ALS User Meeting 2022

This notebook describes methods to extract key information from microCT image stacks

- Create image montage
- Access slices evenly spaced
- Play slices like a video

Created by Dani Ushizima, CAMERA, LBNL - Aug 1st 2022

```
%matplotlib inline
```

```
import numpy as np
import matplotlib.pyplot as plt
from skimage import filters, morphology, io
from glob import glob
```

1. Read a microct image

- from url
- from NERSC
- from Google drive

Read from NERSC

- discard this portion if running in Colab

```
datapath = "/global/cfs/cdirs/als/users/yourname/yourdata/" #update these values
!ls -lt "$datapath"
```

```
image = io.imread(datapath+'bead_pack.tif')
```

Read from Google drive

- discard this portion if running at NERSC

```
from google.colab import drive
drive.mount('/content/drive')
```

```
Mounted at /content/drive
```

```
datapath = "/content/drive/My Drive/Colab Notebooks/ALS User Meeting 2022 colab/data/'
!ls -lt "$datapath"
```

```
total 7842
-rw------- 1 root root 8025493 Aug 11 16:46 bead_pack.tif
drwx------ 2 root root    4096 Aug 11 16:35 concrete
```

Read a multi-tiff file

```
def loadFileNames(path,extension):
  ''' Return filename after using colab files.upload - work for 1 file'''
  fnames = glob(path+extension)
  fnames.sort()
  print(path);
  print(f"Number of files: {len(fnames)}")
  return fnames
```

```
extension = '*tif'
fnames = loadFileNames(datapath+'concrete/',extension)
```

```
/content/drive/My Drive/Colab Notebooks/ALS User Meeting 2022 colab/data/concret
Number of files: 20
```

2. How to create an image montage?

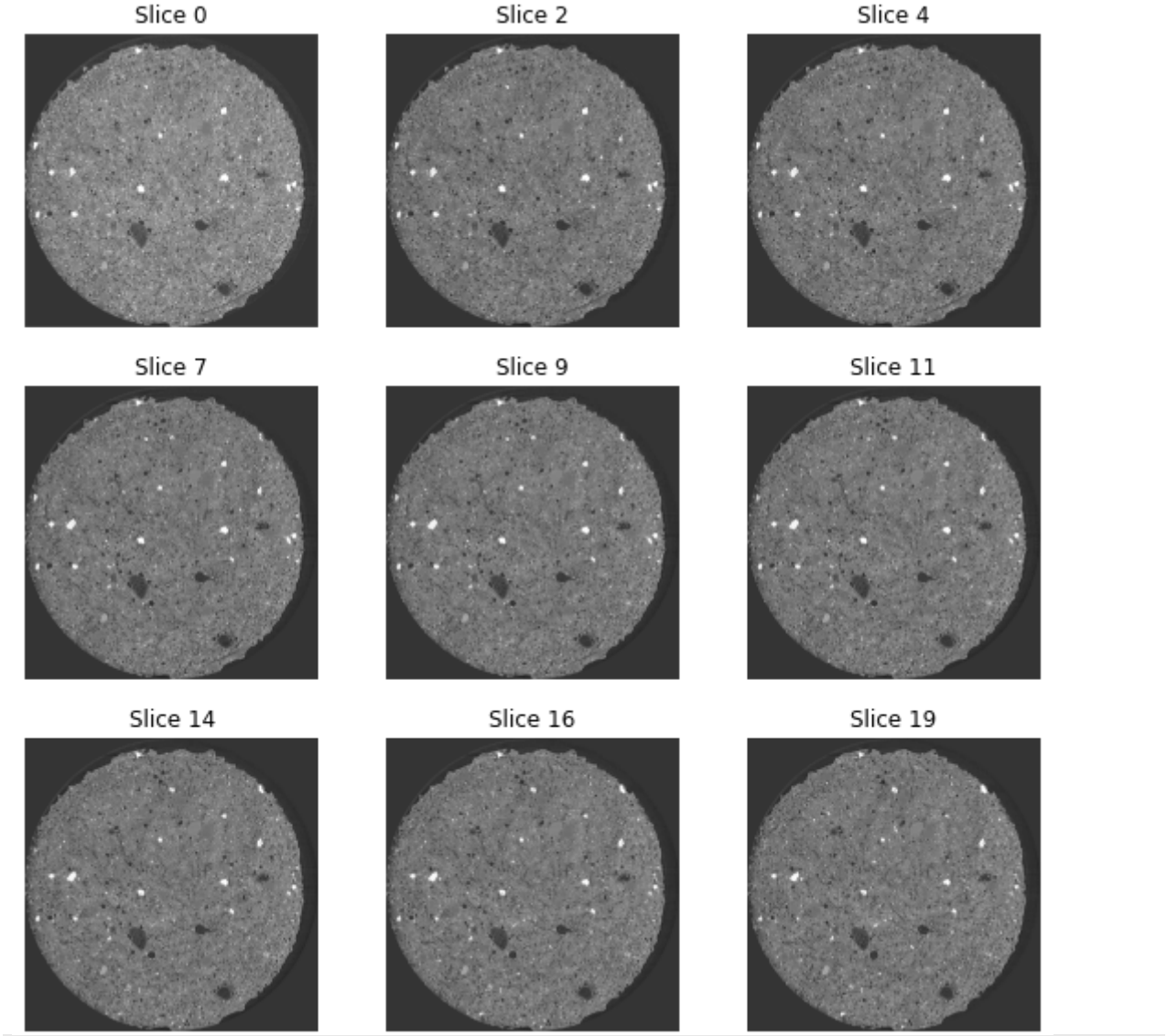
- 3x3
- with all your slices

```
def montage(path,extension,save=False):
        '''Create montage 3x3 '''
        files = loadFileNames(path, extension)
        nfiles = len(files)
        fig, axes = plt.subplots(3, 3, figsize=(10, 10))
        #Plotting subset of slices evenly spaced
        islices = np.linspace(0,nfiles-1,9,dtype=int)
        islices = islices.astype(int)
        for i, n in enumerate(islices):
            img = io.imread(files[n])
            xy = np.unravel_index(i, (3,3))
            axes[xy].imshow(img,cmap='gray')
            axes[xy].set_title('Slice %i'% n)
```

```
            axes[xy].set_axis_off()
            #fig.subplots_adjust( wspace = -0.1,  hspace = 0.1 )
        if(save):
            fig.savefig(path+'MyMontage.png')
            print('Saved@ '+path+'MyMontage.png')


montage(datapath+'concrete/',extension)
```

```
/content/drive/My Drive/Colab Notebooks/ALS User Meeting 2022 colab/data/co
Number of files: 20
```



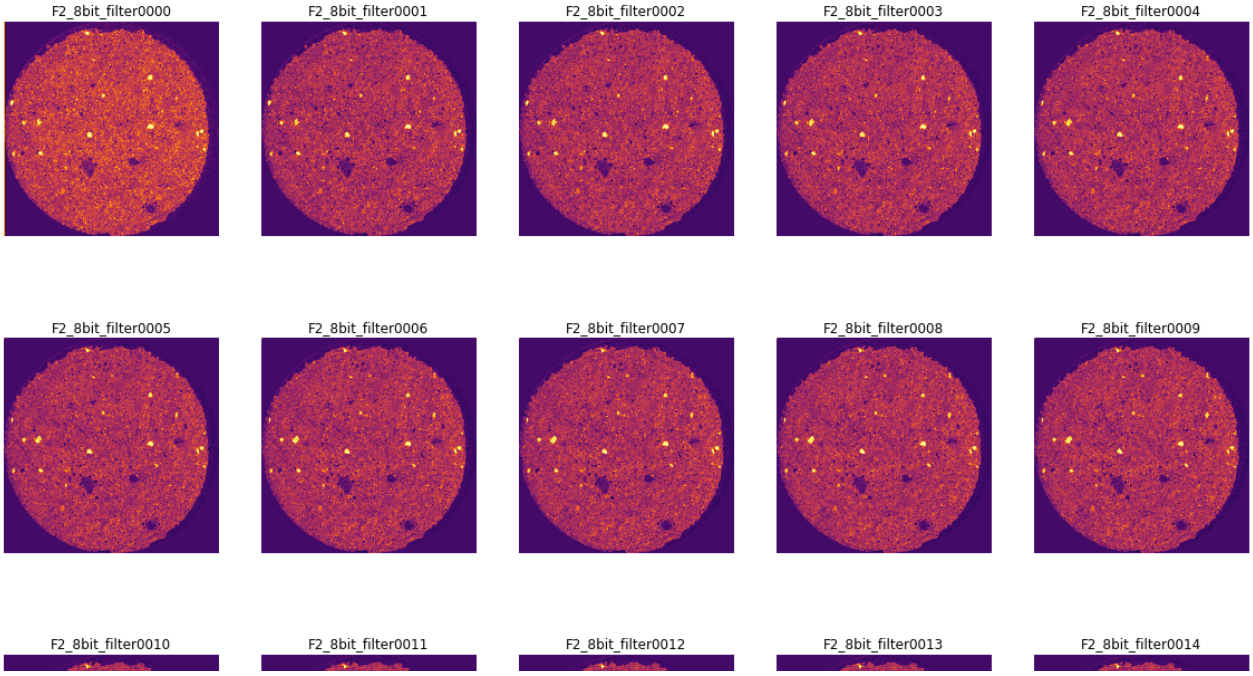

```
def readAndShowMosaic(path,extension):
  '''Read and show ALL images from a folder
     Warning: if nfiles too large, select subset of files
  '''
  files = loadFileNames(path, extension)
  nfiles = len(files)
  y = np.ceil(np.sqrt(nfiles)).astype('int')
  x = np.ceil(nfiles/y).astype('int')
```

```
    f, ax = plt.subplots(x, y, figsize=(20, 20))

    for i,axi in enumerate(ax.flatten()[0:nfiles]):
      img = io.imread(files[i])
      fileCore = str.split(files[i],'/')[-1]
      fileCore = str.split(fileCore,'.')[0]
      axi.imshow(img[::10,::10],cmap='inferno')
      axi.set_axis_off()
      axi.set_title(fileCore)

readAndShowMosaic(datapath+'concrete/',extension)
```

```
/content/drive/My Drive/Colab Notebooks/ALS User Meeting 2022 colab/data/co
Number of files: 20
```



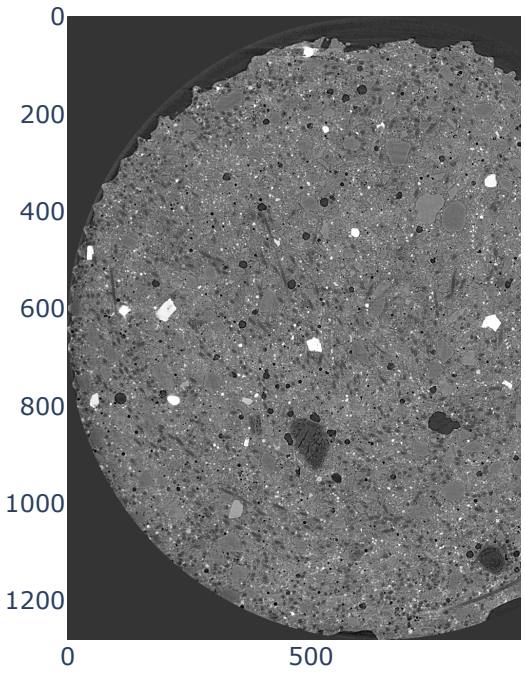
Play your stack

```
!pip install --upgrade plotly
```

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-whe
Requirement already satisfied: plotly in /usr/local/lib/python3.7/dist-packages
Collecting plotly
  Downloading plotly-5.10.0-py2.py3-none-any.whl (15.2 MB)
     |████████████████████████████████| 15.2 MB 3.2 MB/s
Requirement already satisfied: tenacity>=6.2.0 in /usr/local/lib/python3.7/dist-
Installing collected packages: plotly
  Attempting uninstall: plotly
    Found existing installation: plotly 5.5.0
    Uninstalling plotly-5.5.0:
      Successfully uninstalled plotly-5.5.0
Successfully installed plotly-5.10.0
```

```
ic = io.ImageCollection(fnames,conserve_memory=True)
fullstack = ic[0:10].concatenate() #warning: this step might require a lot of mem with
```

```
import plotly.express as px
downsample = 2
fig = px.imshow( fullstack[:,::downsample,::downsample] , animation_frame=0, binary_st
fig.show()
```



microct slice=5

16s completed at 2:35 PM