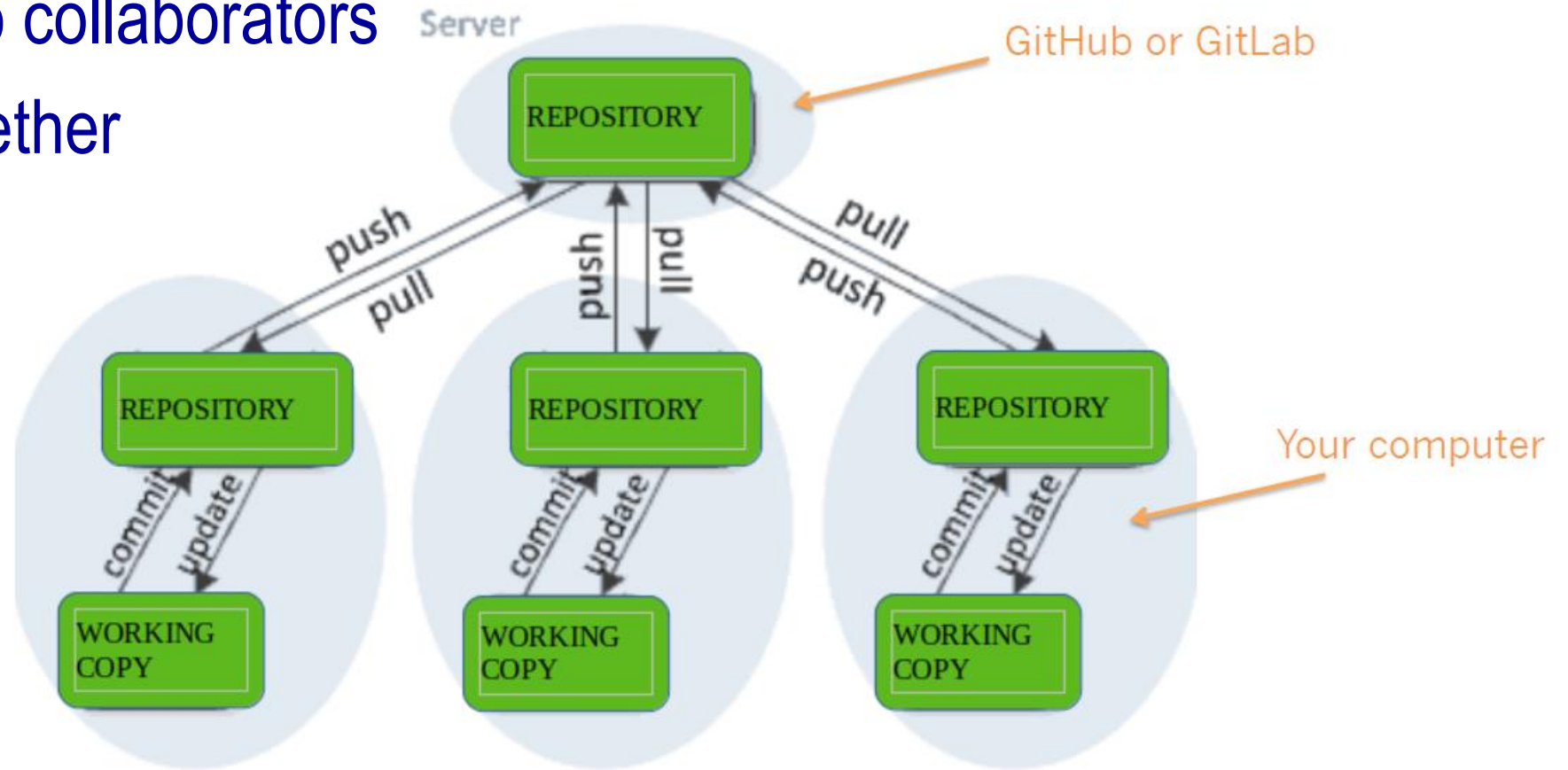
    if len(self._outcome) > 0:
        if len(binary_classes_to_plot) == 2:
            if (binary_classes_to_plot[0] in self._class_label) & (binary_classes_to_plot[1] in self._class_label):
                if not features_to_plot:
                    features_to_plot = self._feature_column
                num_features = []
                for feature in features_to_plot:
                    if self._feature_dataframe[feature].dtype != 'object':
                        num_features.append(feature)
                cols = 4
                rows = len(num_features) // 4 + 1
```

Git

If its an open-source project, online platforms for version control!

Grant an access to collaborators

Learn working together



GitHub

License!

README!

Requirements file!

Gitignore

Welcome to *precision-medicine-toolbox* documentation!

DOI [10.5281/zenodo.6656046](https://doi.org/10.5281/zenodo.6656046) license [BSD-3-Clause](#) docs [passing](#) codefactor [B-](#) pypi [v0.4](#)

precision-medicine-toolbox is an open-source python package for medical imaging data preparation for data science tasks. This package is aimed to provide a tool to curate the imaging data and to perform exploratory feature analysis.

If you are using this toolbox, please, cite the [original paper](#):

Primakov, Sergey, Elizaveta Lavrova, Zohaib Salahuddin, Henry C. Woodruff, and Philippe Lambin. "Precision-medicine-toolbox: An open-source python package for facilitation of quantitative medical imaging and radiomics analysis." arXiv preprint arXiv:2202.13965 (2022).

Comment the code

Explain the variables
Indicate the data types

```
def plot_univariate_roc(self, features_to_plot: list=[], binary_classes_to_plot: list=[], auc_threshold: float=0.75):  
    """Plot univariate ROC curves (with AUC calculation) for threshold binary classifier, based of each feature separately into interactive .html report.  
  
    Arguments:  
        features_to_plot: List of specific features to be selected (otherwise selects all the numerical features).  
        binary_classes_to_plot: List, containing 2 classes of interest in case of multi-class data.  
        auc_threshold: Threshold value for ROC AUC to be highlighted.  
    """
```

Create the examples

Quality control step

This functionality aims to perform a simple quality check of the data and possibly to detect the wrong scans. These might be scans: of wrong imaging modality, with wrong imaging projections, with non-consistent (missing/overlapping) slices, with insufficient amount of slices, with slice thickness inconsistent or out of the defined range, with pixel spacing out of range, with unknown or unacceptable konvolution kernel, with wrong axial plane resolution, with missing slope/intercept tags. To perform this check, the target scanning parameters are to be passed to the function. By removing/commenting some of the input parameters, the corresponding checks are disabled. For each patient, the output table contains the following flags: '1' - check passed, '0' - check failed.

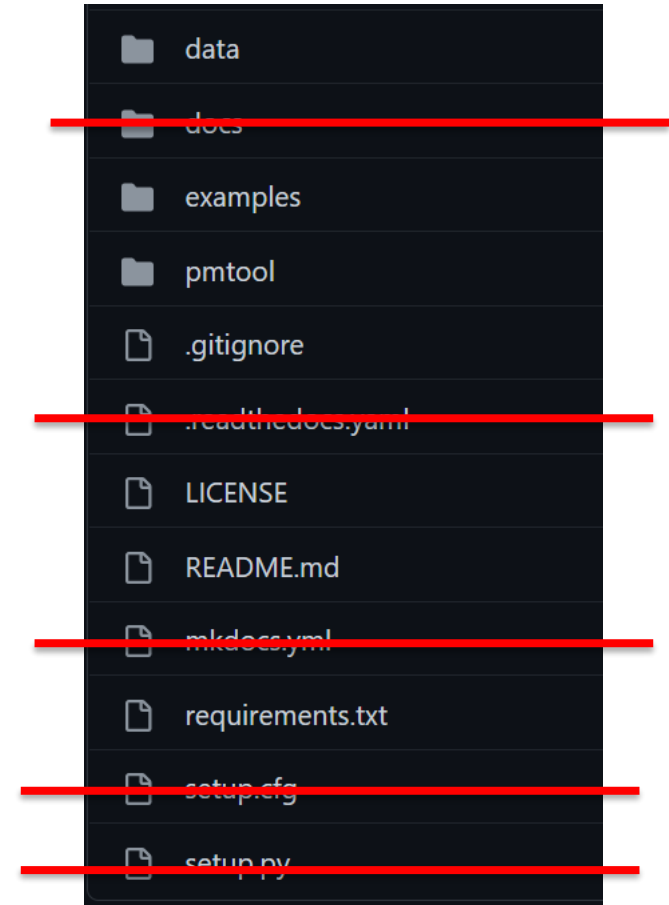
```
In [8]: qc_params = {'specific_modality': 'ct', # target modality: CT
                    'thickness_range': [2,5], # slice thickness should be in range of 2..5 mm
                    'spacing_range': [0.5,1.25], # pixel spacing should be in range of 0.5..1.25 mm
                    'scan_length_range': [5,170], # scan should contain from 5 to 170 slices
                    'axial_res': [512,512], # the axial resolution should be 512x512
                    'kernels_list': ['standard','lung','b19f']} # the following kernels are acceptable
```

```
In [9]: qc_dataframe = ct_dcms.get_quality_checks(qc_params)
```

```
67%|███████████████████████████████████████| 2/3 [00:01<00:00, 1.31it/s]
Cannot perform conv kernel check for pat: LUNG1-002_20180526_CT_1
100%|██████████████████████████████████████████████████████████████████████████████| 3/3 [00:02<00:00, 1.32it/s]
Cannot perform conv kernel check for pat: LUNG1-003_20180209_CT_1
```


Think about the project organization

```
-- CITATION
-- README
-- LICENSE
-- requirements.txt
-- data
|   |-- birds_count_table.csv
-- doc
|   |-- notebook.md
|   |-- changelog.txt
-- results
|   |-- summarized_results.csv
-- src
```



Creating a proper documentation: MkDocs

- Installation: `pip install mkdocs`
- Initialising new project:
`mkdocs new my-project`
- To have a look at every change of your web-site (or even understand if it is compilable): `mkdocs serve`
- Hint: don't stop running it from server, changes appear in real time

```
1  site_name: precision-medicine-toolbox
2  nav:
3    - Home: index.md
4    - Tutorials: tutorials.md
5    - API:
6    - Imaging class: DataSet.md
7    - Imaging module: Toolbox.md
8    - Features class: FeaturesSet.md
9    - Features module: AnalysisBox.md
10 theme: readthedocs
11 docs_dir: 'docs'
12 site_dir: 'site'
13 plugins:
14   - search
15   - mkdocstrings
16 copyright: Copyright &copy; 2019 - 2022, Sergey Primakov & Lisa Lavrova.
```

Creating a proper documentation: MkDocs

- Creating all the necessary pages in md:
 - Tutorials page
 - Separate page for each tutorial with code and description
- Editing an index page, adding:
 - functionality,
 - (future) links to examples and documentation,
 - 3rd party packages,
 - quick guide on installation and quick start,
 - how to contribute,
 - authors, citation, acknowledgements

Tutorials: from Jupyter to MD

Imaging module tutorial

This tutorial shows how to explore the imaging parameters and perform a basic image quality check, convert DICOM into NRRD, check ROI segmentations, pre-process the images, and extract radiomic features.

Importing modules:

```
import os, sys
from pmtool.ToolBox import ToolBox
import pandas as pd
from pandas import DataFrame
import matplotlib.pyplot as plt
import numpy as np
import seaborn as sb

%matplotlib inline
```

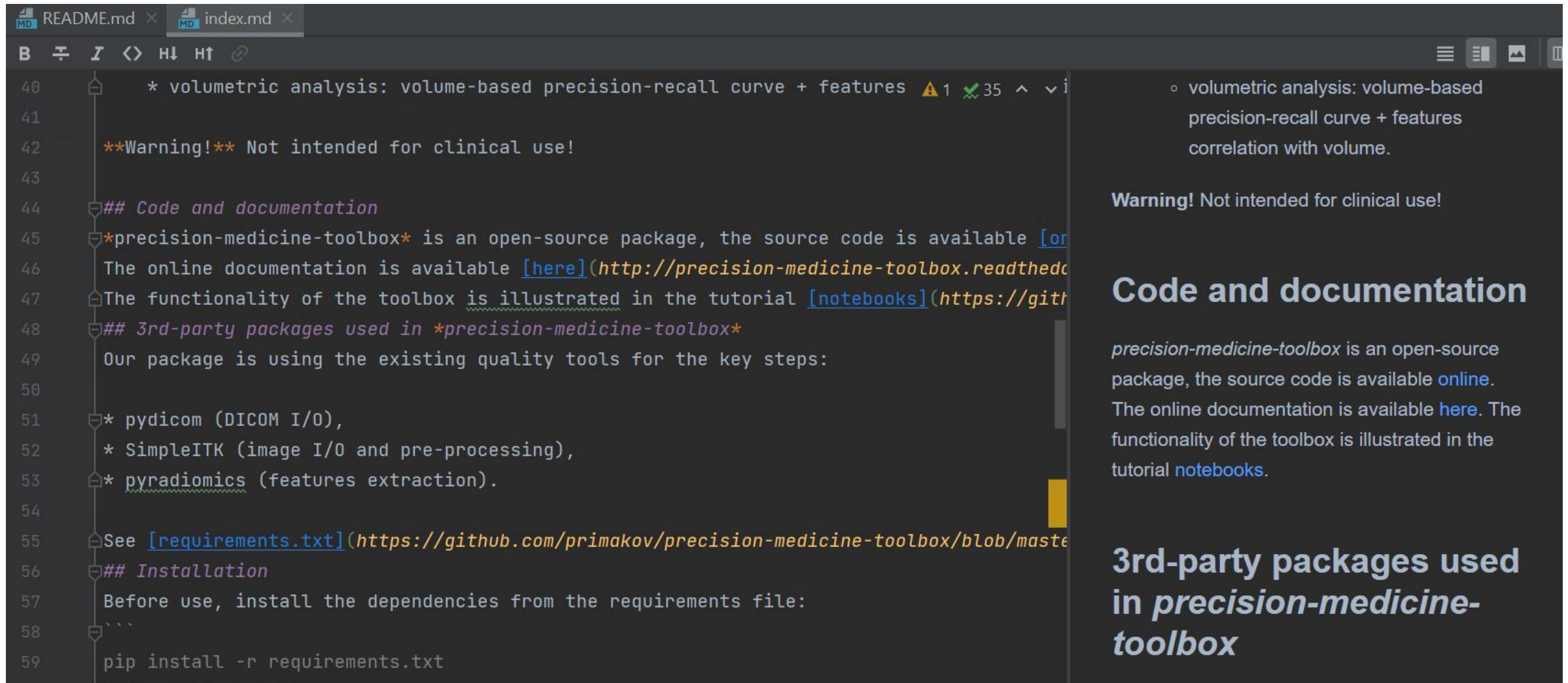
DICOM to NRRD conversion

To convert DICOM dataset to the volume (nrrd) format, set up the parameters:

- 'export_path = ...' export path where the converted NRRDs will be placed,
- 'region_of_interest = ...' if you know exact name of the ROI you want to extract, then write it with the '!' character in front, eg. 'region_of_interest = !gtv1',
- if you want to extract all the GTVs in the rtstruct eg. 'gtv1', 'gtv2', 'gtv_whatever', then just specify the stem word, eg. 'region_of_interest = gtv',
- default value is 'region_of_interest = 'all'', where all ROI's in rtstruct will be extracted.

```
export_path = r'../data/' # the function will create 'converted_nrrd' folder in the specified directory
```

Markdown text example



```
40 * volumetric analysis: volume-based precision-recall curve + features ⚠️ 1 ✅ 35 ^ v i
41
42 **Warning!** Not intended for clinical use!
43
44 ## Code and documentation
45 *precision-medicine-toolbox* is an open-source package, the source code is available [or
46 The online documentation is available [here](http://precision-medicine-toolbox.readthedocs
47 The functionality of the toolbox is illustrated in the tutorial [notebooks](https://gith
48 ## 3rd-party packages used in *precision-medicine-toolbox*
49 Our package is using the existing quality tools for the key steps:
50
51 * pydicom (DICOM I/O),
52 * SimpleITK (image I/O and pre-processing),
53 * pyradiomics (features extraction).
54
55 See [requirements.txt](https://github.com/primakov/precision-medicine-toolbox/blob/master
56 ## Installation
57 Before use, install the dependencies from the requirements file:
58 ```
59 pip install -r requirements.txt
```

◦ volumetric analysis: volume-based precision-recall curve + features correlation with volume.

Warning! Not intended for clinical use!

Code and documentation

precision-medicine-toolbox is an open-source package, the source code is available [online](#). The online documentation is available [here](#). The functionality of the toolbox is illustrated in the tutorial [notebooks](#).

3rd-party packages used in *precision-medicine-toolbox*

We want API description to be generated automatically: mkdocstrings

- Add this plug-in to MkDocs configuration file!
- Create markdown files with handlers linked to every module (script) of the code

```
# Documentation for AnalysisBox

::: pmtool.AnalysisBox.AnalysisBox
  handler: python
  selection:
    members:
      - handle_nan
      - handle_constant
      - plot_distribution
      - plot_correlation_matrix
      - plot_MW_p
      - plot_univariate_roc
      - calculate_basic_stats
      - volume_analysis
  rendering:
    show_root_heading: false
    show_source: false
```

Organise a documentation web-site structure in the MkDocs configuration file (mkdocs.yml)

```
1  site_name: precision-medicine-toolbox
2  nav:
3    - Home: index.md
4    - Tutorials: tutorials.md
5    - API:
6      - Imaging class: DataSet.md
7      - Imaging module: ToolBox.md
8      - Features class: FeaturesSet.md
9      - Features module: AnalysisBox.md
10 theme: readthedocs
11 docs_dir: 'docs'
12 site_dir: 'site'
13 plugins:
14   - search
15   - mkdocstrings
16 copyright: Copyright &copy; 2019 - 2022, Sergey Primakov & Lisa Lavrova.
```


Hosting documentation online: ReadTheDocs.io

- Sign up
- Give it an access to your GitHub
- Import a project
- Build documentation (it has to have requirements file and actually able to be BUILD)
- Be careful with project options (documentation file, python interpreter, disable PDF build for MkDocs)

Code and documentation

3rd-party packages used in
precision-medicine-toolbox

Installation

Quick start

Contributing

Authors and citation

License

Acknowledgements

Tutorials

API

Imaging class

Imaging module

Features class

Features module

Welcome to *precision-medicine-toolbox* documentation!

DOI 10.5281/zenodo.6656046

license BSD-3-Clause

docs passing

codefactor B-

pypi v0.4

precision-medicine-toolbox is an open-source python package for medical imaging data preparation for data science tasks. This package is aimed to provide a tool to curate the imaging data and to perform exploratory feature analysis.

If you are using this toolbox, please, cite the [original paper](#):

Primakov, Sergey, Elizaveta Lavrova, Zohaib Salahuddin, Henry C. Woodruff, and Philippe Lambin.

"Precision-medicine-toolbox: An open-source python package for facilitation of quantitative medical imaging and radiomics analysis." *arXiv preprint arXiv:2202.13965* (2022).


Currently, the toolbox has the following functionality:

- **Dataset exploration.** This function gets the specified metadata from the DICOM files of the dataset and allows for exploration of the diversity degree of the imaging parameters..
- **Dataset quality check.** This function checks every scan in the dataset to be in line with the pre-defined requirements:

Getting DOI for the project: Zenodo




- Sign up
- Give it an access to your GitHub
- Create a release of your code on GitHub
- Get a badge and DOI

Enabled Repositories

 lavrovaliz/plaq-u-net	<input checked="" type="checkbox"/>
DOI 10.5281/zenodo.6953319	

Repositories

If your organization's repositories do not show up in the list, please ensure you have enabled [third-party access](#) to the Zenodo application. Private repositories are not supported.

 lavrovaliz/DuneAI-Automated-detection-and-segmentation-of-non-small-cell-lung-cancer-computed-tomography-images	<input type="checkbox"/>
 lavrovaliz/dlab-scripts	<input type="checkbox"/>
 lavrovaliz/neuroimaging-training	<input type="checkbox"/>

Turning scripts into compiled installable project: PyPi

(or making it possible to be installed with 'pip install')

- Create `__init__.py` in code folder (used when package is built)

```
from pmtool.ToolBox import ToolBox
from pmtool.AnalysisBox import AnalysisBox
from pmtool.GenerateResultBox import GenerateResultBox
```

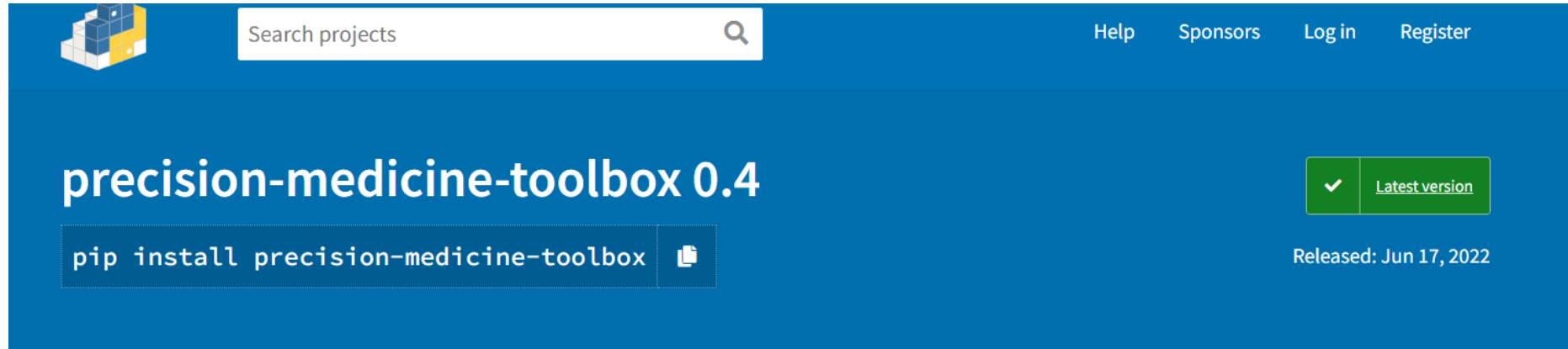
- Main directory: `setup.py` to build a package (should contain a version + you cannot re-release the same version!!!)

setup.py

```
from distutils.core import setup
setup(
    name = 'precision-medicine-toolbox',          # How you named your package folder (MyLib)
    packages = ['pmtool'],      # Chose the same as "name"
    version = '0.4',            # Start with a small number and increase it with every change you make
    license='bsd-3-clause',      # Chose a license from here: https://help.github.com/articles/licensing-a-repository
    description = 'Precision medicine tool-box for medical imaging research', # Give a short description about your library
    author = 'sergey primakov & lisa lavrova',      # Type in your name
    author_email = 'primakov@bk.ru',      # Type in your E-Mail
    url = 'https://github.com/primakov/precision-medicine-toolbox', # Provide either the link to your github or to your website
    download_url = 'https://github.com/primakov/precision-medicine-toolbox/archive/refs/tags/0.0.tar.gz', # I explain this later on
    keywords = ['medical imaging research', 'dicom', 'radiomics', 'statistical analysis', 'features'], # Keywords that define your package best
    install_requires=[
        'SimpleITK',
        'PyWavelets',
        'pykwality',
        'six',
        'tqdm',
        'pydicom',
        'pandas',
        'pyradiomics',
        'scikit-image',
        'ipywidgets',
        'matplotlib',
        'Pillow',
        'scikit-learn',
        'scipy',
```

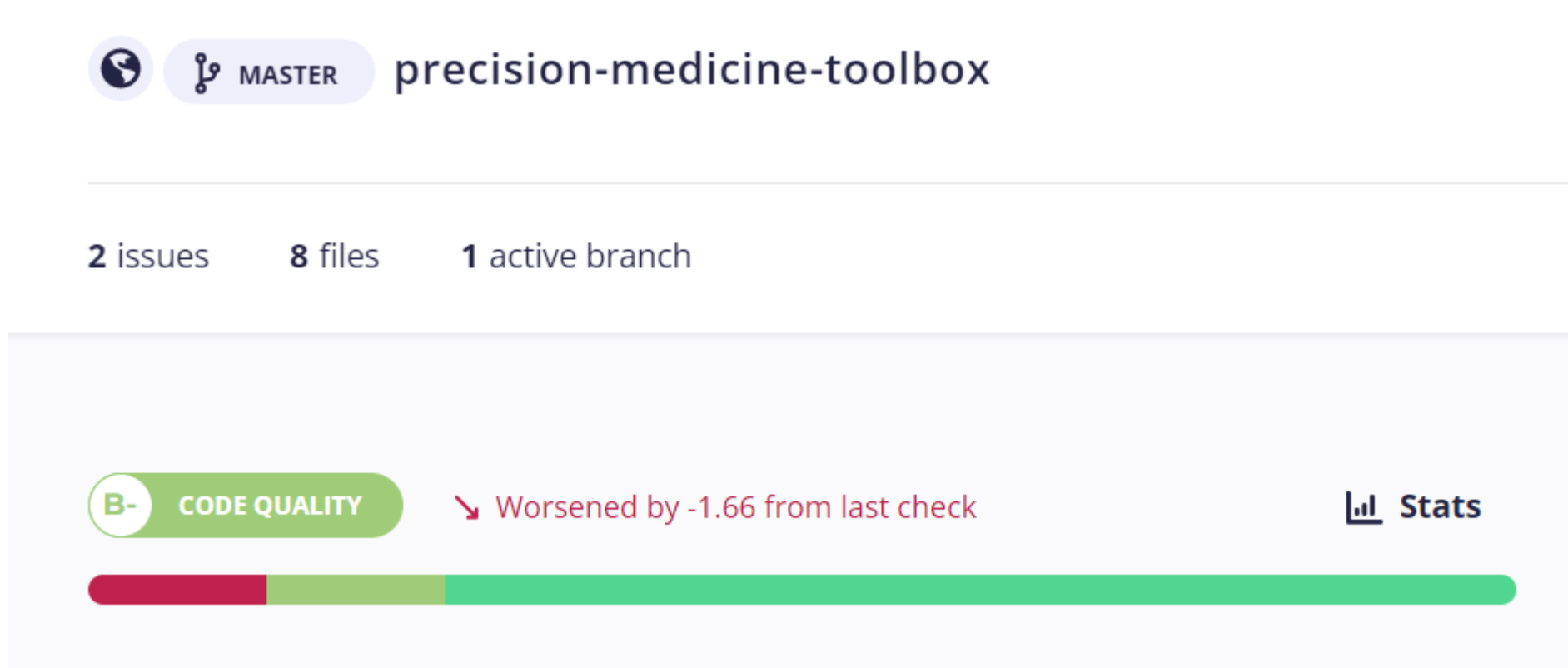
Building a project

- Sign in pypi.org
- pip install setuptools wheel
- pip install twine
- python setup.py bdist_wheel (from project dir)
- twine upload dist/* (you will need your username and password)



Code quality: CodeFactor

Give an access to GitHub



Badges

[![DOI](zenodo link)]

[![License](GitHub link)]

[![Documentation Status]

[![CodeFactor](CodeFactor link)]

[![PyPI](PyPi link)]

Welcome to
*precision-
medicine-
toolbox*
documentati
on!

DOI 10.5281/zenodo.6656046

license BSD-3-Clause

docs passing codefactor B-

pypi v0.4

What's next?

- Preprint/paper
- Maintenance:
 - Troubleshooting
 - Change in code
 - Change in documentation/examples → re-upload documentation
 - New release and new version!
 - New Zenodo DOI
 - New setup.py, building a new release, uploading to pypi.org
- More users → more improvement

THANK YOU