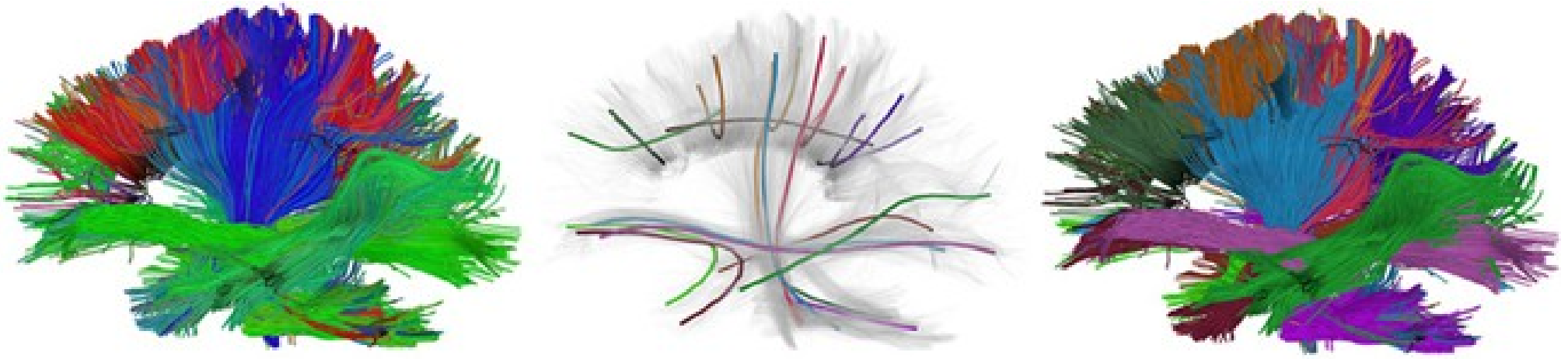
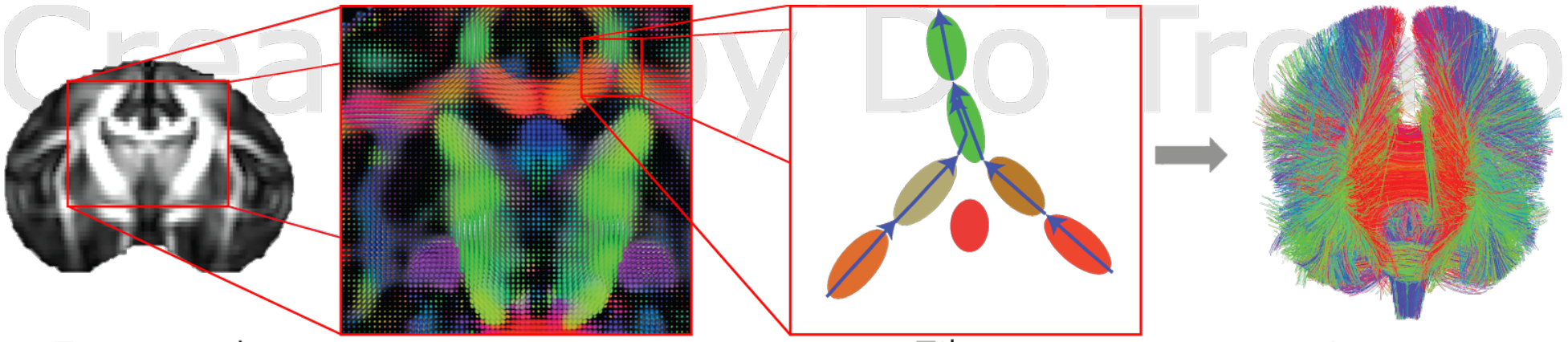
Tractography tutorial



Dr. Alessandro Crimi

Overview



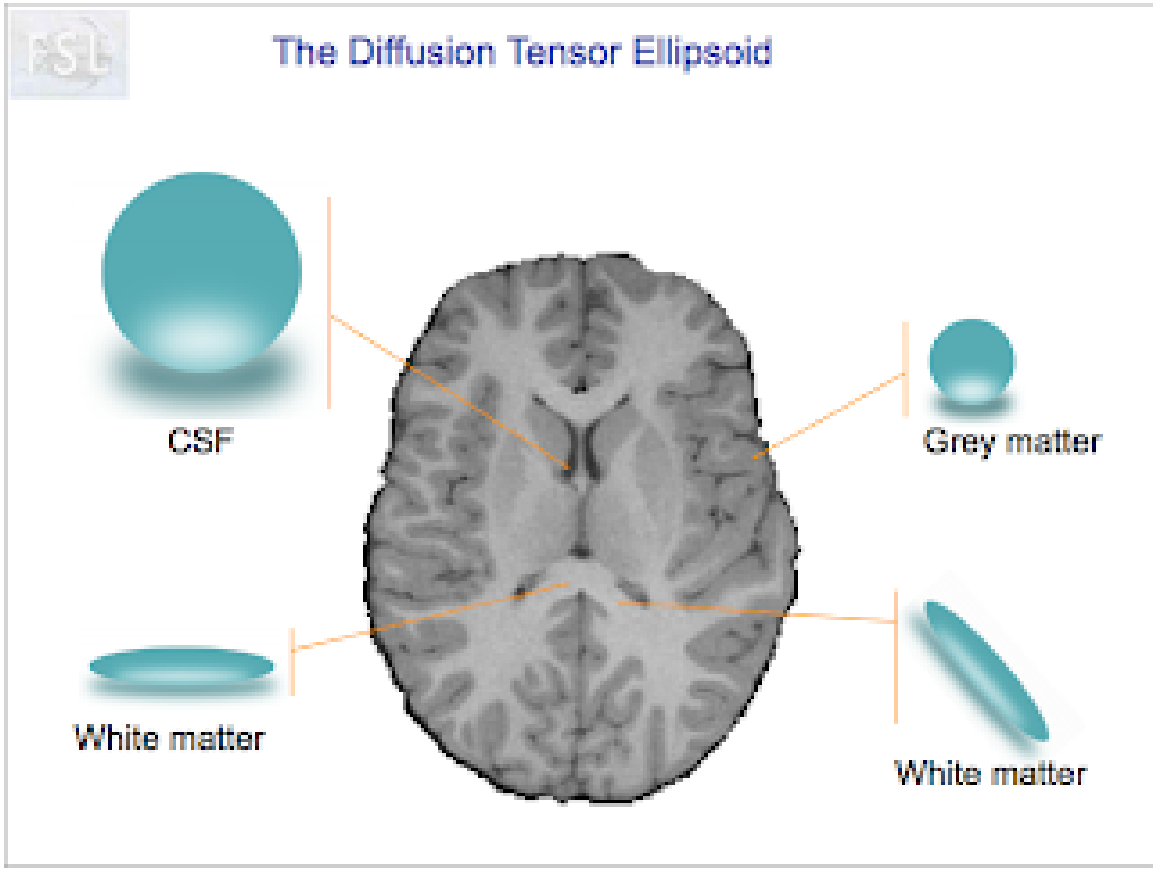
Fractional Anisotropy

Tensors

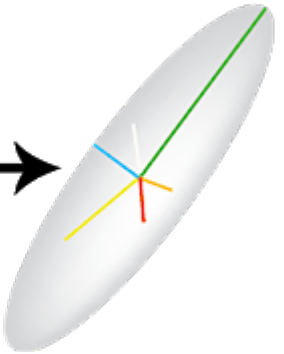
Fiber tracking

White matter delineation

Axons is what we want

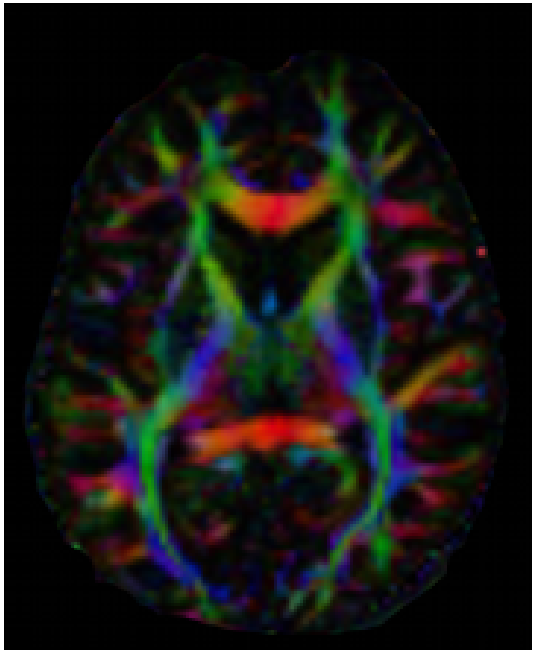


$$\underline{D} = \left[\right] = \begin{bmatrix} D_{xx} & D_{xy} & D_{xz} \\ D_{yx} & D_{yy} & D_{yz} \\ D_{zx} & D_{zy} & D_{zz} \end{bmatrix}$$



Diffusion files

1) A dwidata file.
2) A bvals files contains a scalar value for each applied gradient, corresponding to the respective b-value.
3) A bvecs contains a 3x1 vector for each gradient, indicating the gradient direction. The entries in bvals and bvecs are as many as the number of volumes in the dwidata file. The ith volume in the data corresponds to a measurement obtained after applying a diffusion-sensitising gradient with a b-value given by the ith entry in bvals and a gradient direction given by the ith vector in bvecs.



Jupiter Notebooks

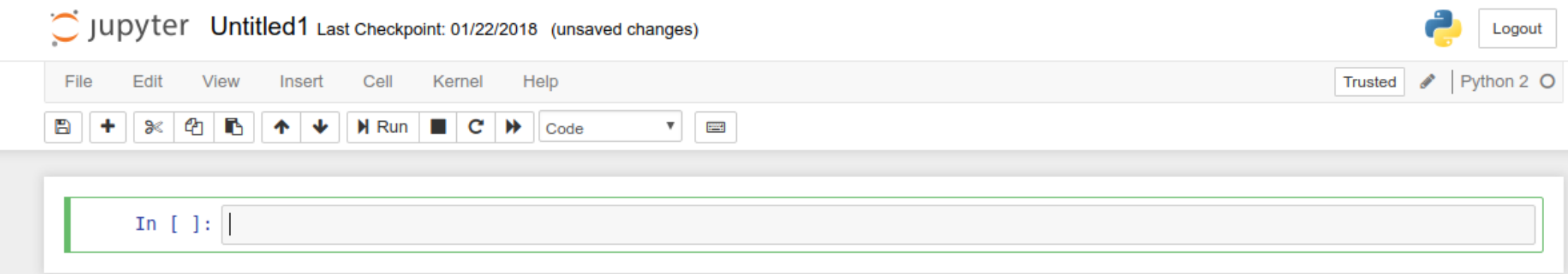
First, install Ipython:
sudo apt-get -y install ipython ipython-notebook

Now we can move on to installing Jupyter Notebook:

sudo -H pip install jupyter

Running Jupyter Notebook

jupyter notebook



Load data

```
import dipy
import numpy as np
import nibabel as nib

fimg = "dti_fm_bet.nii.gz"
img = nib.load(fimg)
data = img.get_data()
affine = img.get_affine()
header = img.get_header()
voxel_size = header.get_zooms()[:3]
mask, S0_mask = median_otsu(data[:, :, :, 0])
fbval = "../../../../bvals"
fbvec = "../../../../bvecs"

bvals, bvecs = read_bvals_bvecs(fbval, fbvec)
gtab = gradient_table(bvals, bvecs)
```

We use the fractional anisotropy (FA) of the DTI model to build a tissue classifier.

-
- ten_model = TensorModel(gtab)
- ten_fit = ten_model.fit(data, mask)
-
- fa = fractional_anisotropy(ten_fit.evals)
- cfa = color_fa(fa, ten_fit.evecs)
-

Constant Solid Angle

The first thing we need to begin fiber tracking is a way of getting directions from this diffusion data set. In order to do that, we can fit the data to a Constant Solid Angle ODF Model. This model will estimate the Orientation Distribution Function (ODF) at each voxel. The ODF is the distribution of water diffusion as a function of direction. The peaks of an ODF are good estimates for the orientation of tract segments at a point in the image.

- csamodel = CsaOdfModel(gtab, 6)
- sphere = get_sphere('symmetric724')
- sphere = get_sphere('symmetric724')
-
- pmd = peaks_from_model(model=csamodel,
- data=data,
- sphere=sphere,
- relative_peak_threshold=.5,
- min_separation_angle=25,
- mask=mask,
- return_odf=False)
-

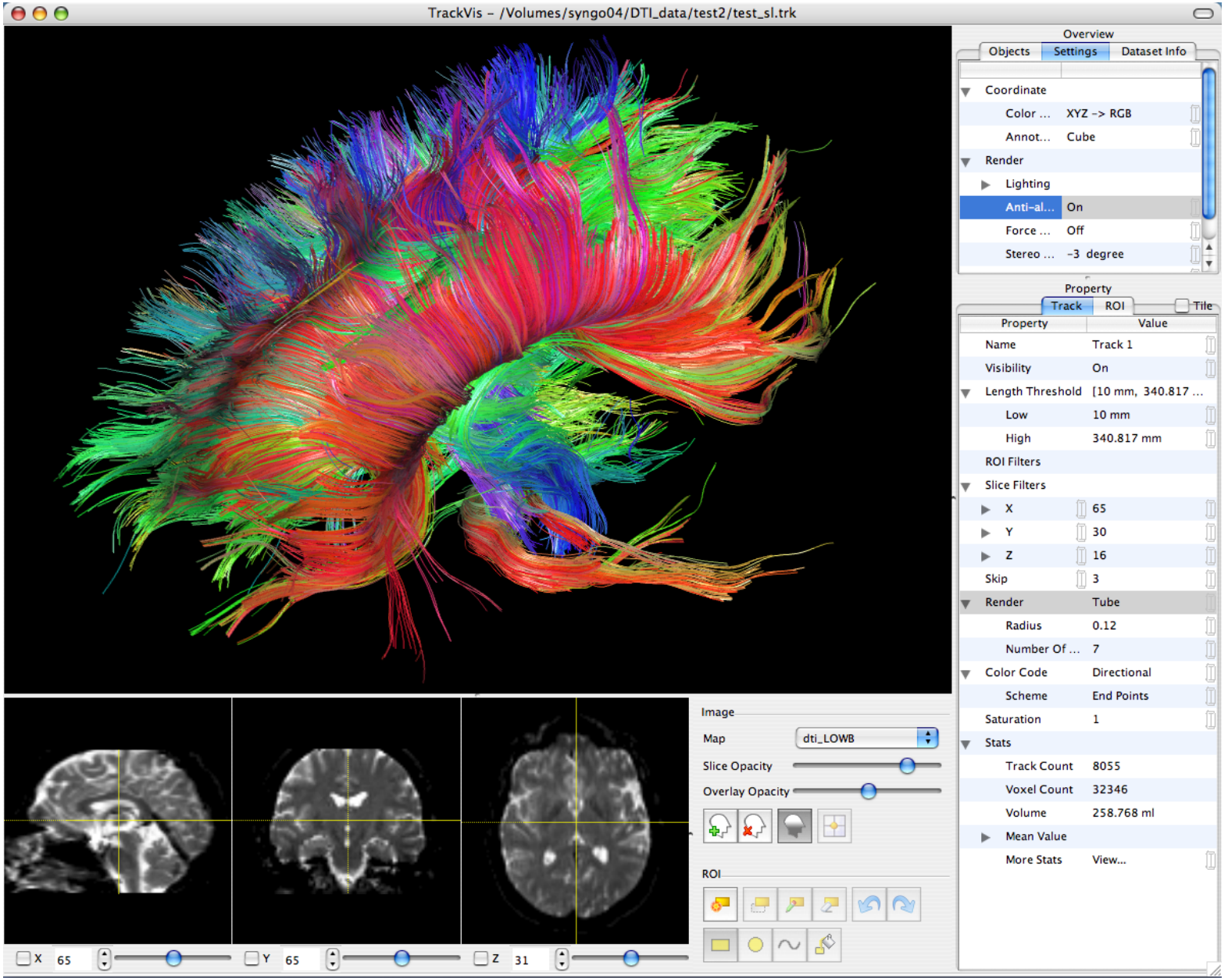
Perform Tractography

- #Deterministic tractography
- eu = EuDX(a=fa, ind=pmd.peak_indices[..., 0], seeds=2000000, odf_vertices=sphere.vertices, a_low=0.01)
- affine = eu.affine
- csd_streamlines= list(eu)
-
- #Remove tracts shorter than 30mm
- #print np.shape(csd_streamlines)
- from dipy.tracking.utils import length
- csd_streamlines=[t for t in csd_streamlines if length(t)>30]
-

Save as a trackvis

- #Trackvis
- hdr = nib.trackvis.empty_header()
- hdr['voxel_size'] = img.get_header().get_zooms()[:3]
- hdr['voxel_order'] = 'LAS'
- hdr['dim'] = fa.shape
- tensor_streamlines_trk = ((sl, None, None) for sl in csd_streamlines)
- ten_sl_fname = 'tensor_streamlines.trk'
- nib.trackvis.write(ten_sl_fname, tensor_streamlines_trk, hdr, points_space='voxel')
-

Trackvis


TrackVis – /Volumes/syngo04/DTI_data/test2/test_sl.trk
Overview
Objects
Settings
Dataset Info
Coordinate
Color ...
XYZ -> RGB
Annot...
Cube
Render
Lighting
Anti-al...
On
Force ...
Off
Stereo ...
-3 degree
Property
Track
ROI
Tile
Property
Value
Name
Track 1
Visibility
On
Length Threshold
[10 mm, 340.817 ...
Low
10 mm
High
340.817 mm
ROI Filters
Slice Filters
X
65
Y
30
Z
16
Skip
3
Render
Tube
Radius
0.12
Number Of ...
7
Color Code
Directional
Scheme
End Points
Saturation
1
Stats
Track Count
8055
Voxel Count
32346
Volume
258.768 ml
Mean Value
More Stats
View...
Image
Map
dti_LOWB
Slice Opacity
Overlay Opacity
ROI
X
65
Y
65
Z
31