

# make\_montage Getting Started Guide

## Table of Contents

1. Installing the Anaconda Environment (skip if you already have a python interpreter).....	2
1.1. Install Anaconda.....	2
1.2. Create the Anaconda Environment.....	2
2. Usage Demonstration.....	6
2.1. Make a montage of the raw images.....	6
2.2. Make a montage of flat-field normalized images.....	7

# 1. Installing the Anaconda Environment (skip if you already have a python interpreter)

## 1.1. Install Anaconda

The `make_montage.py` script is written in python. A python interpreter is required to execute it. If you don't already have one, I recommend installing the Anaconda Platform, which is an open-source (free) package and environment management system (<https://www.anaconda.com/products/distribution>). It is available for Windows, Mac, and Linux. Follow the installation instruction provided on the following website: <https://docs.anaconda.com/anaconda/install/>.

## 1.2. Create the Anaconda Environment

Once you have Anaconda installed, open an Anaconda prompt (<https://docs.anaconda.com/anaconda/install/verify-install/>):

- **Windows:** Click Start, search for Anaconda Prompt, and click to open.
- **macOS:** Use Cmd+Space to open Spotlight Search and type “Navigator” to open the program.
- **Linux–CentOS:** Open Applications > System Tools > terminal.
- **Linux–Ubuntu:** Open the Dash by clicking the Ubuntu icon, then type “terminal”.

To execute a python script, you must first install the required modules. I suggest installing the required modules in an isolated conda environment. On Windows, create the conda environment in one of the following ways:

### Method 1 (create a conda environment from a yml file – easier):

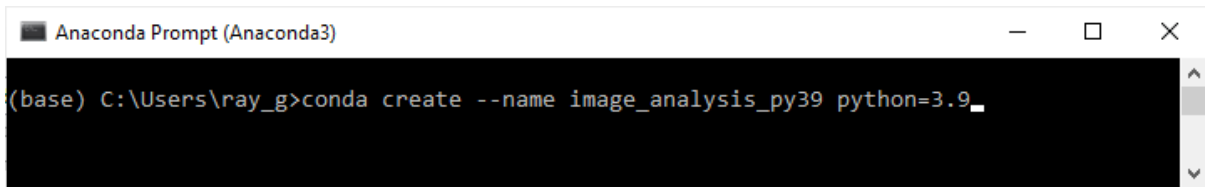
1. On your computer, navigate into the folder where `make_montage_env.yml` is saved.
2. Type:  

```
conda env create -f image_analysis_py39_env.yml
```
3. For detailed instructions, please refer to: <https://conda.io/projects/conda/en/latest/user-guide/tasks/manage-environments.html#creating-an-environment-from-an-environment-yml-file>

### Method 2:

1. First, create a new environment called, *e.g.*, `image_analysis_py39` using the command:

```
conda create --name image_analysis_py39 python=3.9
```

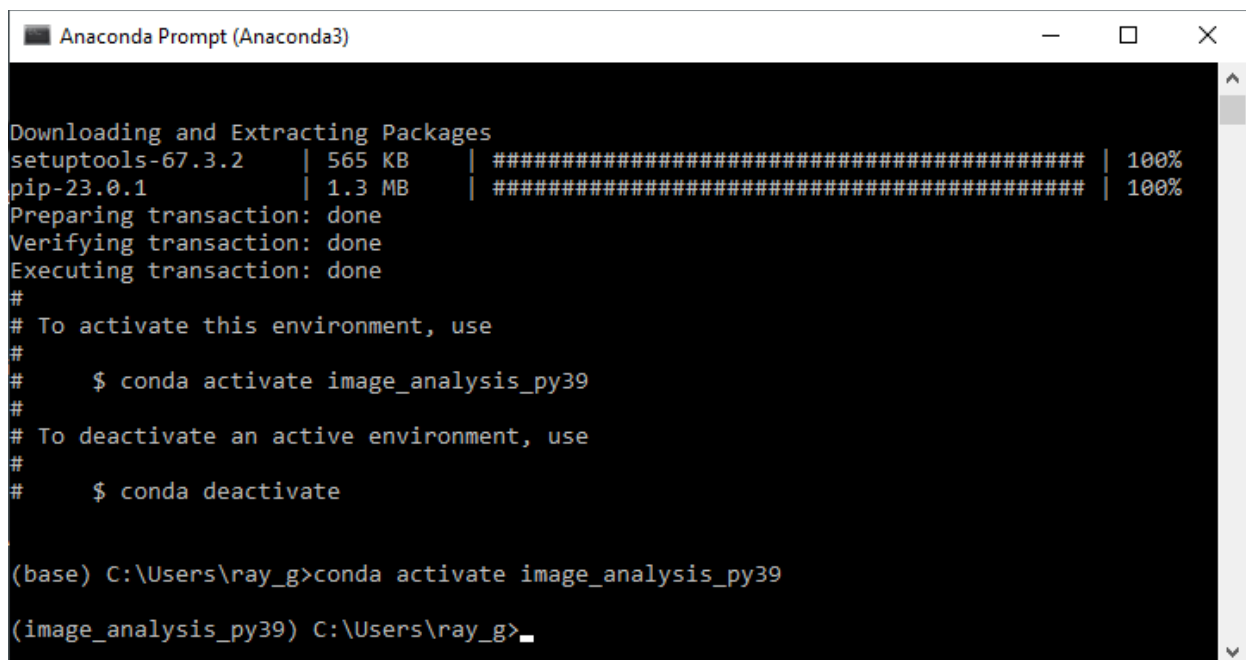


```
Anaconda Prompt (Anaconda3)
(base) C:\Users\ray_g>conda create --name image_analysis_py39 python=3.9_
```

Figure 1. Create a conda environment.

2. Activate the environment using the command:

```
conda activate image_analysis_py39
```



```
Anaconda Prompt (Anaconda3)

Downloading and Extracting Packages
setuptools-67.3.2 | 565 KB | ##### | 100%
pip-23.0.1 | 1.3 MB | ##### | 100%
Preparing transaction: done
Verifying transaction: done
Executing transaction: done
#
# To activate this environment, use
#
#   $ conda activate image_analysis_py39
#
# To deactivate an active environment, use
#
#   $ conda deactivate

(base) C:\Users\ray_g>conda activate image_analysis_py39

(image_analysis_py39) C:\Users\ray_g>
```

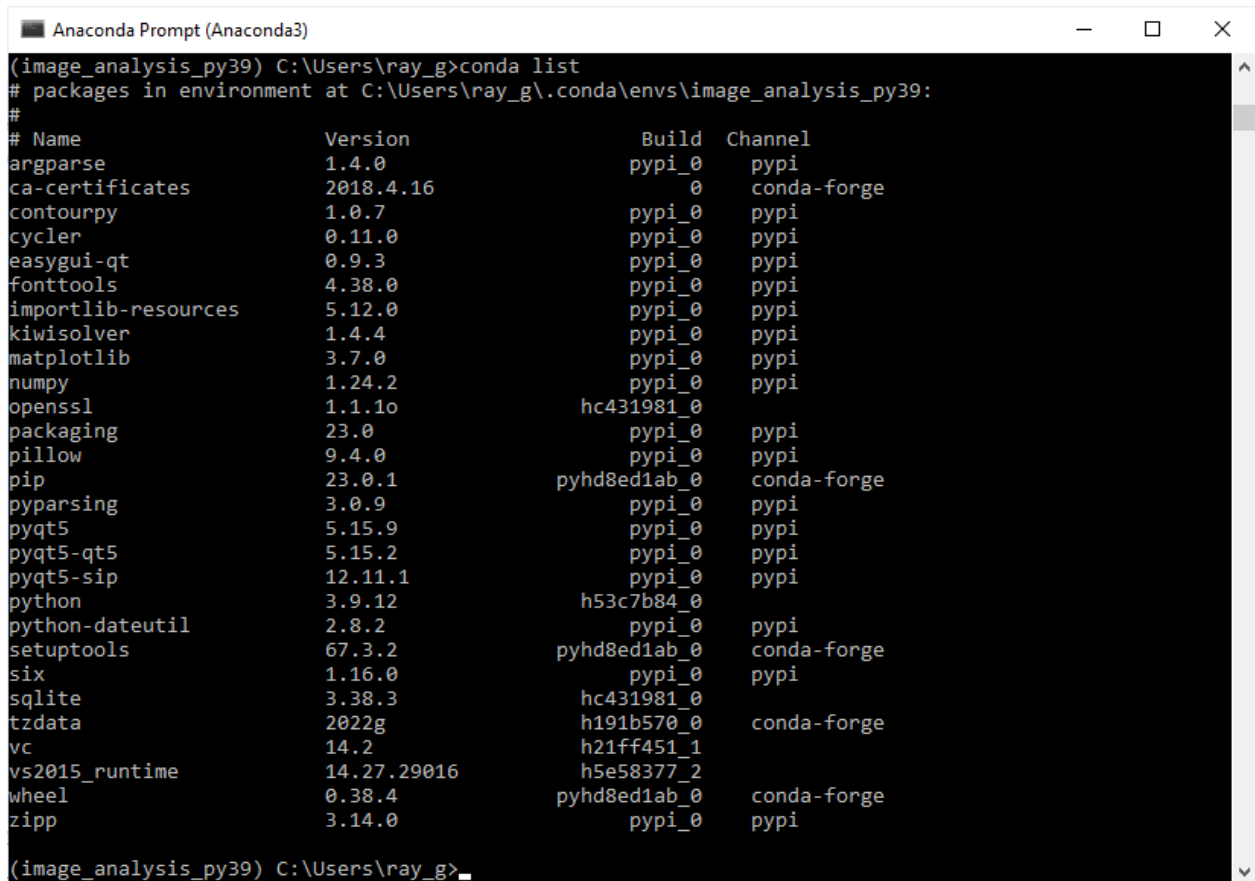
Figure 2. Activate the newly created conda environment.

3. To install the required modules, type:

- pip install matplotlib
- pip install numpy
- pip install PyQt5
- pip install easygui\_qt
- pip install argparse

4. To verify that the required modules have been successfully installed, type:

`conda list`



```
Anaconda Prompt (Anaconda3)
(image_analysis_py39) C:\Users\ray_g>conda list
# packages in environment at C:\Users\ray_g\.conda\envs\image_analysis_py39:
#
# Name                                Version                                Build                                Channel
argparse                              1.4.0                                pypi_0                              pypi
ca-certificates                       2018.4.16                             0                                    conda-forge
contourpy                             1.0.7                                pypi_0                              pypi
cycler                                0.11.0                               pypi_0                              pypi
easygui-qt                            0.9.3                                pypi_0                              pypi
fonttools                             4.38.0                               pypi_0                              pypi
importlib-resources                   5.12.0                               pypi_0                              pypi
kiwisolver                            1.4.4                                pypi_0                              pypi
matplotlib                            3.7.0                                pypi_0                              pypi
numpy                                  1.24.2                               pypi_0                              pypi
openssl                               1.1.1o                               hc431981_0                          pypi
packaging                             23.0                                  pypi_0                              pypi
pillow                                9.4.0                                pypi_0                              pypi
pip                                   23.0.1                               pyhd8ed1ab_0                        conda-forge
pyparsing                             3.0.9                                pypi_0                              pypi
pyqt5                                 5.15.9                               pypi_0                              pypi
pyqt5-qt5                             5.15.2                               pypi_0                              pypi
pyqt5-sip                             12.11.1                              pypi_0                              pypi
python                                3.9.12                               h53c7b84_0                          pypi
python-dateutil                       2.8.2                                pypi_0                              pypi
setuptools                             67.3.2                               pyhd8ed1ab_0                        conda-forge
six                                    1.16.0                               pypi_0                              pypi
sqlite                                 3.38.3                               hc431981_0                          conda-forge
tzdata                                2022g                                h191b570_0                          conda-forge
vc                                     14.2                                  h21ff451_1                          conda-forge
vs2015_runtime                        14.27.29016                          h5e58377_2                          conda-forge
wheel                                  0.38.4                               pyhd8ed1ab_0                        conda-forge
zipp                                   3.14.0                               pypi_0                              pypi

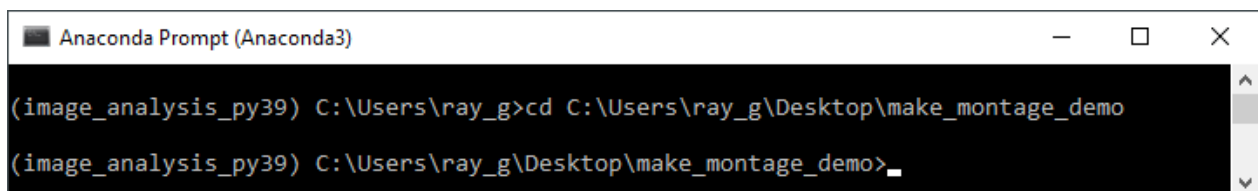
(image_analysis_py39) C:\Users\ray_g>
```

Figure 3. Review installed modules within the `image_analysis_py39` environment.

5. To run `make_montage`:

- First, go to the directory where the script is saved:

`cd C:\Users\ray_g\Desktop\make_montage_demo`

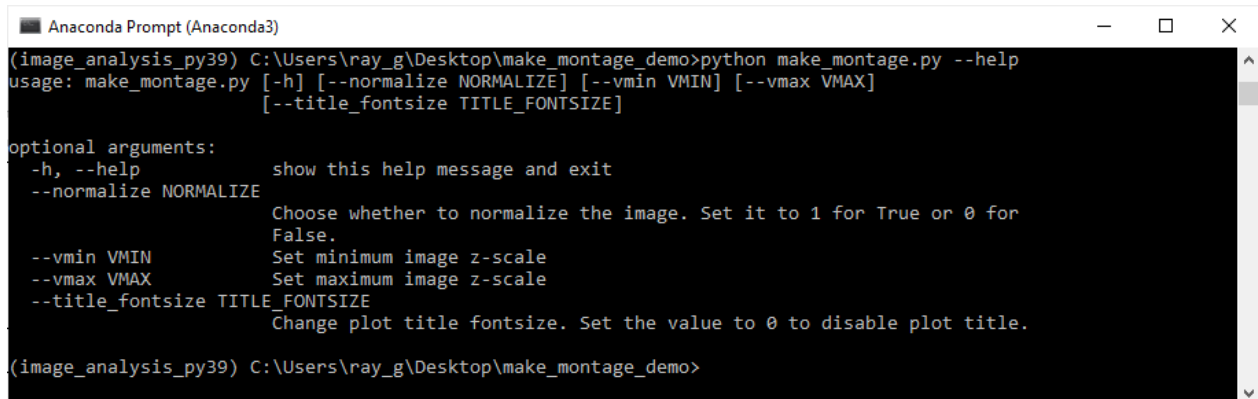


```
Anaconda Prompt (Anaconda3)
(image_analysis_py39) C:\Users\ray_g>cd C:\Users\ray_g\Desktop\make_montage_demo
(image_analysis_py39) C:\Users\ray_g\Desktop\make_montage_demo>
```

Figure 4. Navigate into the directory where `make_montage` is saved.

- To review the parameters for running the script, type:

`python make_montage.py --help`



```
Anaconda Prompt (Anaconda3)
(image_analysis_py39) C:\Users\ray_g\Desktop\make_montage_demo>python make_montage.py --help
usage: make_montage.py [-h] [--normalize NORMALIZE] [--vmin VMIN] [--vmax VMAX]
                        [--title_fontsize TITLE_FONTSIZE]

optional arguments:
  -h, --help            show this help message and exit
  --normalize NORMALIZE  Choose whether to normalize the image. Set it to 1 for True or 0 for
                        False.
  --vmin VMIN           Set minimum image z-scale
  --vmax VMAX           Set maximum image z-scale
  --title_fontsize TITLE_FONTSIZE
                        Change plot title fontsize. Set the value to 0 to disable plot title.

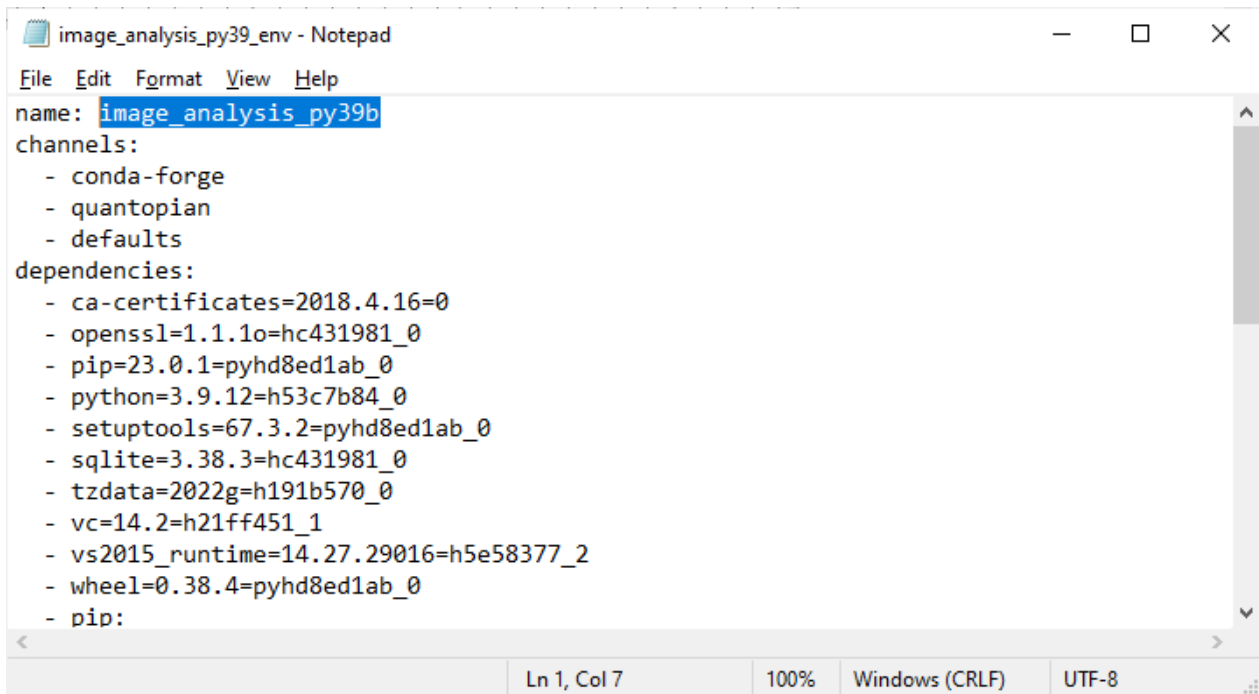
(image_analysis_py39) C:\Users\ray_g\Desktop\make_montage_demo>
```

*Figure 5. Review make\_montage.py parameters.*

6. (Optional) Having successfully created the image\_analysis\_py39 environment, you can save this environment into a yml file using the command:

`conda env export > image_analysis_py39_env.yml`

You may use this yml file to duplicate this environment on another computer as described above. You can also customize the name of the environment within the yml file.



```
image_analysis_py39_env - Notepad
File Edit Format View Help
name: image_analysis_py39b
channels:
  - conda-forge
  - quantopian
  - defaults
dependencies:
  - ca-certificates=2018.4.16=0
  - openssl=1.1.1o=hc431981_0
  - pip=23.0.1=pyhd8ed1ab_0
  - python=3.9.12=h53c7b84_0
  - setuptools=67.3.2=pyhd8ed1ab_0
  - sqlite=3.38.3=hc431981_0
  - tzdata=2022g=h191b570_0
  - vc=14.2=h21ff451_1
  - vs2015_runtime=14.27.29016=h5e58377_2
  - wheel=0.38.4=pyhd8ed1ab_0
  - pip:
```

Figure 6. The content of the `image_analysis_py39.yml` file.

### Removing a conda environment

If for any reasons you will have to remove a conda environment, use the command:

```
conda deactivate
```

Followed by:

```
conda env remove -n ENV_NAME
```

e.g.,

```
conda env remove -n image_analysis_py39
```

## 2. Usage Demonstration

The `make_montage.py` script sorts the sequence of images in one of two ways:

- (a) Assumes a four 2-frame cameras which sorts the images as: C1-frame1, C2-frame1, C3-frame1, C4-frame1, C1-frame2, C2-frame2, C3-frame2, C4-frame2. The script is as follows:

```
if not i%2: #if odd
    plt.subplot(2, int(no_imgs/2), int((i+2)/2))
else: #if even
    plt.subplot(2, int(no_imgs/2), int(4+(i+2)/2))
```

- (b) Assumes a set of sequential images which sorts the images in ascending order by date and time. The script is as follows:

```
plt.subplot(2, int(no_imgs/2), int(i+1))
```

By default, the script is configured to (a). To change the sorting sequence to (b), go to def make\_montage() section of the code, comment (b), and uncomment (a).

## 2.1. Make a montage of the raw images

1. To make a montage of the raw images, type:

```
python make_montage.py --normalize False --vmin 0 --vmax 100 --title_fontsize 8
```

2. When prompted, select the main folder containing the images saved within a subfolder called "Data" and click "Choose".

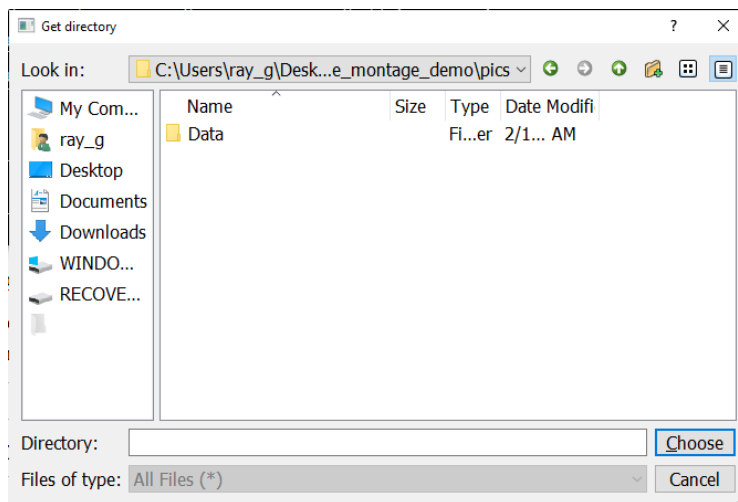
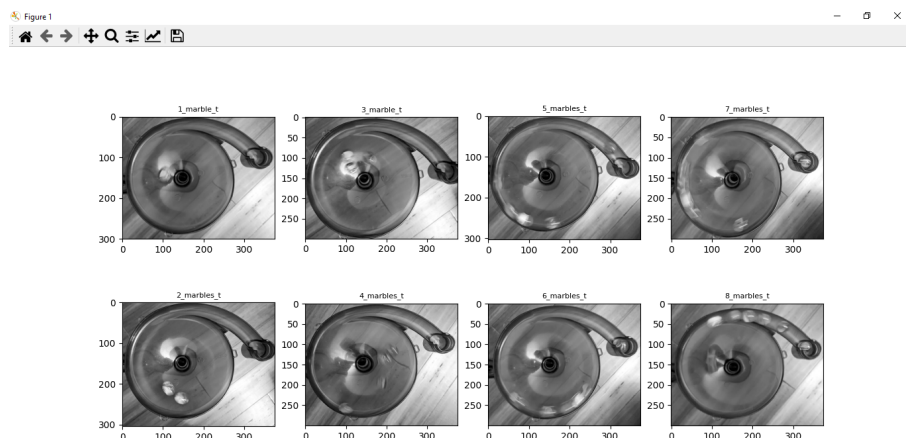


Figure 7. Select directory where pictures are saved.

3. Using the examples image provided in the Data folder, the result is:



*Figure 8. An example of `make_montage` application to a set of eight images without the flat-field normalization. Note that the image sequence is 1,3,5,7,2,4,6,8. In the script – within the `make_montage()` function – you have the option to change the sorting to sequential.*

4. Close the figure window, restart the script and adjust the `vmin` and `vmax` values to vary the contrast.
5. If you like the montage, you can save it by clicking on the save button from the toolbar.

## 2.2. Make a montage of flat-field normalized images

1. Prepare three set of images: raw images, whitefield, and darkfield.
2. Place each set of images into three separate folders called:  
“Data”, “WF”, and “DF”, respectively.
3. To make a montage of the flat-field normalized images, type:  

```
python make_montage.py --normalize True --vmin 0 --vmax 1.5 --title_fontsize 8
```
4. When prompted, select the main folder containing the images saved within the subfolders called “Data”, “WF”, and “DF” and click "Choose" (not shown).
5. Close the figure window, restart the script and adjust the `vmin` and `vmax` values to vary the contrast.
6. If you like the montage, you can save it by clicking on the save button from the toolbar.

~