ALS User Meeting 2022

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

- Read images from different sources
- Show slices of a ndarray
- Plot a slice at random
- Determine stack size

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

```
%matplotlib inline
```

```
import numpy as np
import matplotlib.pyplot as plt
from skimage import img_as_ubyte, io
```

1. Read a microct image

- from url
- from NERSC
- from Google drive

Read from the web

```
#Get an image stack given a url
!wget https://github.com/dani-lbnl/imagexd19/blob/master/dip/data/bead_pack.tif?raw=tr
!mv bead_pack.tif?raw=true bead_pack.tif
```

```
--2022-08-11 18:13:28--  https://github.com/dani-lbnl/imagexd19/blob/master/dip/d
Resolving github.com (github.com)... 13.114.40.48
Connecting to github.com (github.com)|13.114.40.48|:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://github.com/dani-lbnl/imagexd19/raw/master/dip/data/bead_pack.t
--2022-08-11 18:13:29--  https://github.com/dani-lbnl/imagexd19/raw/master/dip/da
Reusing existing connection to github.com:443.
HTTP request sent, awaiting response... 302 Found
Location: https://raw.githubusercontent.com/dani-lbnl/imagexd19/master/dip/data/b
--2022-08-11 18:13:29--  https://raw.githubusercontent.com/dani-lbnl/imagexd19/ma
Resolving raw.githubusercontent.com (raw.githubusercontent.com)... 185.199.111.13
Connecting to raw.githubusercontent.com (raw.githubusercontent.com)|185.199.111.1
```

```
HTTP request sent, awaiting response... 200 OK
Length: 8025493 (7.7M) [application/octet-stream]
Saving to: ‘bead_pack.tif?raw=true’

bead_pack.tif?raw=t 100%[===================>]   7.65M  --.-KB/s    in 0.1s

2022-08-11 18:13:30 (66.9 MB/s) - ‘bead_pack.tif?raw=true’ saved [8025493/802549
```

```
#Double-check you got the image
!ls *.tif
```

```
bead_pack.tif
```

```
#Load the image
image = io.imread('bead_pack.tif')
```

Read from NERSC

- discard this portion if running in Colab

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

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
```

Common steps for NERSC and Colab

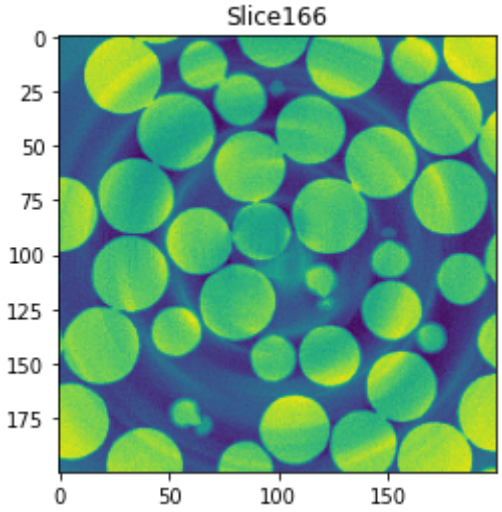
```
image = io.imread(datapath+'bead_pack.tif')

#Check the size before plotting it
print('-----------------------------------------------------------------------------')
print('Image shape is ',image.shape)
print('@CenterSlice: min=',image.min(),',mean=',np.around(image.mean(),decimals=2),',r
print('dtype = ',image.dtype)
print('-----------------------------------------------------------------------------')
```

```
-----------------------------------------------------------------------------
Image shape is  (200, 200, 200)
@CenterSlice: min= 29 ,mean= 142.51 ,max= 239
dtype =  uint8
-----------------------------------------------------------------------------
```

```
#Visualize one slice chose at random
import random
nmax = image.shape[0]
n = random.randint(0,nmax) # want a specific slice? change n
plt.imshow(image[n,:,:])
plt.title('Slice'+str(n))
```

```
Text(0.5, 1.0, 'Slice166')
```



2. How to read several tif files?

- assuming you have a folder with several tif files
- extension is tif and not tiff in this example

```
#Get the list of all files composing your stack
from glob import glob
extension = '*.tif'
slices = glob(datapath+'concrete/'+extension)
slices.sort()
```

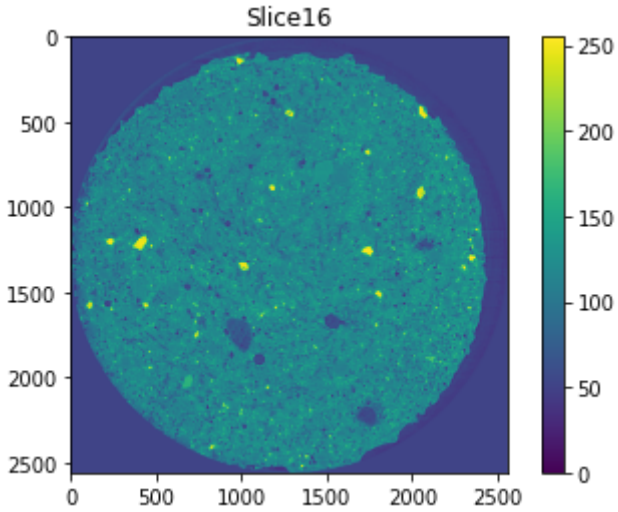
```
print('Folder contains {} files'.format(len(slices)))
```

```
Folder contains 20 files
```

```
ic = io.ImageCollection(slices,conserve_memory=True) #well-suited for large collectior
```

```
nmax = len(slices)
n = random.randint(0,nmax) # want a specific slice? change n
plt.imshow(ic[n])
plt.title('Slice'+str(n))
plt.colorbar()
ic[n].shape
```

```
(2560, 2560)
```




```
image2 = ic[0:10].concatenate()
```

+ Code + Text

```
image2.shape
```

```
(10, 2560, 2560)
```

```
from sys import getsizeof
print("Image size in mem: {}MB".format(round(getsizeof(image2) / 1024 / 1024,2)))
```

```
Image size in mem: 62.5MB
```

✓ 0s completed at 2:26 PM