
pytrip98gui Documentation

Release 0.1.1.post0.dev

Author

October 12, 2016

1	WHAT IS THIS ?	3
1.1	Quick installation guide	3
1.2	More documentation	3
2	Contributing	5
2.1	Types of Contributions	5
2.2	Get Started!	6
2.3	Pull Request Guidelines	6
2.4	Tips	7
3	Credits	9
3.1	Development Lead	9
3.2	Contributors	9
4	Indices and tables	11

Contents:

WHAT IS THIS ?

pytripgui is graphical user interface (GUI) built around TRiP98 planning system and pytrip package. It is capable of visualising patient CT data, dose and LET overlays. pytripgui can make treatment plans using local or remote TRiP98 package.

TRiP98 package is not included here, if you need it, go first to TRiP98 webpage <http://bio.gsi.de/DOCS/TRiP98/NEW/DOCS/trip98.html>

pytripgui works under Linux operating system with necessary packages installed, see installation instructions below.

1.1 Quick installation guide

We recommend that you run a modern Linux distribution, like: **Ubuntu 16.04** or newer, **Debian 9 Stretch** (currently known as testing) or any updated rolling release (archLinux, openSUSE tumbleweed). In this case, be sure you have **python** and **python-pip** installed. To get them on Debian or Ubuntu, type being logged in as normal user:

```
$ sudo apt-get install python-pip
```

Next step is to install required packages:

```
$ sudo apt-get install python-wxgtk3.0 python-tk
```

To automatically download and install the pytripgui, type:

```
$ sudo pip install pytrip98gui
```

NOTE: the pip package is named **pytrip98gui**, while the name of project is **pytripgui**.

This command will automatically download and install pytripgui for all users in your system.

Start it by calling:

```
$ pytripgui
```

More detailed instruction will available soon. pytripgui currently works only with Python 2.7 on Linux. pytripgui does not support Windows and Python3.x (yet).

1.2 More documentation

If you would like to download the code of pytripgui, read first developer documentation.

Contributing

Contributions are welcome, and they are greatly appreciated! Every little bit helps, and credit will always be given. You can contribute in many ways:

2.1 Types of Contributions

2.1.1 Report Bugs

Report bugs at <https://github.com/pytrip/pytripgui/issues>.

If you are reporting a bug, please include:

- Your operating system name and version.
- Any details about your local setup that might be helpful in troubleshooting.
- Detailed steps to reproduce the bug.

2.1.2 Fix Bugs

Look through the GitHub issues for bugs. Anything tagged with “bug” is open to whoever wants to implement it.

2.1.3 Implement Features

Look through the GitHub issues for features. Anything tagged with “feature” is open to whoever wants to implement it.

2.1.4 Write Documentation

pytripgui could always use more documentation, whether as part of the official pytripgui docs, in docstrings, or even on the web in blog posts, articles, and such.

2.1.5 Submit Feedback

The best way to send feedback is to file an issue at <https://github.com/pytrip/pytripgui/issues>.

If you are proposing a feature:

- Explain in detail how it would work.
- Keep the scope as narrow as possible, to make it easier to implement.
- Remember that this is a volunteer-driven project, and that contributions are welcome :)

2.2 Get Started!

Ready to contribute? Here's how to set up *pytripgui* for local development.

1. Fork the *pytripgui* repo on GitHub.
2. Clone your fork locally:

```
$ git clone git@github.com:your_name_here/pytripgui.git
```

3. Install your local copy into a virtualenv. Assuming you have virtualenvwrapper installed, this is how you set up your fork for local development:

```
$ mkvirtualenv pytripgui
$ cd pytripgui/
$ python setup.py develop
```

4. Create a branch for local development:

```
$ git checkout -b name-of-your-bugfix-or-feature
```

Now you can make your changes locally.

5. When you're done making changes, check that your changes pass flake8 and the tests, including testing other Python versions with tox:

```
$ flake8 pytripgui tests
$ python setup.py test
$ tox
```

To get flake8 and tox, just pip install them into your virtualenv.

6. Commit your changes and push your branch to GitHub:

```
$ git add .
$ git commit -m "Your detailed description of your changes."
$ git push origin name-of-your-bugfix-or-feature
```

7. Submit a pull request through the GitHub website.

2.3 Pull Request Guidelines

Before you submit a pull request, check that it meets these guidelines:

1. The pull request should include tests.
2. If the pull request adds functionality, the docs should be updated. Put your new functionality into a function with a docstring, and add the feature to the list in README.rst.
3. The pull request should work for Python 2.6, 2.7, 3.3, 3.4 and 3.5, and for PyPy. Check https://travis-ci.org/pytrip/pytripgui/pull_requests and make sure that the tests pass for all supported Python versions.

2.4 Tips

To run a subset of tests:

```
$ python -m unittest tests.pytrip98gui
```

Credits

3.1 Development Lead

- Leszek Grzanka <leszek.grzanka@gmail.com>

3.2 Contributors

None yet. Why not be the first?

Indices and tables

- `genindex`
- `modindex`
- `search`