

---

# **auTomo Documentation**

***Release 0.2***

**Argonne National Laboratory**

**Nov 01, 2019**



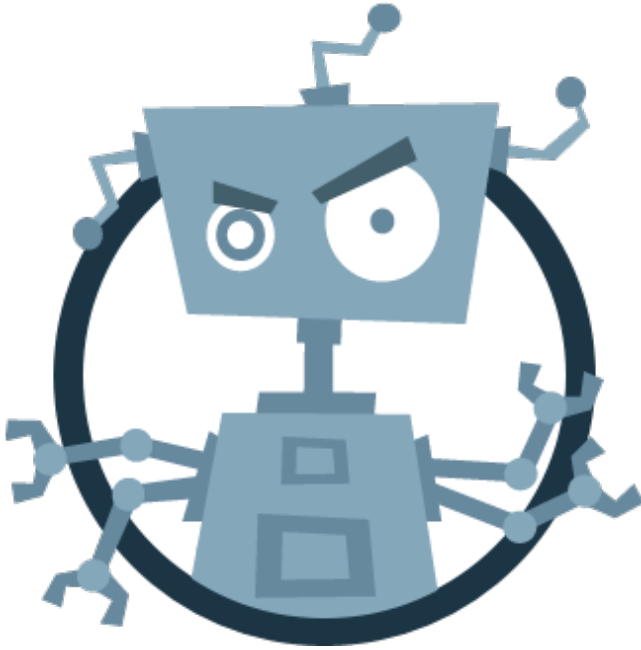
---

## Contents

---

<b>1</b>	<b>Features</b>	<b>3</b>
<b>2</b>	<b>Contribute</b>	<b>5</b>
<b>3</b>	<b>Content</b>	<b>7</b>





auTomo provides automation to several tomography data analysis tasks.



# CHAPTER 1

---

## Features

---

- Automatically organizes the directory, creates an individual folder for HDF5 files associated with each sample
- Comes with preset scripts for tomography processing
- Automatically processes single field-of-view and 360-degree samples
- Generates bash commands that calls scripts for mosaic tomography processing (powered by [Tomosaic](#)).





## CHAPTER 2

---

### Contribute

---

- Documentation: <https://github.com/decarlof/automo/tree/master/doc>
- Issue Tracker: <https://github.com/decarlof/automo/docs/issues>
- Source Code: <https://github.com/decarlof/automo/automo>



### 3.1 Install directions

This section covers the basics of how to download and install **auTomo**.

#### 3.1.1 Installing from source (recommended)

Installing from source can be done easily by running the setup script:

```
git clone https://github.com/decarlof/automo.git automo
cd automo
python setup.py install
```

The script will ask whether you want to add a line in your `.bashrc` file so that bash will automatically add the `macros` folder in the Automo source directory to your `$PATH` variable. This is necessary if you would like to use the feature of calling Automo script directly from bash command line. If you prefer to have the scripts, the configuration file (`automo.ini`), and the parameter setting file (`automo_params.csv`) somewhere else, you need to manually copy them there. For example, to move the files to `~/ .automo`, follow the above commands by

```
mkdir ~/ .automo
cp macros ~/ .automo
export PATH=~/ .automo:$PATH
```

Subsequently, add the last line `export PATH=~/ .automo:$PATH` to your `~/ .bashrc`.

#### 3.1.2 Installing from conda

auTomo has an conda install script that does all the job. Please follow the script:

```
git clone https://github.com/decarlof/automo.git automo
cd automo
conda build .
conda install --use-local automo
```

## 3.2 Overview

Automo creates and/or executes processing operation commands based on the configuration file (default to be `automo.ini`) and the parameter setting file (default to be `automo_params.csv`). You can specify the path to both files when running `automo_process_dir` and `automo_create_commands`.

### 3.2.1 Configuration file

`automo.ini` contains the settings that define the behavior of Automo as well as the processing procedures to be taken for different types of tomography data. Below is the content of a sample file (also available in the `macros` folder):

```
[settings]
default_h5_fname      = data.h5
[robos]
tomo_180              = automo_preview, automo_center, automo_search_center, automo_
↳recon
tomo_360              = automo_preview_360, automo_center_360, automo_search_center, automo_
↳recon_360
tomo_saic              = automo_tomosaic_create_meta, automo_tomosaic_reorg, automo_tomosaic_
↳preview, automo_tomosaic_center, automo_tomosaic_recon
std                   = preview, center, recon
[robos_move]
tomo_180              = new_folder
tomo_360              = new_folder
tomo_saic              = existing_folder
std                   = new_folder
[robos_rename]
tomo_180              = False
tomo_360              = False
tomo_saic              = False
std                   = True
```

The `robos` section specifies the workflow that Automo should adopt for each of the three types of tomography data (the `std` entry is for the exception case where Automo fails to recognize the data files as any of the three categories):

- `tomo_180`: single field-of-view tomography
- `tomo_360`: off-centered 360-degree tomography
- `tomo_saic`: mosaic tomography using the Tomosaic acquisition scheme

The data type is automatically detected according to the naming pattern of the HDF5 files. It is thus necessary to keep a consistent naming style as listed in the table below:

Data type	Naming pattern
<code>tomo_180</code>	<code>*_180_*.h5</code> (must contain <code>_180_</code> )
<code>tomo_360</code>	<code>*_360_*.h5</code> (must contain <code>_360_</code> )
<code>tomo_saic</code>	<code>*_y_?_x_?_*.h5</code> (indices start from 0)

AutoTomo will execute commands (for `tomo_180` and `tomo_360`) or create bash command scripts (for `tomosaic`) following the order at which operations are listed. The name of the operations, such as `automo_preview`, or `automo_tomosaic_reorg`, must match the name of the processing scripts contained in the `macros` directory.

When running `automo_process_dir` in the folder containing all HDF5 data files collected in a beamtime, the `robos_move` section tells AutoTomo whether to move a file into a new directory named exactly after the basename of the file (`new_folder`) or to move it into an existing folder named after the prefix of the filename (`existing_folder`; useful for `tomosaic`). It is recommended to keep the default settings as shown in the sample file above.

The `robos_rename` section specifies whether AutoTomo should rename the moved HDF5 file according to the setting in `settings/default_h5_fname`. Keep this setting as `False` for `tomosaic` unless specifically needed.

## 3.2.2 Parameter file

`automo_params.csv` contains the parameter settings for all operations. The parameters of the same operation must stay in the same line with names and values separated by a semicolon (;). The file should follow the format of

```
operation_name; param_1_name; param_1_value; param_2_name; param_2_value; ...
```

If an optional parameter is not provided in the `csv` file, it will use the default value. Please refer to the docstring of an operation script (for example, `automo_tomosaic_center --help`) to get help on the names and meanings of all its parameters. As in the case of the `ini` file, `operation_name`'s must match operation scripts contained in the `macros` folder.

## 3.3 Demonstration

### 3.3.1 Starting processing

In the directory containing all collected HDF5 data files, run

```
automo_process_dir
```

You can use the `--ini_file` and `--params_file` flags to specify the location and filenames of the configuration and parameter files. Otherwise, AutoTomo will use the files in the `macros` folder of its source directory. After moving each file to the proper folder, AutoTomo does the following:

- If the file is of type `tomo_180` or `tomo_360`, AutoTomo runs operation scripts according to the sequence and parameters specified in the `ini` and `csv` files. After the operations finish, a bash script with executed commands will be created in the folder.
- If the file is of type `tomosaic`, AutoTomo will not execute the commands, but will create a bash scripts of commands for the user to execute manually in bash.

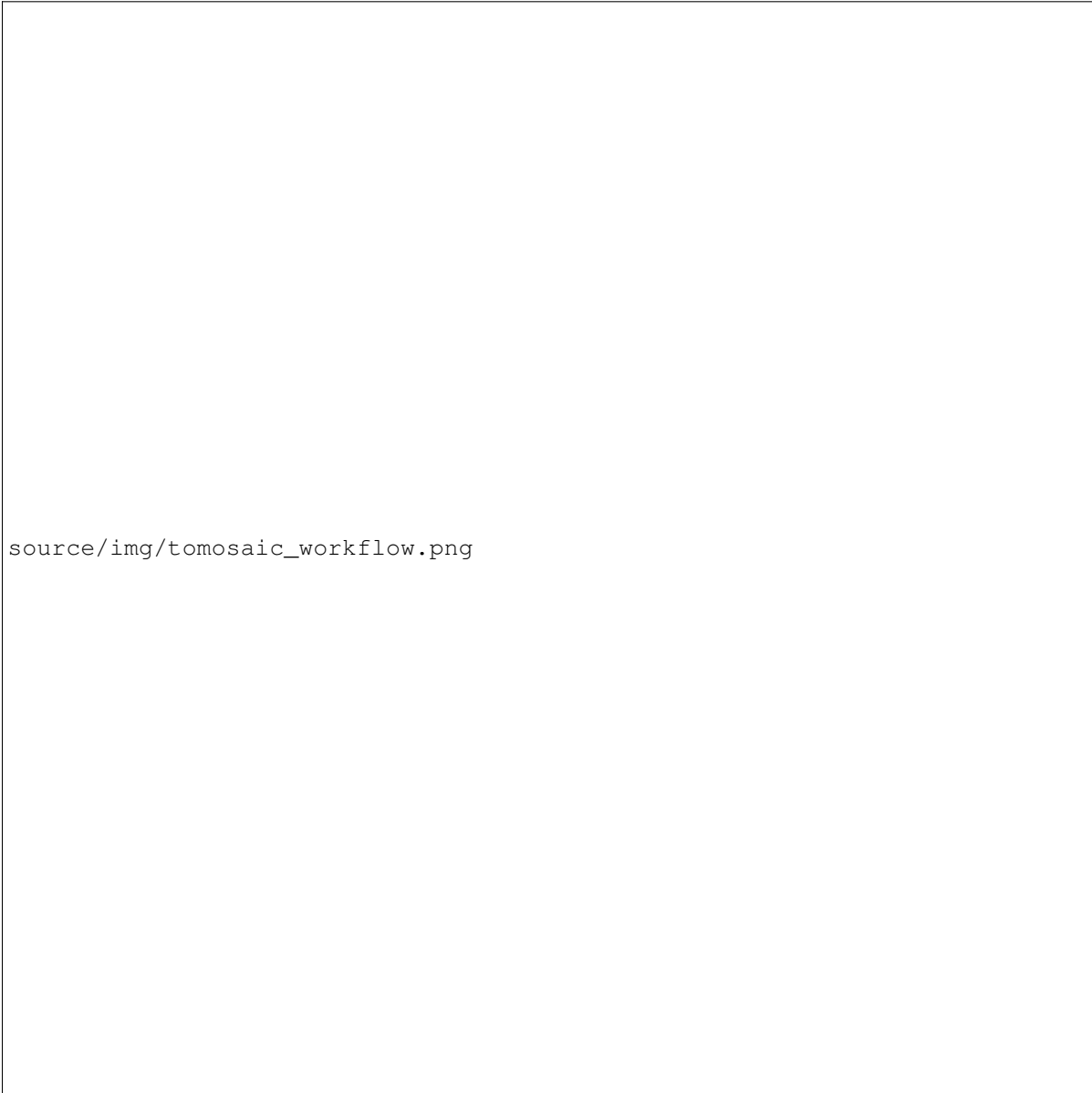
If the directory has already been organized either by hand or by the beamline control software, the `automo_process_dir` can be skipped, and you may want to just generate the bash commands used to run processing operations. For this, type

```
automo_create_commands --type <tomo_180, tomo_360, or tomosaic>
```

This tells AutoTomo to generate a bash script file at the current folder, with the operations and parameters provided in the specified (or default) configuration file and parameter file.

### 3.3.2 Tomosaic workflow

This section talks about the workflow of Tomosaic-type data processing using AutoTomo. Tomosaic allows mosaic tomography reconstruction to be performed either after merging all HDF5 files into a single file with stitching and blending in both x- and y-direction (whole-block mode, WBM), or directly by stitching sinograms in the x-direction (single-slice mode, SSM). While image blending and sub-pixel shifting is currently only supported in WBM, SSM provides a light-weight reconstruction option which demands less memory and hard drive space. The major workflow is summarized in Fig. 2 of the [Tomosaic paper](#).



The general procedure of Tomosaic processing using AutoTomo is described below. Again, you can utilize `autoTomo_create_commands` to generate the bash commands with a series of operations.

1. In the folder containing the data files of all scanned tiles, run `autoTomo_tomosaic_creates_meta` with position offset in x and y specified using the `--x_shift` and `--y_shift` flags. This creates a file called `mosaic_meta.py` in the current directory.
2. Run `autoTomo_tomosaic_reorg --ds 1`. This creates a folder called `data_raw_1x` and moves all

HDF5 files in it. If you want downsampled copies of the data files, modify the `--ds` flag, with different down-sampling levels separated by commas (,). For example, `--ds 1,2,4`. Files downsampled for 2 times will be saved in folder `data_raw_2x`, and same pattern for the rest.

3. Run `automo_tomosaic_preview --frame 0 --pano auto` to create separate projections and flat fields at 0 degree, as well as stitched panoramas at 0 and 180 degrees.
4. (Optional) Run `automo_tomosaic_register` to realign the tiles. This creates a file called `shifts.txt` with each line of it following the format of

```
tile_index_y tile_index_x y_shift_to_right_tile x_shift_to_right_tile y_shift_to_
↳bottom_tile x_shift_to_bottom_tile
```

If this step is skipped (and thus `shifts.txt` doesn't exist), Automo will use the shift values provided in the `mosaic_meta.py` for subsequent procedures.

5. Run `automo_tomosaic_center --center_st auto --center_end auto --row_st 0 --row_end auto --mode discrete`. Automo try to find the rotation centers for specified rows of tiles. If `--center_st` or `--center_end` is set to `auto`, Automo will guess the range for center search using phase correlation. If `--method` is set to `pc`, then the phase correlation results will be final. Otherwise, the range for center searching using manual, entropy, or `vo` will be set to be  $\pm 5$  of the phase correlation guess. For manual and entropy, trial reconstructions will be saved as `center/<row_number>/<center_value>.tiff`.

If `--method` is anything other than `manual`, or if phase correlation is used to provide the initial guess, then Automo will create a file named `center_pos.txt` at the current directory, with each line indicating the row number and the center position at that row. Otherwise, you need to manually create this file in order to the reconstruction script to read.

Since setting `--method` to `manual` or `entropy` lets Aumoto create reconstructions, it is also a way to generate reconstruction previews.

6. (Only needed for WBM) Run `automo_tomosaic_merge` to fuse the HDF5 files into a single one. The default name and location of the created file is `fulldata_flatcorr_1x/fulldata_flatcorr_1x.h5`.
7. Run `automo_tomosaic_recon` for final reconstruction. Make sure the `--mode` flag is set correctly: use `merged` for WBM, and `discrete` for SSM. SSM is recommended if there is only one row of tiles.

By using `automo_create_commands --type tomosaic`, a bash script following the above procedureS (assuming WBM) can be generated:

```
automo_tomosaic_create_meta --x_shift 1000 --y_shift 1000
automo_tomosaic_reorg --ds 1
automo_tomosaic_preview --frame 0 --pano auto
automo_tomosaic_register
automo_tomosaic_center --center_st auto --center_end auto --row_st 0 --row_end 1 --
↳mode discrete --method manual
automo_tomosaic_merge
automo_tomosaic_recon --mode merged --chunk_size 5
```

Correspondingly, `automo.ini` should contain the following in the `robos` section:

```
tomosaic = automo_tomosaic_create_meta, automo_tomosaic_reorg, automo_tomosaic_
↳preview, automo_tomosaic_register, automo_tomosaic_center, automo_tomosaic_merge,
↳automo_tomosaic_recon
```

`automo_params.csv` should have the following settings:

```
automo_tomosaic_reorg; ds; 1
automo_tomosaic_create_meta; x_shift; 1000; y_shift; 1000
automo_tomosaic_preview; frame; 0; pano; auto
automo_tomosaic_register
automo_tomosaic_merge
automo_tomosaic_center; center_st; auto; center_end; auto; row_st; 0; row_end; 1; ↵
↵mode; discrete
automo_tomosaic_recon; mode; merged; chunk_size; 5
```

### 3.3.3 Parallelization

Tomosaic and AutoTomo supports parallel computation using `Mpi4Py`. To enable parallelization, make sure you have `Mpi4py` installed, and then simply add `mpirun -n <num_threads>` in front of each command.

## 3.4 Credits

### 3.4.1 Citations

We kindly request that you cite the following article [cite:01] if you use auTomo and/or Tomosaic.

### 3.4.2 References