# napari-LF plugin

For viewing and processing light field images with the napari viewer

User Guide

Geneva Schlafly, Amitabh Verma, Grant Harris, Josué Page Vizcaino, Patrick La Riviere, Rudolf Oldenbourg
Marine Biological Laboratory, Woods Hole MA
The University of Chicago, Chicago IL
Technische Universität München, Germany
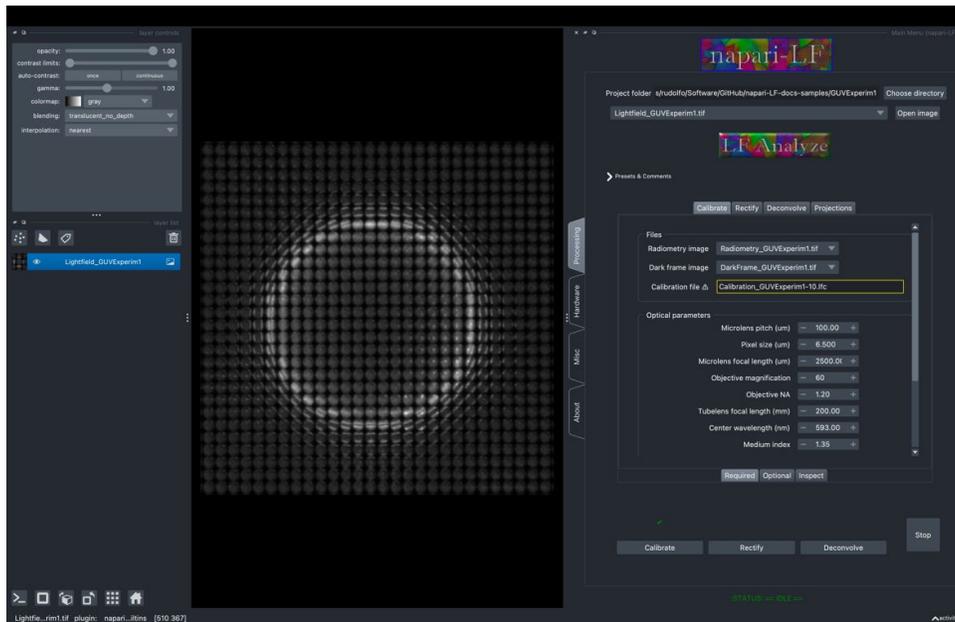
The above screenshot shows an interactive napari viewer with a light field image of a fluorescently labeled giant unilamellar vesicle (GUV). On the right side is the widget for the napari-LF plugin that can be used to view, manipulate, and process light field images.

napari-LF combines existing python code for processing light field images, typically recorded on a microscope. The widget gives access to two processing modes, an iterative deconvolution method called LF-Analyze and a Neural Net deconvolution based on LFMNet or other trained computational neural nets. LF-Analyze was developed by Michael Broxton in the laboratory of Marc Levoy at Stanford University between 2010 and 2013 (Broxton et al., 2013). The Neural Net approach was developed more recently by Josué Page Vizcaino during his PhD project at the Technical University in Munich and the University of Bern (Vizcaino et al, 2020). Both modes provide deconvolution routines for light field images of fluorescently labeled, transparent biological specimens or similar objects whose optical properties are characterized by a single parameter associated with voxels in object space (e.g. fluorescence density or absorption coefficient).

The Github repository for napari-LF is at: https://github.com/PolarizedLightFieldMicroscopy/napari-LF.git
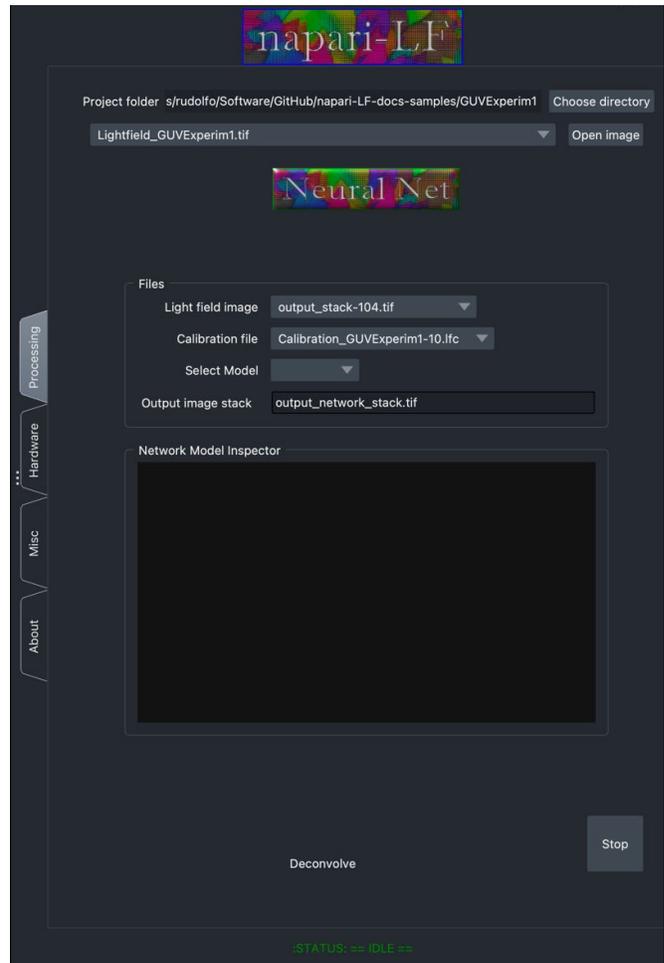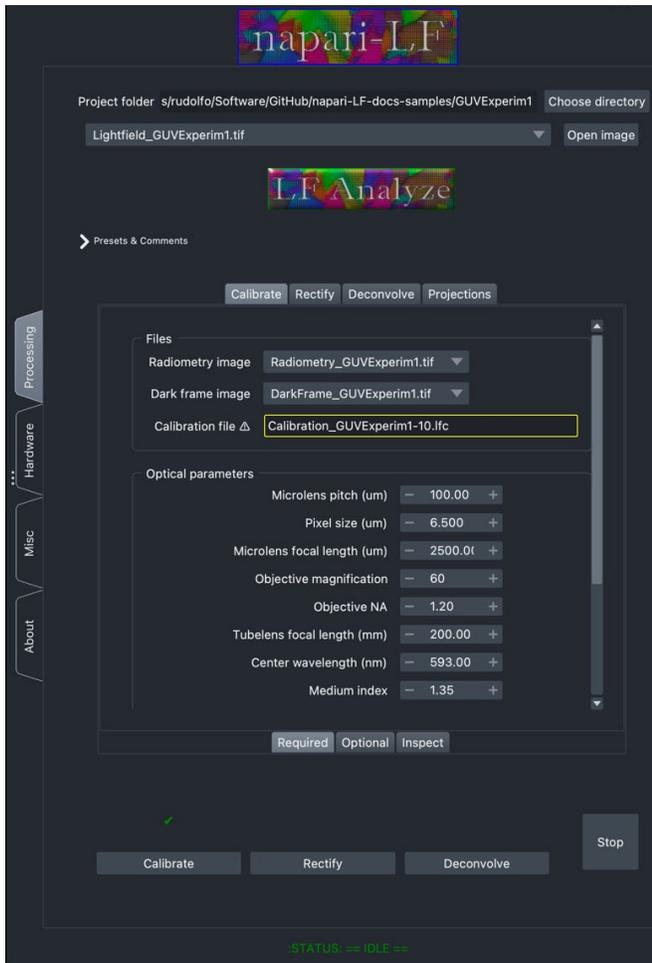
Two states of the napari-LF widget that expose the **LF-Analyze** and the **Neural Net** interface. Clicking on the **LF-Analyze** or **Neural Net** header switches the panels to the other mode.

The two processing modes are independent of each other, and each has its own panels. Given the required supporting data, such as a radiometry image or a trained neural net, both deconvolution approaches can be applied to the same light field images. Therefore, the two lines on top under the **napari-LF** header are used to set the common project directory and for opening image data into the napari Viewer.

In addition, on the left side of the widget are four vertical tabs called **Processing**, **Hardware**, **Misc**ellaneous, and **About**. The tabs **Hardware**, **Misc**, and **About** provide panels to expose hardware and other settings and information that are common to both processing modes (panels not shown).

The **Processing** tab, exposes the parameters and settings that are specific to each processing mode. As already mentioned, clicking on the LF-Analyze and Neural Net headings toggles between the two modes

The next page gives a rough overview of both processing modes, followed by specific instructions, first for LF-Analyze, followed by Neural Net (page 12).

References:

Broxton, M., Grosenick, L., Yang, S., Cohen, N., Andalman, A., Deisseroth, K. and Levoy, M. (2013). Wave optics theory and 3-D deconvolution for the light field microscope. Optics Express, 21(21):25418-25439. https://doi.org/10.1364/OE.21.025418

Vizcaino, J. P., Saltarin, F., Belyaev, Y., Lyck, R. and Favaro, P. 2020. Learning to Reconstruct Confocal Microscopy Stacks from Single Light Field Images. arXiv e-prints. https://arxiv.org/abs/2003.11004

LF Analyze

LF-Analyze provides three basic processes to Calibrate, Rectify, and Deconvolve light field images:

The **Calibrate** process generates a calibration file that represents the optical setup that was used to record the light field images. The same calibration file can be used to rectify and deconvolve all light field images that were recorded with the same optical setup, usually the same microscope and light field camera. The Calibrate process requires as input the radiometry image, dark frame, optical parameters, and volume parameters to generate the calibration file, which is subsequently used to rectify and deconvolve related light field images. The calibration file includes a point spread function (PSF) derived from the optical and volume parameters and is stored in the HDF5 file format with the extension .lfc.

The **Rectify** process uses the calibration file for an affine transformation to scale and rotate experimental light field images that were recorded with a light field camera whose microlens array is typically slightly rotated with respect to the pixel array of the area detector and whose pixel pitch is not commensurate with the microlens pitch. After rectification, the rectified light field has the same integer number of pixels behind each microlens. When the Deconvolve process is called for an experimental light field image, rectifying the light field image is automatically applied before the iterative deconvolution begins. However, the rectified light field image is not saved and is not available for viewing. Therefore, by pushing the Rectify button, only the rectification step is invoked, and the rectified light field image is saved to the project directory.

The **Deconvolve** process uses the calibration file with PSF and a wave optics model to iteratively deconvolve a light field image into a stack of optical sections.

The **Parameter** panels, located in the lower half of the napari-LF widget, allows the user to specify settings for the reconstruction process. Once the appropriate parameters are selected, pushing the Calibrate button followed by the Deconvolve button will generate the deconvolved data as a stack of optical sections.

Neural Net

As an alternative to LF-Analyze, a suitably trained Neural Net can be used to generate deconvolved data as a stack of optical sections.
The widget relies on a trained neural net file that is stored in the project folder and can be selected from a dropdown menu. The widget does not provide means to generate a trained neural net file. However, the code distribution of the napari-LF plugin includes a Jupyter notebook that guides the user through creating a trained network based on algorithms provided by LFMNet https://github.com/pvjosue/LFMNet or VCDNet https://github.com/xinDW/VCD-Net. The user will need to provide an adequate number of experimental or simulated light field images paired with their ground truth data, such as stacks of confocal optical sections or volume data used for simulating the light field images.

The next several pages provide detailed information on LF-Analyze processing steps, followed by similar information on the Neural Net processing steps.

Quickstart

1. Install the napari-LF plugin into your napari environment, as described below under **Installation**.

2. From the napari Plugins menu, select the napari-LF plugin to install its widget into the napari viewer
3. Select the project folder containing the light field, radiometry, dark frame images and/or trained neural net file.
4. For iterative
5. Calibration
    o In the parameters panel, navigate to **Calibrate**, **Required** (top tab **Calibrate**, bottom tab **Required**), which is the default selection.
    o Select **radiometry** and **dark frame** images from pull down menus.
    o Write the name of the **calibration file** that will be produced by the calibration step, e.g. calibration.lfc.
    o Enter the appropriate **optical parameters** according to your microscope and sample
    o Enter the v**olume parameters** you would like for your 3D reconstruction.
    o Push the **Calibrate** button at the bottom of the panel.
6. Deconvolution
    o In the parameters panel, navigate to **Deconvolve, Required**.
    o Select **light field** image and **calibration file** from pull down menus.
    o Write the name of the **output image stack** you would like to produce, e.g. output_stack.tif.
    o Push the **Deconvolve button**. 3D focal stack reconstruction will display in the napari viewer and be saved in your project image folder.
1. Install the napari-LF plugin into your napari environment, as described below under Installation.
2. From the napari Plugins menu, select the napari-LF plugin to install its widget into the napari viewer.

LF Analyze

1. Near the top of the widget, select your project folder containing the following images: light field, radiometry, and dark frame.
2. Calibration
    a. In the processing panel, navigate to **Calibrate**, **Required** (top tab **Calibrate**, bottom tab **Required**), which is the default selection.
    b. Select **radiometry** and **dark frame** images from pull down menus.
    c. Write the name of the **calibration file** you would like to produce, e.g. calibration.lfc.
    d. Enter the appropriate **optical parameters** according to your microscope and sample material.
    e. Enter the **volume parameters** you would like for your 3D reconstuction.
    f. Push the **Calibrate** button.
3. Deconvolution
    a. In the processing panel, navigate to **Deconvolve, Required**.
    b. Select light field image and calibration file from pull down menus.
    c. Write the name of the output image stack you would like to produce, e.g. output_stack.tif.
    d. Push the Deconvolve button. The 3D focal stack reconstruction will display in the napari viewer and be saved in your original project folder.

Neural Net

1. Click on the **LF Analyze** logo to toggle to the **Neural Net** mode.
2. Near the top of the widget, select your project folder containing the light field image and the trained neural net. If you do not already have a trained model, you can train a model using this [Jupyter notebook](#).
3. In the processing panel, select your light field image and neural net model.

4. Write the name of the **output image stack** you would like to produce, e.g. output_stack.tif.
5. Push the **Deconvolve** button. The 3D focal stack reconstruction will display in the napari viewer and be saved in your original project folder.

Getting Help

For details about each parameter, hover over each parameter textbox to read the tooltip description. For additional information about the reconstruction process, see our documentation on GitHub.

Installation

After you have napari installed, you can one of the methods below to install **napari-LF**.

Method 1: You can install **napari-LF** via [pip]:

pip install napari-LF

Method 2: Use the napari plugin menu.
1. Open napari from the command line:
   napari

2. From the napari menu, select **Plugins > Install/uninstall Packages**.
3. Either (a) scroll through the list of available plugins to find **napari-LF**, or (b) drag and drop a down-loaded **napari-LF** directory into the bottom bar.
4. Select **Install** to install the light field plugin.

Method 3: Install the latest development version from the command line.

pip install git+https://github.com/PolarizedLightFieldMicroscopy/napari-LF.git

Lastly, to access the installed plugin, open napari from the command line:

napari

From the napari menu, select **Plugins > Main Menu (napari-LF)**. Note that you may need to close and reo-pen napari for the `napari-LF` to appear.

Installation for developers
Create a virtual environment from the command line for napari with the python libraries necessary for the light field plugin:

conda create --name napari-lf python==3.9
conda activate napari-lf

Clone the github repository:

```
conda install git
git clone https://github.com/PolarizedLightFieldMicroscopy/napari-LF.git
cd napari-LF
pip install -e .
```

Parameter descriptions

The following pages provide screenshots of the various panels of the widget and comment on the data files, parameters, and functions than can be invoked.

Overview of Widget GUI
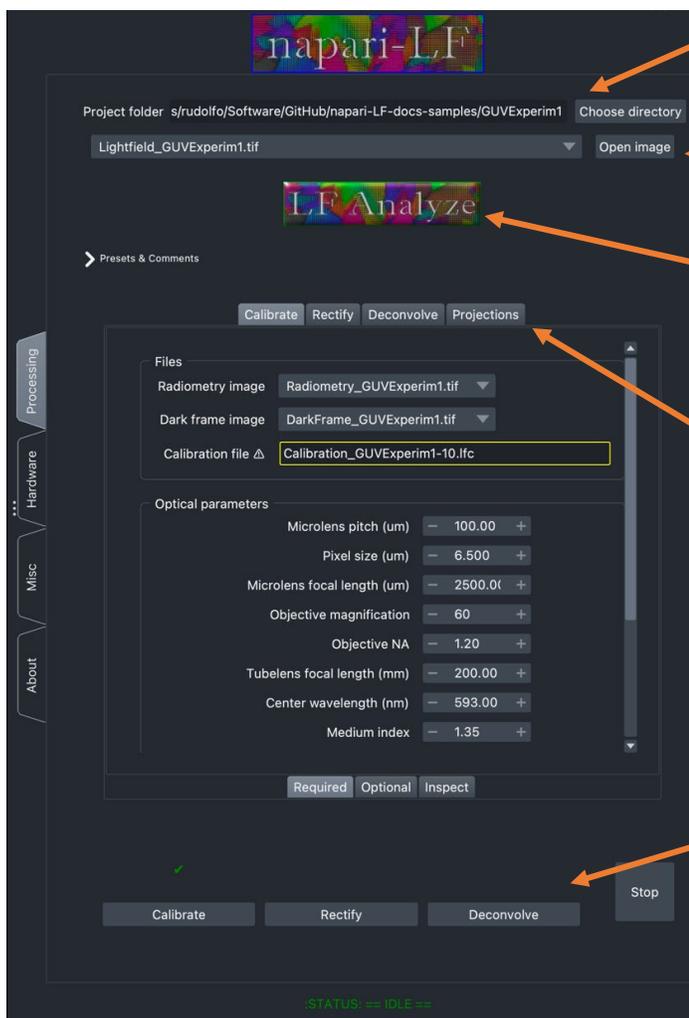
Choose directory, which contains related images and their processed data. All pull-down menus in the widget will populate with file names from this directory.

Select **images in drop-down menu** and **Open Image** in the viewer

Processing mode **LF-Analyze**. Click on the heading to switch between **LF-Analyze** and **Neural Net**

Processing steps for LF-Analyze, namely **Calibrate**, **Rectify**, **Deconvolve**, and **Projections**. The associated panels are used to display, change, and enter parameters used for data processing with the LFAnalyze code. The parameters are explained in more detail on the following pages of this guide.

Calibrate, Rectify, Deconvolve buttons initiates data processing using parameters defined in related panels below.

LF Analyze, Calibrate Required panel

The Calibrate panel defines the required and optional parameters used to generate the calibration file. The calibration file characterizes the microscope setup and includes the instruments point spread function that is used for deconvolving a light field image recorded with the setup. In addition, the calibration file is specific to the number of z-Slices specified as a parameter, their distance in μm, and the Supersample number.

The Supersample number is an integer that specifies the smallest voxel size used for the deconvolution. The smallest voxel size is equal to the microlens pitch divided by the objective lens magnification and by the Supersample integer. Note, however, that the actual deconvolved voxel size is also a function of its distance from the nominal focal plane. The calibration file has the extension .lfc and the HDF5 file format.
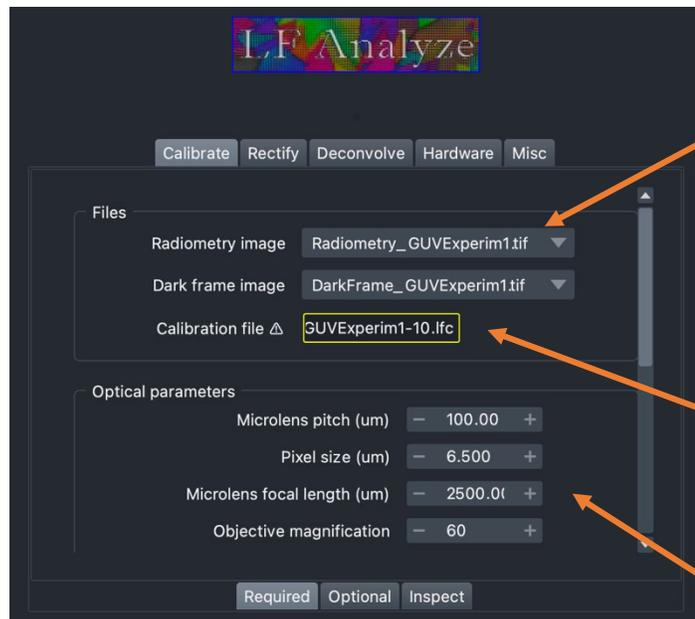

Choose from drop-down menus to select
Radiometry image: Specify a radiometry frame to use for radiometric correction. If no frame is specified, then no radiometric correction is carried out.
Dark frame image: Specify a dark frame image to subtract from the input light-field before processing (This makes radiometric calibration more accurate).

Enter **Calibration file** name. When a file with this name already exists, the field is outlined yellow and the Calibrate button has a green check mark. A new calibration will override the file.

Optical parameters relate to the optical setup used, including the microlens array in front of the area detector, and the microscope objective lens.
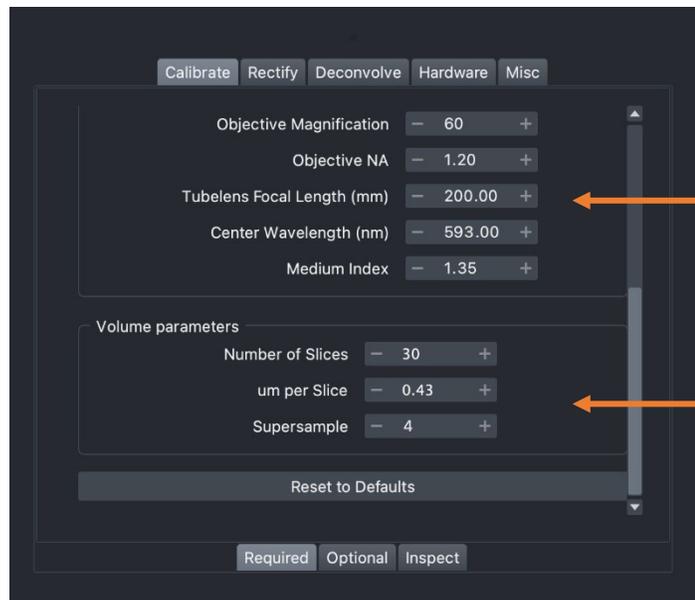
More **Optical parameters**, including the tube lens of the microscope, center wavelength of the light recorded, and the refractive index of the medium embedding the sample structure.

The **Volume parameters** specify the number of z-Slices, their distance in micrometer (um) and the Supersample used. Supersample results in a higher resolution reconstruction up to a point, and interpolation after that point.

Calibrate Optional panel
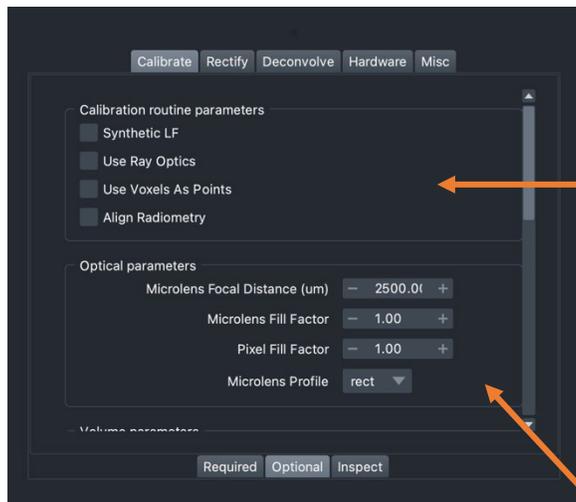
Calibration routine parameters
Synthetic LF: Use this option to create a synthetic light field, i.e. with no calibration image
Use ray optics: Use the less accurate ray optics model, rather than wave optics model
Use voxels as points: Treat voxels as ideal points. This turns off numerical integration that gives the voxel spatial extent (which can be important for anti-aliasing.
**Align radiometry**: Align the radiometry image to the geometric calibration image. Use this option when the radiometry frame was "bumped" before imaging begins.
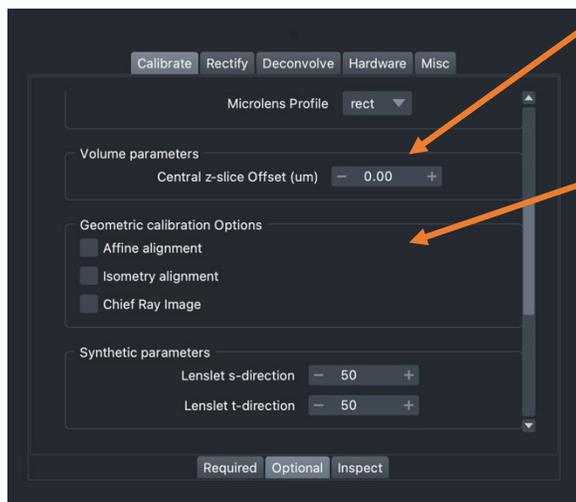
Additional **Optical** and **Volume parameters**

Geometric calibration options
Affine alignment: Use affine warp for correcting geometric distortion (default is cubic).
Isometry alignment: Use isometry warp for correcting geometric distortion (default is cubic).
Chief ray image: Use this flag to indicate that the calibration frame is a chief ray image.
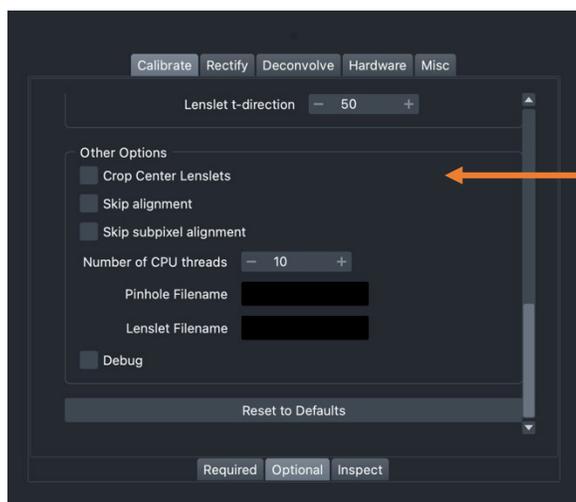
Other Options
Crop center lenslets: For severe aperture vignetting (high NA objectives), use only center lenslets for calibration, and extrapolate outwards.
Skip alignment: Skip the alignment step during geometric calibration. Useful if you are working with an already-rectified light field or a synthetic light field.
Skip subpixel alignment: Skip subpixel alignment for determining lenslet centers.
**Number of CPU threads**: Set the number of CPU threads to use when generating the raydb.
Pinhole filename: After calibrating, save the rectified light field as a rectified sub-aperture image.
Lenslet filename: After calibrating, save the rectified light field as a rectified lenslet image.

Calibrate Inspect panel

The Calibrate Inspect panel prints attributes of the Calibration File selected and the Datasets and Groups of its HDF5 file format
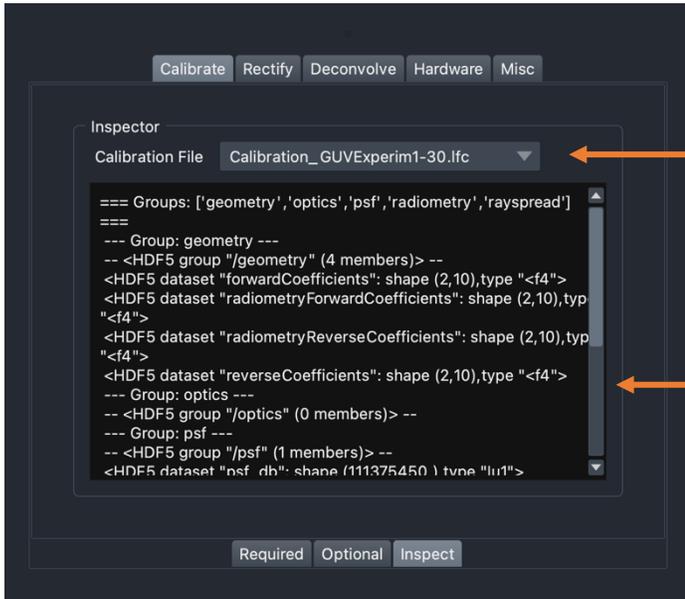
Inspector
Select **Calibration file** to inspect.

Attributes of the HDF5 file format.
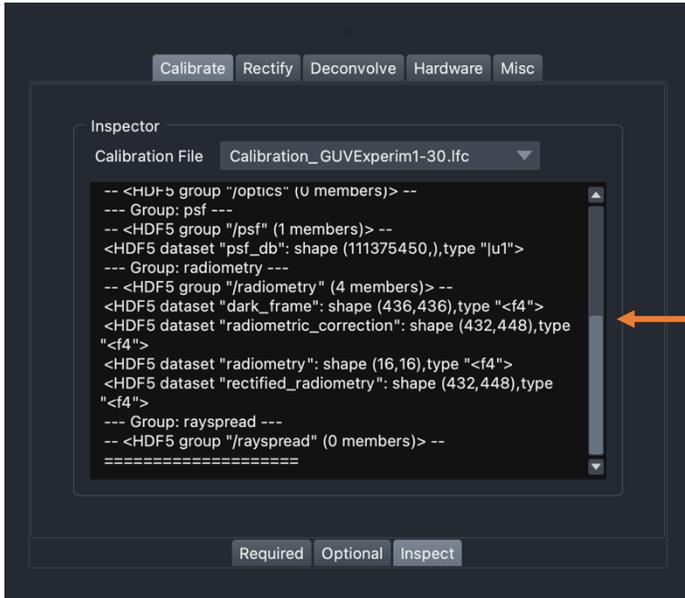
Attributes of the HDF5 file format.

Rectify Required and Optional panels

The calibration file includes instructions for rectifying a light field that was recorded with the microscope setup represented in the calibration file. A rectified light field contains the same information as the original light field, but on a grid which puts an integer number of pixels behind each microlens and generates a light field with an integer number of microlenses in the horizontal and vertical direction. Every deconvolution requires a rectified light field and therefore, rectification is an integral step in the deconvolution process implemented in napari-LF. In addition, the Rectify panel lets one save the rectified light field as an image file for visual inspection.
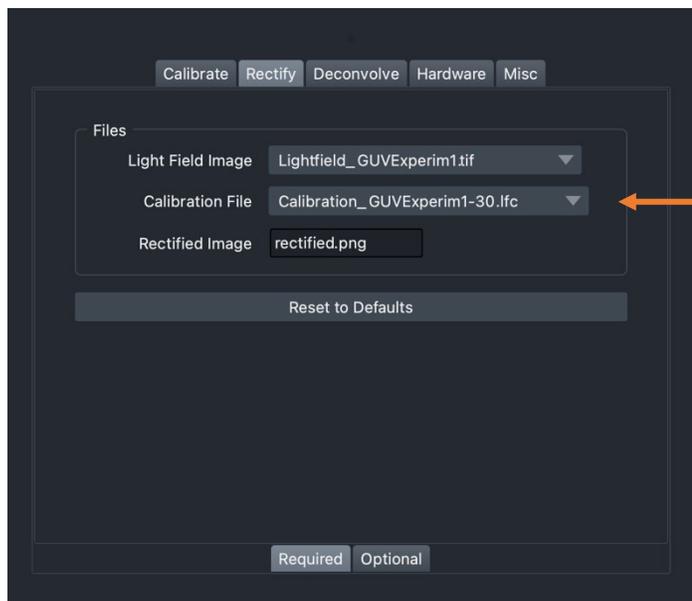
Files:
Choose the original **Light field image** and the **Calibration file** to be used for generating the rectified light field that will be saved using the file name entered under **Rectified image**.
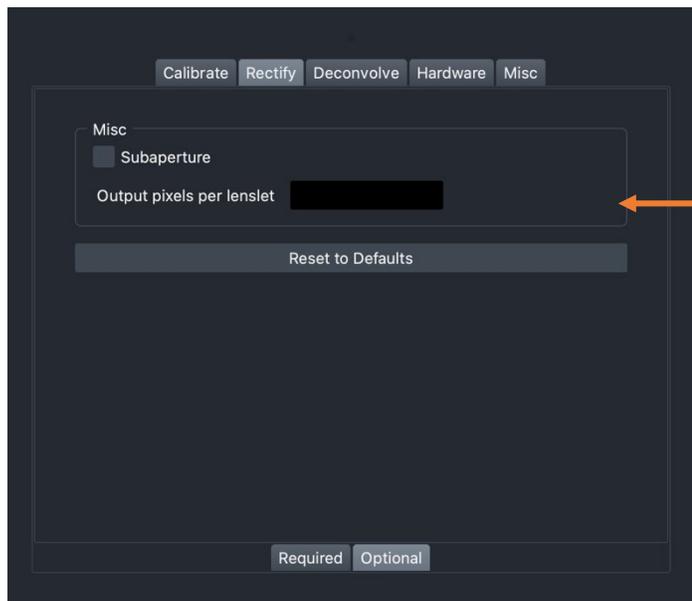
Misc
Subaperture: Save out the light field image as tiled subapertures or pinhole views.
Output pixels per lenslet: Specify the number of pixels per lenslet in the output image.

Deconvolve Required panel
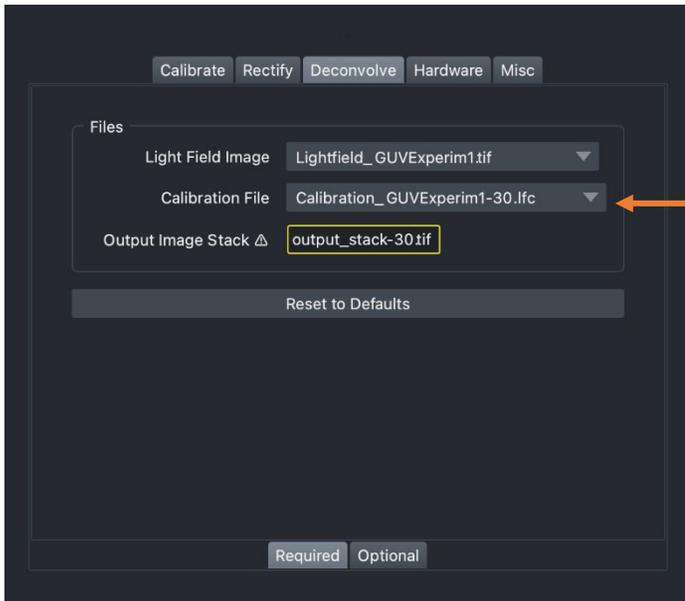
Files
From the drop-down menus, choose the **Light field image** and the **Calibration file** to be used for generating the deconvolved stack that will be saved using the file name entered under **Output image stack**.

Deconvolve Optional panel
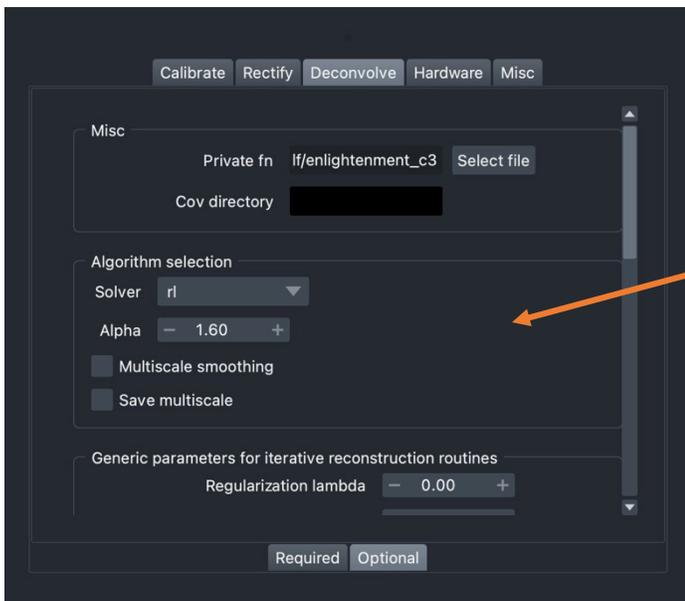
Algorithm selection
Available reconstruction methods are:
amp: Approximate Message Passing with optional multiscale denoising**.**
admm_huber: Alternating Direction Method of Multipliers with Huber loss**.**
admm_tv: Alternating Direction Method of Multipliers with TV penalty.
cg: Conjugate Gradient.
direct: Direct method with Cholesky factorization.
lsqr: Least Squares QR.
kspace: k-space deconvolution.
sirt: Simultaneous Iterative Reconstruction Technique.
mrnsd: MRNSD and
rl: Richardson-Lucy.
Default is currently **rl**.

Alpha: Relaxation parameter for SIRT-based iterative reconstruction.
Multiscale smoothing: Multiscale regularization option for AMP reconstruction.
Save multiscale: Save multilevel decomposition of data.

Deconvolve Optional panel
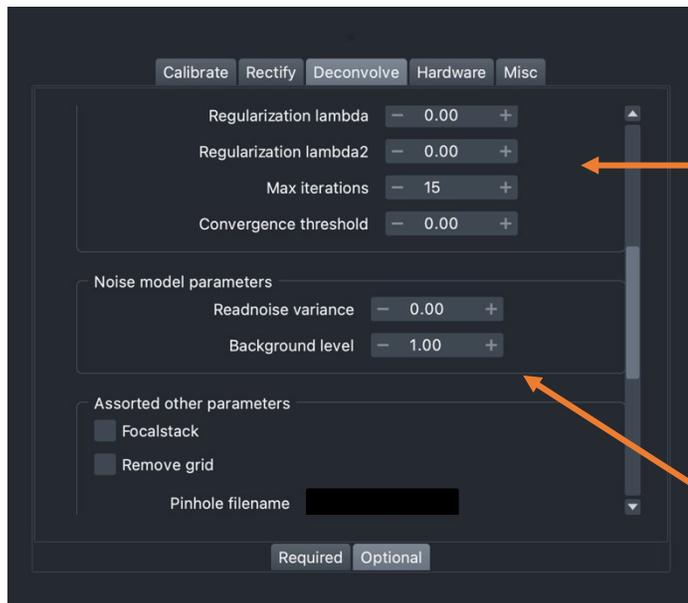
Generic parameters for iterative reconstruction routines
Regularization lambda: Regul. coefficient whose behavior varies by reconstruction algorithm.
Regularization lambda2: Additional coefficient. Behavior varies by algorithm, and not all algorithms use two regularization coefficients.
Max iterations: Maximum number of iterations for SIRT-based reconstruction.
Convergence threshold: Convergence criteria threshold, d/dt (MSE). Try 5e-5 for SIRT, 1e-2 for TV.

Noise model parameters
Read noise variance: Set the variance of the measured camera read noise.
Background level: Set the measured background level of the image.
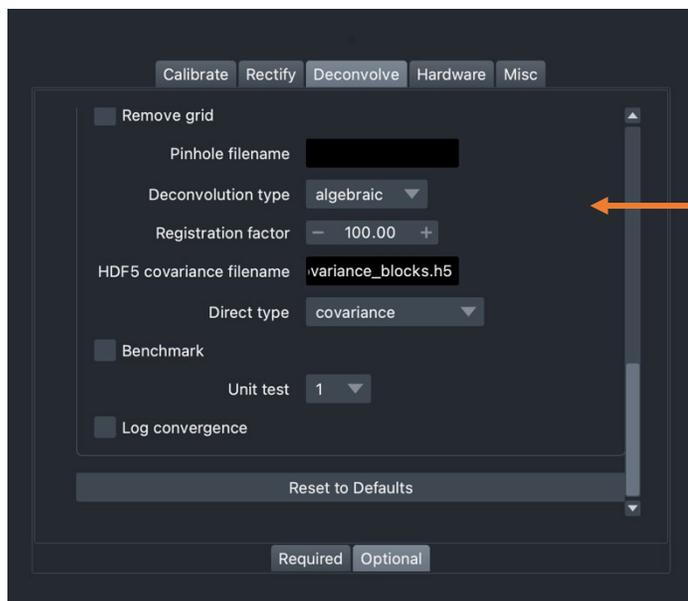
Assorted other parameters
Focal stack: Turn off deconvolution and simply save a focal stack to disk.
Remove grid: Remove grid artifacts in light field image using spectral median filter.
Pinhole filename: After deconvolution, save out a deconvolved light field sub-aperture image.
**Deconvolution type**: Choose deconvolution method, one of algebraic, direct, admm
**Registration factor**: Regularization parameter used in ADMM.
HDF5 covariance filename: Specify filename
**Log convergence**: For logging convergence details
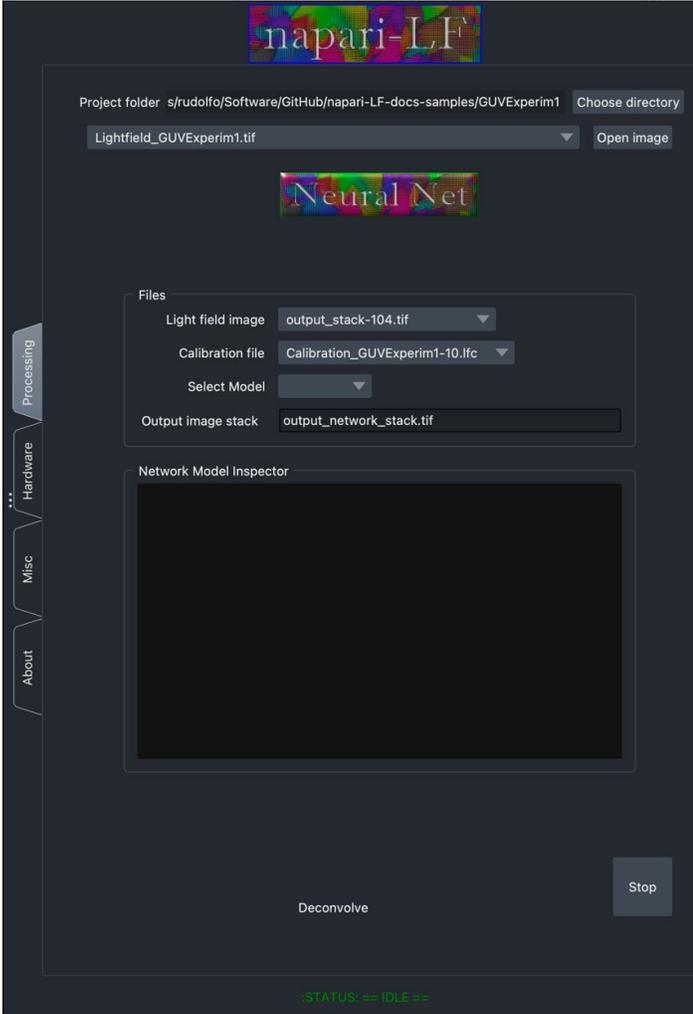
Hardware panel
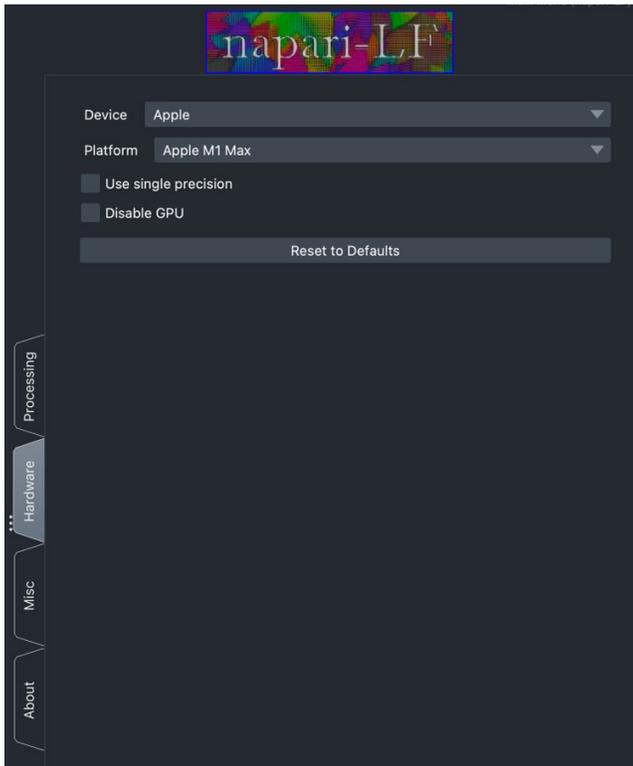
Select device: Force Ifdeconvolve to use a specific GPU on your system. If not supplied, this will default to $USE_GPU_ID or 0; if --gpu-id is not supplied, it will default to the value of the USE_GPU_ID environment variable, if set, or 0 otherwise

Select Platform: Force Ifdeconvolve to use a specific OpenCL Platform on your system.

Use single precision: Use single precision float instead of double.

Disable GPU: Disable GPU deconvolution and use software implementation instead.

Miscellaneous panel
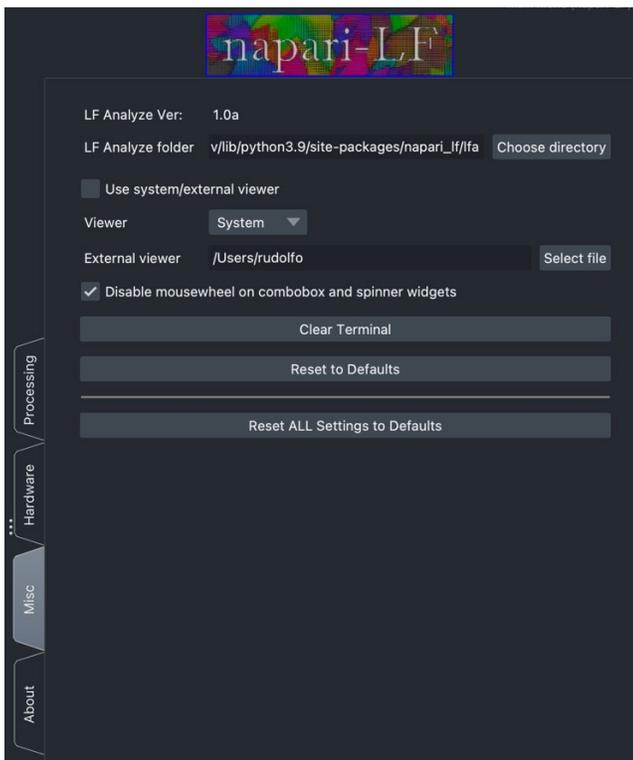
Use system/external viewers: Use system/external viewer for displaying images instead.

System viewer: Chose your viewer, either System: OS default, or External: User selects path under **External viewer**.

Disable mousewheel on combobox and spinner widgets: Disable mousewheel on combobox and spinner widgets to avoid accidentally changing values.