Digital Image Processing in Python

Alon Oyler-Yaniv – alonyan1@gmail.com

https://github.com/alonyan/DIP

Workshop overview

- Day 1
  - Core concepts and motivation.
  - What is a digital image.
  - Pixel-level operations.
- Day 2
  - Pixel-level operations (cont.)
  - Variance balancing (thresholding)
  - Image segmentation
- Day 3
  - Image segmentation (cont.)
  - Feature extraction
  - Intro to modern approaches

Workshop goals

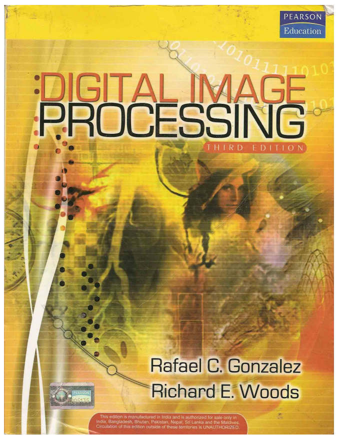
The workshop is an <u>introduction</u> to digital image processing, designed to give you a taste of what's possible with a specific emphasis on microscopy data.

By the end of the workshop, you should:

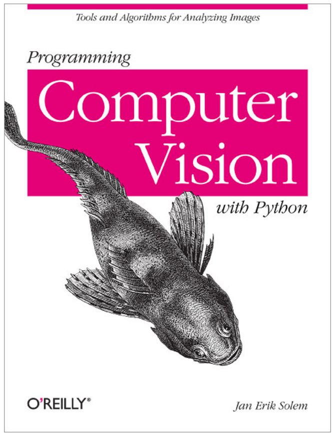
- Acquire basic understanding and familiarity with computer vision.
- Appreciate the importance of rigorous and systematic image analysis for reproducible and quantitative science.

Other resources

MOOC: https://www.mooc-list.com/course/image-analysis-methods-biologists-futurelearn



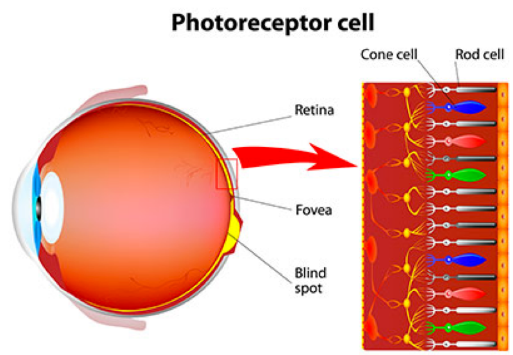
http://web.ipac.caltech.edu/staff/fmasci/home/astro_refs/Digital_Image_Processing_2ndEd.pdf



http://programmingcomputervision.com/

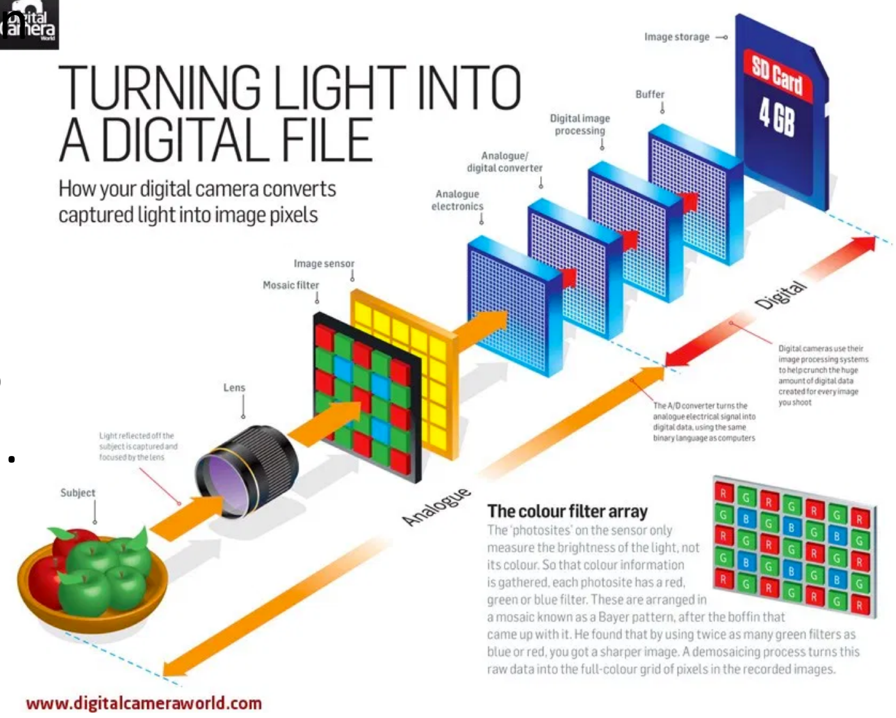
What is an Image?

- Photons are focused by lens (and cornea) and form an image on the retina.
- Activate a 2D array of photoreceptor cells that are sensitive to different colors.
- Information is passed to brain via optic nerves.



What is an Image?

- Photons are focused by lens and form an image on an image sensor.
- Activate (charge) a 2D array of pixels on an image sensor (CCD/CMOS/PMT).
- (Sometimes there are color filters)
- Computers store information in bits , so the information must be digitized (ADC).
- Information is stored as a matrix of numbers.



How is pixel data stored in the computer?

Data type	Range
uint8 – unsigned 8 bit integer	0 to 255
uint16 – unsigned 16 bit integer	0 to 65535
uint32– unsigned 32 bit integer	0 to 2^{32} - 1
float – floating point, single (32 bit) or double (64 bit) precision	-1 to 1 or 0 to 1
int8 - signed 8 bit integer	-128 to 127
int16 - signed 16 bit integer	-32768 to 32767
int32 - signed 32 bit integer	-2^{31} to 2^{31} - 1

- The cost of higher precision is higher memory use.
- Higher precision is sometimes pointless, depending on the data and what you need to get from it.

What is an Image?

- Photons are focused by lens and form an image on an image sensor.
- Activate (charge) a 2D array of pixels on an image sensor (CCD/CMOS).
- (Sometimes there are color filters)
- Computers store information in bits , so the information must be digitized (ADC).
- Information is stored as a matrix of numbers.

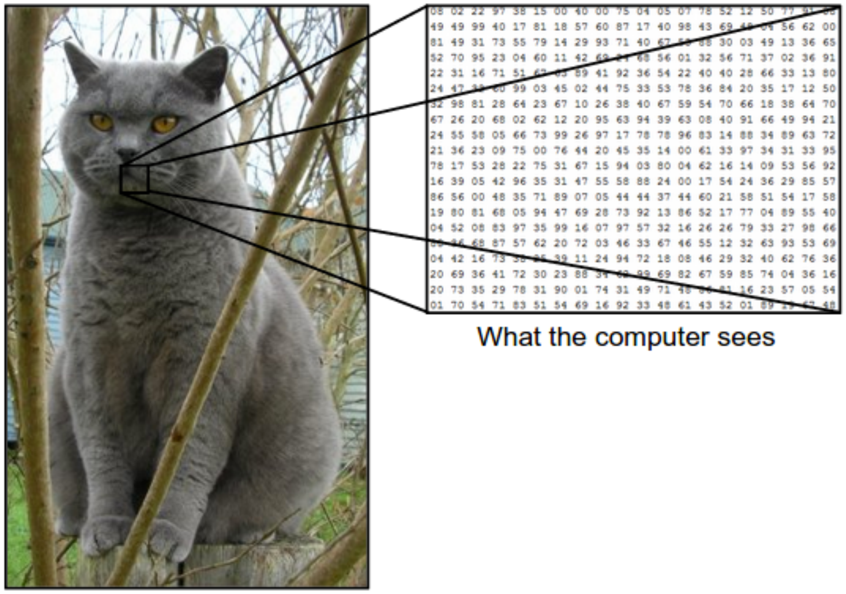


Image Resolution

- Light propagates as a wave.
- Waves can have constructive or destructive interference.
- Lenses are designed to maximize constructive interference at the focal point.

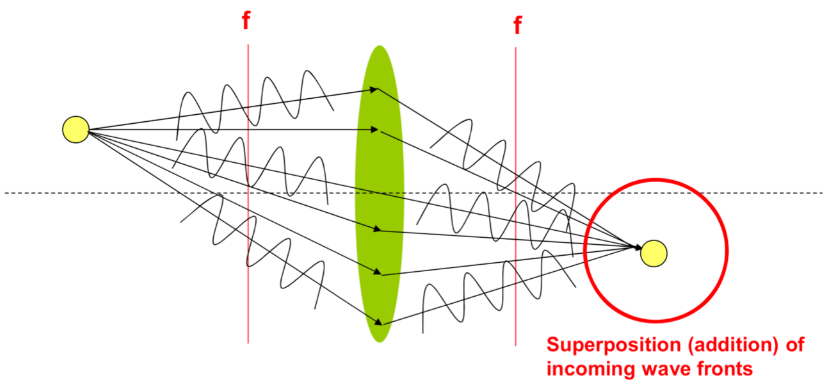


Image Resolution – Point Spread Function

- Image of point source - PSF

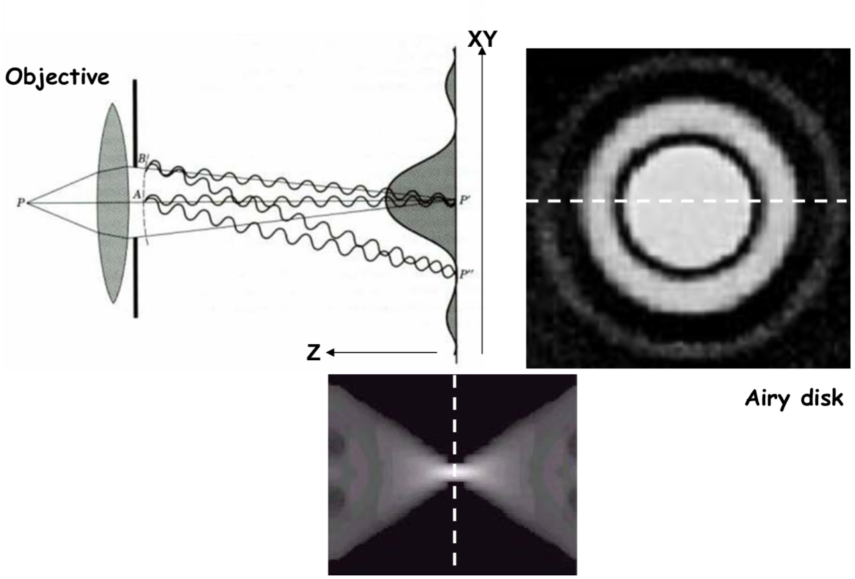


Image Resolution – Effect of Numerical Aperture

- NA = $n \cdot \sin \theta$
- Larger NA = Thinner PSF

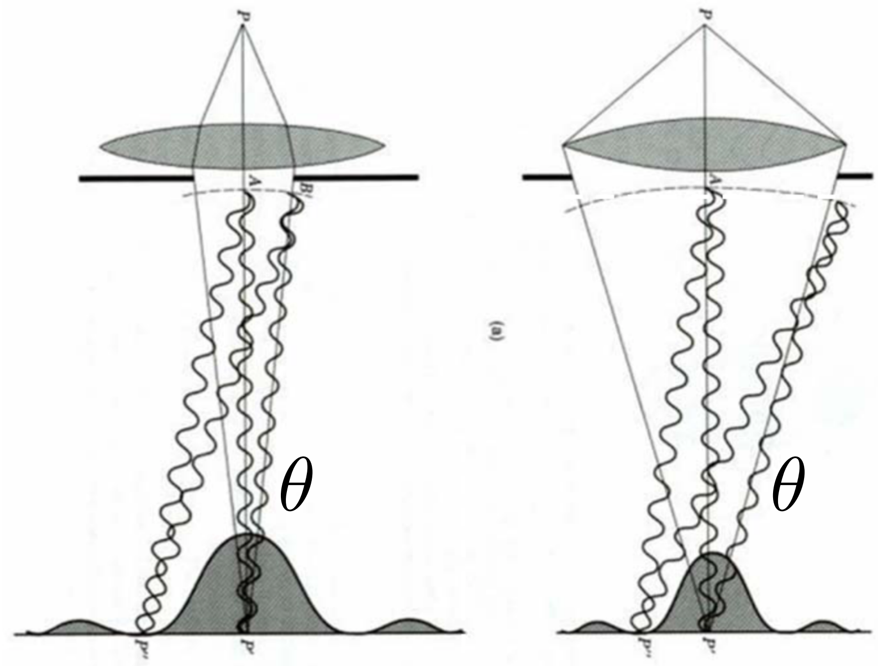
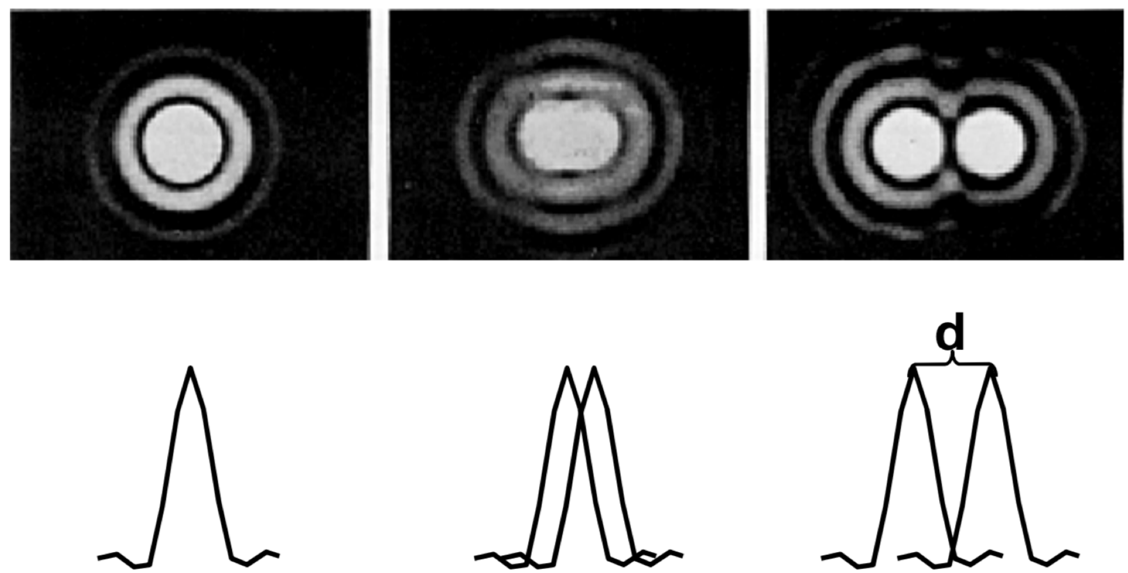


Image Resolution – How close two point can be and still be separable



Lord Rayleigh's criterion: $\delta^R = 0.61\dfrac{\lambda}{NA}$ $\qquad \delta_z^R = 2\dfrac{\lambda n}{NA^2}$

Aka Abbe diffraction limit

Physiologically motivated !!!

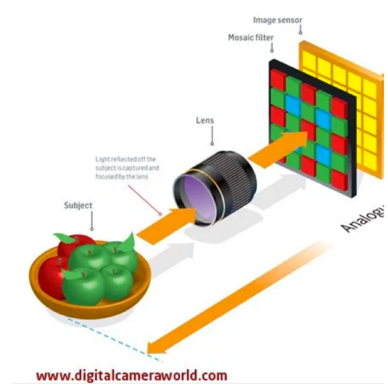
Image Resolution and magnification

- Notice that the objective magnification **does not affect** Abbes resolution formulae:

$$\delta^R = 0.61\frac{\lambda}{NA} \qquad \delta_z^R = 2\frac{\lambda n}{NA^2}$$

- Caveat: $\frac{\text{pixel size}}{\text{Mag}}$ is the area captured by each pixel in the camera.

If that area is larger than Abbes limit than the resolution is set by that. In that case the system is not diffraction limited.



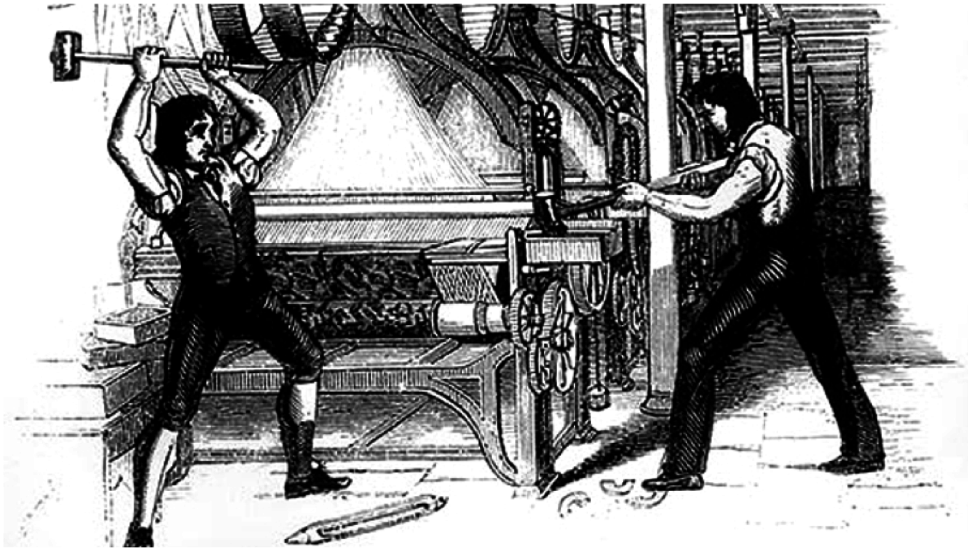
What is Image Processing?

Bunch of monkeys with different attributes

- Our brain is really really good at this!
- Computer vision - teach **computers** to interpret and understand the visual world

Why do we need image processing?

- If we're so good at this, why do we need computers?



(Optical) Microscopy circa y<2000 (b.f.p)

- Primarily antibody based
- Mostly fixed samples
- Low (throughput) and slow

- MB scale data,
  typically ~10 images

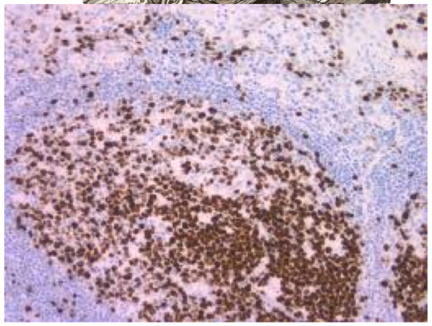


Image analysis was often performed manually,
usually on a subset of the data ("cherry picking")
And heuristic methods that "make sense"

Microscopy circa y>2000 (a.f.p)

- FPs allow multichannel, live, long time imaging.
- New technologies:
  - Imaging beyond Abbe limit (STORM,PALM,STED,SIM…).
  - Imaging of whole organisms in 5D using LSFM (SPIM, Scanning beam, OPM,…) .
  - Multiplexed in situ RNA FISH (MERFISH, SeqFISH).
  - Multiplexed IF (CyCIF, MIBI)

The scale and type of data for the modern microscopist has exploded in the last 2 decades.

GB-TB scale data. In some cases ~10^6 Images/experiment. Manual image analysis becomes impractical.

<u>Imaging has become a big data + high throughput problem.</u>

We need to talk about reproducibility…

- Our goal is to extract meaningful, quantifiable information in a non-biased manner from imaging data.

- Manual, non-systematic, image analysis is prone to bias.

Computational Image processing

- Image processing is about *extracting* the relevant data from your measurements, but not about analyzing them.

Something something computers …

Monkey ID	M/F	size	Coat Color	Attr 1	Attr 2	Attr 3	…
1	F	24	…				
2	F	40					
3	M	87					
4	M	21					
5	F	31					
6	F	24					
7	F	54					

What might an image processing pipeline look like?

Image acquisition

What might an image processing pipeline look like?

Image acquisition

Image Preprocessing
- Aligning
- Tiling
- Denoising

What might an image processing pipeline look like?

Image acquisition

Image Preprocessing
- Aligning
- Tiling
- Denoising

Image Processing
- Background / Foreground
- Segmentation
- Feature extraction
- Conversion to data

What might an image processing pipeline look like?

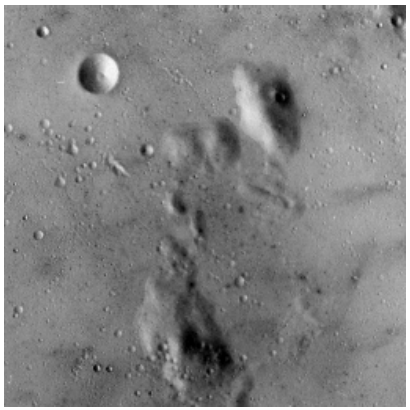
Image acquisition

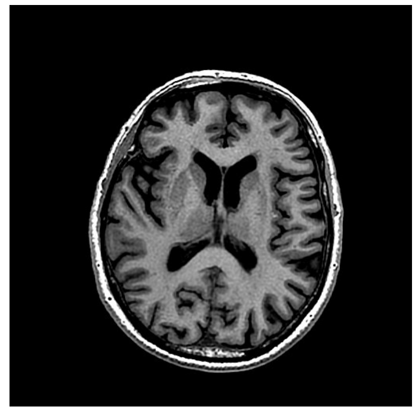
Image Preprocessing
- Aligning
- Tiling
- Denoising

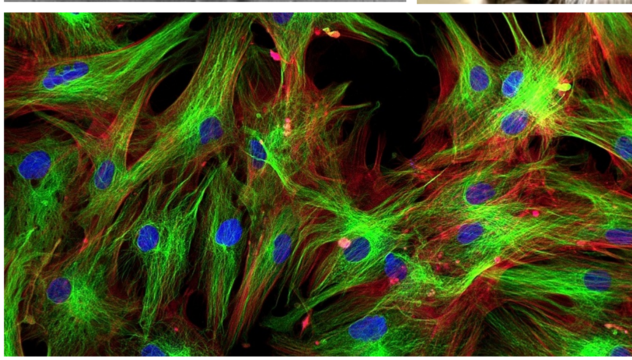
Image Processing
- Background / Foreground
- Segmentation
- Feature extraction
- Conversion to data

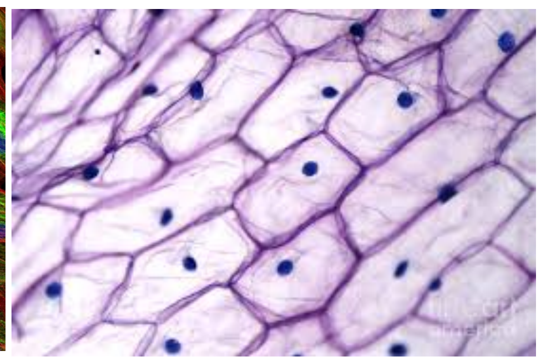
Data Analysis
- Anything that can be learned from the data

What kinds of images might we look at?









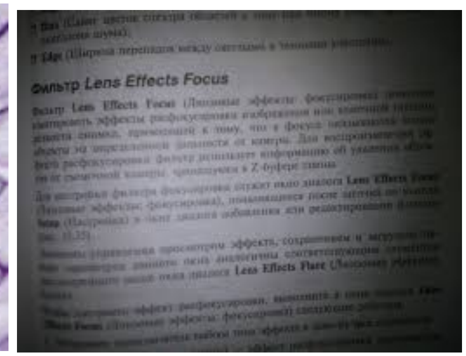


Image formats and compression

- It is important to have your data in the right format for the analysis you would like to perform.
- Some formats are considered **lossy** due to some compression when converting and saving in this format.
- Others are **lossless**, and uncompressed. This is more important when you would like to extract quantitative data from your image. They also tend to have larger file sizes.
- Many microscope manufacturers have their own proprietary image types and these will require specific tools to extract data from.

Some common data formats

- JPG or JPEG – Typically lossy and small bit depth (values range from 0 to 255). Can be grayscale or RGB, most common on standard digital cameras.
- PNG – Lossless image format but not high bit depth (0 to 255). Can also contain a transparency channel (alpha). One of the most common image formats.
- TIFF or TIF – Most common format in scientific image. Generally lossless and can have a bit depth up to 16 bit (0 to 65,535) in both grayscale and color. TIFF files can also be image stacks.
- CZI (zeiss) and LIF (Leica) – Proprietary image formats that can contain extra information about imaging and microscopy conditions.
- HDF5 / N5 - Hierarchical Data Format, designed to store very large amounts of structured data.

Some common data formats

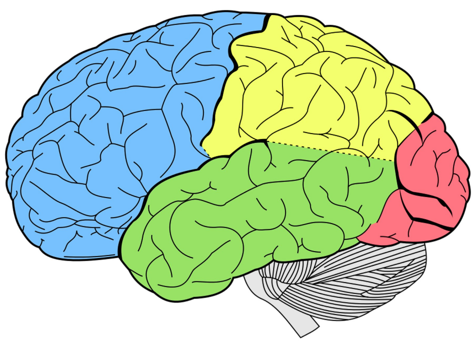
- JPG or JPEG – Typically lossy and small bit depth (values range from 0 to 255). Can be grayscale or RGB, most common on standard digital cameras.
- PNG – Lossless image format but not high bit depth (0 to 255). Can also contain a transparency channel (alpha). One of the most common image formats.
- TIFF or TIF – Most common format in scientific image. Generally lossless and can have a bit depth up to 16 bit (0 to 65,535) in both grayscale and color. TIFF files can also be image stacks.
- CZI (zeiss) and LIF (Leica) – Proprietary image formats that can contain extra information about imaging and microscopy conditions.
- HDF5 / N5 - Hierarchical Data Format, designed to store very large amounts of structured data.

Tools used in this workshop

python™

stackoverflow

scikit-image
image processing in python



NumPy

jupyter

Libraries we will be using

One of python's major advantages is it's extensive libraries that have been developed for a variety of different applications.

I'll be focusing this workshop on the libraries shown here.

NumPy

matplotlib

scikit-image
image processing in python

pandas
$y_{it} = \beta' x_{it} + \mu_i + \epsilon_{it}$

NumPy

Advanced matrix calculator and high-level math

Graphing, plotting, and viewing images

scikit-image

image processing in python

High-level image processing

pandas

$y_{it} = \beta' x_{it} + \mu_i + \epsilon_{it}$

Data management, manipulation and analysis

What we’ll be doing

The course has two tracks

1. Fully completed notebooks
2. Fill in the blanks notebooks

- If you have basic experience programing, you should be able to fill in the blanks.
- If you don’t, you can either tag along with someone who’s more experienced, or just follow along with a completed notebook.

  We’ll be going over everything together, leaving time for you to try and figure stuff out on your own.

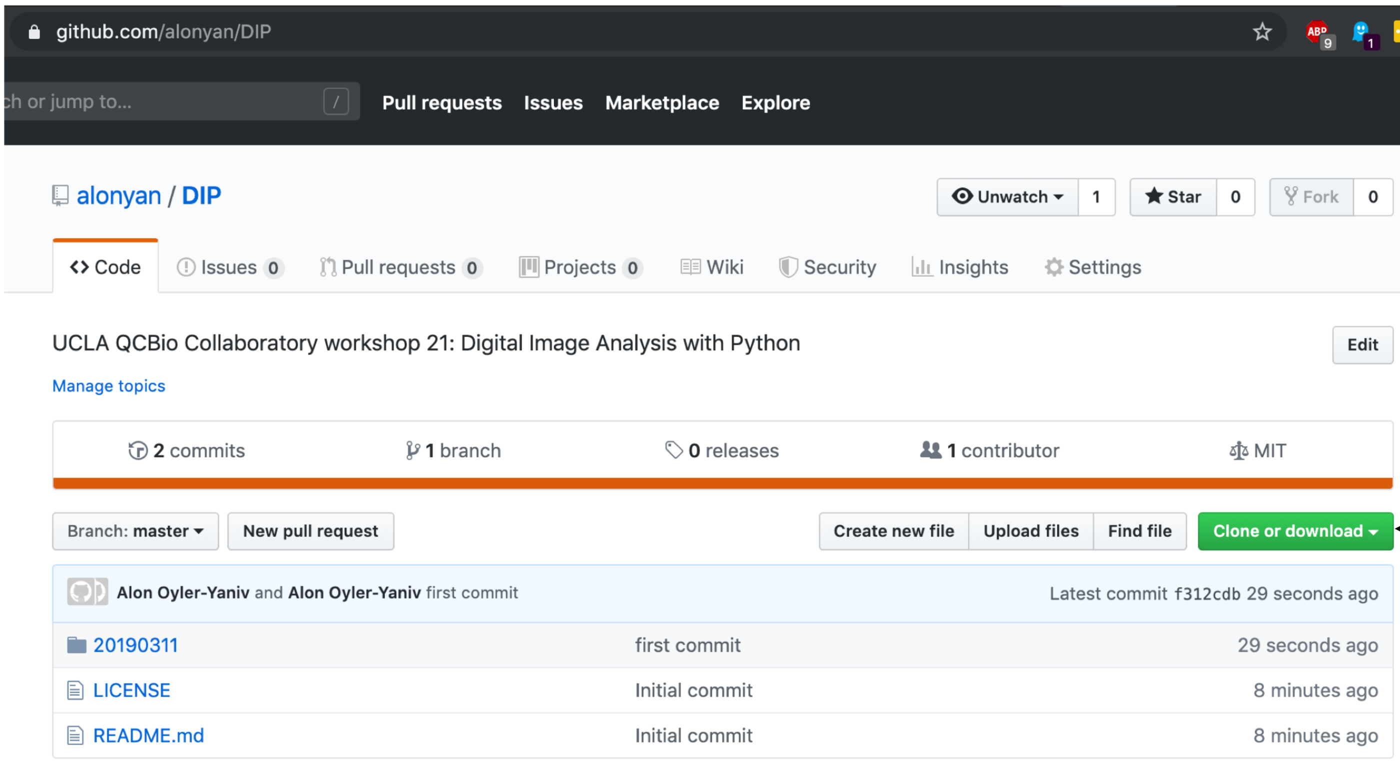
Setup

Let’s get going!

We need to download/clone the relevant course material and access it using jupyter notebook

Cloning/Downloading the course repository

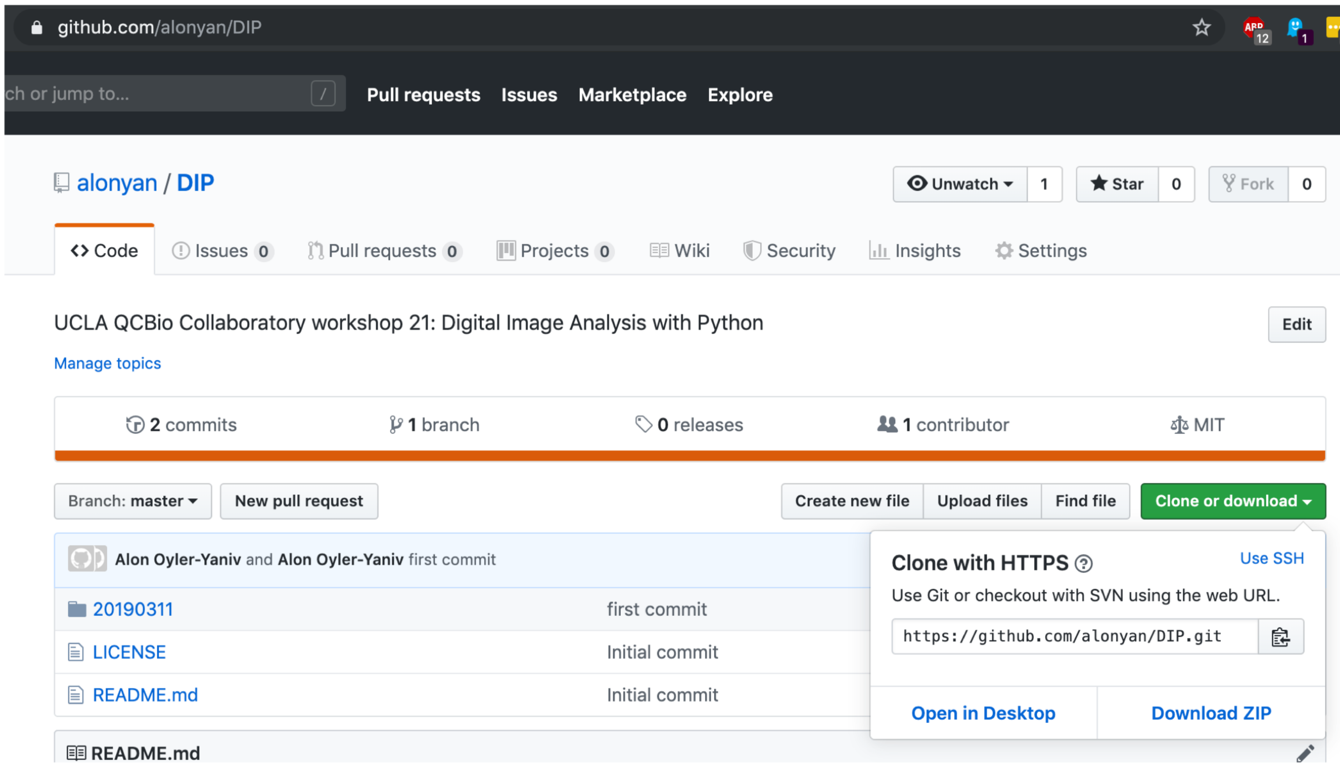
- Go to https://github.com/alonyan/DIP



Download the repo

Cloning/Downloading the course repository

- Go to https://github.com/alonyan/DIP



Download the repo

Cloning/Downloading the course repository

- Move the .ZIP file to wherever you want it (/Documents is a good spot) and unzip

- You can also navigate (cd) in the terminal/command line to wherever you want it and type:

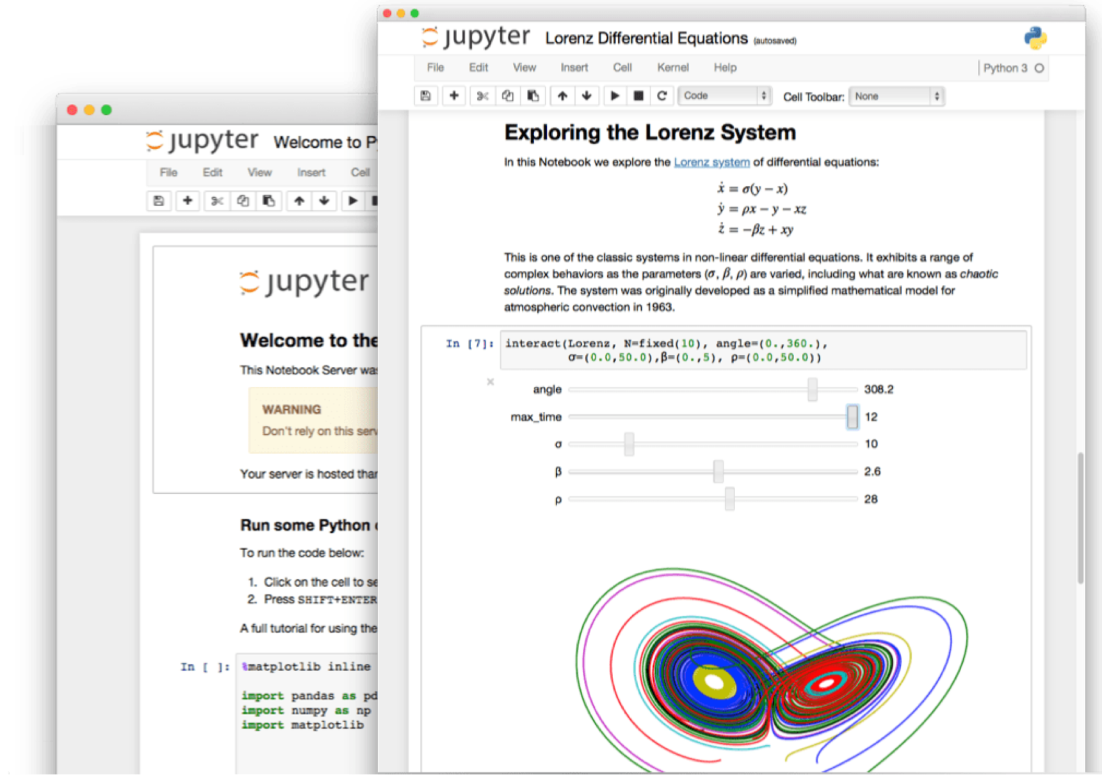
    git clone https://github.com/alonyan/DIP

Getting started with Jupyter Notebooks

Jupyter notebooks

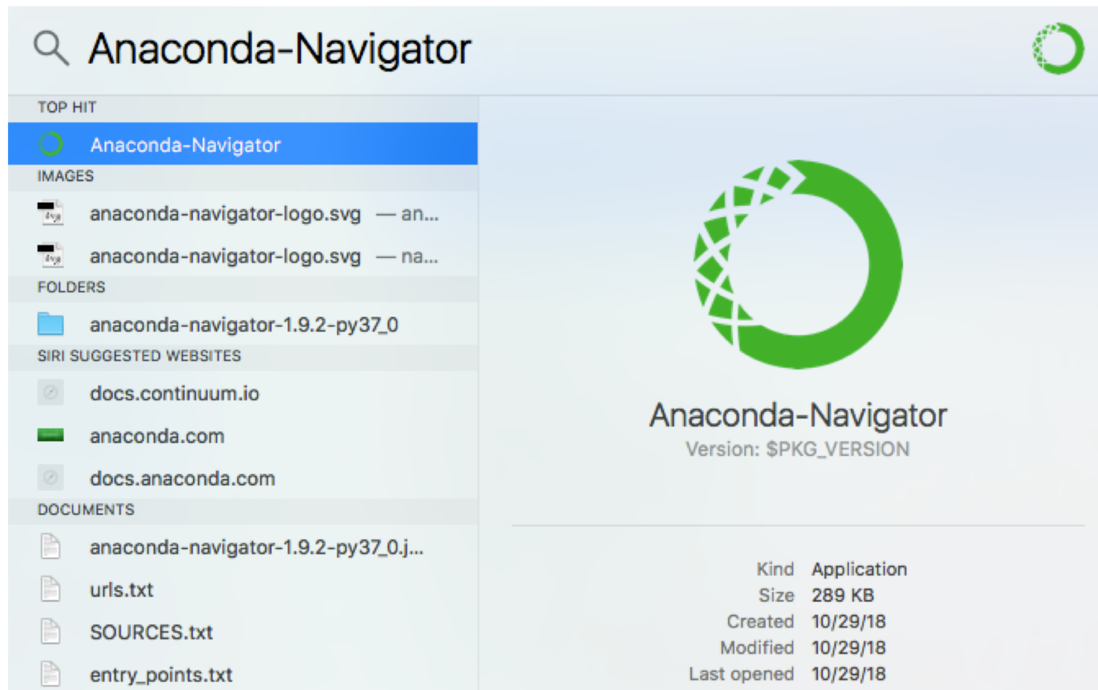
Jupyter notebooks are special documents that allow you to create and share documents **live code**.

They are a great way to keep your coding endeavors organized and share useful analysis with others.



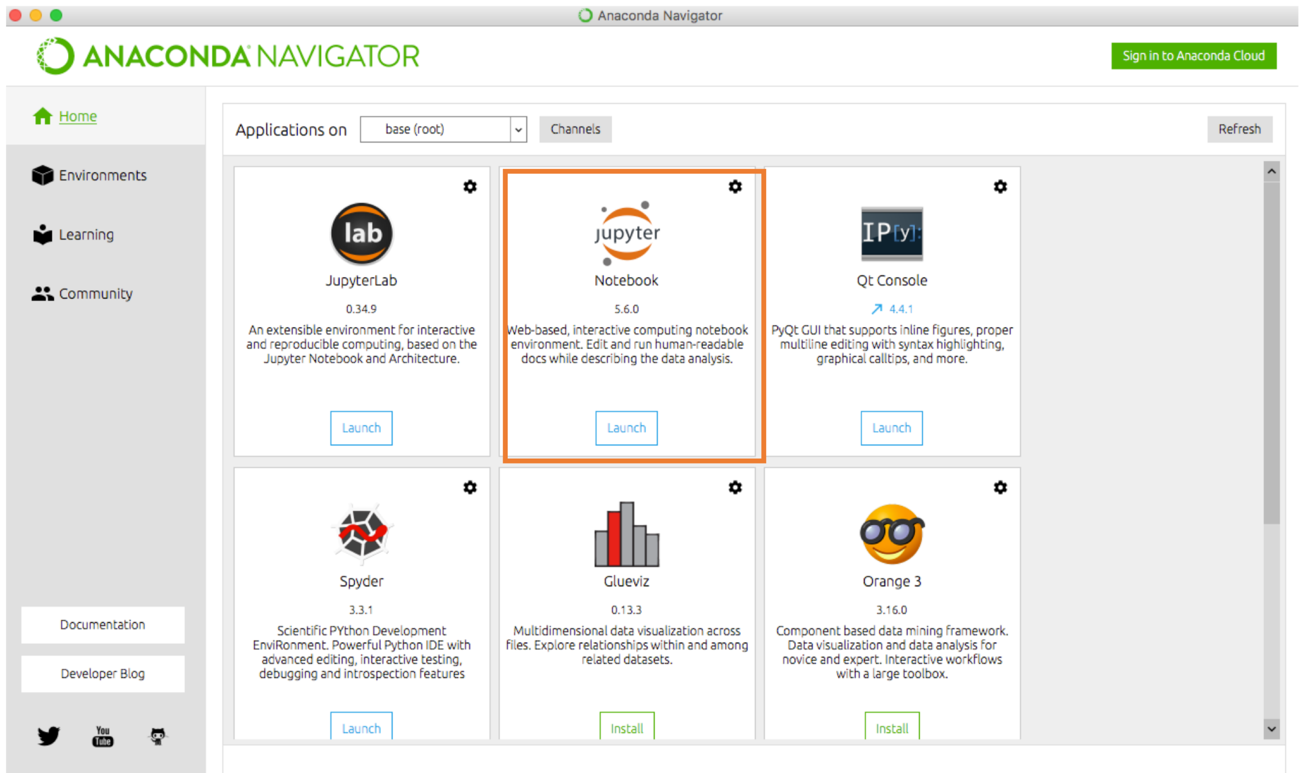
Getting started in Anaconda

- Launch the Anaconda application.
- MAC: Hit command+<space bar> and type "Anaconda"
- WINDOWS: Look for Anaconda on your start menu
- This should start you in the home directory.

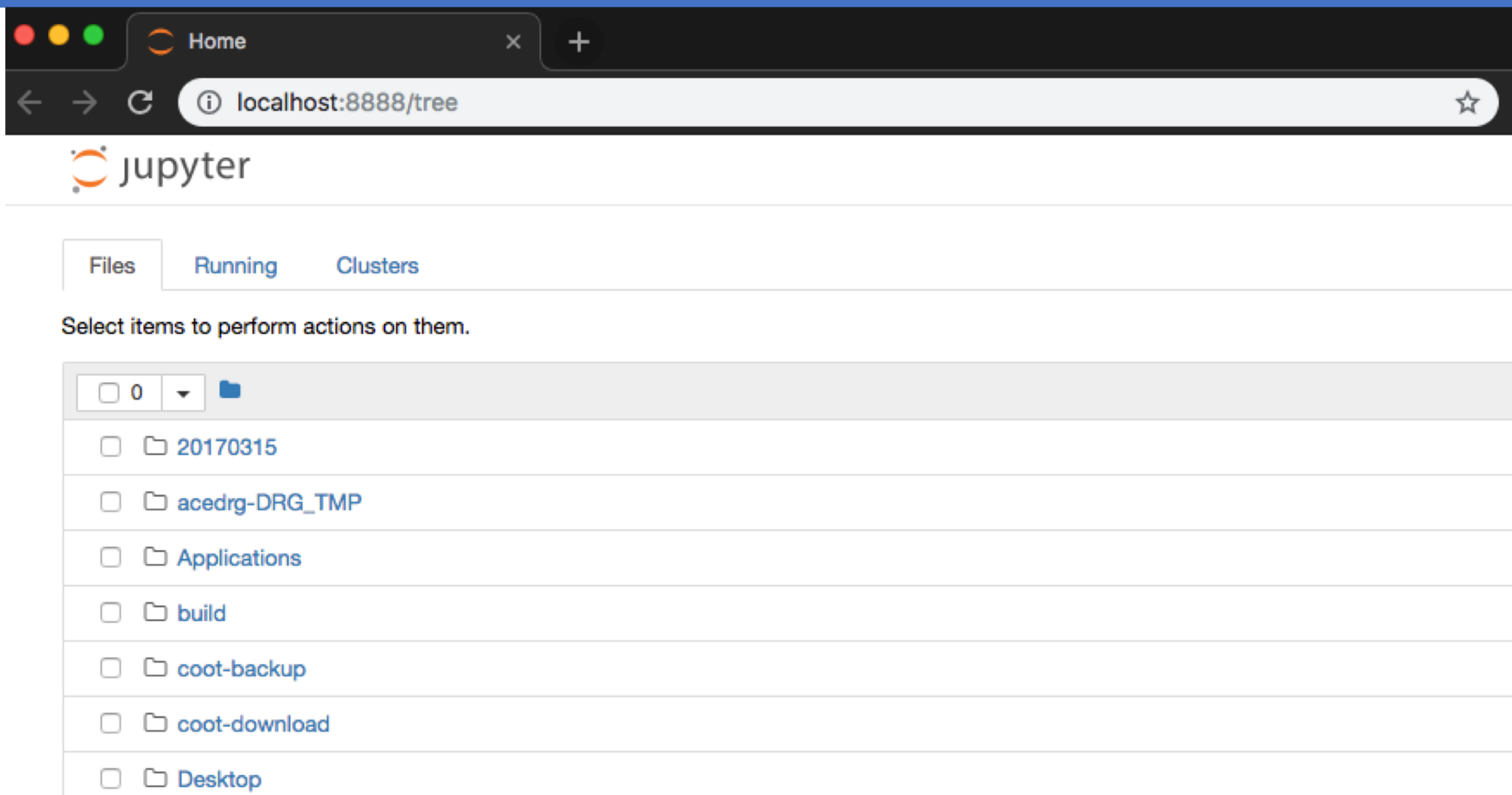


Getting started from Anaconda

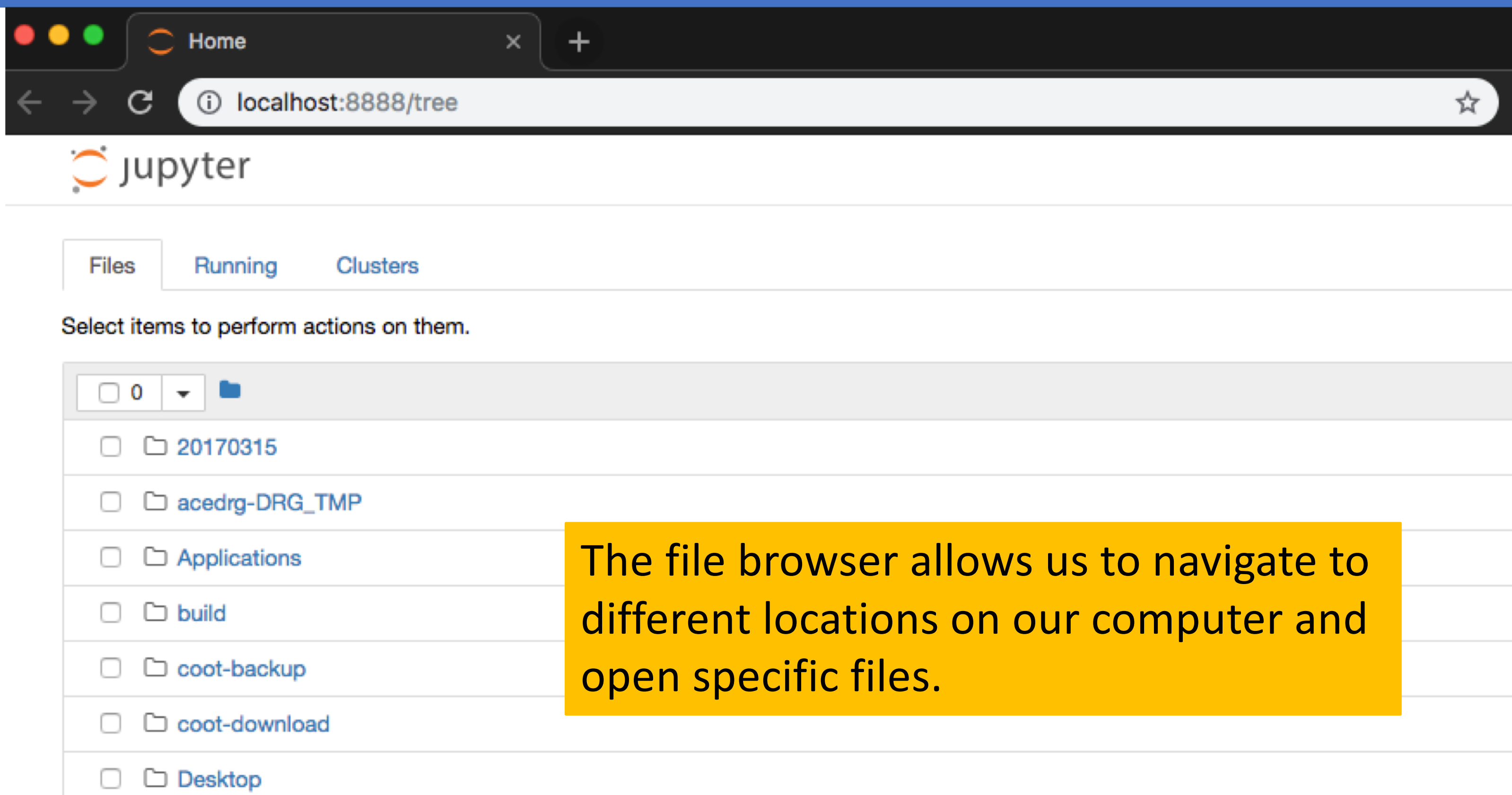
Click on the launch button under the jupyter notebook icon. After a few moments the dashboard should open in your default web browser.



The jupyter dashboard

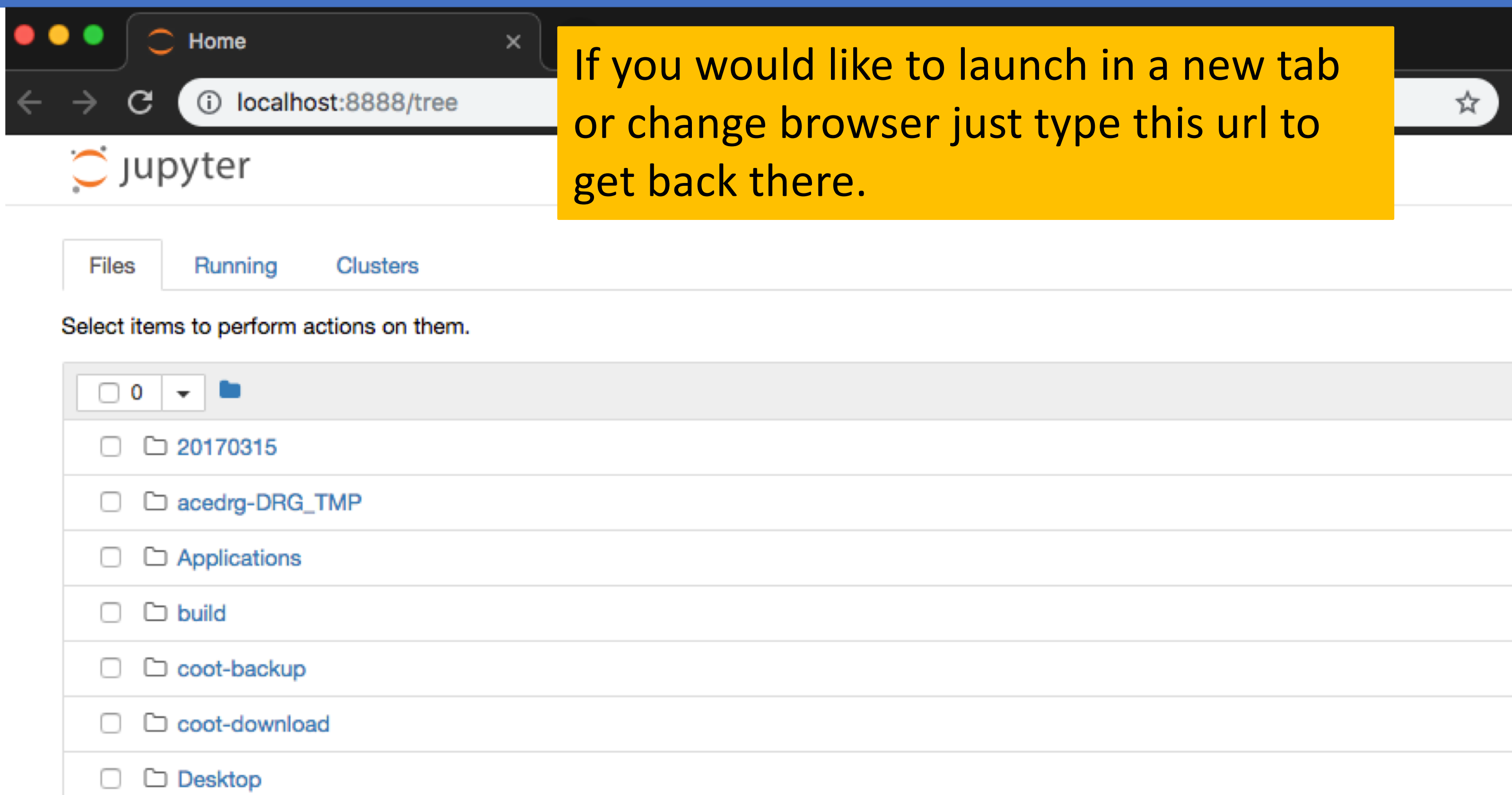


The jupyter dashboard



The file browser allows us to navigate to different locations on our computer and open specific files.

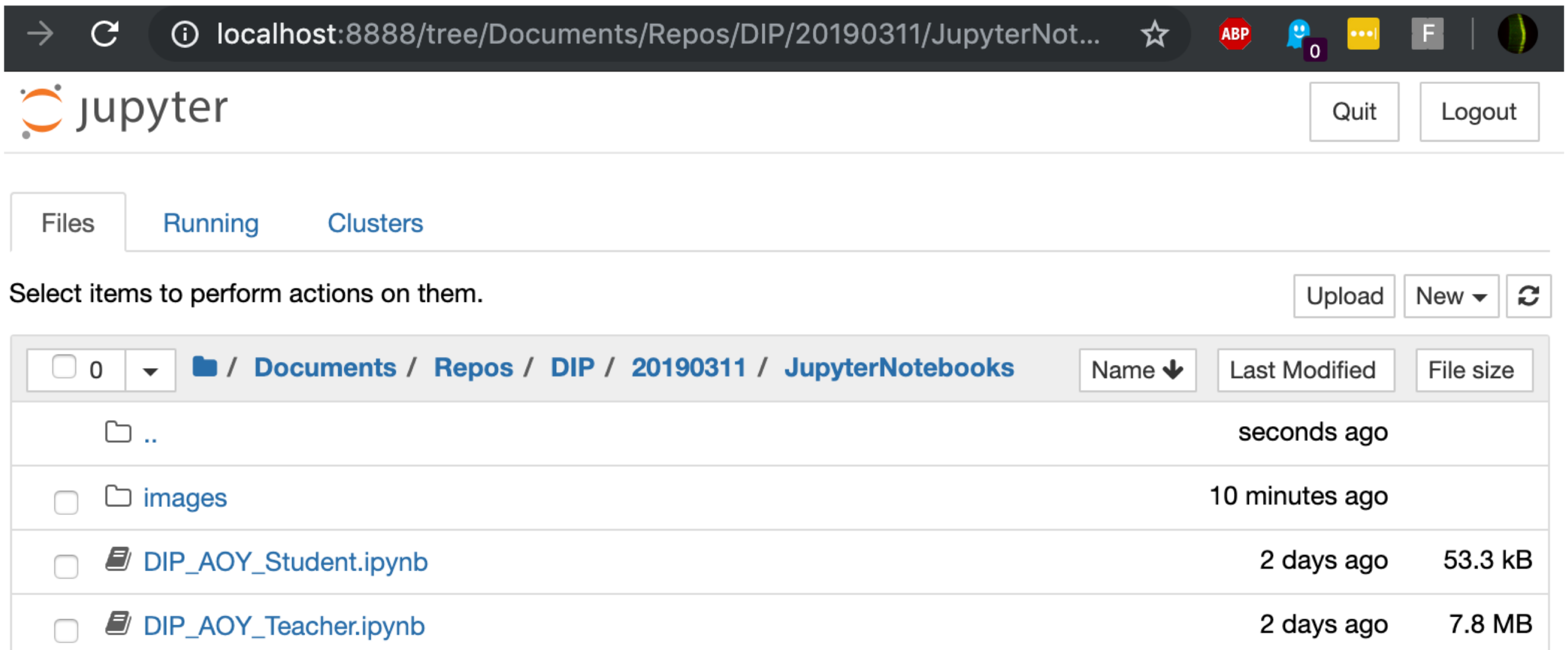
The jupyter dashboard



If you would like to launch in a new tab or change browser just type this url to get back there.

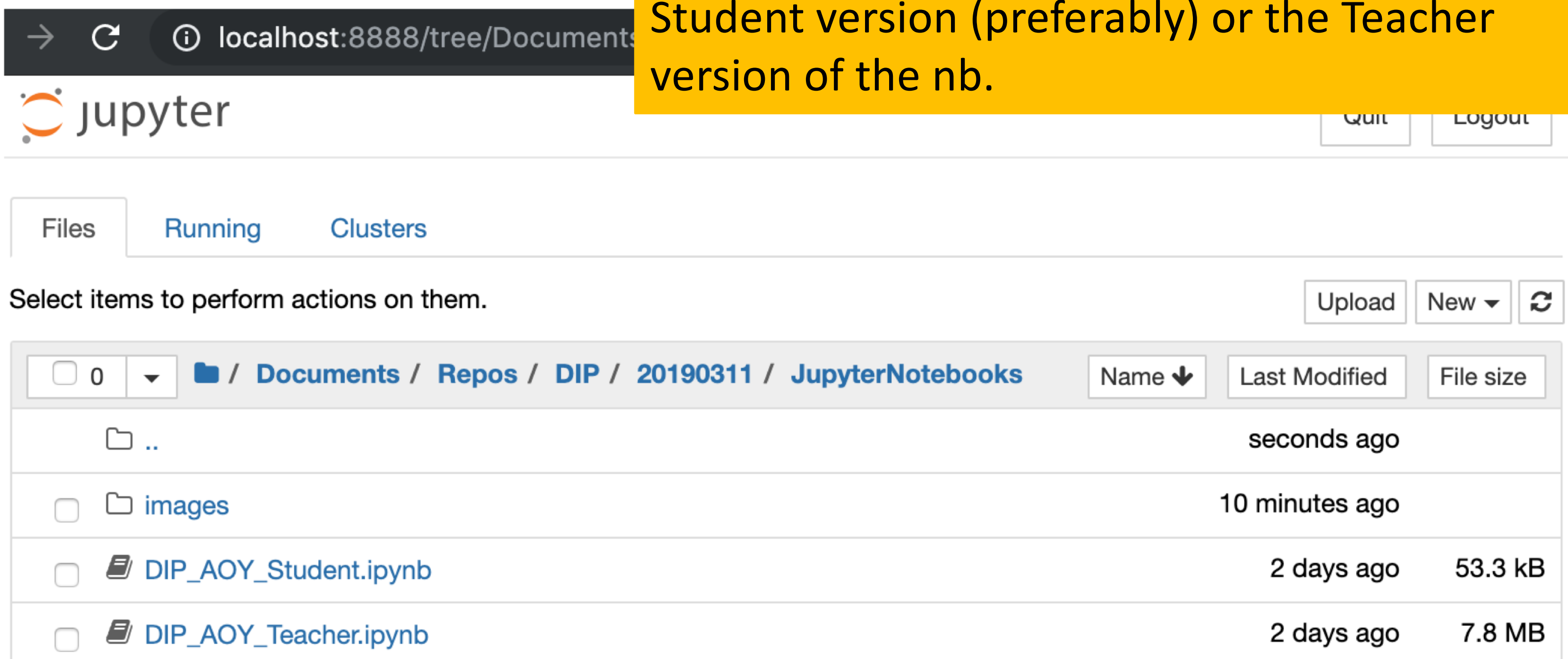
The jupyter dashboard

Navigate to wherever you saved the repository



The jupyter dashboard

*.ipynb is the suffix of a notebook. Open the Student version (preferably) or the Teacher version of the nb.



Time to process