Fractal Analytics Platform Napari Plugins

Release 1.0.0

Friedrich Miescher Institute for Biomedical Research
University of Zurich

August 4, 2021

USER DOCUMENTATION

CHAPTER

ONE

NAPARI PLUGINS

1.1 About

This documentation is designed for end-users of the Fractal Analytics Platform's napari plugins. Topics for developers and system administrations are covered in separate documents.

1.2 Prerequisites

This section enlists prerequisites required by the napari plugins.

1.2.1 Python

All napari plugins are written in Python which requires a Python interpreter for execution. We recommend to install Python from one of the following sources...

Provider	URL	Description
Python Software Foundation	https://www.python.org/downloads/	Python interpreter (no additional packages)
Anaconda Inc.	https://docs.conda.io/en/latest/miniconda.html	Python interpreter + conda package manager (no additional packages)
Anaconda Inc.	https://docs.anaconda.com/anaconda/install/	Python interpreter + conda package manager + 100+ packages

While the Python Software Foundation provides a lightweight installer without any packages Anaconda Inc. provides a distribution with over 100 packages which, in return, needs more than 3Gb of disk space.

> Note: To avoid conflicts between napari plugins and other Python packages, it is recommended to create/use a new virtual environment. Please refer to "The Python Tutorial" for further information.

1.2.2 Git

By now, all napari plugins are available via private GitHub repositories. Hence, a git client is required for authentication and downloading/installing Python packages. The git command line interface (CLI) client can be downloaded from https://git-scm.com/download/. However, if you prefer a graphical user interface (GUI) you can find a full list of available GUI clients here.

If you are new to git please refer to the official documentation and the introduction videos.

> Note: The documentation assumes as you installed the CLI client and as you use a terminal to execute the git commands. This way, we can keep the documentation tidy as each GUI client is different in its behaviour and has a different feature set. Please refer to your GUI client's docunmentation on how to perform individual git commands via the graphical user interface.

1.2.3 Napari

Napari is a fast, interactive, multi-dimensional image viewer for Python.

Minimal requirements

Operating System	Python
Windows, Linux, MacOS	v3.7 and later

Napari can be installed via pip:

```
pip install napari[all]
```

Note: napari[all] includes the PyQt5 GUI backend which is required for the graphical user interface.

A full installation instruction is available via https://napari.org/.

1.3 Installation

All napari plugins are available via https://github.com/fractal-napari-plugins-collection.

Minimal requirements:

Operating System	Python
Windows, Linux, MacOS	v3.7 and later

The plugins can be installed via pip...

```
pip install git+https://github.com/fractal-napari-plugins-collection/<napari-plugin>.
↪git
```

... or by using git manually:

```
git clone https://github.com/fractal-napari-plugins-collection/<napari-plugin>
cd <napari-plugin>
pip install .
```

1.4 Usage

The following section contains usage instructions for the Napari plugins. For more details on how to use Napari itself please refer to the Napari tutorials.

Note: Please verify as all plugins were installed correctly. You should find them in Napari's "Plugins/List Installed Plugins..." menu.

1.4.1 PTIF Reader

Note: Reading fractions of an image (a.k.a. lazy loading) requires advanced data structures which Napari only supports recently. For a smooth user-experience it is recommended to activate the NAPARI_OCTREE flag **before** running Napari. You can do so from the command line **before** running napari.

Windows	Linux/Mac

`SET NAPARI_OCTREE=1`
`napari`
`SET NAPARI_OCTREE=0`
	`export NAPARI_OCTREE=1`
	`napari`
	`export NAPARI_OCTREE=0`

We recommend to create a batch/shell script with the above commands. Deactivating the octree feature at the end avoids conflicts when using Napari with non-multi-scale images later on. If you are not familiar with batch/shell scripts please refer to the following tutorials for Windows and Linux/Mac.

Given you have a multi-scale (pyramidal) TIF file with a PTIF extension, you can simple drag-n-drop the image into Napari or open the file from within Napari via the "File/Open File(s)…" menu.

Note: Fractal does not yet provide a tool to convert large-scale images into multi-scale images. For testing purposes, we recommend to install ImageMagick and convert e.g. the test001.tif image into a test001.ptif image using the following command:

```
convert.exe test001.tif -define tiff:tile-geometry=1024x1024 -compress jpeg␣
↪ptif:test001.ptif
```

Please note as ImageMagick is a very powerful tool which allows you to specify many more options via the command line. For more details please refer to the official reference.

1.4.2 XML Reader

The XML file format is an intermediate solution for reading a series of images. The reader expects a valid XML file structure with a list of "image" elements. Following a simple example:

```
<?xml version="1.0" ?>
<images>
    <image file=".\test001.tif"/>
    <image file=".\test002.tif"/>
    <image file=".\test003.tif"/>
</images>
```

Please note as all images must have the same dimensions. For more information/updates please refer to the following GitHub Issue.

Given you have a conform XML file, you can simple drag-n-drop the file into Napari or open the file from within Napari via the "File/Open File(s)…" menu.

1.4.3 ROI Reader

The ImageJ/FIJI ROI file format is an intermediate solution for reading shape information. You can create ROIs via ImageJ's/FIJI's ROI manager. By selecting/exporting multiple ROIs ImageJ/FIJI creates a ZIP archive which contains all selected ROIs as individual files.

> Note: The ROI reader plugin currently support the following ROI types: polygon, rect, oval, line, freeline and freehand.

Given a ROI file or ZIP file contianing multiple ROIs, you can simple drag-n-drop the file into Napari or open the file from within Napari via the "File/Open File(s)…" menu.

1.4.4 Error Handling

TBD