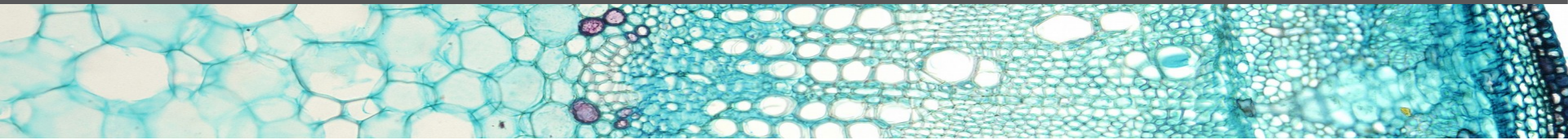


Programming bioimage analysis workflows in python and R using Jupyter Notebook

MCB



MRI

Montpellier Ressources
Imagerie



Inserm



MUSE

Outline



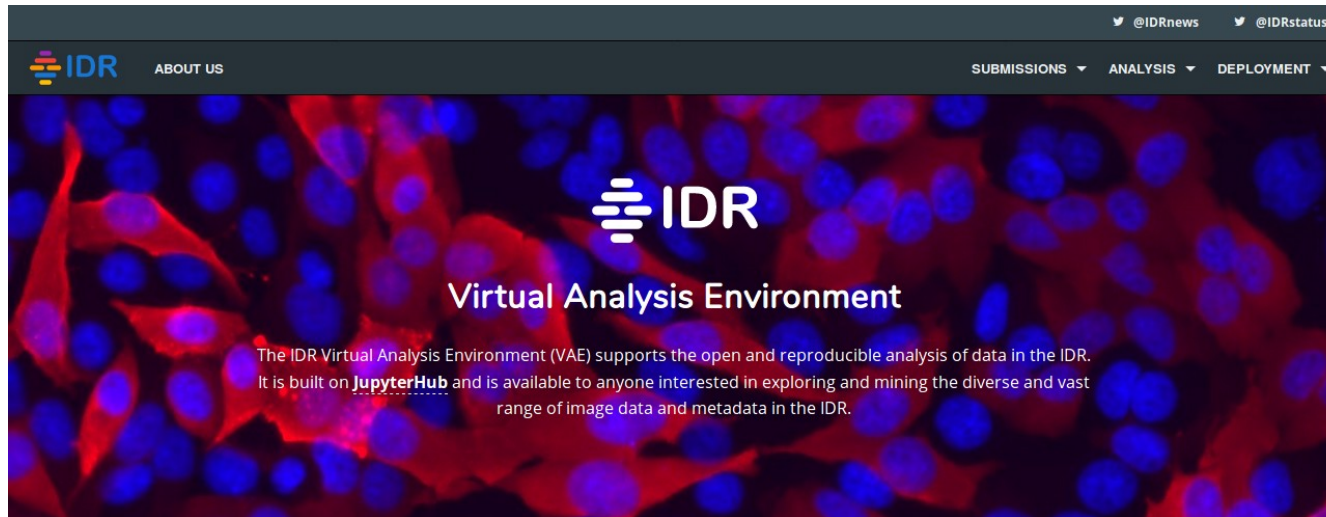
- 1) Motivation
- 2) Example IDR
- 3) Jupyter Notebook
- 4) OMERO
- 5) Python client for the OMERO Blitz API
- 6) Python in image analysis
- 7) R in statistical analysis

Interest of using Jupyter



- ✓ Reproducible Research
- ✓ Education
- ✓ Keep track of data analysis work
- ✓ Prototyping
- ✓ Share notebooks
- ✓ Open-source project
- ✓ Multi-language
- ✓ Interactive Output





- x Public data repository
 - ✓ Data from published scientific studies
- ✓ Image Database - OMERO
 - ✓ Data that is frequently accessed and cited
 - ✓ Links to public genetic or chemical database
 - ✓ Links to cell and tissue phenotype
- x Analyze data – Jupyter
 - ✓ Enable re-analysis
 - ✓ Analysis of gene networks

Eleanor Williams, Josh Moore, Simon W Li, Gabriella Rustici, Aleksandra Tarkowska, Anatole Chessel, Simone Leo, Bálint Antal, Richard K Ferguson, Ugis Sarkans, Alvis Brazma, Rafael E Carazo Salas, Jason R Swedlow. Image Data Resource: a bioimage data integration and publication platform. Nature Methods, 2017; DOI: [10.1038/nmeth.4326](https://doi.org/10.1038/nmeth.4326)

source: <https://idr.openmicroscopy.org/about/>



Browser tabs: jupyter-workshop-mifol x | JupyterLab x | IDR: Image Data Resource x | +

Address bar: <https://idr-analysis.openmicroscopy.org/public/user/3a4286e9-076f-43b0-ab1a> | Rechercher

Navigation: Les plus visités | Getting Started

File Edit View Run Kernel Hub Tabs Settings Help

Files: + | notebooks

Name	Last Modified
docker	5 months ago
includes	5 months ago
CalculateSharpness.ipynb	3 minutes ago
Figure_1_Sampling_of...	5 months ago
GeneNetwork.ipynb	5 months ago
GenesToPhenotypes.I...	2 minutes ago
Getting_Started.ipynb	5 months ago
IDR_API_example_sc...	2 minutes ago
PCAAanalysisOfCharm...	5 months ago
QueryIDRWithGeneLis...	5 months ago
README.ipynb	5 months ago
RohnPhenotypeCluste...	5 months ago
SysgroOverview.ipynb	5 months ago
SysgroRoilLength.ipynb	5 months ago
Using_Jupyter.ipynb	5 months ago
Dockerfile	5 months ago
README.md	5 months ago

Launcher x | CalculateSharp x | GenesToPheno x | IDR_API_exam x

Markdown v | IDR Python 2

Get phenotypes associated with a list of genes from high content screens

This notebook takes a list of gene symbols and queries the IDR for phenotypes associated with the genes in high content screens.

```
In [1]: import json
import csv
import pandas as pd
```

Set up where to query and session

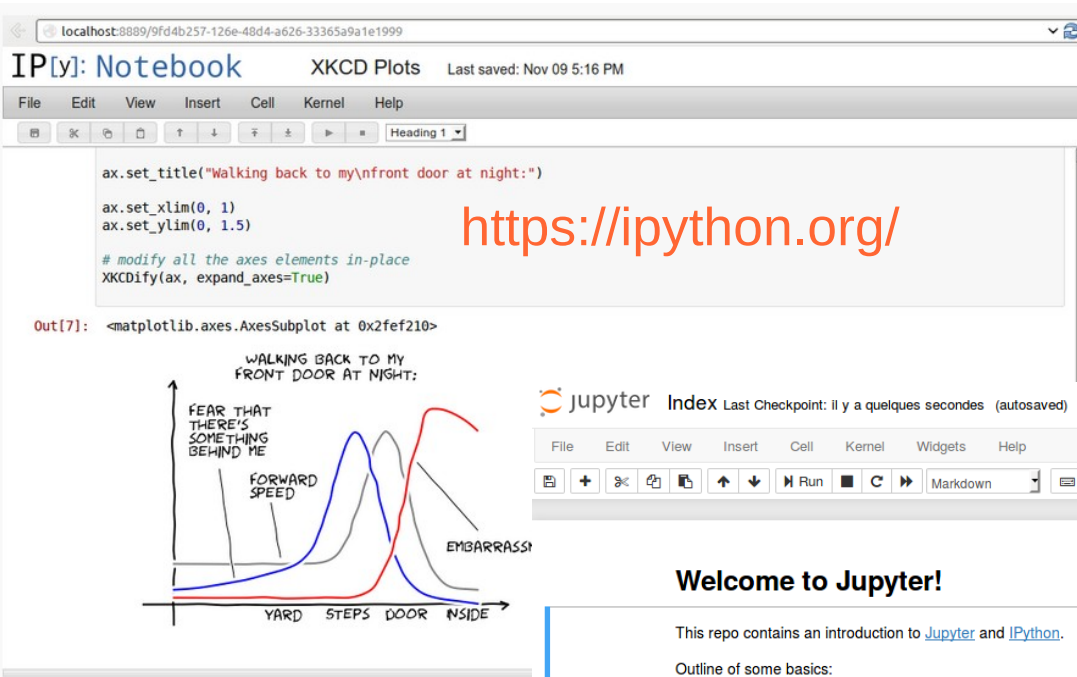
```
In [2]: import requests

INDEX_PAGE = "http://idr.openmicroscopy.org/webclient/?experimenter=-1"

# create http session
with requests.Session() as session:
    request = requests.Request('GET', INDEX_PAGE)
    prepped = session.prepare_request(request)
    response = session.send(prepped)
    if response.status_code != 200:
        response.raise_for_status()
```

source: <https://idr.openmicroscopy.org/about/>

Jupyter Notebook



<https://ipython.org/>

jupyter Index Last Checkpoint: il y a quelques secondes (autosaved)

File Edit View Insert Cell Kernel Widgets Help

Trusted Python 3

Run

Welcome to Jupyter!

This repo contains an introduction to [Jupyter](#) and [IPython](#).

Outline of some basics:

- [Notebook Basics](#)
- [IPython - beyond plain python](#)
- [Markdown Cells](#)
- [Rich Display System](#)
- [Custom Display logic](#)
- [Running a Secure Public Notebook Server](#)
- [How Jupyter works](#) to run code in different languages.

<http://jupyter.org/index.html>

You can also get this tutorial and run it on your laptop:

```
git clone https://github.com/ipython/ipython-in-depth
```

Install IPython and Jupyter:

with [conda](#):

```
conda install ipython jupyter
```

Jupyter Notebook



← → ↺ 🏠 ⓘ localhost:8889/notebooks/image_analysis.ipynb ⌵ ⋮ ⌵ ☆ 🔍 Rechercher 📁 📄 📖 ☰

⚙️ Les plus visités 🌐 Getting Started

jupyter image_analysis Last Checkpoint: 19/09/2018 (autosaved) Logout

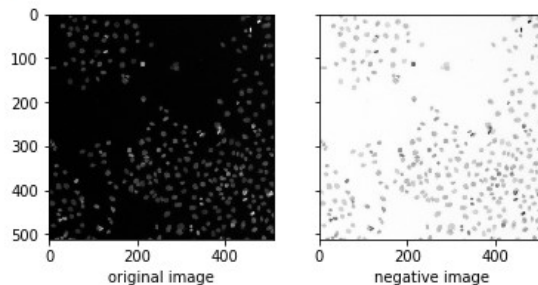
File Edit View Insert Cell Kernel Widgets Help Trusted Python 2 ○

📁 + 🔍 📄 ⬆️ ⬆️ ▶️ Run 🛑 ↺ ▶️ Code 🗒️

Manipulating images

Being numeric arrays, images can be manipulated by any of python's arithmetic operators. For example, we can produce a **negative image** by simply subtracting the image from its maximum value.

```
In [6]: nuc_neg = nuc.max() - nuc
fig, axes = plt.subplots(ncols=2)
ax0, ax1 = axes.flatten()
ax0.imshow(nuc, cmap='gray')
ax0.set_xlabel("original image")
ax1.imshow(nuc_neg, cmap='gray')
ax1.set_xlabel("negative image")
plt.setp(ax1.get_yticklabels(), visible=False)
plt.show()
```



Jupyter Notebook

What is it ?



localhost:8889/notebooks/image_analysis.ipynb

Rechercher

Les plus visités Getting Started

Jupyter image_analysis Last Checkpoint: 19/09/2018 (autosaved)

Logout

File Edit View Insert Cell Kernel Widgets Help

Trusted Python 2

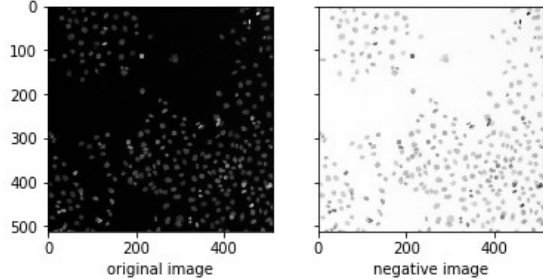
Run

Manipulating images

Being numeric arrays, images can be manipulated by any of python's arithmetic operators. For example, we can produce a **negative image** by simply subtracting the image from its maximum value.

```
In [6]: nuc_neg = nuc.max() - nuc
fig, axes = plt.subplots(ncols=2)
ax0, ax1 = axes.flatten()
ax0.imshow(nuc, cmap='gray')
ax0.set_xlabel("original image")
ax1.imshow(nuc_neg, cmap='gray')
ax1.set_xlabel("negative image")
plt.setp(ax1.get_yticklabels(), visible=False)
plt.show()
```

Live Code



Jupyter Notebook

What is it ?



localhost:8889/notebooks/image_analysis.ipynb

Rechercher

Les plus visités Getting Started

jupyter image_analysis Last Checkpoint: 19/09/2018 (autosaved)

Logout

File Edit View Insert Cell Kernel Widgets Help

Trusted Python 2

Code

Manipulating images

Being numeric arrays, images can be manipulated by any of python's arithmetic operators. For example, we can produce a **negative image** by simply subtracting the image from its maximum value.

```
In [6]: nuc_neg = nuc.max() - nuc
fig, axes = plt.subplots(ncols=2)
ax0, ax1 = axes.flatten()
ax0.imshow(nuc, cmap='gray')
ax0.set_xlabel("original image")
ax1.imshow(nuc_neg, cmap='gray')
ax1.set_xlabel("negative image")
plt.setp(ax1.get_yticklabels(), visible=False)
plt.show()
```

Visualization

Jupyter Notebook

What is it ?



localhost:8889/notebooks/image_analysis.ipynb

Rechercher

Les plus visités Getting Started

Jupyter image_analysis Last Checkpoint: 19/09/2018 (autosaved)

Logout

File Edit View Insert Cell Kernel Widgets Help

Trusted Python 2

Code

Manipulating images

Being numeric arrays, images can be manipulated by any of python's arithmetic operators. For example, we can produce a **negative image** by simply subtracting the image from its maximum value.

Rich text for explanation

```
In [6]: nuc_neg = nuc.max() - nuc
fig, axes = plt.subplots(ncols=2)
ax0, ax1 = axes.flatten()
ax0.imshow(nuc, cmap='gray')
ax0.set_xlabel("original image")
ax1.imshow(nuc_neg, cmap='gray')
ax1.set_xlabel("negative image")
plt.setp(ax1.get_yticklabels(), visible=False)
plt.show()
```

original image negative image

Jupyter Notebook



jupyter Introduction (autosaved)

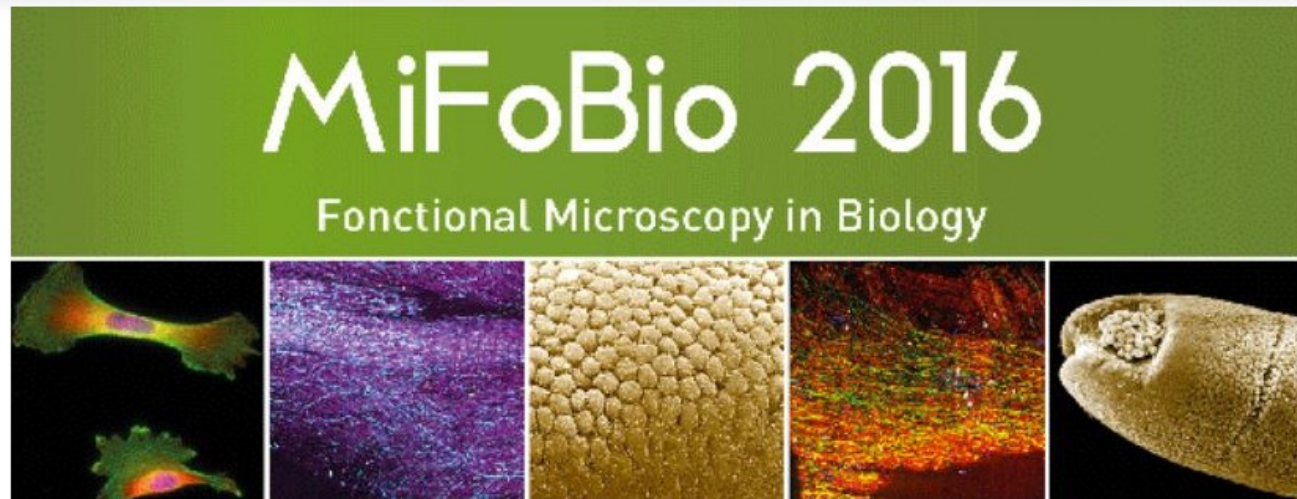


Logout

File Edit View Insert Cell Kernel Widgets Help

Not Trusted

Python 3



Edit Metadata

Equations


LaTeX

Equations can be used inline, so you can have short equations like $E = mc^2$ within the text. Longer equations should be separated from the text:

$$f(x) = \sum_{n=0}^{\infty} A_n \cos\left(\frac{n\pi x}{L}\right) + \sum_{n=1}^{\infty} B_n \sin\left(\frac{n\pi x}{L}\right)$$

Jupyter Notebook



 jupyter tuto_image_analysis_pipeline (autosaved)



Logout

File Edit View Insert Cell Kernel Widgets Help

Not Trusted

Python 3

New Notebook

Open...

Make a Copy...

Rename...

Save and Checkpoint

Revert to Checkpoint

Print Preview

Download as

Trust Notebook

Close and Halt

Notebook (.ipynb)

Python (.py)

HTML (.html)

Reveal.js slides (.html)

Markdown (.md)

reST (.rst)

LaTeX (.tex)

PDF via LaTeX (.pdf)

import modules & packages

able to use declarations and definitions of an existing module, you should use the *import* clause.

ort module [as alias]

mple

/2018/mifobio/workshop/exercices

Use to import additional modules or packages for the future use in this pipeline.

```
In [2]: #the matplotlib package is a great tool for plotting
import matplotlib.pyplot as plt

#the numpy package for scientific computing is also a great tool for manipulating arrays
import numpy

#the image processing package skimage and ndimage from scipy
import skimage
import scipy.ndimage
```



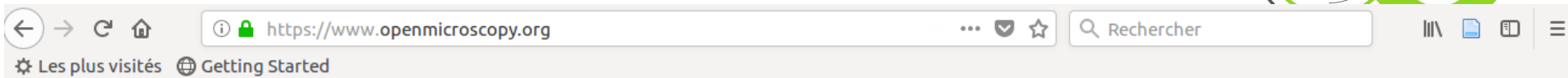
Jupyter notebook is a web application that allows you to run live code, embed visualization and explanatory text all in one place



EXERCICES

01-Introduction.ipynb

OMERO



OMERO is client-server software for managing, visualizing and analyzing microscopy images and associated metadata.



OMERO for

Scientists

Your microscopy images are securely stored but shareable and available from anywhere you have internet access.



OMERO for

Developers

Join the OME community and extend OMERO's functionality to suit your individual needs.



OMERO for

Your Institution

OMERO securely stores image data and enables all of your users to manage the data from the same platform.

source: <https://www.openmicroscopy.org/>

OMERO

What is it ?

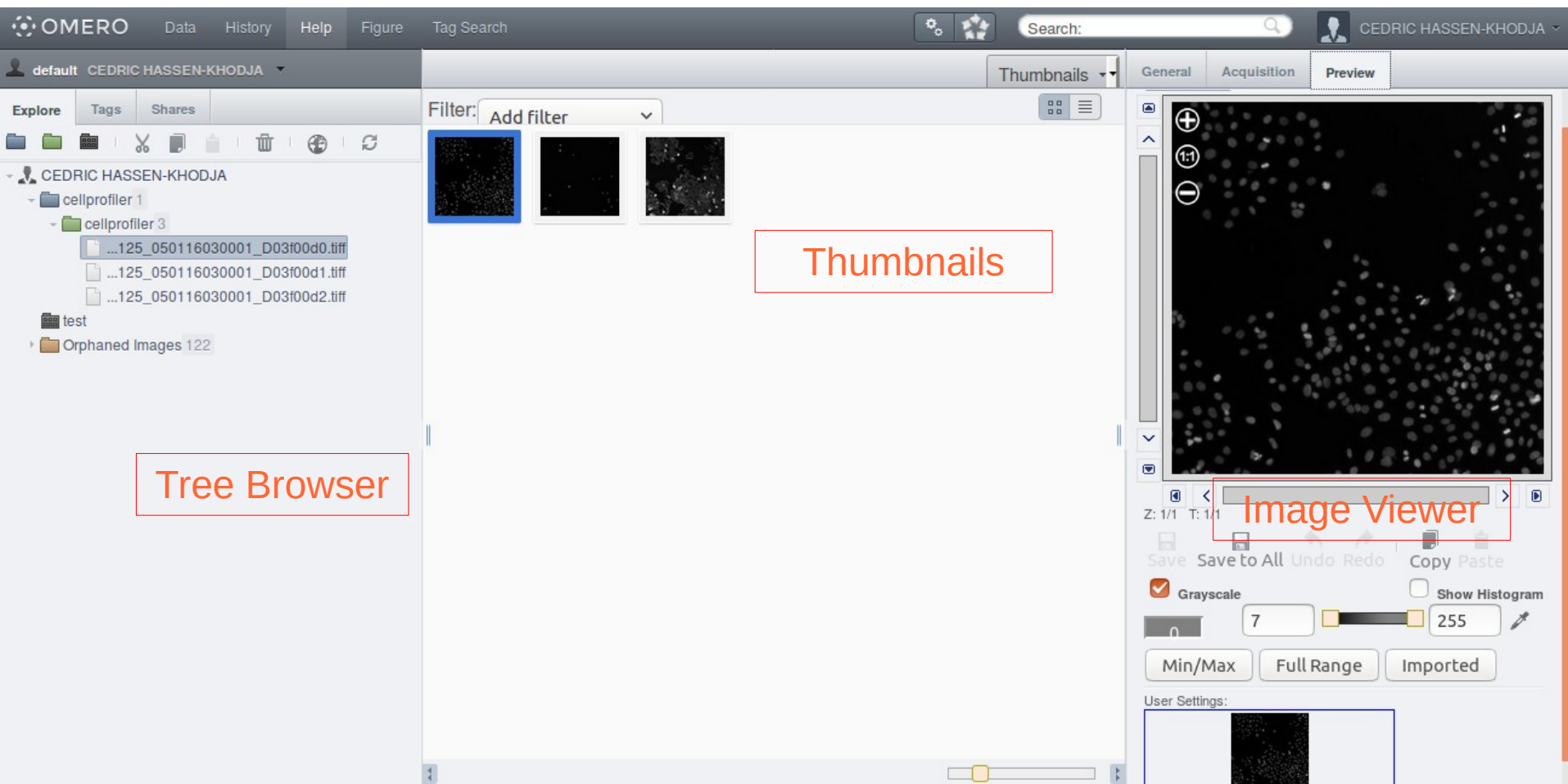


The screenshot displays the OMERO web interface. The top navigation bar includes links for Data, History, Help, Figure, and Tag Search. The main interface is divided into three primary sections:

- Tree Browser (left):** A hierarchical view of the data repository. It shows a user profile for CEDRIC HASSEN-KHODJA, followed by a folder named 'cellprofiler 1', which contains a sub-folder 'cellprofiler 3'. Inside 'cellprofiler 3', there are three image files: '...125_050116030001_D03f00d0.tiff', '...125_050116030001_D03f00d1.tiff', and '...125_050116030001_D03f00d2.tiff'. Below these are a 'test' folder and an 'Orphaned Images 122' folder.
- Thumbnails (center):** A grid of image thumbnails. The first thumbnail is highlighted with a blue border. A red box labeled 'Thumbnails' is overlaid on this section.
- Metadata (right):** A panel showing detailed information about the selected image. It includes fields for Image ID (441563), Owner (CEDRIC HASSEN-KHODJA), and various technical specifications: Import Date (2016-09-19 08:51:16), Dimensions (XY) (512 x 512), Pixels Type (uint8), Pixels Size (XYZ) (352.78 x 352.78 x 1), Z-sections/Timepoints (1 x 1), Channels (0), and ROI Count (0). Below these are sections for Tags, Key-Value Pairs, Tables, Attachments, Comments, and Ratings, all showing a count of 0. A red box labeled 'Metadata' is overlaid on the bottom right of this panel.

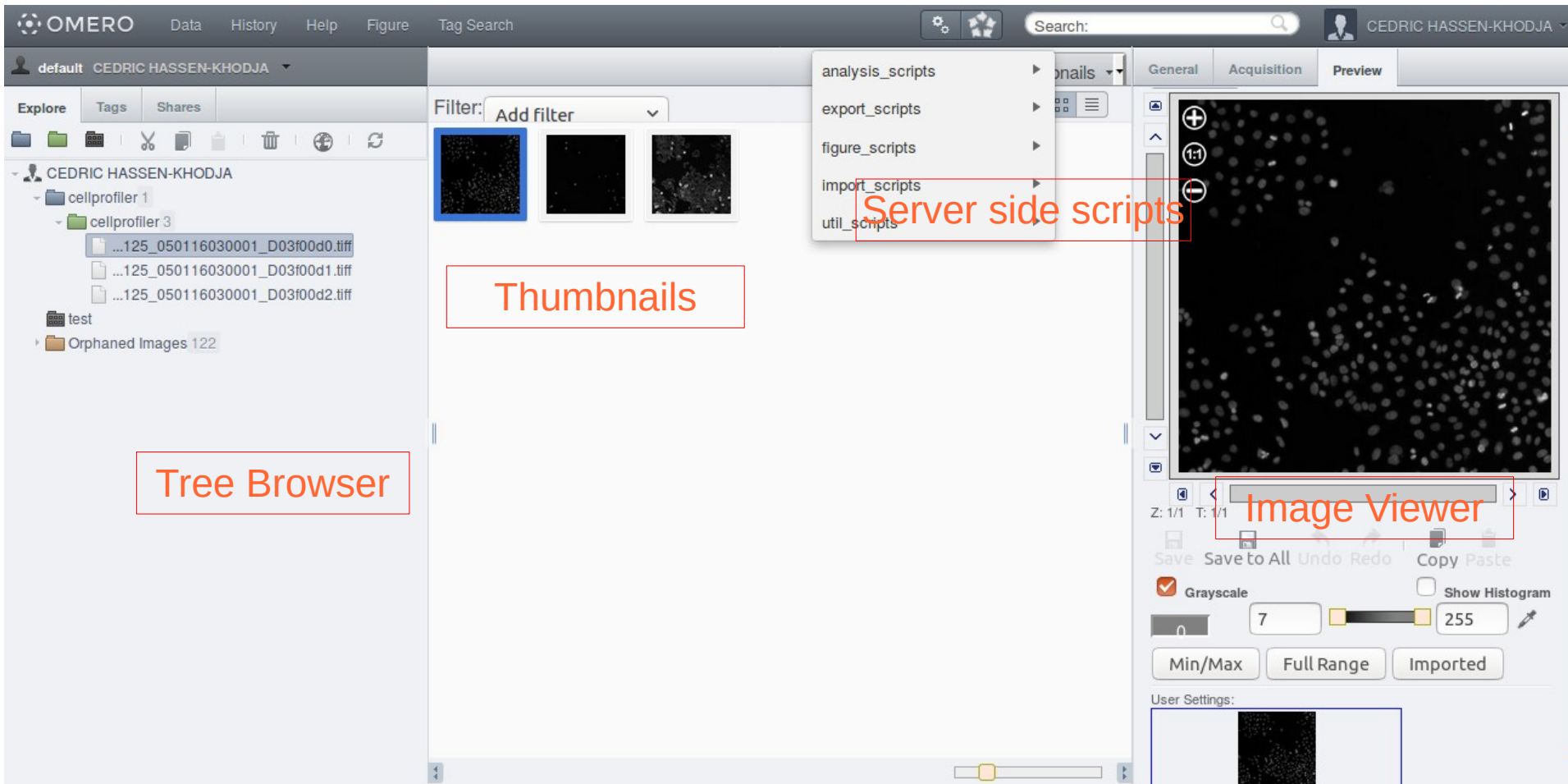
OMERO

What is it ?



OMERO

What is it ?



Python Blitz API



PYTHON BLITZ API

An easy-to-use API for OMERO making it easier to work with your data in Python.



<https://docs.openmicroscopy.org/omero/5.4.8/developers/Python.html>

[View Developer Guide](#)

source: <https://www.openmicroscopy.org/omero/features/analyze/>

Python Blitz API



Connect to omero server:

- › import **omero.gateway** package
- › Use the wrapper **BlitzGateway**:

```
class omero.gateway._BlitzGateway(username=None, passwd=None,  
                                   client_obj=None, group=None,  
                                   clone=False, try_super=False,  
                                   host=None, port=None,  
                                   | extra_config=None, secure=False,  
                                   anonymous=True, useragent=None,  
                                   userip=None)
```

Get user information:

- › Use the **getUser()** function to return current experimenter
- › Use **getName()** function to return the experimenter name
- › Use **getFullName()** function to return the full name of this experimenter
- › **getGroupsMemberOf()** function return current users groups
- › **getGroupFromContext()** function return your current group

source: <https://www.openmicroscopy.org/omero/features/analyze/>



Get user information:

- Use the `getUser()` function to return current experimenter
- Use `getName()` function to return the experimenter name
- Use `getFullName()` function to return the full name of this experimenter
- `getGroupsMemberOf()` function return current users groups
- `getGroupFromContext()` function return your current group

Get Project / Dataset information:

- Use `getObjects()` function to retrieve Objects by type, e.g. “Image” returns generator of appropriate BlitzObjectWrapper type, e.g. ImageWrapper

```
getObjects(obj_type, ids=None, params=None, attributes=None,  
           respect_order=False, opts=None)
```

- `listChildren()` lists available child objects

```
listChildren(ns=None, val=None, params=None)
```

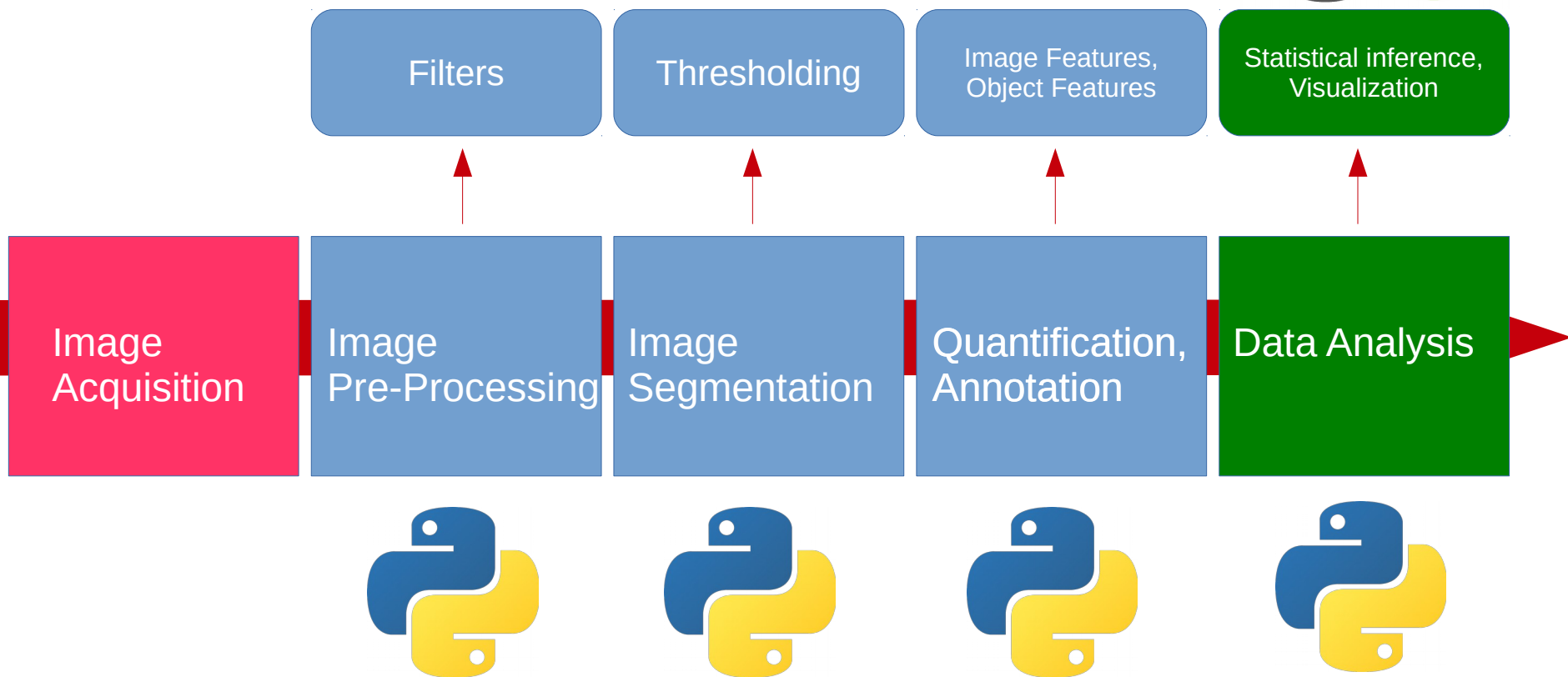
source: <https://www.openmicroscopy.org/omero/features/analyze/>



EXERCICES

02-Introduction_Omero_PartI.ipynb

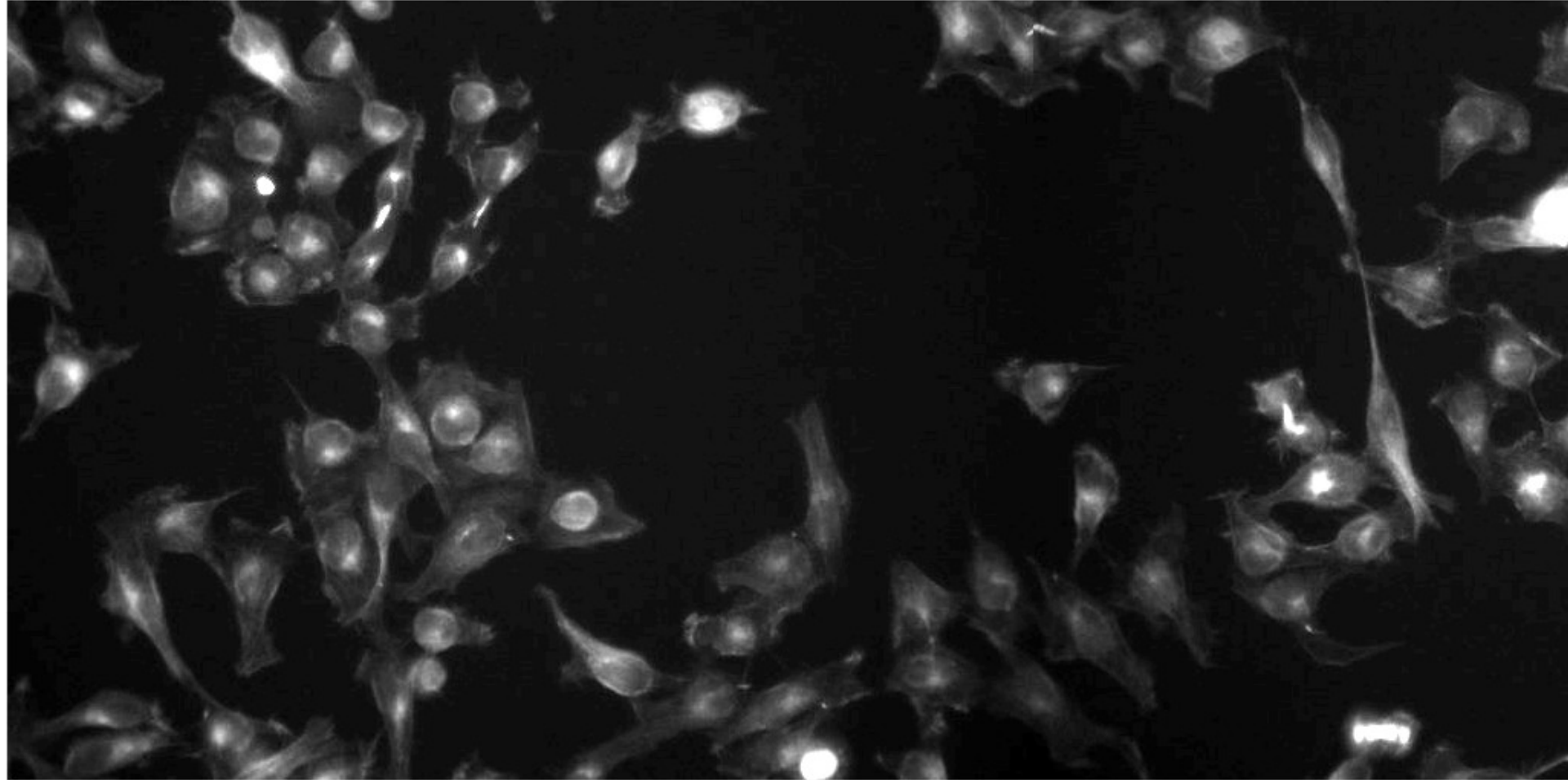
Python for image analysis



Digital Images



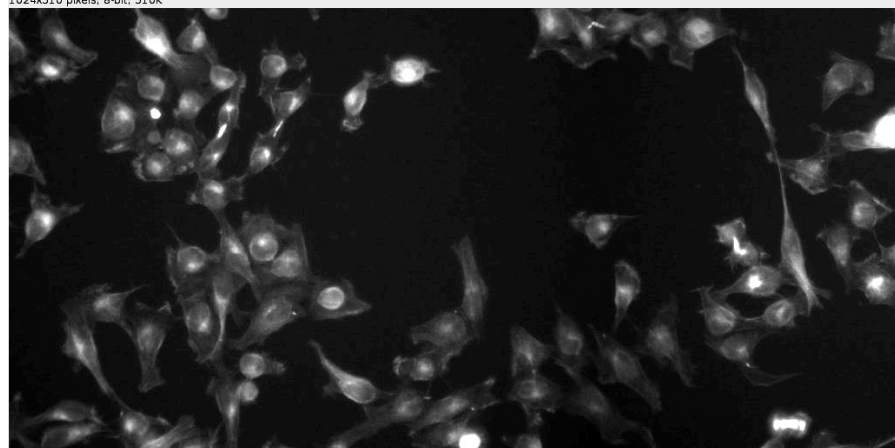
1024x510 pixels; 8-bit; 510K



Digital Images



1024x510 pixels; 8-bit; 510K



21	21	21	22	22	22	22	23	25	25
21	21	21	22	22	22	22	22	24	24
21	21	21	22	22	22	22	22	23	24
21	21	21	21	22	22	22	22	23	24
21	21	21	21	22	22	22	22	23	24
21	21	21	21	21	22	22	22	22	23
21	21	21	21	21	22	22	22	21	23
20	21	21	21	21	22	22	22	21	22
23	22	22	21	21	21	21	20	22	22
22	22	21	21	21	21	21	20	21	22

21	21	21	22	22	22	22	23	25	25
21	21	21	22	22	22	22	22	24	24
21	21	21	22	22	22	22	22	23	24
21	21	21	21	22	22	22	22	23	24
21	21	21	21	22	22	22	22	23	24
21	21	21	21	21	22	22	22	22	23
21	21	21	21	21	22	22	22	22	23
21	21	21	21	21	22	22	22	21	23
20	21	21	21	21	22	22	22	21	22
23	22	22	21	21	21	21	20	22	22
22	22	21	21	21	21	21	20	21	22

Image (y,x)

21	21	21	22	22	22	22	23	25	25		
21	21	21	21	22	22	22	22	23	25	25	
21	21	21	21	21	22	22	22	22	23	25	25
21	21	21	21	21	22	22	22	22	22	24	24
21	21	21	21	21	22	22	22	22	22	23	24
21	21	21	21	21	21	22	22	22	22	23	24
21	21	21	21	21	21	22	22	22	22	23	24
20	21	21	21	21	21	21	22	22	22	22	23
23	20	21	21	21	21	21	22	22	22	21	23
22	23	20	21	21	21	21	22	22	22	21	22
	22	23	22	22	21	21	21	21	20	22	22
		22	22	21	21	21	21	21	20	21	22

Stack (z,y,x)

21	21	21	22	22	22	22	23	25	25		
21	21	21	21	22	22	22	22	23	25	25	
21	21	21	21	21	22	22	22	22	23	25	25
21	21	21	21	21	22	22	22	22	22	24	24
21	21	21	21	21	22	22	22	22	22	23	24
21	21	21	21	21	21	22	22	22	22	23	24
21	21	21	21	21	21	22	22	22	22	23	24
20	21	21	21	21	21	21	22	22	22	22	23
23	20	21	21	21	21	21	22	22	22	21	23
22	23	20	21	21	21	21	22	22	22	21	22
	22	23	22	22	21	21	21	21	20	22	22
		22	22	21	21	21	21	21	20	21	22

Channel (c,y,x)

Image (y,x)

Time series (t)

Time series (t,y,x)

Data types & unexpected output



uint8	Unsigned integer (0 to 255)
uint16	Unsigned integer (0 to 65535)
float32	Single precision float: sign bit, 8 bits exponent, 23 bits mantissa

in python 2

$20 / 3 = 6$

from __future__ import division

$20 / 3 = 6.666666666666667$

$20 // 3 = 6$

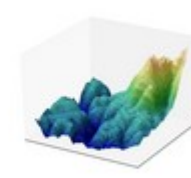
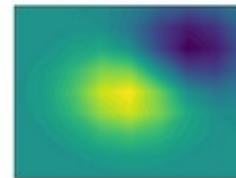
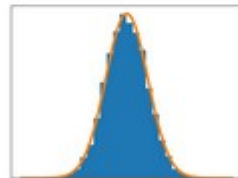
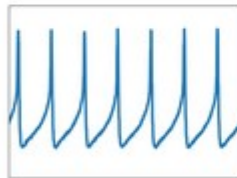
source: <https://docs.scipy.org/doc/numpy-1.13.0/user/basics.types.html>

Visualization



[home](#) | [examples](#) | [tutorials](#) | [API](#) | [docs](#) »

Matplotlib is a Python 2D plotting library which produces publication quality figures in a variety of hardcopy formats and interactive environments across platforms. Matplotlib can be used in Python scripts, the Python and **IPython** shells, the **Jupyter** notebook, web application servers, and four graphical user interface toolkits.



Matplotlib tries to make easy things easy and hard things possible. You can generate plots, histograms, power spectra, bar charts, errorcharts, scatterplots, etc., with just a few lines of code. For examples, see the [sample plots](#) and [thumbnail gallery](#).

For simple plotting the `pypLOT` module provides a MATLAB-like interface, particularly when combined with IPython. For the power user, you have full control of line styles, font properties, axes properties, etc, via an object oriented interface or via a set of functions familiar to MATLAB users.

source: <https://matplotlib.org/>

Visualization



matplotlib.pyplot ¶

`matplotlib.pyplot` is a state-based interface to matplotlib. It provides a MATLAB-like way of plotting.

pyplot is mainly intended for interactive plots and simple cases of programmatic plot generation:

<code>figure([num, figsize, dpi, facecolor, ...])</code>	Create a new figure.
<code>imshow(X[, cmap, norm, aspect, ...])</code>	Display an image, i.e.
<code>show(*args, **kw)</code>	Display a figure.

source: <https://matplotlib.org/>



Get images metadata

- `image = conn.getObject("Image", imageId)`
 - `image.getSizeX()`
 - `image.getSizeY()`
 - `image.getSizeZ()`
 - `image.getSizeC()`
 - `Image.getSizeT()`

Display images

- `rendered_image = image.renderImage(z, t, compression=0.9)`
 - get a compressed and rendered version of the slice z and frame t of the image
- `rendered_image.show()`

source: <https://www.openmicroscopy.org/omero/features/analyze/>



EXERCICES

Import_Handle_Local_Image_Data.ipynb
Introduction_Omero_PartII.ipynb



Image Filtering is used to:

- Remove noise
- Sharpen contrast
- Highlight contours
- Detect edges

Preprocessing - Filters



$\frac{1}{9}$	$\frac{1}{9}$	$\frac{1}{9}$
$\frac{1}{9}$	$\frac{1}{9}$	$\frac{1}{9}$
$\frac{1}{9}$	$\frac{1}{9}$	$\frac{1}{9}$

A: 3 × 3
square

$\frac{1}{25}$	$\frac{1}{25}$	$\frac{1}{25}$	$\frac{1}{25}$	$\frac{1}{25}$
$\frac{1}{25}$	$\frac{1}{25}$	$\frac{1}{25}$	$\frac{1}{25}$	$\frac{1}{25}$
$\frac{1}{25}$	$\frac{1}{25}$	$\frac{1}{25}$	$\frac{1}{25}$	$\frac{1}{25}$
$\frac{1}{25}$	$\frac{1}{25}$	$\frac{1}{25}$	$\frac{1}{25}$	$\frac{1}{25}$
$\frac{1}{25}$	$\frac{1}{25}$	$\frac{1}{25}$	$\frac{1}{25}$	$\frac{1}{25}$

B: 5 × 5
square

0	0	$\frac{1}{13}$	0	0
0	$\frac{1}{13}$	$\frac{1}{13}$	$\frac{1}{13}$	0
$\frac{1}{13}$	$\frac{1}{13}$	$\frac{1}{13}$	$\frac{1}{13}$	$\frac{1}{13}$
0	$\frac{1}{13}$	$\frac{1}{13}$	$\frac{1}{13}$	0
0	0	$\frac{1}{13}$	0	0

C: Circular,
radius = 1.5

0	$\frac{1}{21}$	$\frac{1}{21}$	$\frac{1}{21}$	0
$\frac{1}{21}$	$\frac{1}{21}$	$\frac{1}{21}$	$\frac{1}{21}$	$\frac{1}{21}$
$\frac{1}{21}$	$\frac{1}{21}$	$\frac{1}{21}$	$\frac{1}{21}$	$\frac{1}{21}$
$\frac{1}{21}$	$\frac{1}{21}$	$\frac{1}{21}$	$\frac{1}{21}$	$\frac{1}{21}$
0	$\frac{1}{21}$	$\frac{1}{21}$	$\frac{1}{21}$	0

D: Circular,
radius = 2

A kernel resembles another (usually small and rectangular) image in which each pixel is known as a filter coefficient and these correspond to the the weights used for scaling

Preprocessing - Filters



<code>skimage.morphology.square</code> (width[, dtype])	Generates a flat, square-shaped structuring element.
<code>skimage.morphology.rectangle</code> (width, height)	Generates a flat, rectangular-shaped structuring element.
<code>skimage.morphology.diamond</code> (radius[, dtype])	Generates a flat, diamond-shaped structuring element.
<code>skimage.morphology.disk</code> (radius[, dtype])	Generates a flat, disk-shaped structuring element.
<code>skimage.morphology.cube</code> (width[, dtype])	Generates a cube-shaped structuring element.
<code>skimage.morphology.octahedron</code> (radius[, dtype])	Generates a octahedron-shaped structuring element.
<code>skimage.morphology.ball</code> (radius[, dtype])	Generates a ball-shaped structuring element.
<code>skimage.morphology.octagon</code> (m, n[, dtype])	Generates an octagon shaped structuring element.
<code>skimage.morphology.star</code> (a[, dtype])	Generates a star shaped structuring element.

You can create your own filter of convolution

source: <http://scikit-image.org/docs/dev/api/skimage.morphology.html>

Preprocessing - Filters



<code>generic_laplace(input, derivative2[, ...])</code>	N-dimensional Laplace filter using a provided second derivative function :Parameters: input : array_like Input array to filter.
<code>laplace(input[, output, mode, cval])</code>	N-dimensional Laplace filter based on approximate second derivatives.
<code>maximum_filter(input[, size, footprint, ...])</code>	Calculates a multi-dimensional maximum filter.
<code>maximum_filter1d(input, size[, axis, ...])</code>	Calculate a one-dimensional maximum filter along the given axis.
<code>median_filter(input[, size, footprint, ...])</code>	Calculates a multidimensional median filter.
<code>minimum_filter(input[, size, footprint, ...])</code>	Calculates a multi-dimensional minimum filter.
<code>minimum_filter1d(input, size[, axis, ...])</code>	Calculate a one-dimensional minimum filter along the given axis.
<code>percentile_filter(input, percentile[, size, ...])</code>	Calculates a multi-dimensional percentile filter.
<code>prewitt(input[, axis, output, mode, cval])</code>	Calculate a Prewitt filter.
<code>rank_filter(input, rank[, size, footprint, ...])</code>	Calculates a multi-dimensional rank filter.
<code>sobel(input[, axis, output, mode, cval])</code>	Calculate a Sobel filter.
<code>uniform_filter(input[, size, output, mode, ...])</code>	Multi-dimensional uniform filter.
<code>uniform_filter1d(input, size[, axis, ...])</code>	Calculate a one-dimensional uniform filter along the given axis.

Segmentation - Thresholding



Thresholding is used to:

- Identify interesting objects
- Binarizing an image

How ?:

- Global vs Adaptive

Segmentation - Thresholding



Otsu's Thresholding Method (1979)

- Based on a very simple idea: Find the threshold that *minimizes the weighted within-class variance*.
- This turns out to be the same as *maximizing the between-class variance*.
- Operates directly on the gray level histogram [*e.g.* 256 numbers, $P(i)$], so it's fast (once the histogram is computed).
- I've used it with considerable success in “murky” situations.

Nobuyuki Otsu (1979). "A threshold selection method from ----- histograms". IEEE Trans. Sys., Man., Cyber. 9 (1): 62–66.

Segmentation - Thresholding



threshold_otsu

```
skimage.filters.threshold_otsu(image, nbins=256)
```

[\[source\]](#)

Return threshold value based on Otsu's method.

Parameters:

image : (N, M) ndarray

Grayscale input image.

nbins : int, optional

Number of bins used to calculate histogram. This value is ignored for integer arrays.

Returns:

threshold : float

Upper threshold value. All pixels with an intensity higher than this value are assumed to be foreground.

Raises:

ValueError

If image only contains a single grayscale value.

source: http://scikit-image.org/docs/dev/api/skimage.filters.html#skimage.filters.threshold_otsu

PostProcessing – Morphological operation



Erosion will make objects smaller

Dilation will make objects bigger

Goal: clean up binary image

Opening consists of an erosion followed by a dilation

Closing is the opposite of opening, i.e. a dilation followed by an erosion



Original

Erosion

Dilation

Opening

Closing

<code>skimage.morphology.erosion (image[, selem, ...])</code>	Return greyscale morphological erosion of an image.
<code>skimage.morphology.dilation (image[, selem, ...])</code>	Return greyscale morphological dilation of an image.
<code>skimage.morphology.opening (image[, selem, out])</code>	Return greyscale morphological opening of an image.
<code>skimage.morphology.closing (image[, selem, out])</code>	Return greyscale morphological closing of an image.

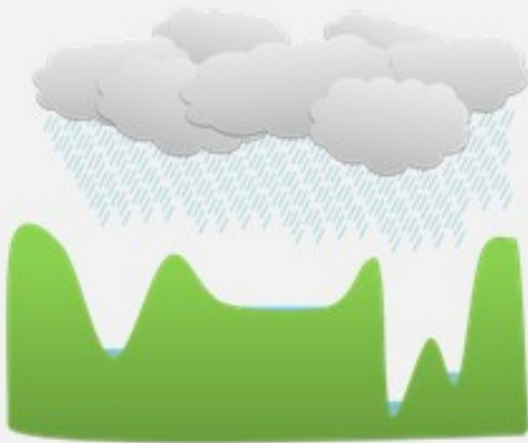
source: <http://scikit-image.org/docs/dev/api/skimimage.morphology.html>



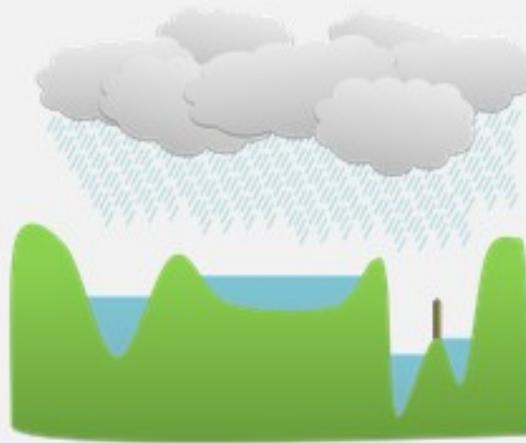
EXERCICES

nuclei_analysis.ipynb

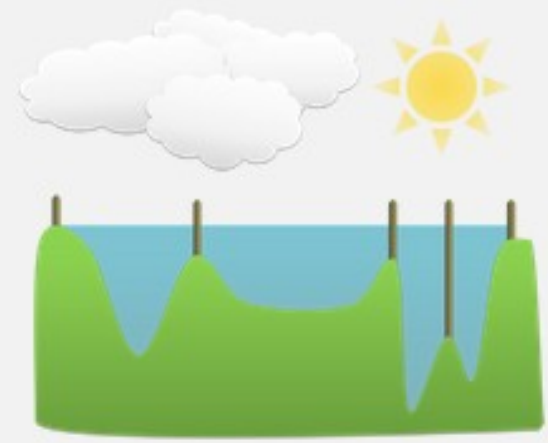
Segmentation - Watershed



A: Water starts to fill the deepest regions first



B: As the water rises, dams are built to prevent overflow



C: The water continues rising until reaching its maximum

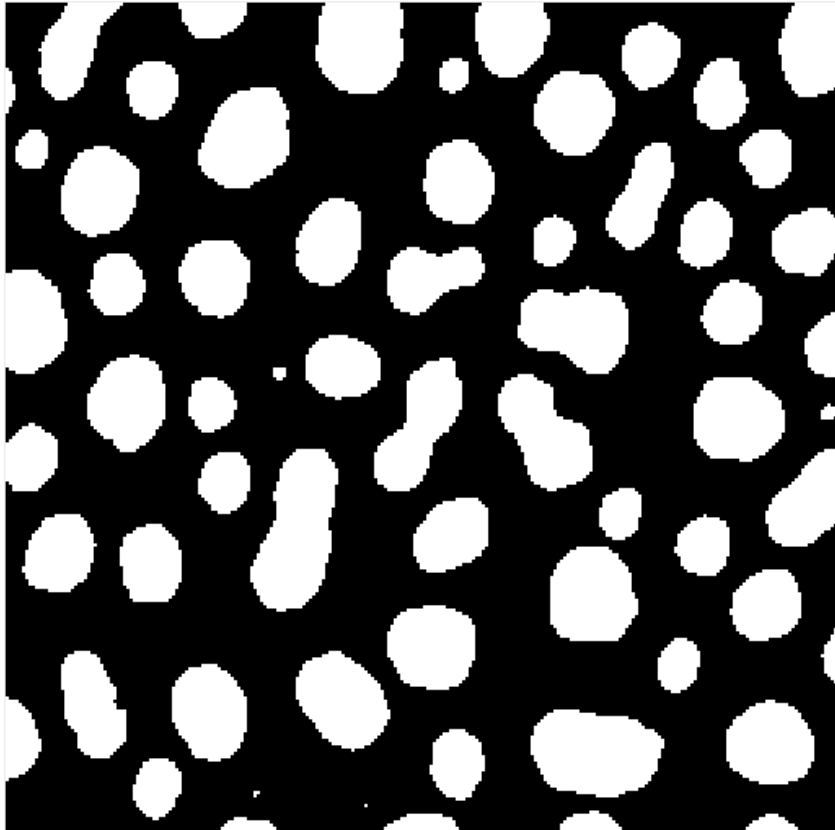
source: Analyzing fluorescence microscopy images with ImageJ. / Bankhead, Peter

Segmentation - Watershed

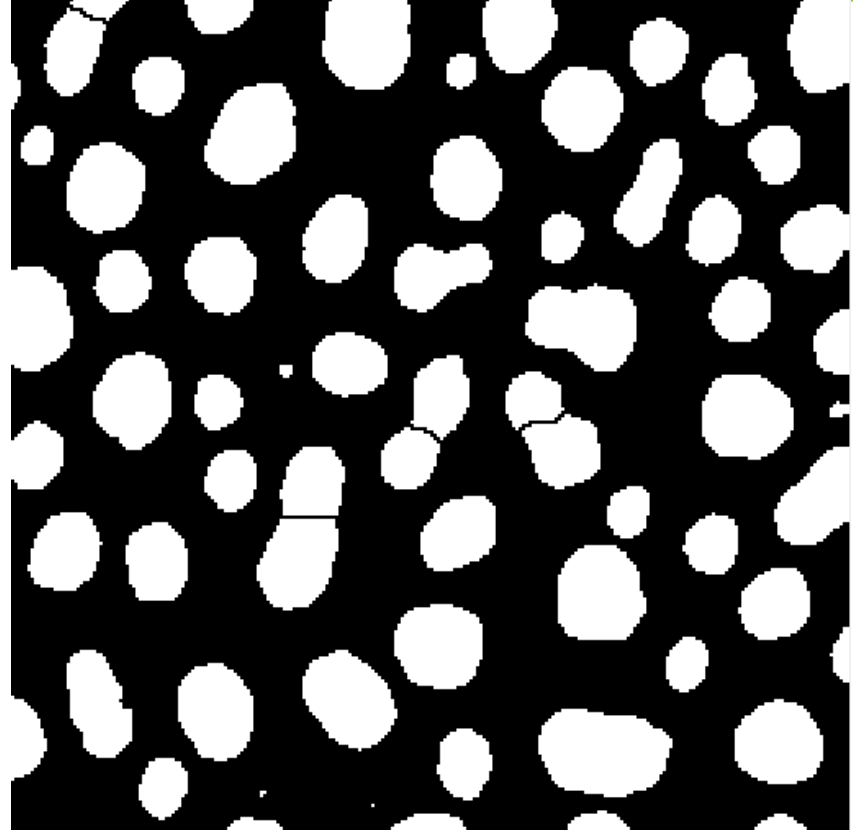


`skimage.morphology.watershed` (image, markers)

Find watershed basins in image flooded from given markers.

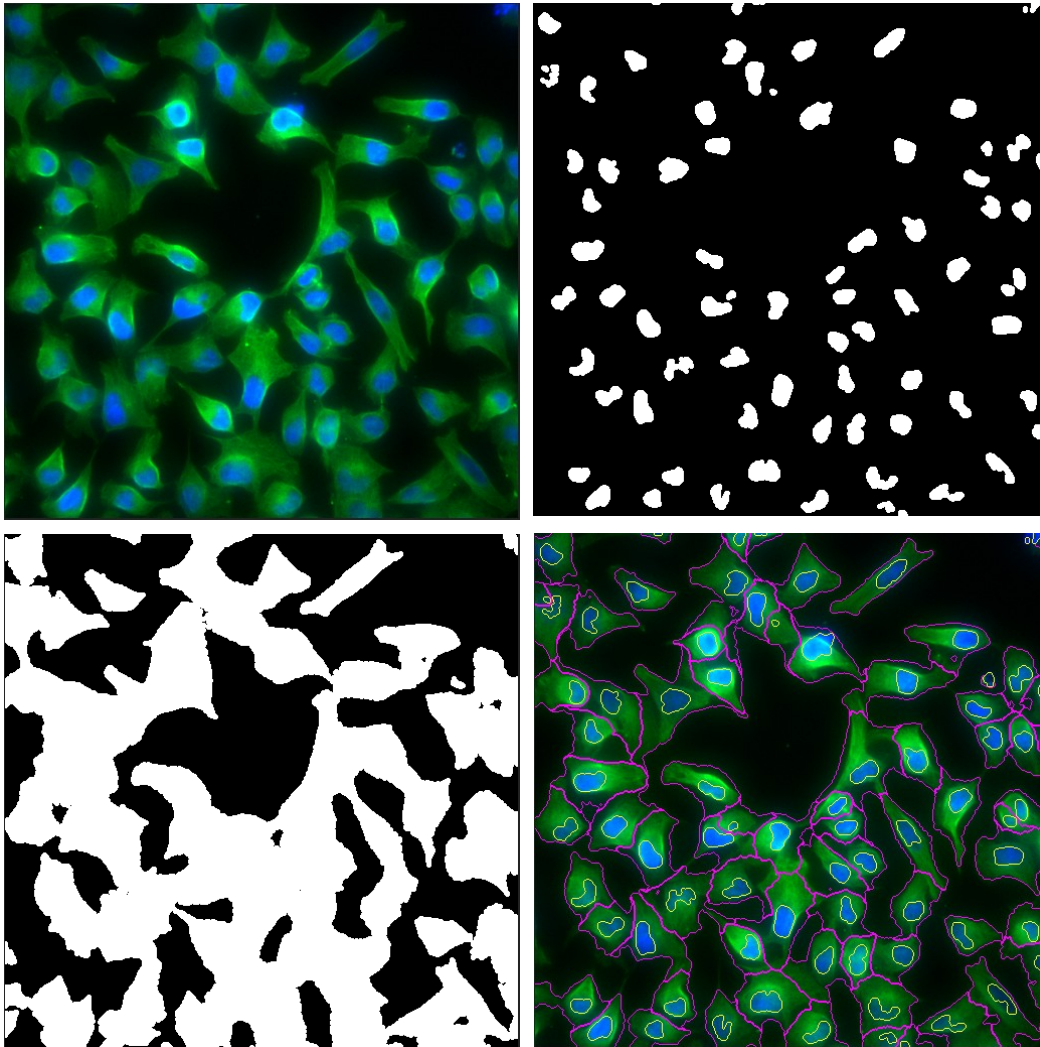


Original



Watershed segmentation
on original image

Segmentation - Voronoi based segmentation on image manifolds



T. Jones, A. Carpenter and P. Golland, "Voronoi-Based Segmentation of Cells on Image Manifolds", CVBIA05 (535-543), 2005



EXERCICES

image_analysis.ipynb

Quantification



- Extract features on interested objects
 - ✓ area, perimeter, circularity, intensities...
- Statistics
 - ✓ Descriptive
 - ✓ Regression models
 - ✓ Classification



EXERCICES

image_analysis.ipynb
(Exercises 6, 7 & 8)

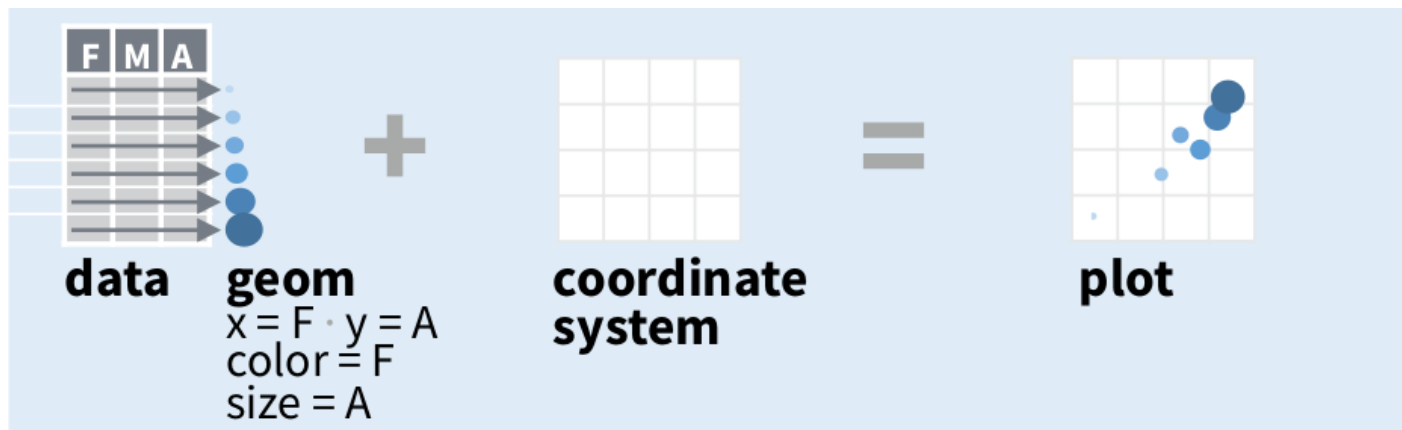
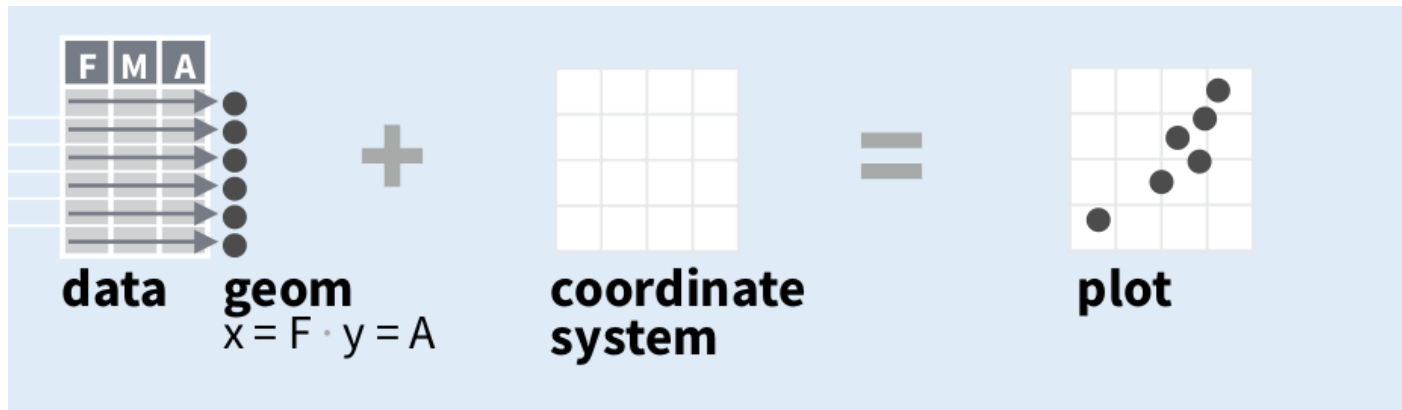
Data analysis - Descriptive statistics in R



- Descriptive statistics offers:
 - ✓ Graphics
 - Qualitative var → bar chart / pie chart
 - Quantitative var → bar chart / histogram
 - ✓ Indicators
 - Mean
 - Median
 - Quantiles
 - Standard deviation & variance

R code: `summary(result)`
`sd(result$var)`
`var(result$var)`

Data analysis - Visualization in R



R code: `ggplot(mpg, aes(hwy, cty)) +
geom_point(aes(color=cyl)) +
geom_smooth(method = "lm") + coord_cartesian() +
scale_color_gradient() + theme_bw()`

source: <https://ggplot2.tidyverse.org/>

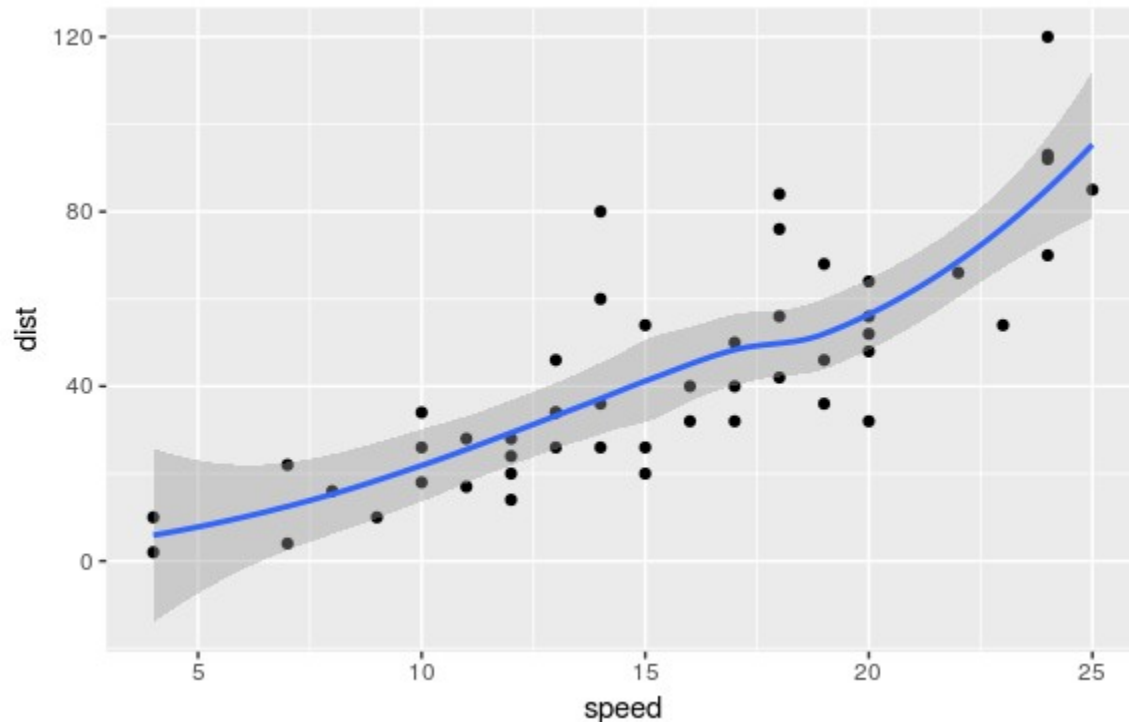
Data analysis - Regression models in R



$$y = b0 + b1*x + e$$

- $b0$ -> **intercept** of the regression line
- $b1$ -> **slope** of the regression line
- e -> residuals errors

Data analysis - Regression models in R



$\text{dist} = b_0 + b_1 * \text{speed}$

```
model <- lm(dist ~ speed, data = cars)
```

Call:

```
lm(formula = dist ~ speed, data = cars)
```

Coefficients:

(Intercept)	speed
-17.579	3.932

Interpretation

* $\text{dist} = -17.579 + 3.932 * \text{speed}$

* for a speed equal to 20 mph, we can expect an increase of 78.64 ft.

* That is, $\text{dist} = -17.579 + 3.932 * 20 = 61.061$ ft.

Before using this formula to predict future dist, you should make sure that this model is statistically significant, that is:

- ✓ there is a statistically significant relationship between the predictor and the outcome variables
- ✓ the model that we built fits very well the data in our hand.

Data analysis - Regression models in R



```
summary(model)
```

```
Call:
```

```
lm(formula = dist ~ speed, data = cars)
```

```
Residuals:
```

Min	1Q	Median	3Q	Max
-29.069	-9.525	-2.272	9.215	43.201

```
Coefficients:
```

	Estimate	Std. Error	t value	Pr(> t)	
(Intercept)	-17.5791	6.7584	-2.601	0.0123	*
speed	3.9324	0.4155	9.464	1.49e-12	***

```
---
```

```
Signif. codes:
```

```
0 *** 0.001 ** 0.01 * 0.05 . 0.1 1
```

```
Residual standard error: 15.38 on 48 degrees of freedom
```

```
Multiple R-squared: 0.6511, Adjusted R-squared: 0.6438
```

```
F-statistic: 89.57 on 1 and 48 DF, p-value: 1.49e-12
```

Interpretation

A statistically significant coefficient indicates that there is an association between the predictor (x) and the outcome (y) variable. In our example, both the p-values for the intercept and the predictor variable are significant, so we can reject the null hypothesis and accept the alternative hypothesis, which means that there is a significant association between the predictor and the outcome variables.

Data analysis - Regression models in R



Call:

```
lm(formula = dist ~ speed, data = cars)
```

Residuals:

Min	1Q	Median	3Q	Max
-29.069	-9.525	-2.272	9.215	43.201

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)	
(Intercept)	-17.5791	6.7584	-2.601	0.0123	*
speed	3.9324	0.4155	9.464	1.49e-12	***

Signif. codes:

0 *** 0.001 ** 0.01 * 0.05 . 0.1 1

Residual standard error: 15.38 on 48 degrees of freedom

Multiple R-squared: 0.6511, Adjusted R-squared: 0.6438

F-statistic: 89.57 on 1 and 48 DF, p-value: 1.49e-12

Interpretation

- In our example, **RSE = 15.38**, meaning that the observed dist values deviate from the true regression line by approximately 15.38 units in average.
- we can calculate the percentage error = 35.78312%
- The R2 measures, how well the model fits the data. For a simple linear regression, R2 is the square of the Pearson correlation coefficient.



EXERCICES

statistical_analysis.ipynb