DIRECTORY TREE:

Root directory: ERC-project-webviz/
Subject directories: right now we only have one subject, but expect 5 in total.

- Data subdirectories: mri, histology, histology_hr

Generic files:

- readme_ERC_website.pdf/docx: the readme file of the data directory

- *FS_label_mapping.csv:* ontology grouping labels in a hierarchical way

- WholeHemisphereFS.txt: color-code to represent the different structure labels

- image_ontology_table.csv: table that relates each label with a representative block and section from the dataset. The (x,y,z) coordinates indicate.
  - x: rows of the label block
  - y: columns of the label block.
  - z: slice number within the block

  If a label is missing in a brain, the table is filled with -1 at each column.

- image_ontology_table_FS.csv: similar to the previous one with a different structure hierarchy.

Directory overview:

```
ERC-project-webviz/
    P57-16/
        mri/
        histology/
            01/
            02/
            …
        histology_hr/
            01/
            02/
            …
    … (x4)
```

Within each subject, you'll find **mri, histology, histology_hr** directories explained below:

****
MRI
****

mri.nii.gz: Nifti file with 3D volume.

slices_sagittal: subdirectory with sagittal slices in JPG format. These were extracted from the 3D MRI volume as: slice_i = np.squeeze(VOLUME[i,:,:])

slices_coronal: subdirectory with sagittal slices in JPG format. These were extracted from the 3D MRI volume as: slice_i = np.squeeze(VOLUME[:,i,:])

slices_axial: subdirectory with axial slices in JPG format. These were extracted from the 3D MRI volume as: slice_i = VOLUME[:,:,i]

indices.nii.gz: Nifti file with index volume (it tells you, for every voxel, which block you are mapping to, i.e., what subdirectory under "HISTOLOGY" to look into, and which matrix under "matrices" to use). Zero means background, i.e., no block available.

indices_sagittal: subdirectory with sagittal indices in JPG format. These were extracted from the 3D INDEX index volume as: slice_i = np.squeeze(VOLUME[i,:,:])

indices_coronal: subdirectory with sagittal indices in JPG format. These were extracted from the 3D INDEX volume as: slice_i = np.squeeze(VOLUME[:,i,:])

indices_axial: subdirectory with axial indices in JPG format. These were extracted from the 3D INDEX volume as: slice_i = VOLUME[:,:,i]

matrices: subdirectory with a bunch of 4x4 matrices (one per block) that enable you to go from MRI voxel coordinates (i_m, j_m, k_m) to histology coordinates (i_h, j_h, k_h). It's based on homogeneous coordinates, i.e., you need to add a 4th element to the coordinate vectors that is always a 1. The relationship is thus: [i_h; j_h; k_h; 1] = MATRIX * [i_m; j_m; k_m; 1].

matrices_hr: subdirectory with a bunch of 4x4 matrices (one per block) that enable you to go from MRI voxel coordinates (i_m, j_m, k_m) to high-resolution histology coordinates (i_h, j_h, k_h). It's based on homogeneous coordinates, i.e., you need to add a 4th element to the coordinate vectors that is always a 1. The relationship is thus: [i_h; j_h; k_h; 1] = MATRIX * [i_m; j_m; k_m; 1].

****
HISTOLOGY
****

Here you will find one file with the lookup table and a bunch of subdirectories corresponding to the different blocks;

lookup_table.txt: the lookup table. Please note that label 0 (background) should always be fully transparent. Please note that every label has four associated values: red, green, blue, and alpha. You can ignore the alpha, which is always zero anyway.

Subdirectories XX/: the nomenclature is just XX, a double digit corresponding to the block number.

Within each subdirectory, you will find:

LFB.nii.gz: Nifti file with 3D volume of LFB stain. Please note that, as opposed to the MRI, this is actually a 4D matrix, where the 4th dimension has three channels for red, green and blue (this is because the MRI is grayscale, but this one is in color).

HE.nii.gz: same as LFB.nii.gz but with HE stain.

MRI.nii.gz: the same as the previous two, but with the resampled MRI. This one is in grayscale so it has three rather than four dimensions.

labels.nii.gz: Nifti file with 3D volume of labels. In order to go from labels to colors, you need the lookup table provided in the parent directory. Also, note that you can use Freeview (the viewer bundled with FreeSurfer) to open these labels on top of the other 3 volumes, using the provided lookup table (FreeSurfer's default table won't work).

matrix.txt: this the matrix that goes from the voxel coordinates of this block (i_b, j_b, k_b) to the voxel coordinates of the MR (i_m, j_m, k_m). Specifically: [i_m; j_m; k_m; 1] = MATRIX * [i_b; j_b; k_b; 1]. As you would expect, this matrix is simply the inverse of the matrix corresponding to this block found in MRI/matrices.

And also four sub-subdirectories

slices_LFB: contains the LFB slices in JPG format. These have been obtained with slice_i = np.squeeze(VOLUME[:,:,i,:])

slices_HE: same thing for H&E.

slices_MRI: same thing for MRI, where slice_i = VOLUME[:,:,i]

slices_labels: these are the individual slices for the labels. Since they have values over 255 and since you cannot afford compression loss here, we use numpy arrays for them, instead of JPGs.

****

HISTOLOGY_HR

****

This directory is equivalent to the HISTOLOGY directory but contains high resolution images. You'll find a bunch of different subdirectories corresponding to different blocks and data for LFB, HE and labels. MRI images are NOT saved (can be upsampled from the low resolution). *Currently, there is only one block (A1.3)*

Subdirectories XX/: the nomenclature is just XX, a double digit corresponding to the block number.

Within each subdirectory, you will find:

matrix.txt: this is the matrix that goes from the voxel coordinates of this block (i_b, j_b, k_b) to the voxel coordinates of the MR (i_m, j_m, k_m). Specifically: [i_m; j_m; k_m; 1] = MATRIX * [i_b; j_b; k_b; 1]. As you would expect, this matrix is simply the inverse of the matrix corresponding to this block found in MRI/matrices.

And also three sub-subdirectories

slices_LFB: contains the LFB slices in JPG format. These have been obtained with slice_i = np.squeeze(VOLUME[:,:,i,:])

slices_HE: same thing for H&E.

slices_labels: these are the individual slices for the labels. Since they have values over 255 and since you cannot afford compression loss here, we use numpy arrays for them, instead of JPGs.