

6 months into my 1st software development journey

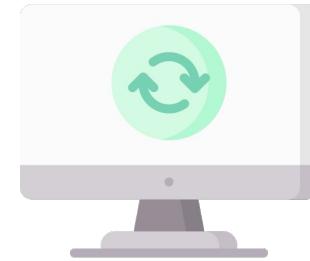
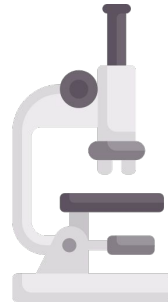
Ana Stojiljkovic

Data Science Lab (DSL) - University of Bern





About me





Topics

Usability



Documentation



Outreach



Build the GUI





Build the GUI



Find your Community!!!



<https://github.com/search?q=cookiecutter&type=Repositories&p=1>



Build the GUI



Find your Community!!!



napari

```
napari-growth-cone-finder/  
├── .github  
│   └── workflows  
│       └── test_and_deploy.yml  
├── LICENSE  
├── MANIFEST.in  
├── napari_growth_cone_finder  
│   ├── __init__.py  
│   ├── _widget.py  
│   ├── _reader.py  
│   ├── napari.yaml  
│   └── _tests  
│       ├── __init__.py  
│       ├── test_widget.py  
│       └── test_reader.py  
├── pyproject.toml  
├── README.md  
├── setup.cfg  
└── tox.ini
```

<https://github.com/napari/cookiecutter-napari-plugin>



Tools



Summer Trainings

The Data Science Lab offers every summer in June and September four weeks of trainings in various areas of programming, computation, and digital skills in general. The courses are available to all university members. Click on the links below to learn more about each course and register via Ilias (don't forget to log in)!

2023-06-05 Introduction to Git and GitHub (09:00-17:00)

2023-06-06 Tools for Python: GitHub, Jupyter and conda (09:00-17:00)

2023-06-07 Create your personal Website (09:00-17:00)

2023-06-08 Introduction to Linux for users (09:00-17:00)

2023-06-09 Advanced Python (09:00-17:00)

2023-06-12 High Performance Computing (HPC) on UBELIX (09:00-12:30)

2023-06-13 Advanced HPC Topics (09:00-12:30)

2023-06-14 Data Science and ML with MATLAB (09:00-17:00)

2023-06-15 Machine Learning with scikit-learn (09:00-17:00)

2023-06-16 Deep Learning with MATLAB (09:00-17:00)



Data Science Lab - Unibe

<https://www.dsl.unibe.ch/training/upcoming/>



Good-practice

- Visual
- Interactivity
- Workflow





Good-practice



Copy your Community!!!

napari

Usage **Plugins** Community API Reference napari hub [🔗](#)

stable (0.4.18) [📄](#)

search 🔍

- Finding and installing a napari plugin
- Your First Plugin
- Manifest Reference
- Contributions Reference
- Contribution Guides
- Test and Deploy
- Best practices**
- Debugging during plugin development
- npe2** migration guide
- 1st Gen Plugin Guide (*Deprecated*)
- ▶ In-depth guide to plugin testing
- ▶ Tools and tips when building napari plugins or: firehose of stuff I wish I knew earlier

Best practices

There are a number of good and bad practices that may not be immediately obvious when developing a plugin. This page covers some known practices that could affect the ability to install or use your plugin effectively.

Don't include PySide2 or PyQt5 in your plugin's dependencies.

This is important!

Napari supports *both* PyQt and PySide backends for Qt. It is up to the end-user to choose which one they want. If they installed napari with `pip install napari[all]`, then the `[all]` extra will (currently) install `PyQt5` for them from pypi. If they installed via `conda install napari`, then they'll have `PyQt5`, but via anaconda cloud instead of pypi. Lastly, they may have installed napari with PySide2.

Here's what can go wrong if you also declare one of these backends in the `install_requires` section of your plugin metadata:

ON THIS PAGE

- Don't include PySide2 or PyQt5 in your plugin's dependencies.**

Try not to depend on packages that require C compilation if these packages do not offer wheels

Don't import heavy dependencies at the top of your module

Don't leave resources open

Write extensive tests for your plugin!

Set style for additional windows in your plugin

Do not package your tests as a top-level package

License issues when including code from 3rd parties



My GUI



The screenshot displays the napari GUI with a central image of a grayscale wave pattern overlaid with a radial mask. The mask consists of concentric circles in blue, red, green, and magenta. The interface is divided into several panels:

- Layer Controls (Left):** Contains settings for the selected layer, including label (1), opacity (0.70), brush size (10), blending (translucent), color mode (direct), contour (0), n edit dim (2), and checkboxes for contiguous, preserve labels, show selected, and show selected.
- Layer List (Bottom Left):** Lists the layers: radial_mask (selected), point of interest, and wave_image_ch_01.
- Dynamic Panel (Right):** A workflow for generating a radial sector with the following steps:
 - Add a point of interest: Add ROI
 - Select the point or label layer from which the mask will be generated: point of interest
 - Generate a sector of the desired geometry using the tabs: Radial sector, Angular sector, Annular sector, Manual sector. Below this, there are sliders for Sector width (pixels) set to 43 and Number of sectors set to 5, followed by a "Generate radial sector!" button.
 - Select the channel to analyze: wave_image_ch_01
 - Select the sector mask to apply: radial_mask
 - Compute plot
 - Set axes range
 - Save plot data
 - Save all combinations



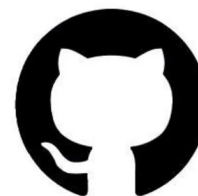
Documentation



jupyter {book}



GitHub Actions



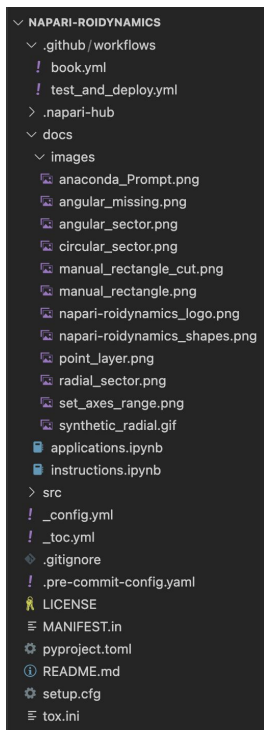
GitHub Pages



<https://jupyterbook.org/en/stable/publish/gh-pages.html>



Documentation



You can install Jupyter Book via `pip` :

```
pip install -U jupyter-book
```

or via `conda-forge` :

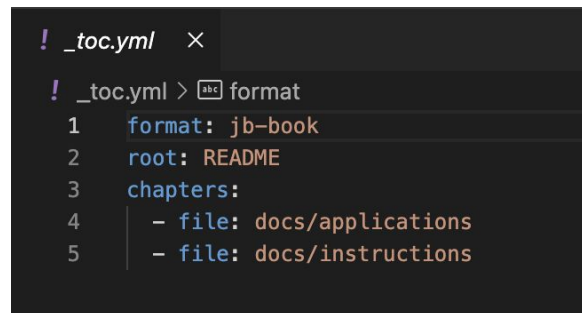
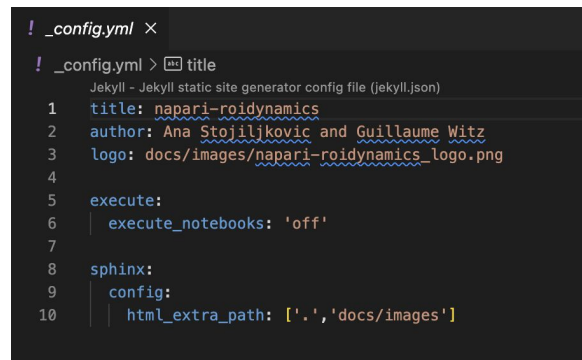
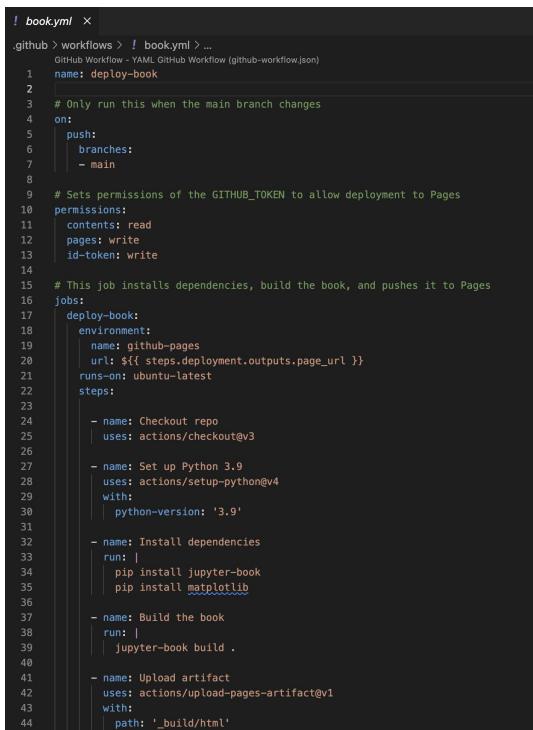
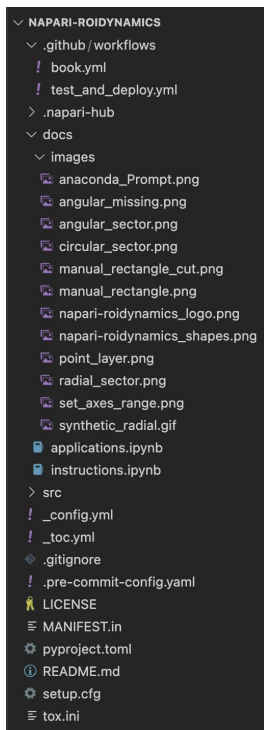
```
conda install -c conda-forge jupyter-book
```

This will install everything you need to build a Jupyter Book locally.

- **book.yml** in github/workflows
- folder containing images
- [Jupyter notebooks\(.ipynb\)](#) or [markdowns\(.md\)](#)
→ will become sections of the documentation website
- **_config.yml**
- **_toc.yml**




Documentation



Documentation



← → ↻ stojijkovicvetana.github.io/napari-roidynamics/README.html# 🔍 📄 ⚙️ 📄 📄 📄



napari-roidynamics

[napari-roidynamics](#)

Application

Step-by-step guide

napari-roidynamics

This [napari](#) plugin was generated with [Cookiecutter](#) using [@napari's cookiecutter-napari-plugin](#) template.


license package or version not found pypi package or version not found python package or version not found tests passing

codecov unknown napari hub plugin not found

Plugin to use the python package 'roidynamics' within napari.

The [roidynamics](#) package provides interesting tools to analyze the dynamics of intensity in time-lapse images split into regions of specific geometries.

Here we have wrapped the roidynamics functions into an easy-to-use napari plugin, which enables to interactively setup points of interest, which are used to generate masks of different geometries. In the current version you can select three different geometries:



radial angular annular

We also allow to combine one of these three masks with any user-defined shape. This increases the flexibility of the plugin. However, if you wish to see a new shape please [file an issue](#).

These specific masks are then used to measure intensity on different channels of the time-lapse images. The intensity data can be viewed in napari or saved as plots and csv data along with the masks.

Contents

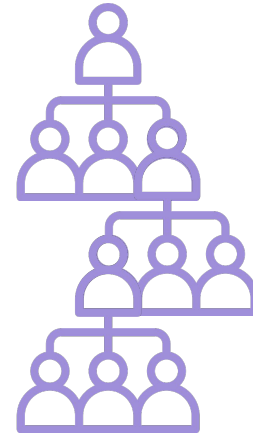
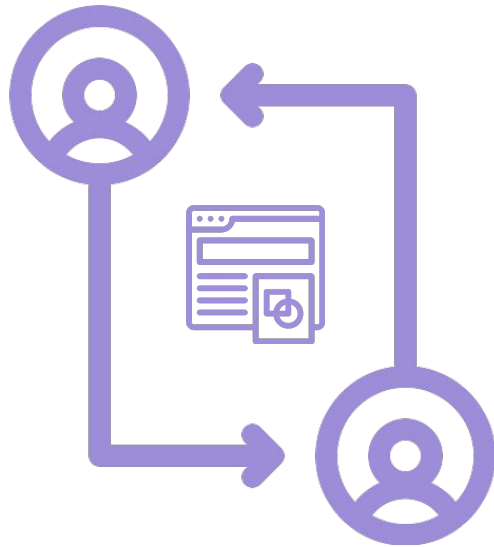
- Installation
 - First create a python environment containing napari
 - Install napari-roidynamics
 - Use napari-roidynamics
- Authors
- License
- Issues



Outreach - reach out!



Ask your Community!!!





Keep track of the feedback



Product ▾ Solutions ▾ Open Source ▾ Pricing

Search or jump to... / Sign in Sign up

guiwitz / napari-morphodynamics Public

Notifications Fork 1 Star 0

<> Code Issues 4 Pull requests Actions Projects Security Insights

is:open is:issue Labels 9 Milestones 0 New issue

Clear current search query, filters, and sorts

4 Open ✓ 5 Closed	Author ▾	Label ▾	Projects ▾	Milestones ▾	Assignee ▾	Sort ▾
Progress bar enhancement	#6 opened on Aug 29 by StojiljkovicVetAna					
Prompt to use set an export folder enhancement	#5 opened on Aug 29 by guiwitz					
Remove predict all frames enhancement	#4 opened on Aug 29 by guiwitz					
Save model by default enhancement	#3 opened on Aug 29 by guiwitz					