

Cuboid

Motivation

Developping code in neuro-imaging implies handling n-dimensionnal data (typically 3D images, with sagittal, coronal, axial axes) which also is associated with meta data. Usually, the data representation consists of two detached objects:

- an array containing the actual measurements
- a "header" gathering all meta data.

This twofold form is used in most common file format but also for object design. In Python, measurements are typically stored in a `numpy.ndarray` and meta data in a dictionary-like structure.

In terms of design, it seems that some information available in meta-data are more closely related to measurements than others, namely: axes information which consists of their names/labels (eg sagittal) and the mapping between slice indexes in the data array and a "real world" value (eg distance for spatial axes, duration for temporal axes).

Features

- gathers extra info for ndarray: axes labels, axes domains, value label, meta_data (eg Nifti header, affine)
- array orientation from axes labels
- array expansion based on axes labels
- array slicing based on axes labels and domain values
- stacking
- nifti I/O (TODO: gifti)

Example script

TODO: split into features ...

[source:pyhrf-free/trunk/script/example/cuboid.py](https://source.pyhrf-free/trunk/script/example/cuboid.py)

```
import pyhrf
from pyhrf.ndarray import Cuboid, MRI3Daxes, MRI4Daxes, stack_cuboids

mask_fn = pyhrf.get_data_file_name('audio-video_nalocalizer__roimask.nii.gz')
bold_fn = pyhrf.get_data_file_name('audio-video_nalocalizer__session0.nii.gz')

print 'Nibabel convention for axes orientation (nifti format):'
print MRI4Daxes

## Data load examples:
cmask = Cuboid.load(mask_fn)
print 'mask loaded as Cuboid:'
print cmask.descrip()

affine, nifti_header = cmask.meta_data
print 'affine:'
print affine

cbold = Cuboid.load(bold_fn)
print 'BOLD loaded as Cuboid:'
print cbold.descrip()

cbold_flat = cbold.cflatten(cmask, 'voxel')
cbold_flat.set_orientation(['time', 'voxel'])
print 'Flatten BOLD (reoriented):'
print cbold_flat.descrip()

cbold2 = cbold_flat.cexpand(cmask, 'voxel')
cbold2.set_orientation(MRI4Daxes)
print 'Re-expanded BOLD :'
```

```

print cbold2.descrip()

time_axis_id = cbold.get_axis_id('time')
bold_mean = cbold.data.mean(time_axis_id)
bold_var = cbold.data.var(time_axis_id)

# Cuboid init synopsis:
# c = Cuboid(self, narray, axes_names, axes_domains, value_label, meta_data)

cbold_mean = Cuboid(bold_mean, MRI3Daxes, value_label='intensity',
                    meta_data=cbold.meta_data)
print 'Mean of BOLD:'
print cbold_mean.descrip()

cbold_var = Cuboid(bold_var, MRI3Daxes, value_label='intensity',
                  meta_data=cbold.meta_data)
print 'Var of BOLD'
print cbold_var.descrip()

cbold_vm = stack_cuboids([cbold_mean, cbold_var], 'mv_axis', ['mean','var'])
cbold_vm.set_orientation(MRI3Daxes + ['mv_axis'])
print 'Stack of BOLD mean and var in a single Cuboid:'
print cbold_vm.descrip()

```

Output of this script:

Nibabel convention for axes orientation (nifti format):

```
['sagittal', 'coronal', 'axial', 'time']
```

mask loaded as Cuboid:

```
* shape : (53, 63, 46)
* orientation: ['sagittal', 'coronal', 'axial']
* value label: intensity
* axes domains:
  'sagittal': arange(0,52,1)
  'coronal': arange(0,62,1)
  'axial': arange(0,45,1)

```

affine:

```
[[ -3.   0.   0.  78. ]
 [  0.   3.   0. -93. ]
 [  0.   0.   3. -67.5]
 [  0.   0.   0.   1. ]]
```

BOLD loaded as Cuboid:

```
* shape : (53, 63, 46, 125)
* orientation: ['sagittal', 'coronal', 'axial', 'time']
* value label: intensity
* axes domains:
  'sagittal': arange(0,52,1)
  'coronal': arange(0,62,1)
  'axial': arange(0,45,1)
  'time': arange(0,124,1)

```

Flatten BOLD (reoriented):

```
* shape : (125, 1272)
* orientation: ['time', 'voxel']
* value label: intensity
* axes domains:
  'time': arange(0,124,1)
  'voxel': arange(0,1271,1)

```

Re-expanded BOLD :

```
* shape : (53, 63, 46, 125)
* orientation: ['sagittal', 'coronal', 'axial', 'time']
* value label: intensity
* axes domains:
  'sagittal': arange(0,52,1)

```

```
'coronal': arange(0,62,1)
'axial': arange(0,45,1)
'time': arange(0,124,1)
```

Mean of BOLD:

```
* shape : (53, 63, 46)
* orientation: ['sagittal', 'coronal', 'axial']
* value label: intensity
* axes domains:
  'sagittal': arange(0,52,1)
  'coronal': arange(0,62,1)
  'axial': arange(0,45,1)
```

Var of BOLD

```
* shape : (53, 63, 46)
* orientation: ['sagittal', 'coronal', 'axial']
* value label: intensity
* axes domains:
  'sagittal': arange(0,52,1)
  'coronal': arange(0,62,1)
  'axial': arange(0,45,1)
```

Stack of BOLD mean and var in a single Cuboid:

```
* shape : (53, 63, 46, 2)
* orientation: ['sagittal', 'coronal', 'axial', 'mv_axis']
* value label: intensity
* axes domains:
  'sagittal': arange(0,52,1)
  'coronal': arange(0,62,1)
  'axial': arange(0,45,1)
  'mv_axis': array(['mean', 'var'],
    dtype='|S4')
```