-
-
-

-
-
-

Installing and using Nibabel Python Library

```
1  from numpy import concatenate, zeros
2
3  from matplotlib.pyplot import subplots, tight_layout, show
4
5  import nibabel as nib
```

Load images and get data

```
1  img_nibabel = nib.load("data/T1_mask.nii")
2
3  type(img_nibabel)
```

```
1  <class 'nibabel.nifti1.Nifti1Image'>
```

```
1  meta_info = img_nibabel.header
2
3  print(meta_info)
```

```
 1  <class 'nibabel.nifti1.Nifti1Header'> object, endian='<'
 2  sizeof_hdr      : 348
 3  data_type       : b''
 4  db_name         : b''
 5  extents         : 0
 6  session_error   : 0
 7  regular         : b'r'
 8  dim_info        : 0
 9  dim             : [  3 128 128  70   1   1   1   1]
10  intent_p1       : 0.0
11  intent_p2       : 0.0
12  intent_p3       : 0.0
13  intent_code     : none
14  datatype        : float32
15  bitpix          : 32
16  slice_start     : 0
17  pixdim          : [-1.   2.   2.   2.2  0.   0.   0.   0. ]
```

```
18  vox_offset       : 0.0
19  scl_slope        : nan
20  scl_inter        : nan
21  slice_end        : 0
22  slice_code       : unknown
23  xyzt_units       : 10
24  cal_max          : 0.0
25  cal_min          : 0.0
26  slice_duration   : 0.0
27  toffset          : 0.0
28  glmax            : 0
29  glmin            : 0
30  descrip          : b'5.0.11'
31  aux_file         : b''
32  qform_code       : scanner
33  sform_code       : scanner
34  quatern_b        : 0.0
35  quatern_c        : 1.0
36  quatern_d        : 0.0
37  qoffset_x        : 125.5061
38  qoffset_y        : -109.38977
39  qoffset_z        : -86.742615
40  srow_x           : [ -2.        0.        0.      125.5061]
41  srow_y           : [   0.        2.        0.     -109.38977]
42  srow_z           : [   0.        0.        2.2     -86.742615]
43  intent_name      : b''
44  magic            : b'n+1'
```

```python
1  print(meta_info.get_xyzt_units())
```

```python
1  ('mm', 'sec')
```

```python
1  img1 = img_nibabel.get_fdata()
2
3  print(type(img1), img1.shape)
```

```python
1  <class 'numpy.memmap'> (128, 128, 70)
```

```python
1  img_nibabel = nib.load("data/b0_mask.nii")
2
3  img2 = img_nibabel.get_fdata()
```

```python
1  img1.shape
```

```python
1  (128, 128, 70)
```

plotting Images

```
1  img_slice = 30
```

```
1  fig, ax = subplots(ncols=2, figsize=(15, 5))
2
3  f1 = ax[0].imshow(img1[:, :, img_slice], cmap="gray")
4  f2 = ax[1].imshow(img2[:, :, img_slice], cmap="gray")
5
6  fig.colorbar(f1, ax=ax[0])
```

```
1  <matplotlib.colorbar.Colorbar object at 0x7f5e28ecb670>
```

```
1  fig.colorbar(f2, ax=ax[1]);
2
3  ax[0].set_xlabel('T1 Image', fontsize=16);
4  ax[1].set_xlabel('B0 Image', fontsize=16);
5
6  show()
```

Data pre-processing

```
1  img1_slice = img1[:, :, img_slice]
2  img2_slice = img2[:, :, img_slice]
```

Remove zeros

```
1  fig, ax = subplots(ncols=2, figsize=(20, 5))
2
3  ax[0].hist(img1_slice.flatten(), bins=50)
```

```
1  (array([1.2374e+04, 3.0000e+00, 6.0000e+00, 1.0000e+01, 1.1000e+01,
2         2.4000e+01, 3.2000e+01, 2.5000e+01, 4.3000e+01, 3.5000e+01,
3         3.8000e+01, 3.3000e+01, 2.9000e+01, 3.0000e+01, 2.6000e+01,
4         4.7000e+01, 4.6000e+01, 6.6000e+01, 5.3000e+01, 6.6000e+01,
5         6.8000e+01, 6.4000e+01, 9.7000e+01, 9.5000e+01, 1.2100e+02,
6         1.2700e+02, 1.3100e+02, 1.2600e+02, 1.5600e+02, 1.3700e+02,
7         1.4000e+02, 1.2900e+02, 1.1900e+02, 1.1900e+02, 1.3100e+02,
8         1.4600e+02, 1.5400e+02, 1.4300e+02, 1.7000e+02, 1.2800e+02,
9         1.1100e+02, 1.1100e+02, 1.0900e+02, 1.0500e+02, 1.0500e+02,
10        1.2600e+02, 9.9000e+01, 5.5000e+01, 4.2000e+01, 2.3000e+01]),
          array([  0.       ,   9.6287677 ,  19.2575354 ,  28.8863031
            ,
11         38.5150708 ,  48.1438385 ,  57.7726062 ,  67.4013739 ,
12         77.0301416 ,  86.6589093 ,  96.287677  , 105.9164447 ,
13        115.5452124 , 125.1739801 , 134.8027478 , 144.4315155 ,
14        154.0602832 , 163.6890509 , 173.3178186 , 182.9465863 ,
```

```
15        192.575354  , 202.2041217 , 211.8328894 , 221.4616571 ,
16        231.0904248 , 240.7191925 , 250.34796021, 259.97672791,
17        269.60549561, 279.23426331, 288.86303101, 298.49179871,
18        308.12056641, 317.74933411, 327.37810181, 337.00686951,
19        346.63563721, 356.26440491, 365.89317261, 375.52194031,
20        385.15070801, 394.77947571, 404.40824341, 414.03701111,
21        423.66577881, 433.29454651, 442.92331421, 452.55208191,
22        462.18084961, 471.80961731, 481.43838501]), <BarContainer object
             of 50 artists>)
```

```
1  ax[1].hist(img2_slice.flatten(), bins=50);
2
3  show()
```

```
1  mask = (img1_slice>0) & (img2_slice>0)
2
3  img1_nz = img1_slice[mask]
4  img2_nz = img2_slice[mask]
5
6  fig, ax = subplots(nrows=1, ncols=2, figsize=(20, 5))
7
8  ax[0].hist(img1_nz, bins=50)
```

```
1  (array([  7.,    9.,   11.,   18.,   35.,   19.,   39.,   33.,   32.,   41.,
      29.,
2         25.,   27.,   29.,   52.,   58.,   52.,   48.,   66.,   71.,   57.,
            96.,
3         88.,  119.,  124.,  122.,  125.,  149.,  136.,  121.,  137.,  112.,
           113.,
4        135.,  116.,  151.,  156.,  135.,  159.,  113.,  108.,  100.,  107.,
           104.,
5        109.,  114.,   88.,   53.,   38.,   23.]), array([ 17.50374222,
            26.78243507,  36.06112793,  45.33982079,
6         54.61851364,  63.8972065 ,  73.17589935,  82.45459221,
7         91.73328506, 101.01197792, 110.29067078, 119.56936363,
8        128.84805649, 138.12674934, 147.4054422 , 156.68413506,
9        165.96282791, 175.24152077, 184.52021362, 193.79890648,
10       203.07759933, 212.35629219, 221.63498505, 230.9136779 ,
11       240.19237076, 249.47106361, 258.74975647, 268.02844933,
12       277.30714218, 286.58583504, 295.86452789, 305.14322075,
13       314.4219136 , 323.70060646, 332.97929932, 342.25799217,
14       351.53668503, 360.81537788, 370.09407074, 379.3727636 ,
15       388.65145645, 397.93014931, 407.20884216, 416.48753502,
16       425.76622787, 435.04492073, 444.32361359, 453.60230644,
17       462.8809993 , 472.15969215, 481.43838501]), <BarContainer object
             of 50 artists>)
```

```
1  ax[1].hist(img2_nz, bins=50);
2
3  show()
```

Scaling

Standard Scaler: removing the mean and scaling to unit variance

```
1  from sklearn.preprocessing import StandardScaler
```

```
1  Error: ModuleNotFoundError: No module named 'sklearn'
```

```
1  scaler = StandardScaler()
```

```
1  Error: NameError: name 'StandardScaler' is not defined
```

```
1  img1_scaled = scaler.fit_transform(img1_nz.reshape(-1, 1))
```

```
1  Error: NameError: name 'scaler' is not defined
```

```
1  img2_scaled = scaler.fit_transform(img2_nz.reshape(-1, 1))
```

```
1  Error: NameError: name 'scaler' is not defined
```

```
1  img1_scaled.shape
```

```
1  Error: NameError: name 'img1_scaled' is not defined
```

Visualise and Concatenate

Seaborn: https://seaborn.pydata.org

c.f. pair grid example https://seaborn.pydata.org/examples/pair_grid_with_kde.html

kdeplot documentation https://seaborn.pydata.org/generated/seaborn.kdeplot.html

```
1   fig, ax = subplots(1, 3, figsize=(20, 6))
2
3   # Scatter plot
4   ax[0].scatter(img1_nz, img2_nz)
5
6   # 2D Histogram
7   ax[1].hist2d(img1_nz, img2_nz, bins=50, vmax=10);
8
9   from seaborn import kdeplot
10
11  # Density Plot
```

```
1  Error: ModuleNotFoundError: No module named 'seaborn'
```

```
1  kdeplot(x=img1_nz, y=img2_nz, ax=ax[2]);
```

```
1  Error: NameError: name 'kdeplot' is not defined
```

```
1  show()
```

Scaled Images

```
1  fig, ax = subplots(1, 3, figsize=(20, 6))
2
3  # Scatter plot
4  ax[0].scatter(img1_scaled[:,0], img2_scaled[:,0])
5
6  # 2D Histogram
```

```
1  Error: NameError: name 'img1_scaled' is not defined
```

```
1  ax[1].hist2d(img1_scaled[:,0], img2_scaled[:,0], bins=50, vmax=10);
```

```
1  Error: NameError: name 'img1_scaled' is not defined
```

```
1  from seaborn import kdeplot
2
3  # Density Plot
```

```
1  Error: ModuleNotFoundError: No module named 'seaborn'
```

```
1  kdeplot(x=img1_scaled[:,0], y=img2_scaled[:,0], ax=ax[2]);
```

```
1  Error: NameError: name 'kdeplot' is not defined
```

```
1  show()
```

```
1  all_img_scaled = concatenate([img1_scaled, img2_scaled], axis=1)
```

```
1  Error: NameError: name 'img1_scaled' is not defined
```

```
1  all_img_scaled.shape
```

```
1  Error: NameError: name 'all_img_scaled' is not defined
```

Segmenting images with Gaussian Mixtures

GMM clustering

```
1  from sklearn.mixture import GaussianMixture
```

```
1  Error: ModuleNotFoundError: No module named 'sklearn'
```

```
1  n_components = 3
2
3  RANDOM_STATE = 12345
4
5  gmm = GaussianMixture(n_components=n_components,
6                        random_state=RANDOM_STATE)
```

```
1  Error: NameError: name 'GaussianMixture' is not defined
```

```
1  all_img_labels = gmm.fit_predict(all_img_scaled)
```

```
1  Error: NameError: name 'gmm' is not defined
```

```
1  all_img_labels[0]
```

```
1  Error: NameError: name 'all_img_labels' is not defined
```

```
1  fig, ax = subplots(figsize=(8, 8))
2
3  ax.scatter(img1_nz, img2_nz, c=all_img_labels, s=100)
```

```
1  Error: NameError: name 'all_img_labels' is not defined
```

```
1  ax.set_xlabel('Image 1', fontsize=16)
2  ax.set_ylabel('Image 2', fontsize=16);
3
4  show()
```

```
1  all_img_labels_mapped = zeros(img1_slice.shape)
2
3  all_img_labels_mapped[mask] = all_img_labels
```

```
1  Error: NameError: name 'all_img_labels' is not defined
```

```
1  fig, ax = subplots(figsize=(20, 10))
2
3  ax.imshow(all_img_labels_mapped);
4
5  show()
```

Keypoints

-
-
-