

# 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

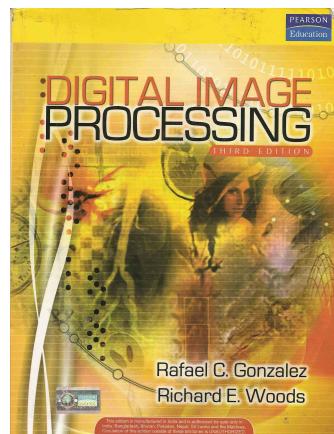
The workshop is an introduction to digital image processing, designed to give you a taste of what's possible with a specific emphasis on microscopy data.

By the end of the workshop, you should:

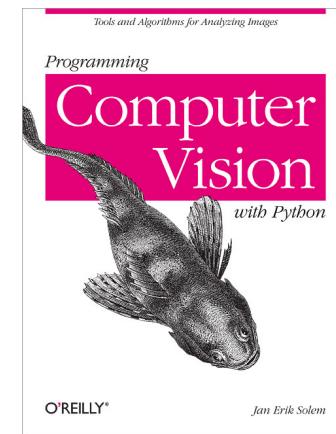
- Acquire basic understanding and familiarity with computer vision.
- Appreciate the importance of rigorous and systematic image analysis for reproducible and quantitative science.

# Other resources

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



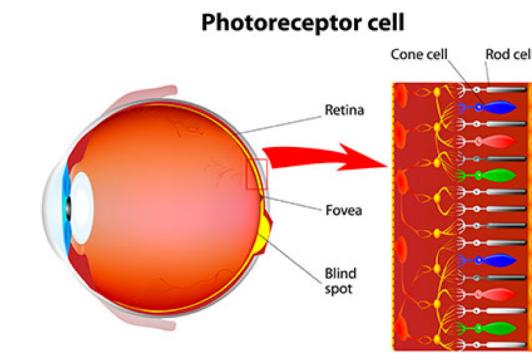
[http://web.ipac.caltech.edu/staff/fmasci/home/astro\\_refs/Digital Image Processing 2ndEd.pdf](http://web.ipac.caltech.edu/staff/fmasci/home/astro_refs/Digital_Image_Processing_2ndEd.pdf)



<http://programmingcomputervision.com/>

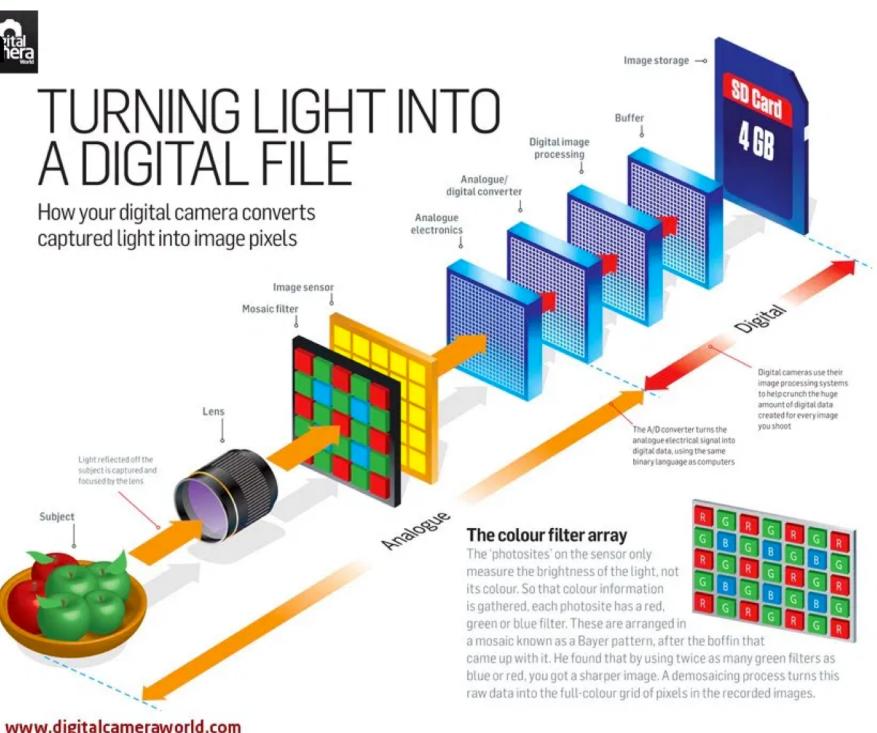
# What is an Image?

- 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.



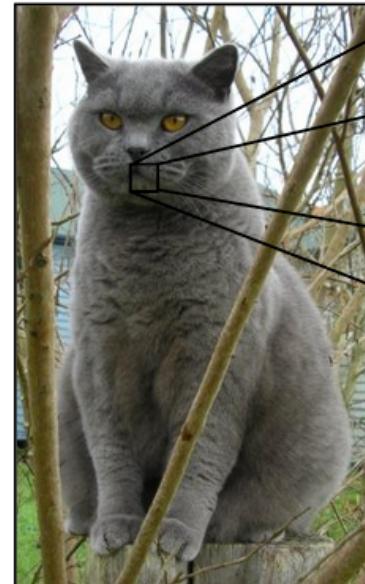
# How is pixel data stored in the computer?

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.

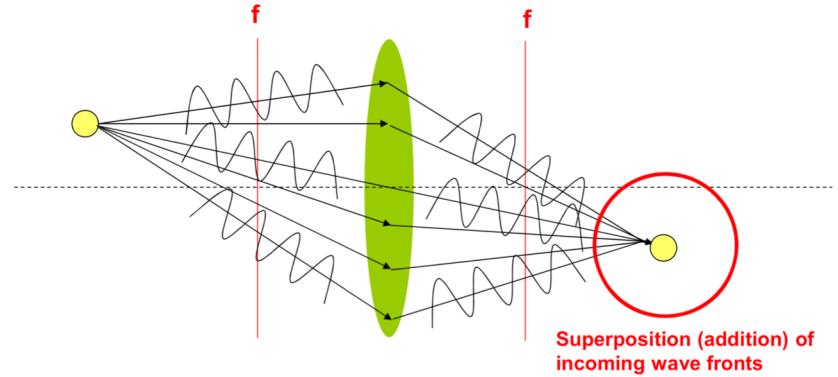


08	02	22	97	38	15	00	40	00	75	04	05	07	78	52	12	50	77	01	00
49	49	99	40	17	81	18	57	60	87	17	40	98	43	69	4	05	56	62	00
01	49	31	73	55	79	14	29	93	71	40	67	51	18	30	03	49	13	36	65
52	70	95	23	04	60	11	42	63	11	65	56	01	32	56	71	37	02	36	91
22	31	16	71	51	67	05	89	41	92	36	54	22	40	40	28	66	33	13	80
24	47	31	60	39	03	45	02	44	75	33	53	78	36	84	20	35	17	12	50
32	98	81	28	64	23	67	10	26	38	40	67	59	54	70	66	18	38	64	70
67	26	20	68	02	62	12	20	95	63	94	39	63	08	40	91	66	49	94	21
24	55	53	05	66	73	99	26	97	17	78	78	96	83	14	80	34	89	63	72
21	36	23	09	75	00	76	44	20	45	35	14	00	61	33	97	34	31	33	95
78	17	53	28	22	75	35	67	15	94	03	80	04	62	16	14	09	53	56	92
16	39	05	42	96	35	31	47	55	58	88	24	00	17	54	24	36	29	85	57
86	56	00	48	35	71	89	07	05	44	44	37	44	50	21	58	51	54	17	58
19	80	81	68	05	94	47	69	28	73	92	13	86	52	17	77	04	89	55	40
04	52	08	83	97	35	99	16	07	97	57	32	16	26	26	79	33	27	98	66
54	68	87	57	62	20	72	03	46	33	67	46	55	12	32	63	93	53	69	
04	42	16	73	36	85	39	11	24	94	72	18	05	46	29	32	40	62	76	36
20	69	36	41	72	30	23	68	34	71	03	69	82	67	59	85	74	04	36	16
20	73	35	29	72	31	98	01	74	31	49	71	45	16	16	23	57	05	54	
01	70	54	71	83	51	54	69	16	92	33	48	61	43	52	01	87	54	41	

What the computer sees

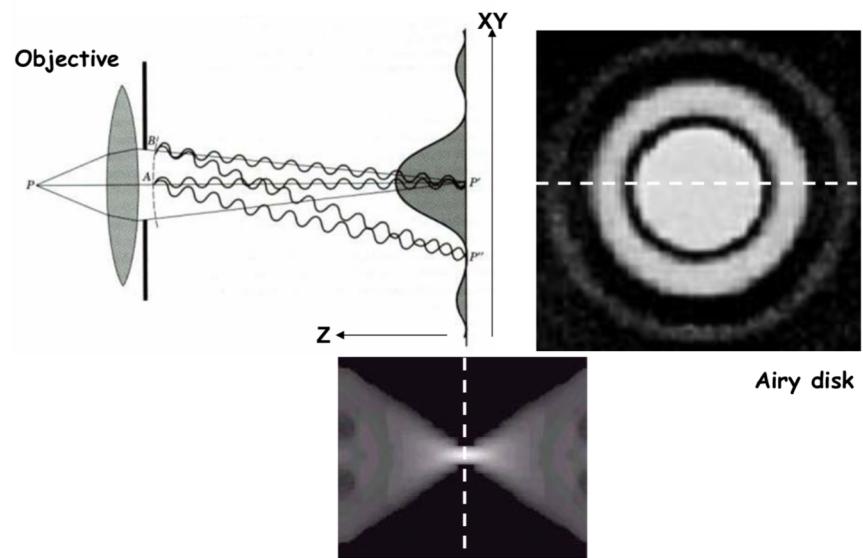
# Image Resolution

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



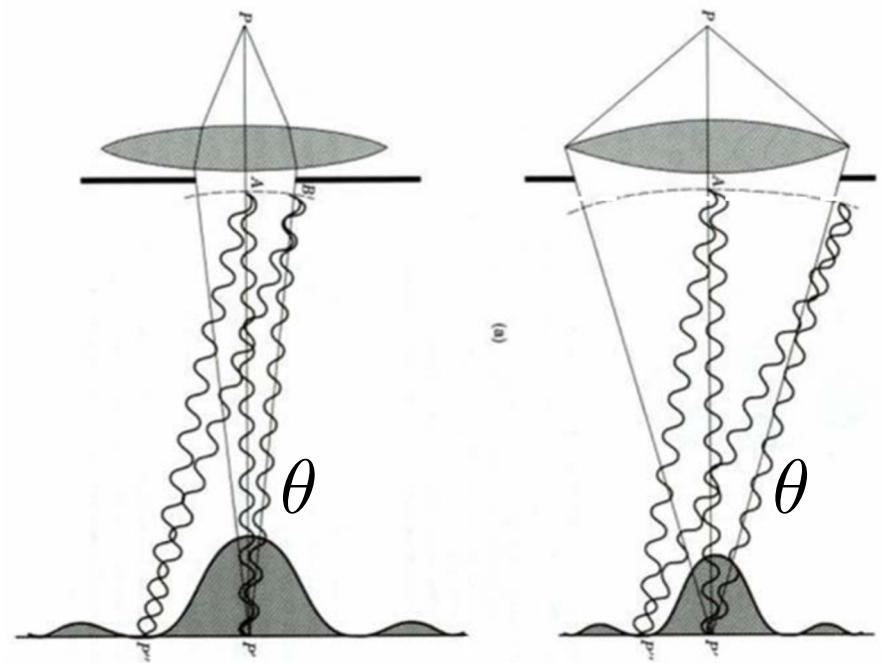
# Image Resolution – Point Spread Function

- Image of point source - PSF

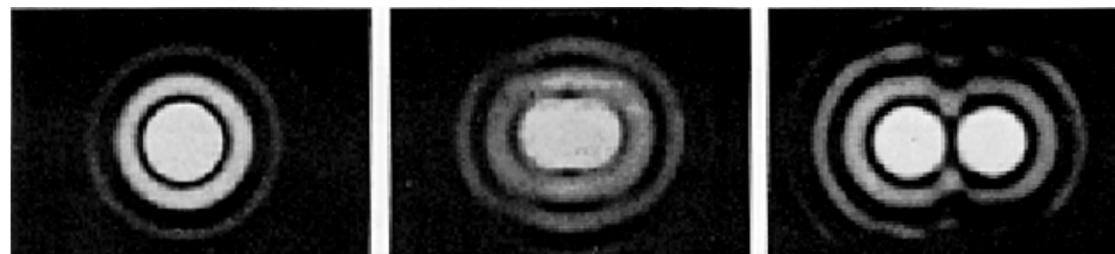


# Image Resolution – Effect of Numerical Aperture

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



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



$$\text{Lord Rayleigh's criterion: } \delta^R = 0.61 \frac{\lambda}{\text{NA}}$$

Physiologically motivated !!!

$$\delta_z^R = 2 \frac{\lambda n}{\text{NA}^2}$$

Aka Abbe  
diffraction  
limit

# Image Resolution and magnification

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

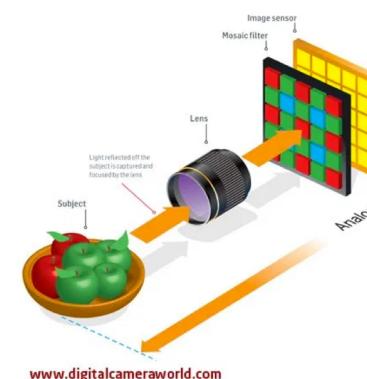
$$\delta^R = 0.61 \frac{\lambda}{NA} \quad \delta_z^R = 2 \frac{\lambda n}{NA^2}$$

Magnification



- 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?

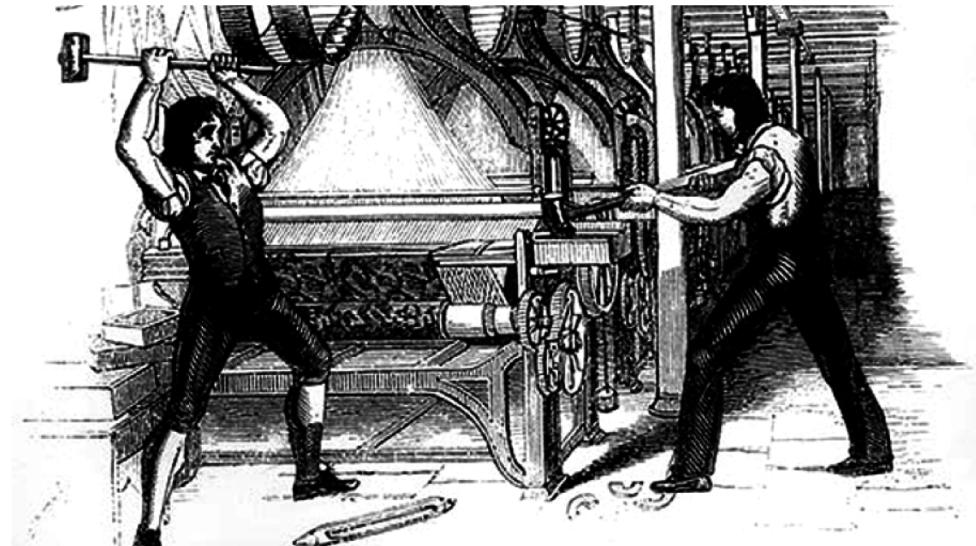


Bunch of monkeys with  
different attributes

- 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

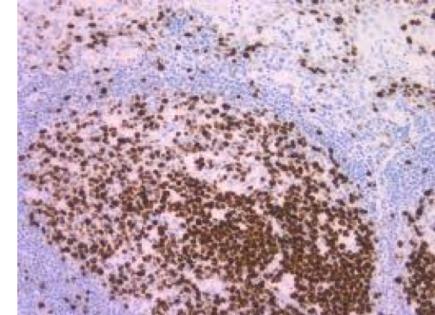
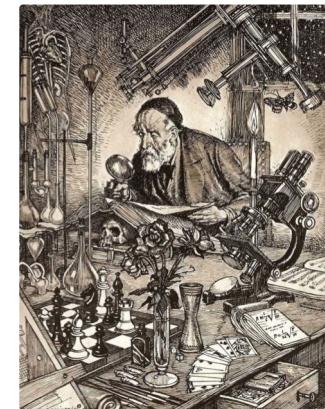


Image analysis was often performed manually,  
usually on a subset of the data (“cherry picking”)  
And heuristic methods that “make sense”

# 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  $\sim 10^6$  Images/experiment. Manual image analysis becomes impractical.

**Imaging has become a big data + high throughput problem.**

# 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

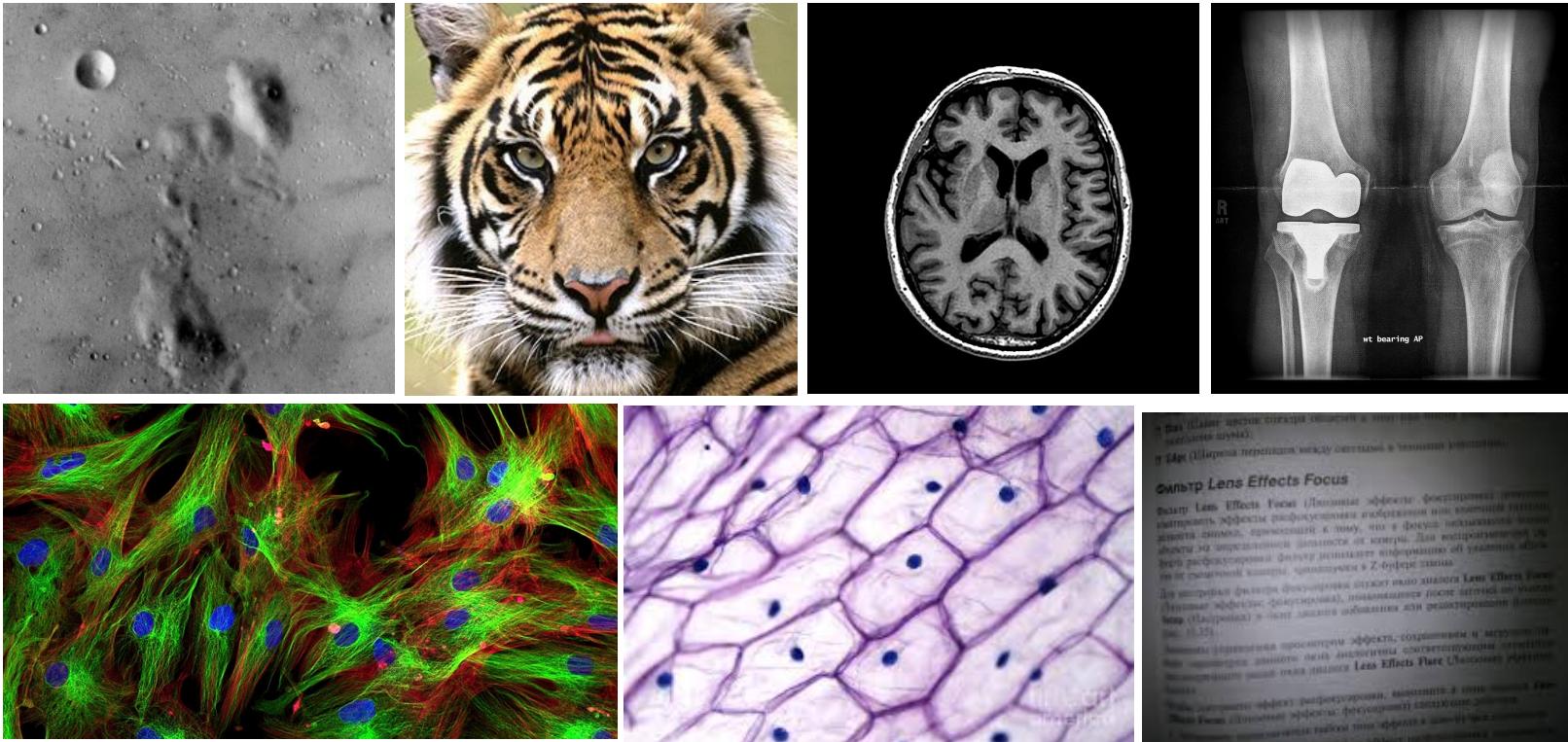
Image Processing

- 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.

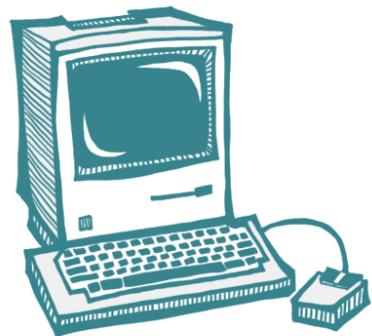
## Some common data formats

- 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.

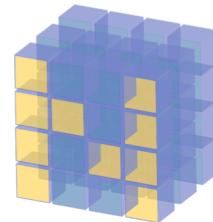
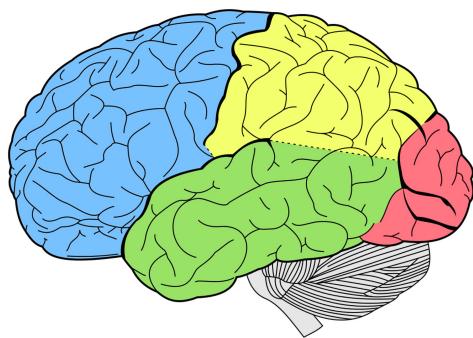
## Some common data formats

- 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



scikit-image  
image processing in python



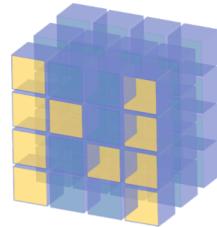
NumPy



# 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



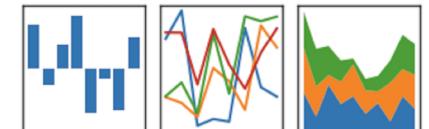
matplotlib

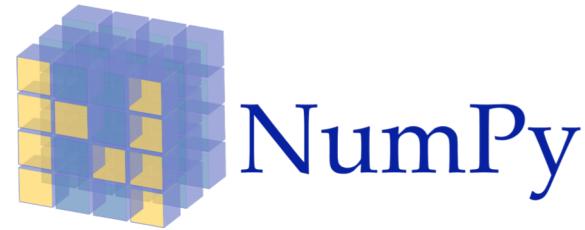


scikit-image  
image processing in python

pandas

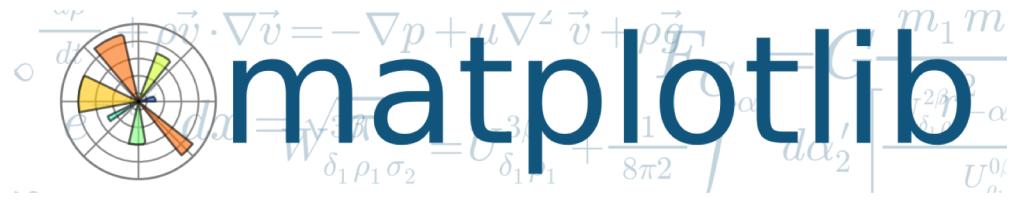
$$y_{it} = \beta' x_{it} + \mu_i + \epsilon_{it}$$





NumPy

Advanced matrix calculator and  
high-level math



Graphing, plotting, and viewing  
images

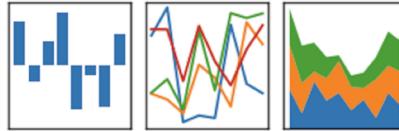


scikit-image  
image processing in python

# 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>

The screenshot shows the GitHub repository page for 'alonyan / DIP'. The page title is 'alonyan / DIP' and the description is 'UCLA QCBio Collaboratory workshop 21: Digital Image Analysis with Python'. Key statistics shown include 2 commits, 1 branch, 0 releases, 1 contributor, and an MIT license. A yellow callout box with the text 'Download the repo' points to the 'Clone or download' button.

alonyan / DIP

UCLA QCBio Collaboratory workshop 21: Digital Image Analysis with Python

Manage topics

2 commits 1 branch 0 releases 1 contributor MIT

Branch: master ▾ New pull request Create new file Upload files Find file Clone or download ↗

Alon Oyler-Yaniv and Alon Oyler-Yaniv first commit

20190311 first commit 29 seconds ago

LICENSE Initial commit 8 minutes ago

README.md Initial commit 8 minutes ago

Download the repo ↗

# Cloning/Downloading the course repository

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

The screenshot shows the GitHub repository page for 'alonyan / DIP'. The page includes the repository name, a description ('UCLA QCBio Collaboratory workshop 21: Digital Image Analysis with Python'), and various statistics like 2 commits, 1 branch, 0 releases, 1 contributor, and an MIT license. At the bottom, there's a table of files: '20190311' (first commit), 'LICENSE' (Initial commit), 'README.md' (Initial commit), and another 'README.md' entry. A prominent green button labeled 'Clone or download' is visible, with a dropdown menu showing options for cloning with HTTPS or SSH, and links to 'Open in Desktop' and 'Download ZIP'. A yellow callout box with the text 'Download the repo' and an arrow points to the 'Clone or download' button.

# 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:

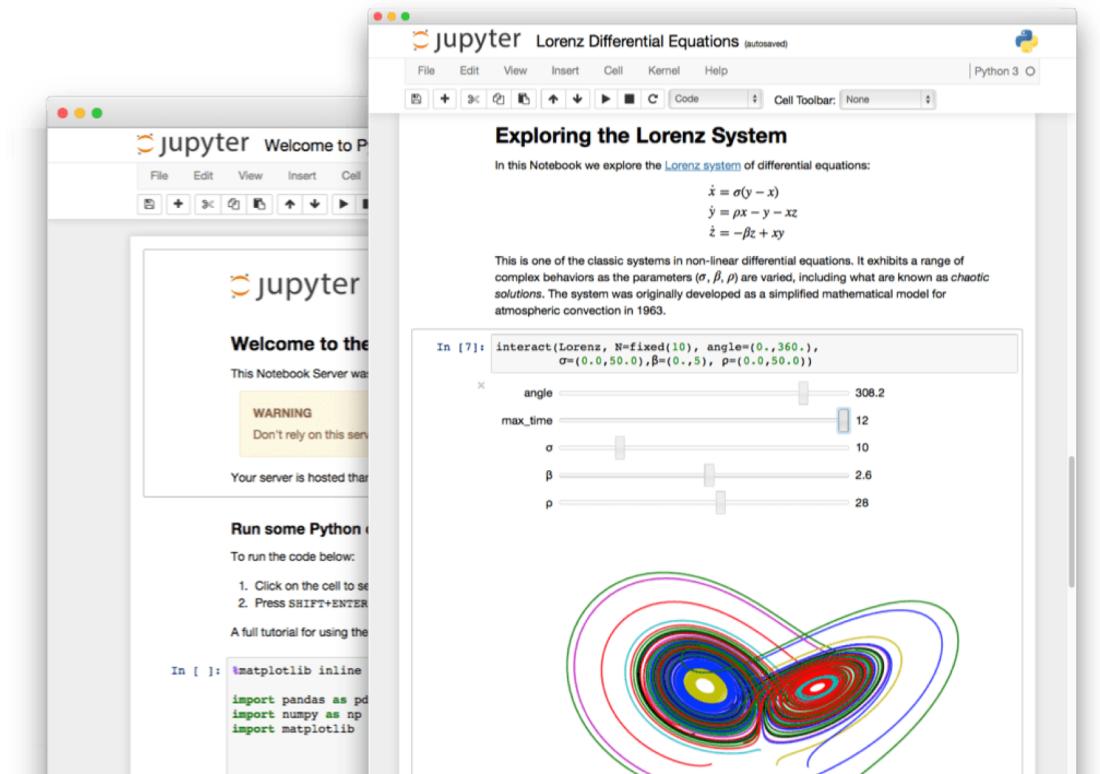
*git clone <https://github.com/alonyan/DIP>*

# Getting started with Jupyter Notebooks

# Jupyter notebooks

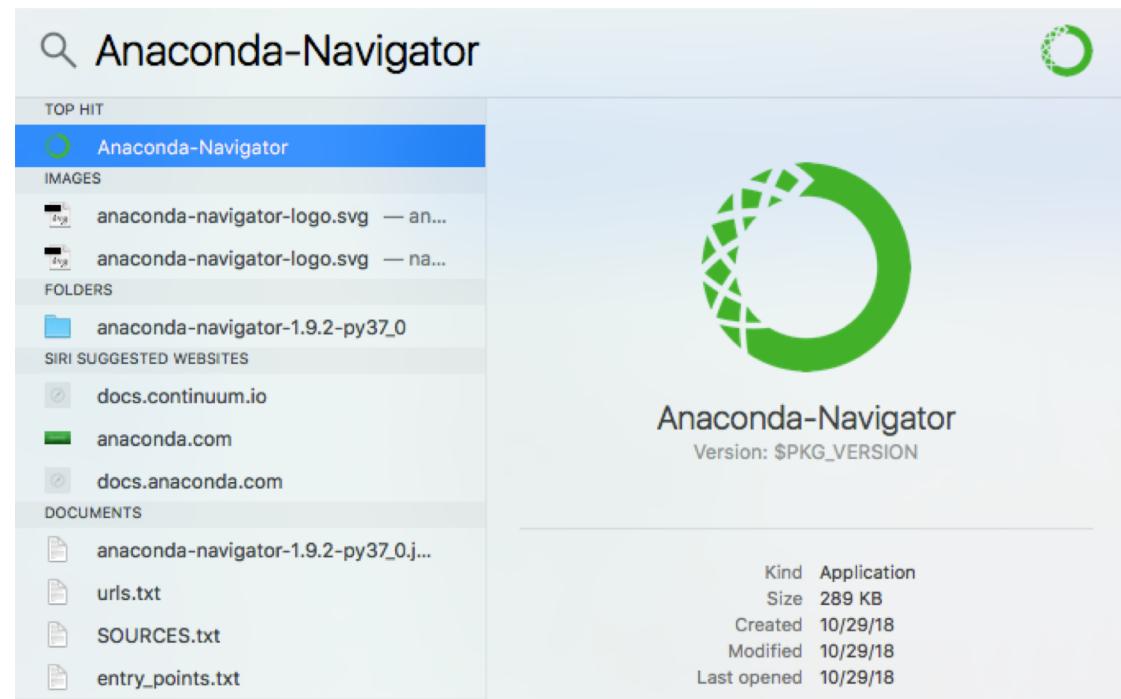
Jupyter notebooks are special documents that allow you to create and share documents **live code**.

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



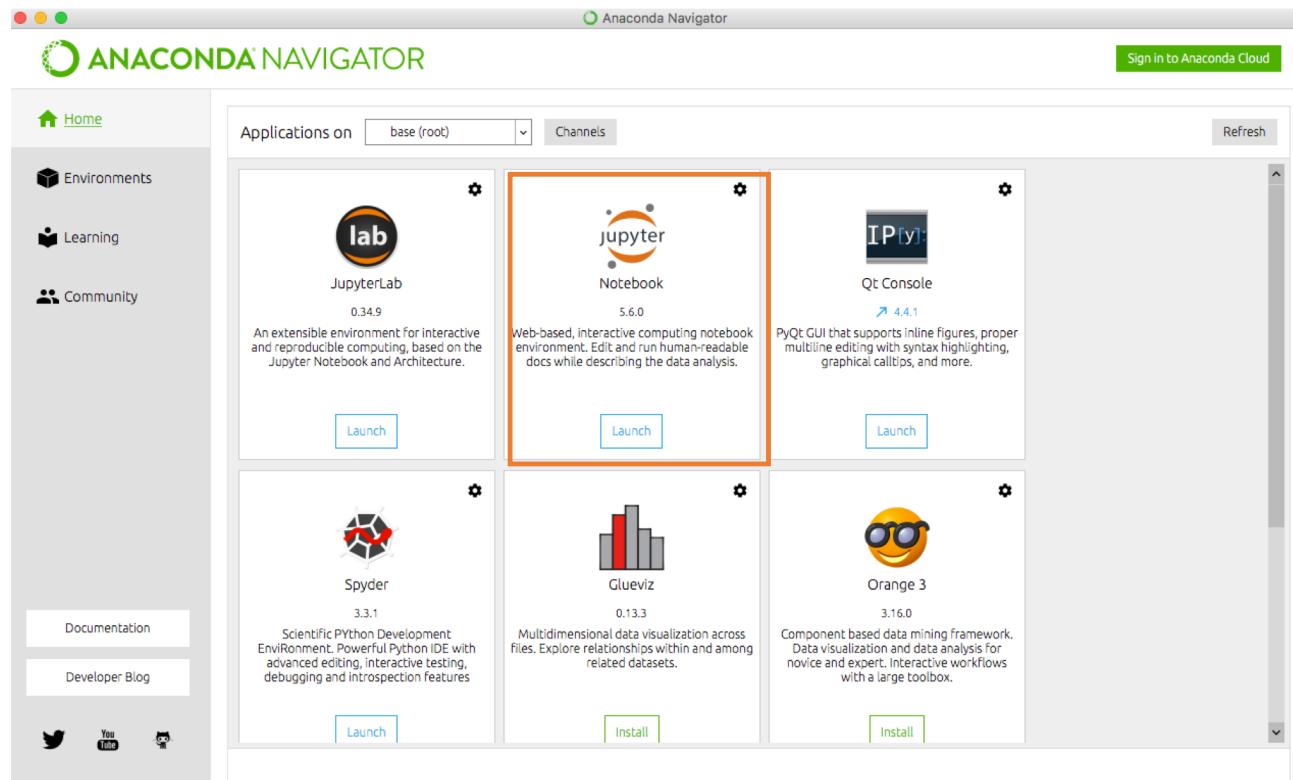
# Getting started in Anaconda

- 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.

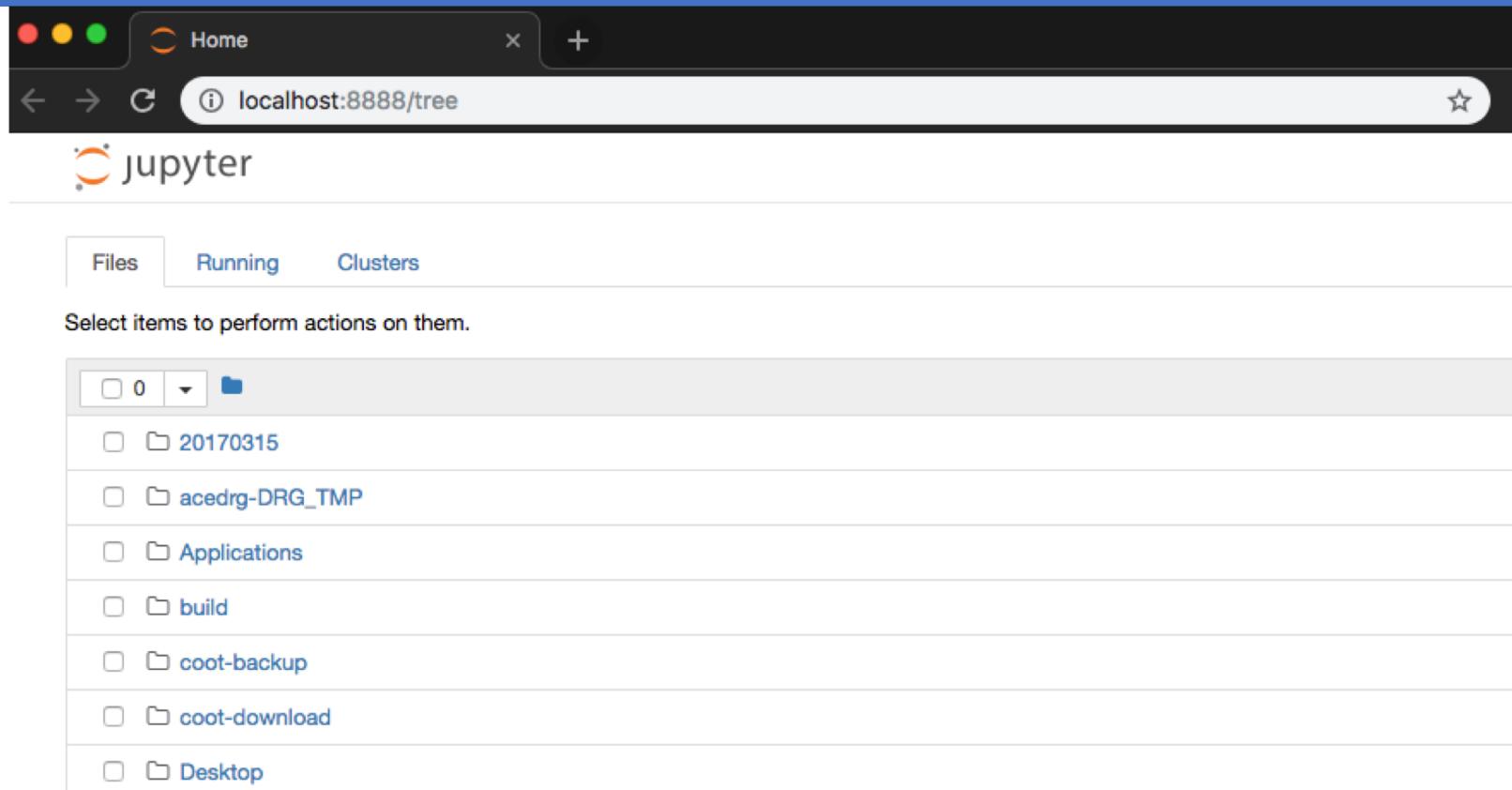


# Getting started from Anaconda

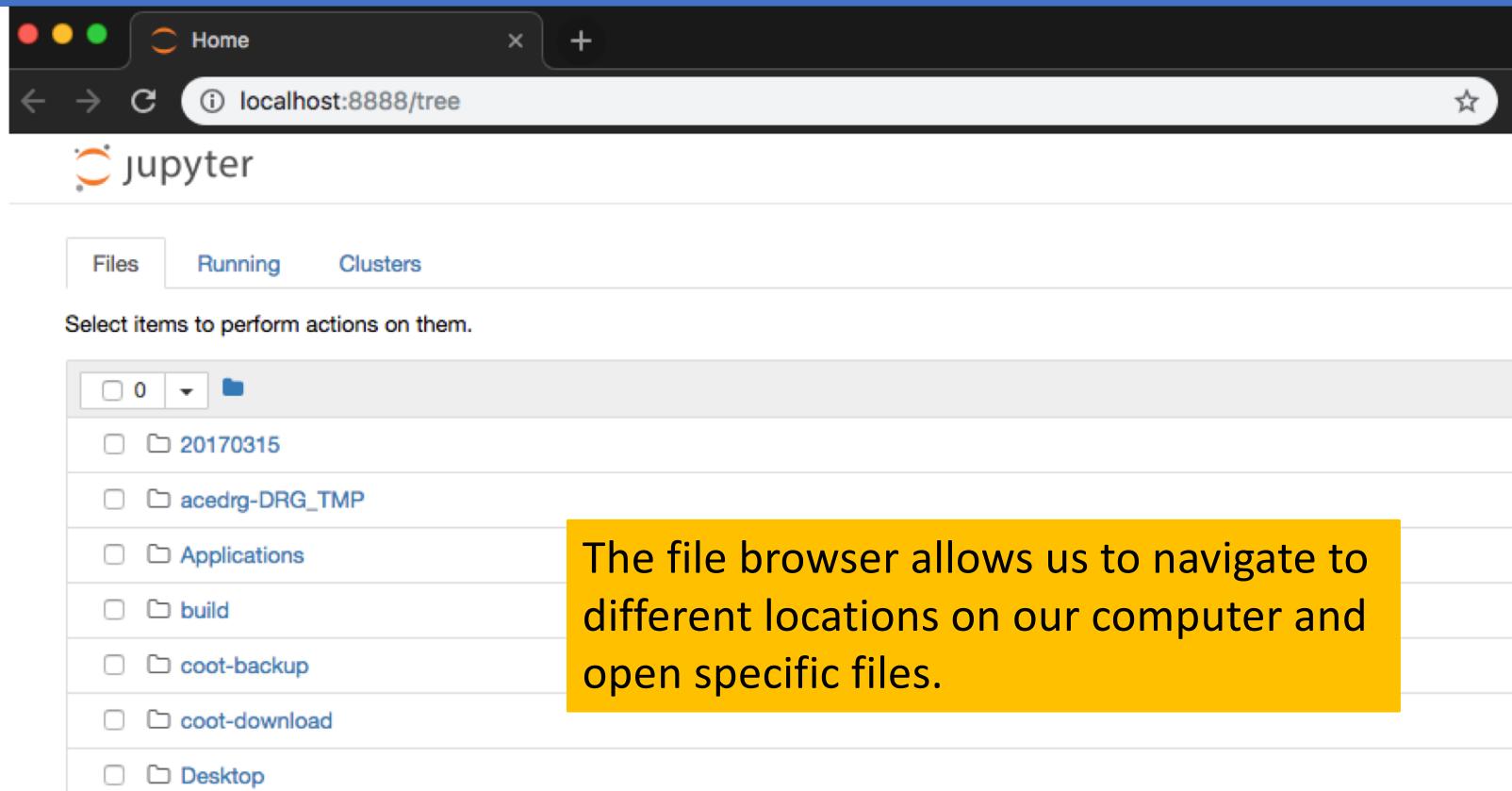
Click on the launch button under the **jupyter notebook** icon.  
After a few moments the dashboard should open in your default web browser.



# The jupyter dashboard



# The jupyter dashboard



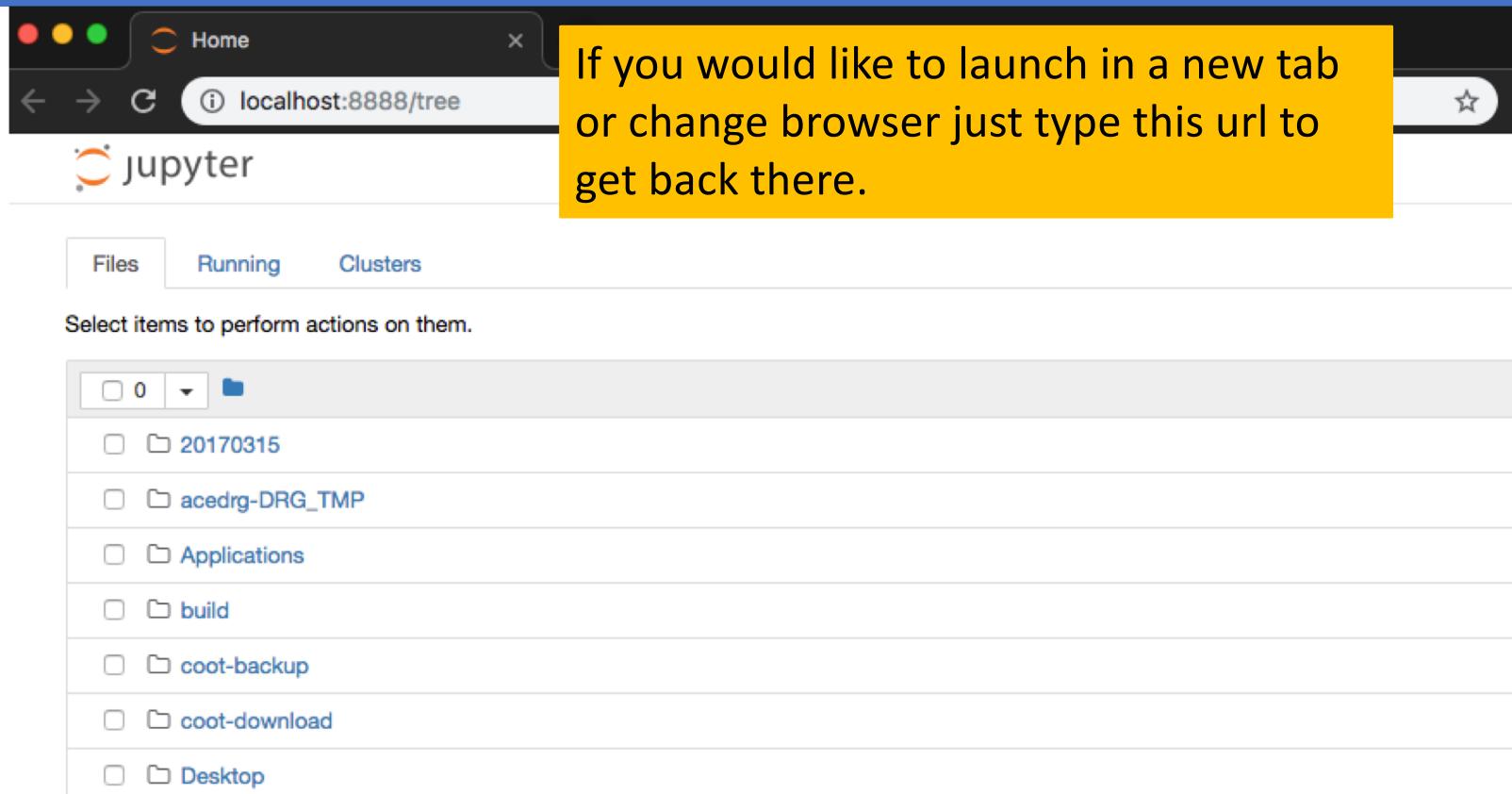
A screenshot of the Jupyter Notebook dashboard. At the top, there's a dark header bar with a window control area (red, yellow, green buttons), a "Home" button, a close button ("x"), a plus sign for new tabs, and a back/forward navigation bar. Below the header, the URL "localhost:8888/tree" is displayed. The main content area has a blue header "jupyter". Below it, there are three tabs: "Files" (selected), "Running", and "Clusters". A message "Select items to perform actions on them." is shown above a file browser. The file browser lists several directories: "20170315", "acedrg-DRG\_TMP", "Applications", "build", "coot-backup", "coot-download", and "Desktop". To the right of the file browser, a yellow callout box contains the text: "The file browser allows us to navigate to different locations on our computer and open specific files."

0

- [20170315](#)
- [acedrg-DRG\\_TMP](#)
- [Applications](#)
- [build](#)
- [coot-backup](#)
- [coot-download](#)
- [Desktop](#)

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

# The jupyter dashboard



# The jupyter dashboard

Navigate to wherever you saved the repository

→ C ⓘ localhost:8888/tree/Documents/Repos/DIP/20190311/JupyterNot... ☆ ABP 0 ... F | 🌐

jupyter

Files Running Clusters

Select items to perform actions on them.

Upload New ↘

<input type="checkbox"/>	0	▼	📁 / Documents / Repos / DIP / 20190311 / JupyterNotebooks	Name ↓	Last Modified	File size
<input type="checkbox"/>	...				seconds ago	
<input type="checkbox"/>	📁 images				10 minutes ago	
<input type="checkbox"/>	📝 DIP_AOY_Student.ipynb				2 days ago	53.3 kB
<input type="checkbox"/>	📝 DIP_AOY_Teacher.ipynb				2 days ago	7.8 MB

# The jupyter dashboard

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

The screenshot shows the Jupyter Notebook dashboard interface. At the top, there's a blue header bar with the title 'The jupyter dashboard'. Below it is a yellow banner containing the text: '\*.ipynb is the suffix of a notebook. Open the Student version (preferably) or the Teacher version of the nb.' The main area has a dark grey header with a back arrow, a refresh icon, and the URL 'localhost:8888/tree/Documents'. Below the header is the Jupyter logo. The dashboard features three tabs: 'Files' (selected), 'Running', and 'Clusters'. Under the 'Files' tab, there's a message 'Select items to perform actions on them.' followed by a set of buttons for 'Upload', 'New ▾', and a refresh icon. A file list table is shown, with columns for checkboxes, file count (0), dropdown menu, folder path ('/ Documents / Repos / DIP / 20190311 / JupyterNotebooks'), sorting options ('Name ↓', 'Last Modified', 'File size'), and file details ('seconds ago', '10 minutes ago', '2 days ago 53.3 kB', '2 days ago 7.8 MB').

	0	...	/ Documents / Repos / DIP / 20190311 / JupyterNotebooks	Name ↓	Last Modified	File size
<input type="checkbox"/>	0	...			seconds ago	
<input type="checkbox"/>	<input type="checkbox"/>	images			10 minutes ago	
<input type="checkbox"/>	<input checked="" type="checkbox"/>	DIP_AOY_Student.ipynb			2 days ago	53.3 kB
<input type="checkbox"/>	<input checked="" type="checkbox"/>	DIP_AOY_Teacher.ipynb			2 days ago	7.8 MB

# Time to process

