nifti_io

January 10, 2022

I use the nibabel package to handle my nifti loading and saving. Here's the link to their site. You can install it using `pip` or `conda`

https://nipy.org/nibabel/

[39]:
```
import nibabel as nib
import numpy as np
```

[28]:
```
# First load the nifti image. The argument should be a filepath to a .nii or .
 ↪nii.gz image.
# I'm using an image on my desktop for this example.

nii = nib.load('/Users/ryanellis/Desktop/GOALS9014-2_masked.nii.gz')
```

[29]:
```
# Reorient image to RAS+ (first axis left-right, second axis␣
 ↪posterior-anterior, third axis inferior-superior)
nii = nib.as_closest_canonical(nii)

# This can be helpful to ensure consitent image indexing.
# For example: img[:, :, 0:20] would always index the first 20 axial slices
```

[36]:
```
# Convert to a numpy array
img = nii.get_fdata()

# Get affine matrix
affine = nii.affine

print("Image shape:", img.shape)
print("Image type:", type(img), img.dtype)
print("Affine Matrix\n", affine)
```

```
Image shape: (208, 320, 320)
Image type: <class 'numpy.ndarray'> float64
Affine Matrix
 [[   0.70000005    0.            0.          -64.35308146]
 [   0.            0.69999999    0.          -76.8066864 ]
 [   0.            0.            0.69999999 -140.09138489]
 [   0.            0.            0.            1.        ]]
```

[38]:
```
# Saving a nifti

# create nifti image object
nii_out = nib.Nifti1Image(img, affine=affine)

# Save the nifti to a specified filepath
nii_out.to_filename("output_file.nii.gz")

# An image and affine matrix are required to create an nifti image object.
# You can also pass the entire header to the output image.
nii_out = nib.Nifti1Image(img, affine=nii.affine, header=nii.header)

# If you're creating an image and don't have an affine matrix, you could just␣
 ↪use:
affine = np.eye(4)
print("Identity Matrix:\n", affine)
```

```
Identity Matrix:
 [[1. 0. 0. 0.]
 [0. 1. 0. 0.]
 [0. 0. 1. 0.]
 [0. 0. 0. 1.]]
```

[ ]: