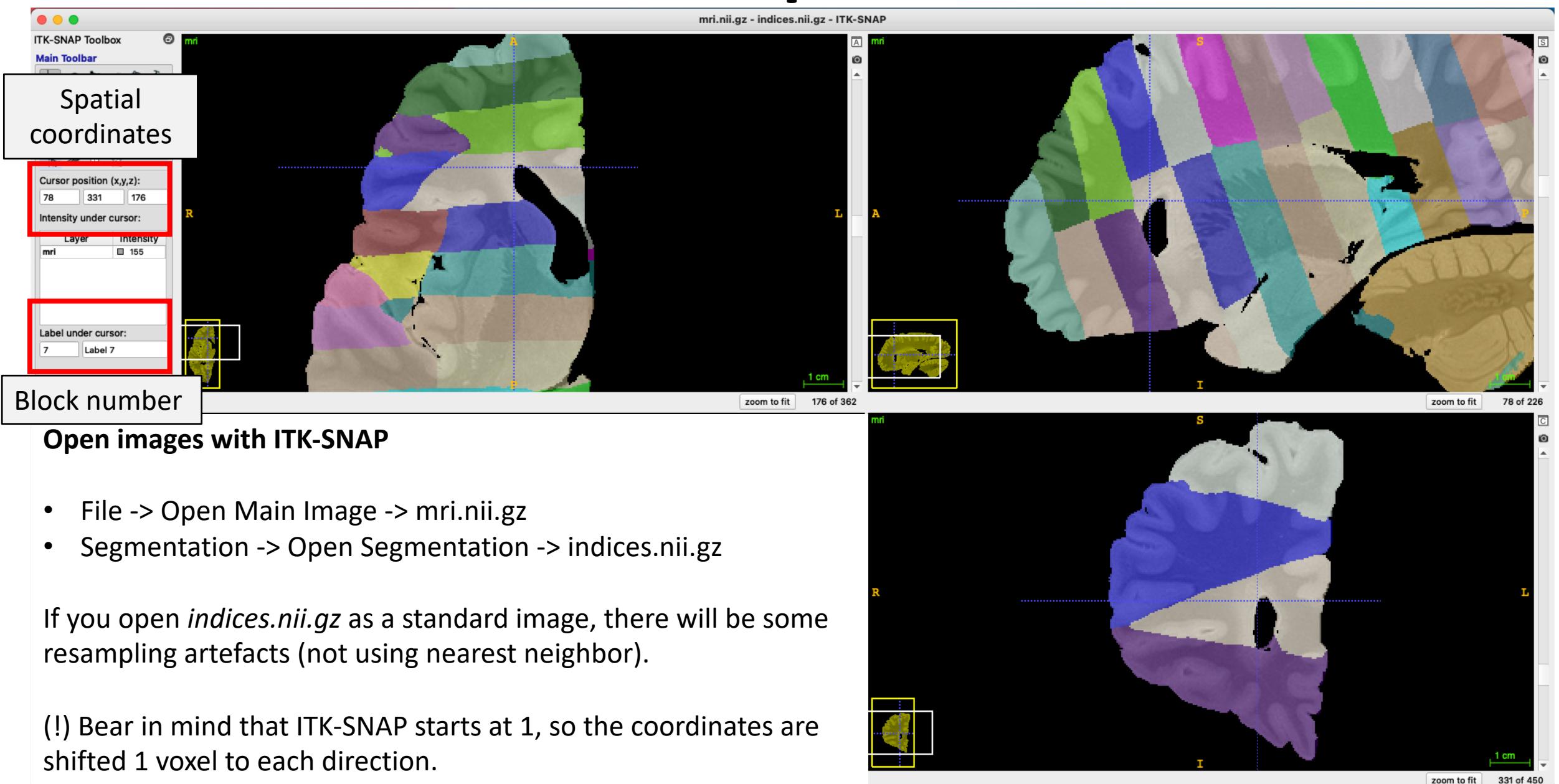


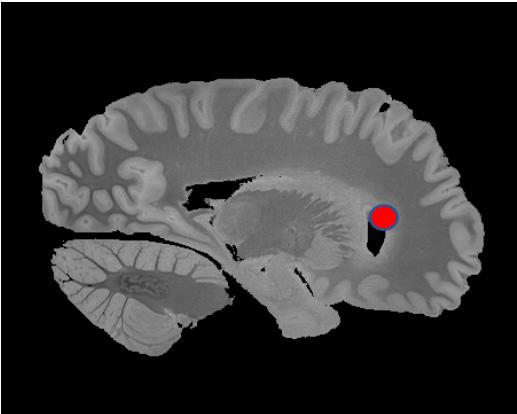
Histology coordinates

Recap: How to compute histology coordinates

1. Get the (i,j,k) from input images.
 - a. You could use the notebook from Eugenio to go from the rotated to the original coordinate system.
 - b. But I believe you use it to display the images, so the (i,j,k) can be directly inferred from the slice names:
 - i=sagittal/slice_i.png
 - j=coronal/slice_j.png
 - k=axial/slice_k.png
2. Multiply by the matrix m :
 - mri_coordinates = np.asarray([i, j, k, 1])
 - m = load_from(matrices/block_index.txt)
 - i',j',k',1 = np.dot(m, mri_coordinates)
3. Get histology coordinates
 - i'=mouseY (rows of the image)
 - j'=mouseX (columns of the image)
 - k'=section number (histology/slices_HE/slice_k'.jpg)

Example #1

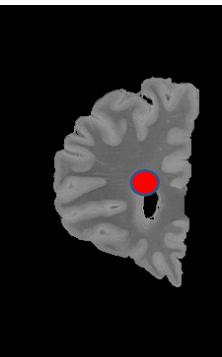




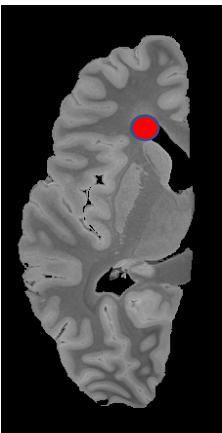
mouseX (columns) = 330
mouseY (rows) = 362 - 175 =
187

MRI Coordinates

- I = 77 (saggital, slice_077), ITK-SNAP: X=78
- J = 330 (coronal, slice_330), ITK-SNAP: Y=331
- K = 175 (axial, slice_175), ITK-SNAP: Z=176



mouseX (columns) = 226 - 77 = 149
mouseY (rows) = 362 - 175 = 187



mouseX (columns) = 226 - 77 = 149
mouseY (rows) = 450 - 330 = 120

BLOCK: 7

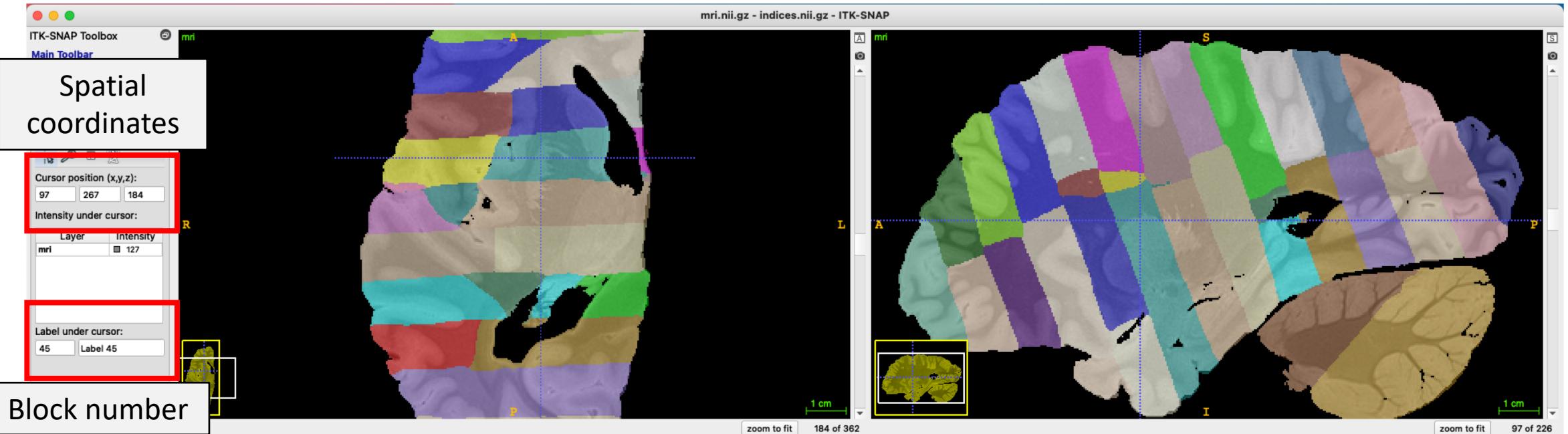
HISTO Coordinates

- I = 322.67 (Y, rows)
- J = 127.99 (X, columns)
- K = 23.839 = 24 (slice number)

How to compute histology coordinates

1. Get the (i,j,k) from input images. You already know these coordinates because you loaded the images and they are slice_XXX.png numbers from saggittal (i), coronal (j), axial (k)
 - i = 77, j= 330, k = 175
2. Mutiply by the matrix (in *matrices/block_07.txt*):
 - mri_coordinates = np.asarray([77, 330, 175, 1])
 - m = np.asarray([[3.92, -0.54, -0.54, 293.53],[-0.66, -1.26, -3.76, 1252.61],[0.097, 1.1, -0.4, -276.63],[0.0, 0.0, 0.0, 1.0]])
3. Get histology coordinates
 - i'=322
 - j'=128
 - k'=24

Example #2

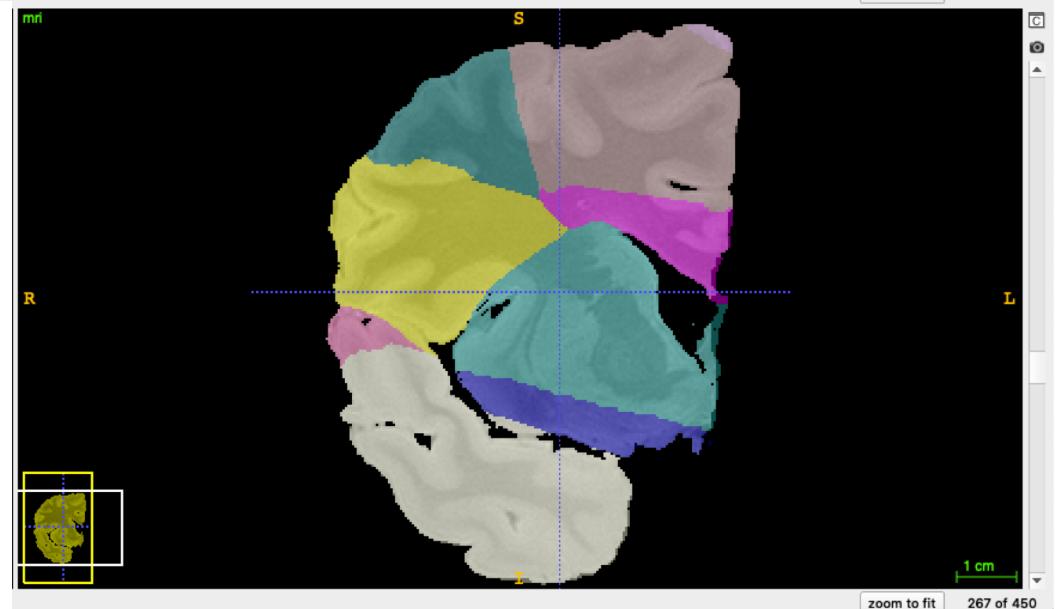


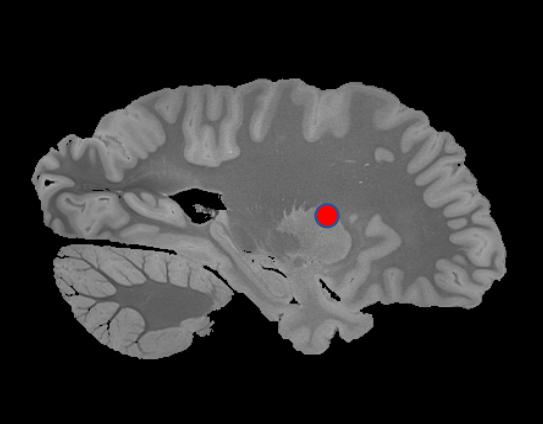
Open images with ITK-SNAP

- File -> Open Main Image -> mri.nii.gz
- Segmentation -> Open Segmentation -> indices.nii.gz

If you open *indices.nii.gz* as a standard image, there will be some resampling artefacts (not using nearest neighbor).

(!) Bear in mind that ITK-SNAP starts at 1, so the coordinates are shifted 1 voxel to each direction.

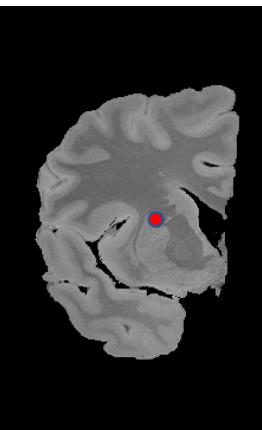




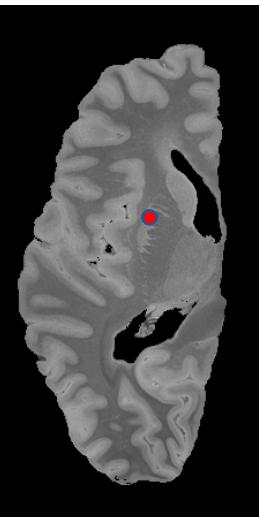
mouseX (columns) = 266
mouseY (rows) = 362 - 183 =
179

MRI Coordinates

- I = 96 (saggital, slice_096), ITK-SNAP: X=97
- J = 266 (coronal, slice_266), ITK-SNAP: Y=267
- K = 183 (axial, slice_183), ITK-SNAP: Z=184



mouseX (columns) = 226 - 96 = 130
mouseY (rows) = 362 - 183 = 179



mouseX (columns) = 226 - 96 = 130
mouseY (rows) = 450 - 266 = 184



BLOCK: 45

HISTO Coordinates

- I = 348.87 (Y, rows)
- J = 132.96 (X, columns)
- K = 16.59 = 17 (slice number)

How to compute histology coordinates

1. Get the (i,j,k) from input images. You already know these coordinates because you loaded the images and they are slice_XXX.png numbers from saggittal (i), coronal (j), axial (k)
 - i = 73, j= 315, k = 189
2. Mutiply by the matrix (in *matrices/block_45.txt*):
 - mri_coordinates = np.asarray([96, 266, 183, 1])
 - m= np.asarray([[3.56, -0.58, -1.74, 479.81],[-1.78, -1.23, -3.38, 1249.56],[0.014, 1.1, -0.41, -202.33],[0.0, 0.0, 0.0, 1.0]])
1. Get histology coordinates
 - i'=349
 - j'=133
 - k'=17

Q: How are *slices* and *indices* computed?

1. import nibabel
import numpy as np
from PIL import Image -> (to save grayscale/RGB images)
2. Use nibabel to load the *mri.nii.gz* and the *indices.nii.gz* volumes
3. For every dimension (saggittal, coronal, axial) do:
 - For all slices in that dimension do:
 - if 'saggittal':
 - a. mri_i = np.rot90(mri_i)
 - b. slice index i = np.rot90(index_i)
 - else:
 - a. mri_i = np.fliplr(np.rot90(mri_i))
 - b. index_i = np.fliplr(np.rot90(index_i))
 - 4. Save slices:
 1. img.save(mri_i, 'slices_X/slice_i.png') -> using Pillow library
 2. np.save(index_i, 'indices_X/slice_i.npy')