

Advanced Biomedical Imaging Analysis:

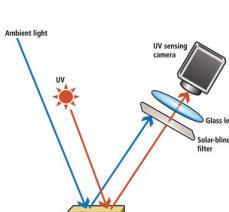
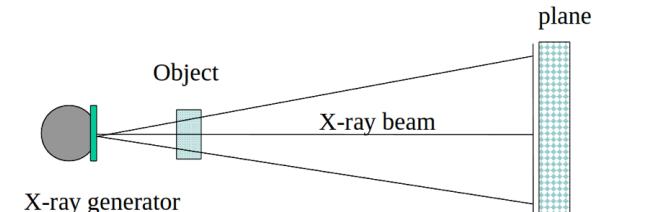
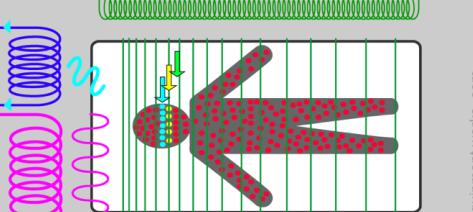
The Road Toward Deep Learning: Neuroimage Preprocessing and Postprocessing

Presented by Christian McDaniel

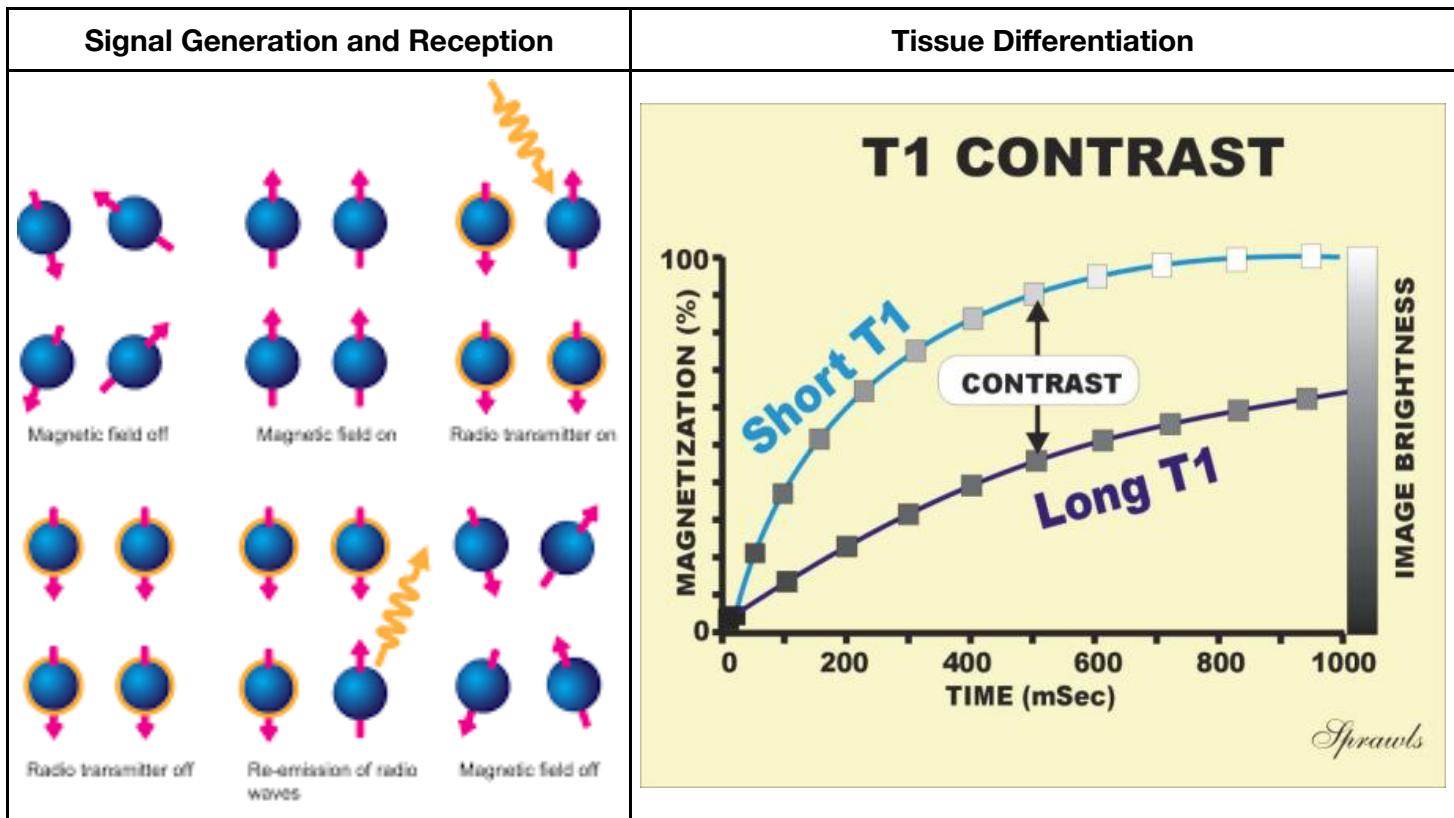
Background

Background:

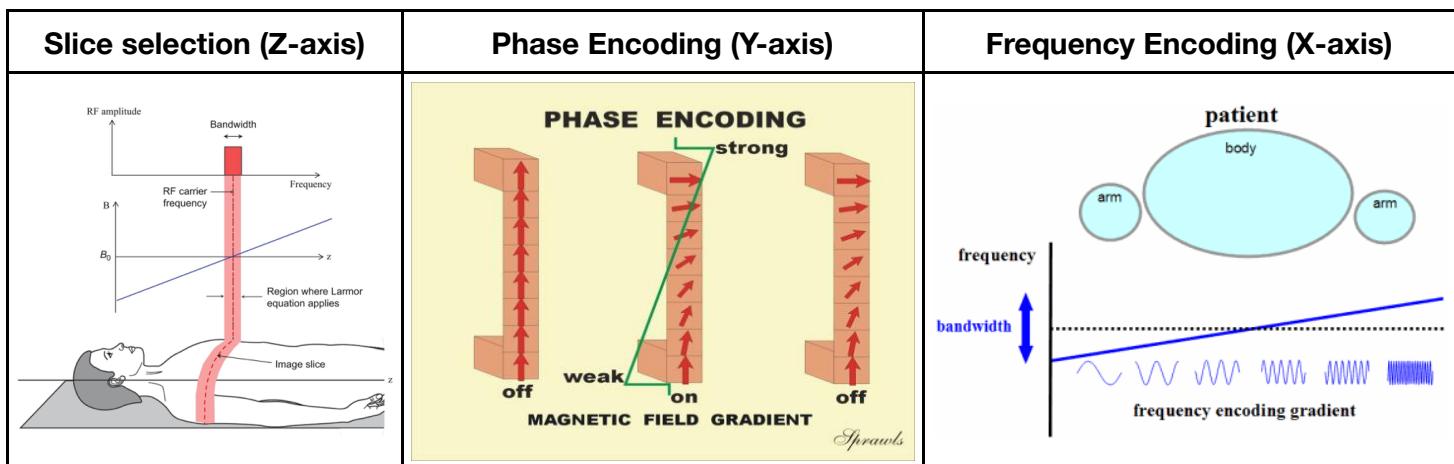
Magnetic Resonance-Based Imaging

Optical Imaging	X-Ray Imaging	Magnetic Resonance Imaging
		 <p>how equipment works .com</p>

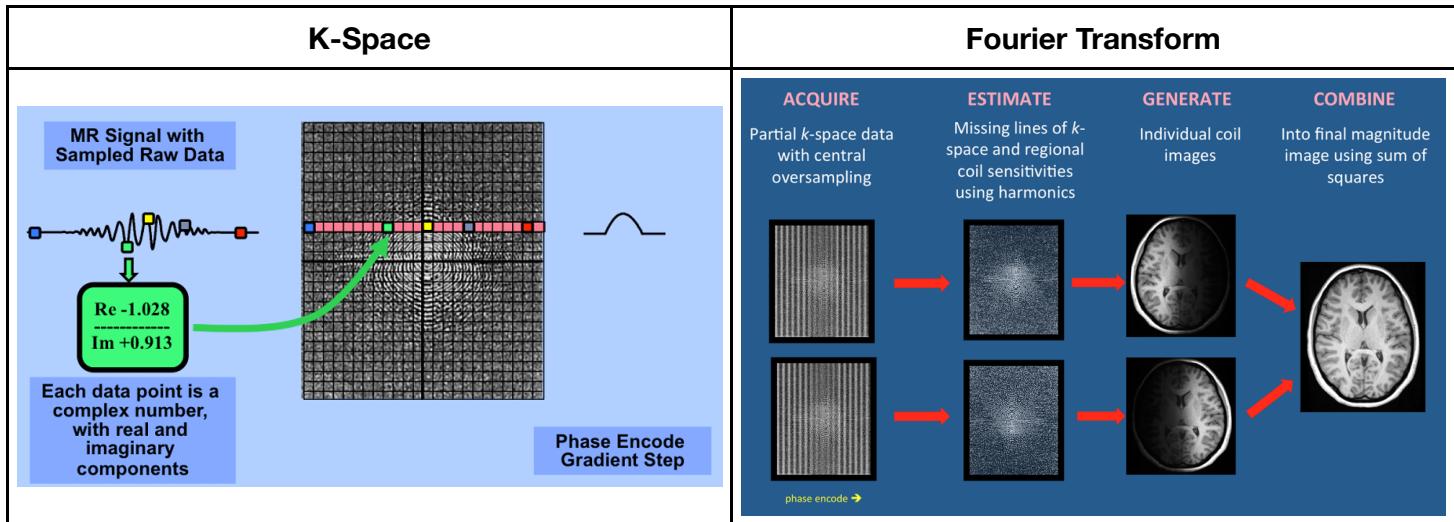
Magnetic Resonance Imaging



Magnetic Resonance Imaging - Signal Localization

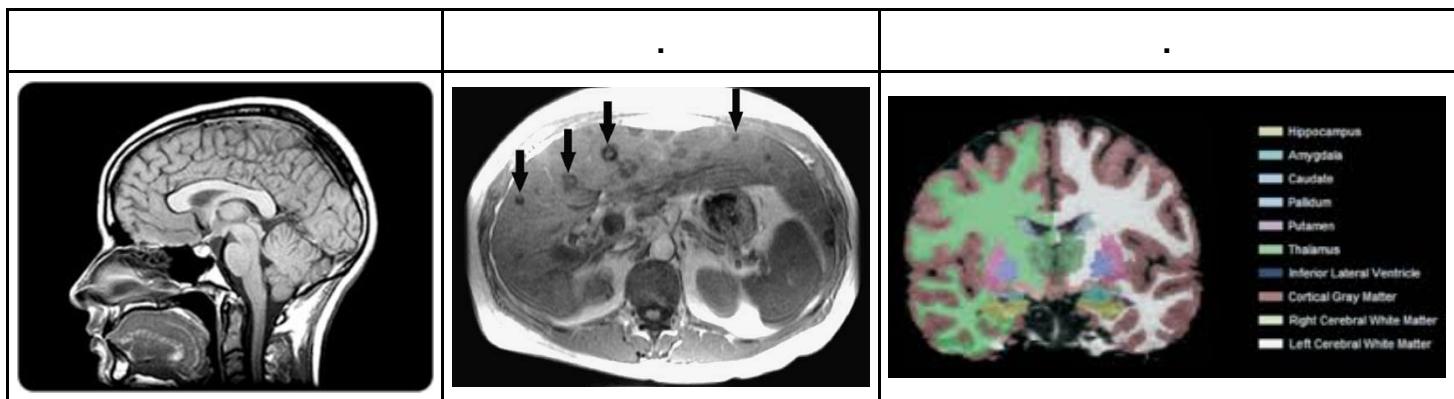


Magnetic Resonance Imaging - Image Construction

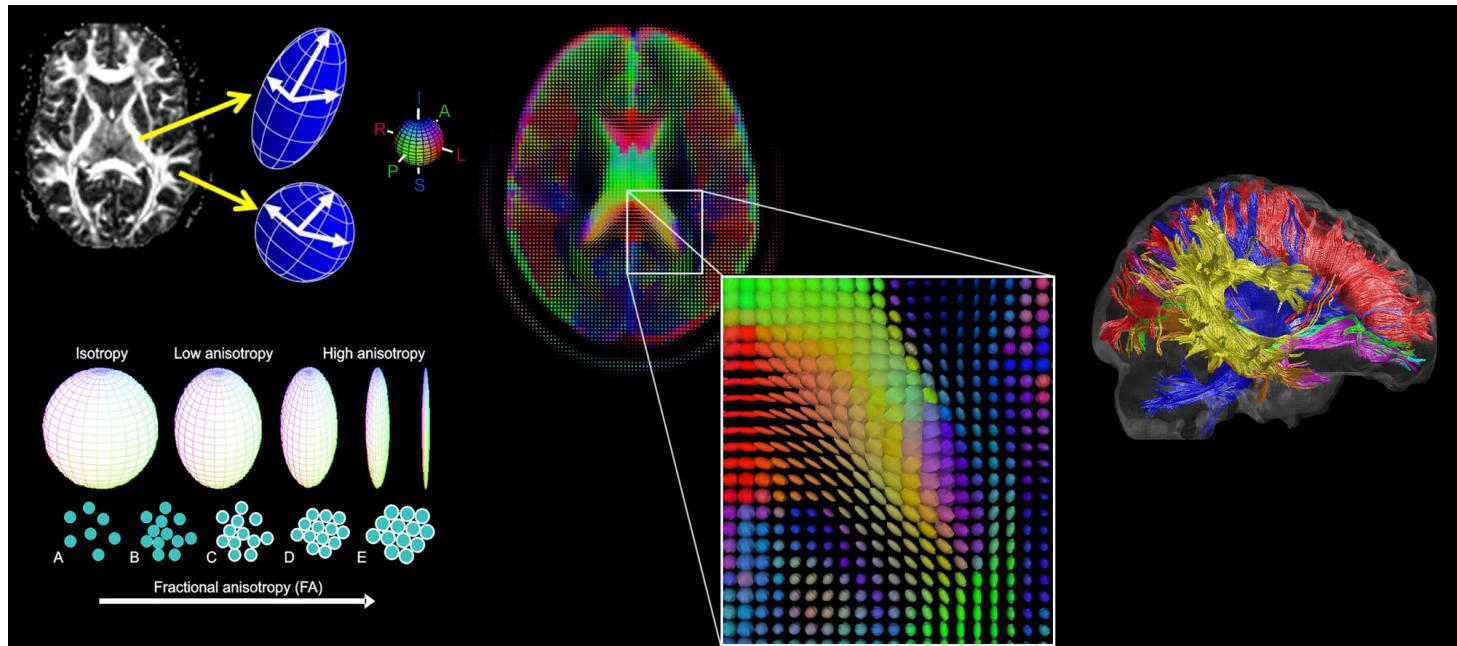


MRI Modalities

MRI: Anatomical Insights



Diffusion Weighted MR Imaging: Connectivity Insights



function MR Imaging: Functional Insights

experiment paradigm	voxel-wise tracking	network inference														
<p>fMRI block design paradigm</p>	<p>fMRI data time series</p>	<p>Cognitive key networks</p> <table border="1"> <thead> <tr> <th>aDMN</th> <th>pvDMN</th> <th>pdDMN</th> <th>accSN</th> <th>insSN</th> <th>ICEN</th> <th>rCEN</th> </tr> </thead> <tbody> <tr> <td>[−5,45,6]</td> <td>[6,−59,34]</td> <td>[5,−67,34]</td> <td>[−5,30,6]</td> <td>[−2,16,−2]</td> <td>[−42,30,40]</td> <td>[45,18,43]</td> </tr> </tbody> </table> <p>MNI-coordinates [x,y,z]</p>	aDMN	pvDMN	pdDMN	accSN	insSN	ICEN	rCEN	[−5,45,6]	[6,−59,34]	[5,−67,34]	[−5,30,6]	[−2,16,−2]	[−42,30,40]	[45,18,43]
aDMN	pvDMN	pdDMN	accSN	insSN	ICEN	rCEN										
[−5,45,6]	[6,−59,34]	[5,−67,34]	[−5,30,6]	[−2,16,−2]	[−42,30,40]	[45,18,43]										

MRI combines...

MRI combines...

... a symphony of magnetic transmissions ...

MRI combines...

... a symphony of magnetic transmissions ...

... physical properties at the atomic level ...

MRI combines...

... a symphony of magnetic transmissions ...

... physical properties at the atomic level ...

... digitization and mathematical computation ...

MRI combines...

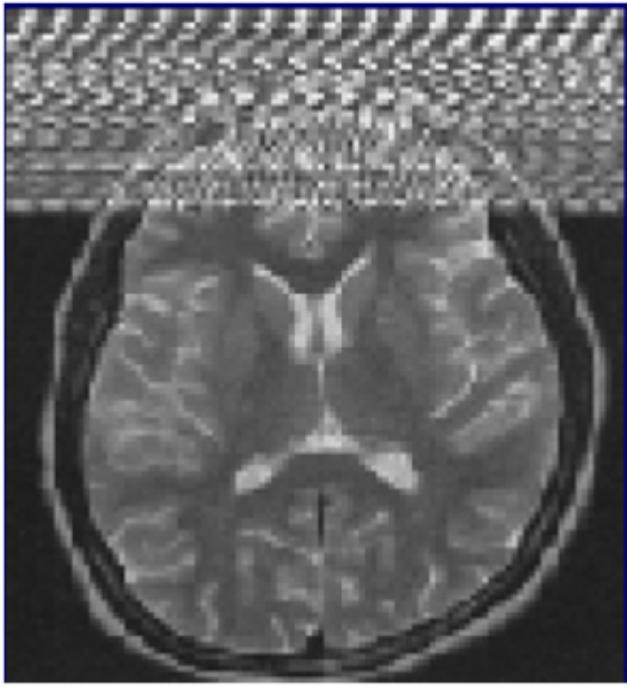
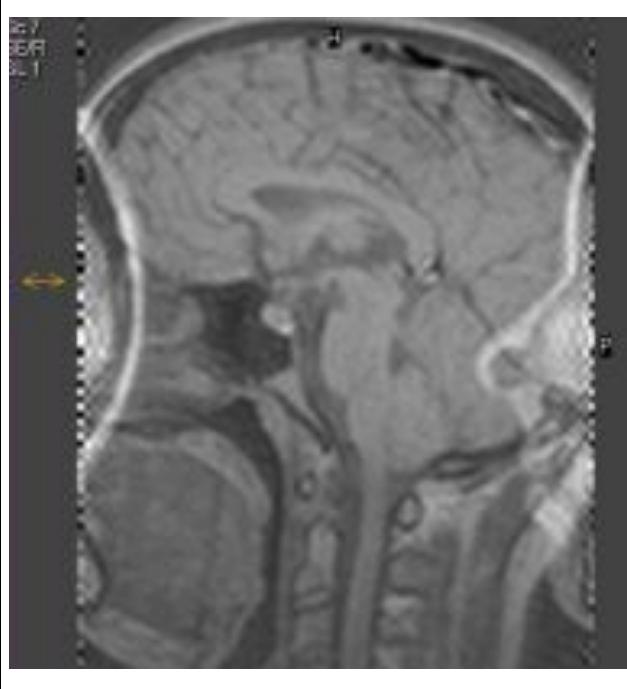
... a symphony of magnetic transmissions ...

... physical properties at the atomic level ...

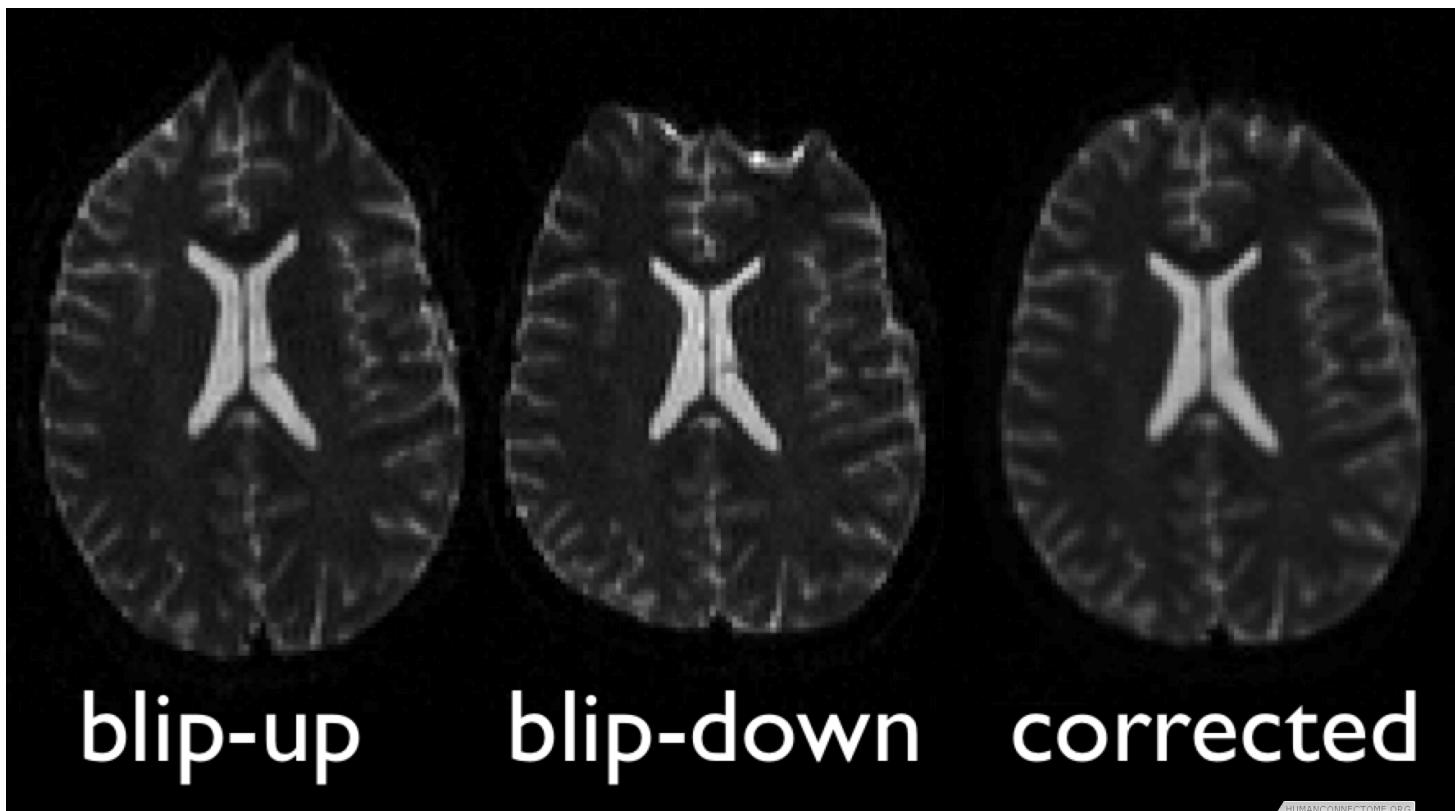
... digitization and mathematical computation ...

... and each introduces a source of noise!

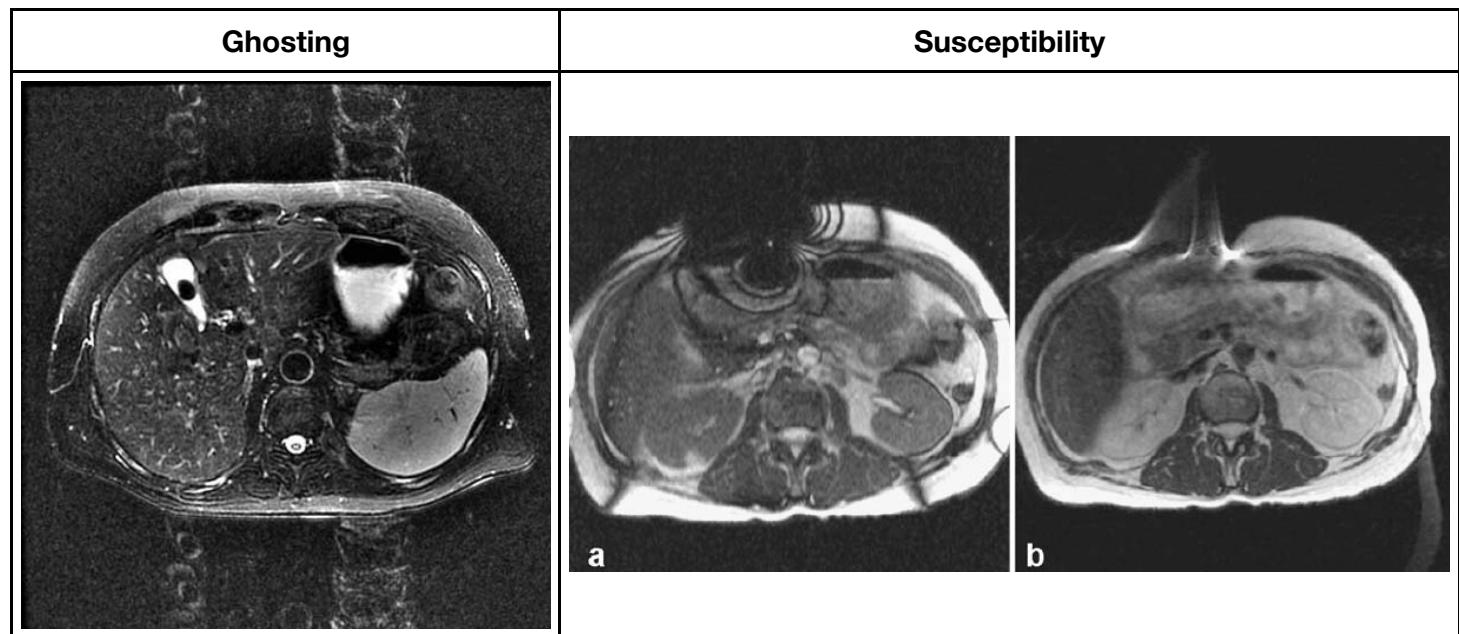
Noise: MRI Artifacts

Signal Transmission - Zipper Effect	Signal Acquisition - Aliasing
	

Noise: MRI Artifacts - Gradient-Related Artifacts

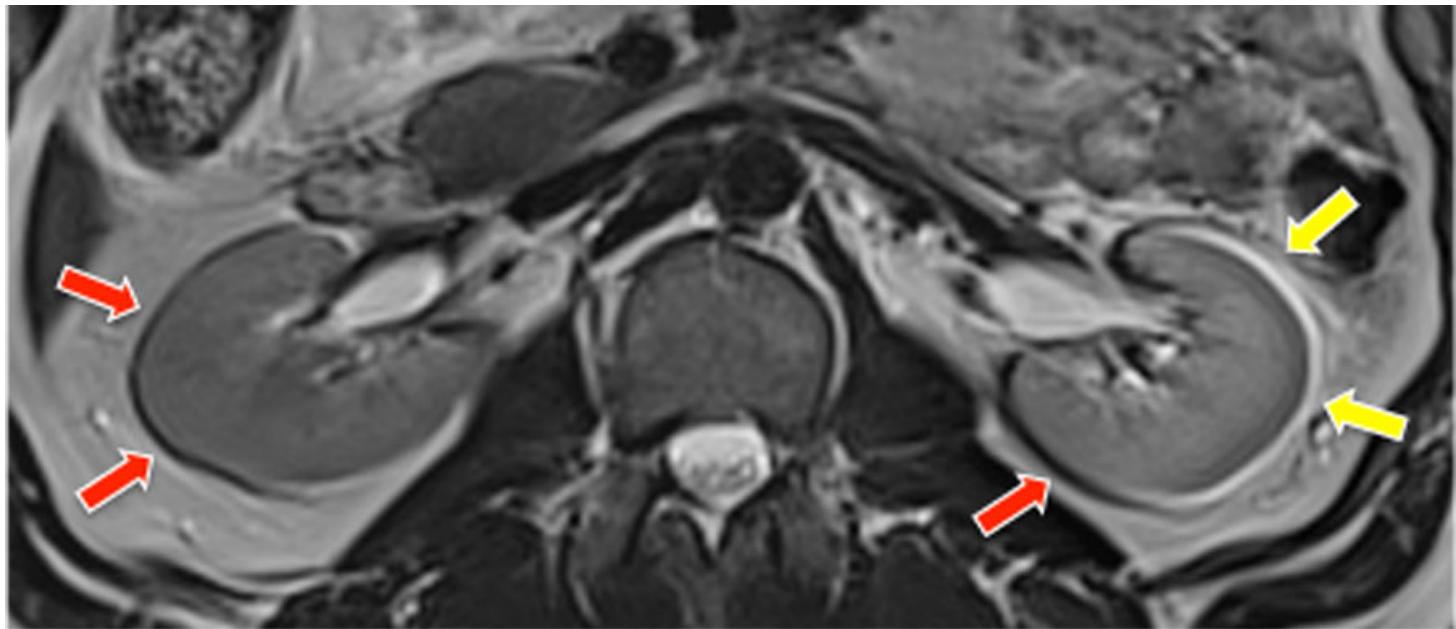


Noise: MRI Artifacts - Magnetization



Noise: MRI Artifacts - Magnetization

Chemical Shift:



The acquisition physics are directly tied to the artifacts.....

The acquisition physics are directly tied to the artifacts.....

..... which are directly tied to the problematic processing

E.g., *Hidden Cues: Deep Learning for Alzheimer's Disease (2016)*

	tSNE
+ Used tri-axial RAW MRI data + Transfer learning from AlexNet + data augmentation via mirror flip + 66.51% accuracy re: classification of AD v. HC	

So...

let's clean things up

Preprocessing

Preprocessing

First, let's look at what we get from the image lab...

Preprocessing

Step 1: Dicom to Nifti

Preprocessing

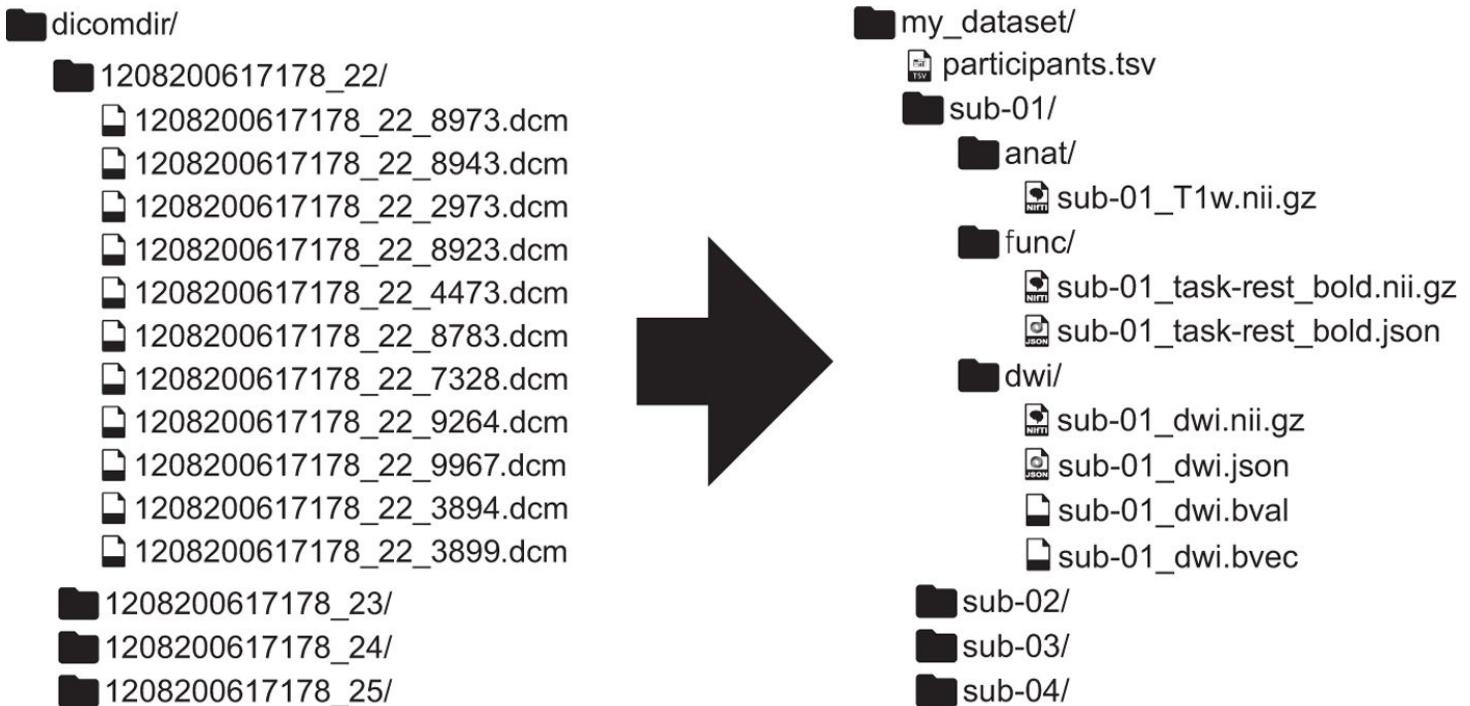
Step 1: Dicom to Nifti

DICOM	NIFTI
Digital Imaging and Communications in Medicine	Neuroimaging Informatics Technology Initiative
Many volumes are taken for one image	Processing is concerned with the composite
The emphasis is acquisition and interchange standards	The emphasis is compatibility and interpretability

Preprocessing

Step 1: Dicom to Nifti

BIDS: Brain Imaging Data Structure

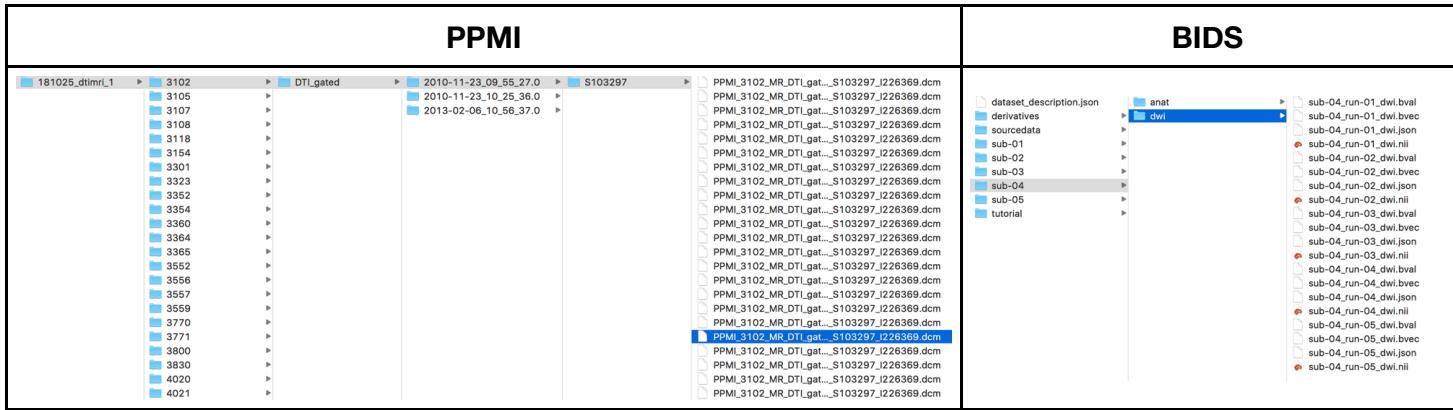


```
import os
import subprocess

"""
Create a folder for each **subject**, then for each **modality**,
and then distinguish each imaging 'run'.
"""

# create list for a single modality
rawdiffs = list([rootr for rootr,dirnamesr,filenamesr in os.walk(subject) \
                 if len(rootr.split('/'))==6 and 'DTI' in rootr.split('/')[3]])

# generate the file structure and naming; and convert dcm to nii
for rawdiff in rawdiffs:
    run+=1
    newfn = sub_name+'_run-0{}_dwi'.format(run) # new filename
    newdir = os.path.join(main_dir,sub_name,'dwi/') # new path
    # perform bash command dcm2niix
    cmd = "dcm2niix -b y -o {} -f {} {}".format(newdir,newfn,rawdi
ff)
    process = subprocess.Popen(cmd.split(), stdout=subprocess.PIPE
)
    output, error = process.communicate()
    # move the file
    if not os.path.exists(srcdata+sub_name):
        os.mkdir(srcdata+sub_name)
    os.rename(rawdiff,srcdata+sub_name+'/rawdcm_run-0{}/'.format(r
un))
    run=0
```



In [13]:

```
"""
dcm2niix creates a json file with pertinent header information
"""

{
    "Modality": "MR",
    "MagneticFieldStrength": 3,
    "Manufacturer": "Siemens",
    "ManufacturersModelName": "TrioTim",
    "InstitutionName": "Baylor_College_of_Medicine",
    "DeviceSerialNumber": "4TUMzYL029kw",
    "PatientPosition": "HFS",
    "SoftwareVersions": "syngo_MR_B15",
    "MRAcquisitionType": "2D",
    "SeriesDescription": "DTI",
    "ProtocolName": "DTI",
    "ScanningSequence": "EP",
    "SequenceVariant": "SK_SP",
    "ScanOptions": "PFP_FS",
    "SequenceName": "_ep_b0",
    "ImageType": ["ORIGINAL", "PRIMARY", "M", "ND", "NORM", "MOSAI
C"],
    "SeriesNumber": 3,
    "AcquisitionTime": "10:36:41.432500",
    "AcquisitionNumber": 1,
    "SliceThickness": 2,
    "SpacingBetweenSlices": 2,
    "SAR": 0.2452,
    "EchoTime": 0.088,
    "RepetitionTime": 9,
    "FlipAngle": 90,
    "PartialFourier": 0.75,
    "BaseResolution": 116,
    "ShimSetting": [
        14415,
        -12729,
        3293,
    ]
}
```

```
        400,
        -102,
        -700,
        -291,
        -99      ],
    "TxRefAmp": 332.863,
    "PhaseResolution": 1,
    "ReceiveCoilName": "HeadMatrix",
    "PulseSequenceDetails": "%SiemensSeq%_ep2d_diff",
    "PercentPhaseFOV": 100,
    "PhaseEncodingSteps": 90,
    "AcquisitionMatrixPE": 116,
    "ReconMatrixPE": 116,
    "BandwidthPerPixelPhaseEncode": 23.618,
    "ParallelReductionFactorInPlane": 2,
    "EffectiveEchoSpacing": 0.000365005,
    "DerivedVendorReportedEchoSpacing": 0.00073001,
    "TotalReadoutTime": 0.0419756,
    "PixelBandwidth": 1595,
    "DwellTime": 2.7e-06,
    "PhaseEncodingDirection": "j-",
    "SliceTiming": [
        4.5175,
        0,
        4.6425,
        0.125,
        4.7675,
        0.25,
        4.895,
        0.3775,
        5.02,
        0.5025,
        5.145,
        0.6275,
        5.27,
        0.7525,
        5.395,
        0.8775,
        5.5225,
        1.005,
        5.6475,
        1.13,
        5.7725,
        1.255,
        5.8975,
        1.38,
        6.0225,
        1.505,
        6.15,
        1.6325,
```

```
6.275,
1.7575,
6.4,
1.8825,
6.525,
2.0075,
6.65,
2.1325,
6.7775,
2.26,
6.9025,
2.385,
7.0275,
2.51,
7.1525,
2.635,
7.2775,
2.76,
7.405,
2.8875,
7.53,
3.0125,
7.655,
3.1375,
7.78,
3.2625,
7.905,
3.3875,
8.0325,
3.515,
8.1575,
3.64,
8.2825,
3.765,
8.4075,
3.89,
8.5325,
4.015,
8.66,
4.14,
8.785,
4.2675,
8.91,
4.3925 ],
"ImageOrientationPatientDICOM": [
  1,
  -1e-16,
  0,
  1e-16,
  0.944949,
```

```
        -0.327218      ],
    "InPlanePhaseEncodingDirectionDICOM": "COL",
    "ConversionSoftware": "dcm2niix",
    "ConversionSoftwareVersion": "v1.0.20180622 (JP2:OpenJPEG) GCC
6.1.0"
}
```

```
Out[13]: {'AcquisitionMatrixPE': 116,
'AcquisitionNumber': 1,
'AcquisitionTime': '10:36:41.432500',
'BandwidthPerPixelPhaseEncode': 23.618,
'BaseResolution': 116,
'ConversionSoftware': 'dcm2niix',
'ConversionSoftwareVersion': 'v1.0.20180622 (JP2:OpenJPEG) GCC6.1.0
',
'DerivedVendorReportedEchoSpacing': 0.00073001,
'DeviceSerialNumber': '4TUMzYL029kw',
'DwellTime': 2.7e-06,
'EchoTime': 0.088,
'EffectiveEchoSpacing': 0.000365005,
'FlipAngle': 90,
'ImageOrientationPatientDICOM': [1, -1e-16, 0, 1e-16, 0.944949, -0.
327218],
'ImageType': ['ORIGINAL', 'PRIMARY', 'M', 'ND', 'NORM', 'MOSAIC'],
'InPlanePhaseEncodingDirectionDICOM': 'COL',
'InstitutionName': 'Baylor_College_of_Medicine',
'MRAcquisitionType': '2D',
'MagneticFieldStrength': 3,
'Manufacturer': 'Siemens',
'ManufacturersModelName': 'TrioTim',
'Modality': 'MR',
'ParallelReductionFactorInPlane': 2,
'PartialFourier': 0.75,
'PatientPosition': 'HFS',
'PercentPhaseFOV': 100,
'PhaseEncodingDirection': 'j-',
'PhaseEncodingSteps': 90,
'PhaseResolution': 1,
'PixelBandwidth': 1595,
'ProtocolName': 'DTI',
'PulseSequenceDetails': '%SiemensSeq%_ep2d_diff',
'ReceiveCoilName': 'HeadMatrix',
'ReconMatrixPE': 116,
'RepetitionTime': 9,
'SAR': 0.2452,
'ScanOptions': 'PFP_FS',
'ScanningSequence': 'EP',
'Sequencename': '_ep_b0',
'SequenceVariant': 'SK_SP',
'SeriesDescription': 'DTI',
```

```
'SeriesNumber': 3,  
'ShimSetting': [14415, -12729, 3293, 400, -102, -700, -291, -99],  
'SliceThickness': 2,  
'SliceTiming': [4.5175,  
 0,  
 4.6425,  
 0.125,  
 4.7675,  
 0.25,  
 4.895,  
 0.3775,  
 5.02,  
 0.5025,  
 5.145,  
 0.6275,  
 5.27,  
 0.7525,  
 5.395,  
 0.8775,  
 5.5225,  
 1.005,  
 5.6475,  
 1.13,  
 5.7725,  
 1.255,  
 5.8975,  
 1.38,  
 6.0225,  
 1.505,  
 6.15,  
 1.6325,  
 6.275,  
 1.7575,  
 6.4,  
 1.8825,  
 6.525,  
 2.0075,  
 6.65,  
 2.1325,  
 6.7775,  
 2.26,  
 6.9025,  
 2.385,  
 7.0275,  
 2.51,  
 7.1525,  
 2.635,  
 7.2775,  
 2.76,  
 7.405,
```

```
2.8875,
7.53,
3.0125,
7.655,
3.1375,
7.78,
3.2625,
7.905,
3.3875,
8.0325,
3.515,
8.1575,
3.64,
8.2825,
3.765,
8.4075,
3.89,
8.5325,
4.015,
8.66,
4.14,
8.785,
4.2675,
8.91,
4.3925],
'SoftwareVersions': 'syngo_MR_B15',
'SpacingBetweenSlices': 2,
'TotalReadoutTime': 0.0419756,
'TxRefAmp': 332.863}
```

```
In [2]: import subprocess
```

```
def bash_command(string_cmd):
    process = subprocess.Popen(string_cmd.split(), stdout=subprocess.PIPE)
    output, error = process.communicate()
    return output
```

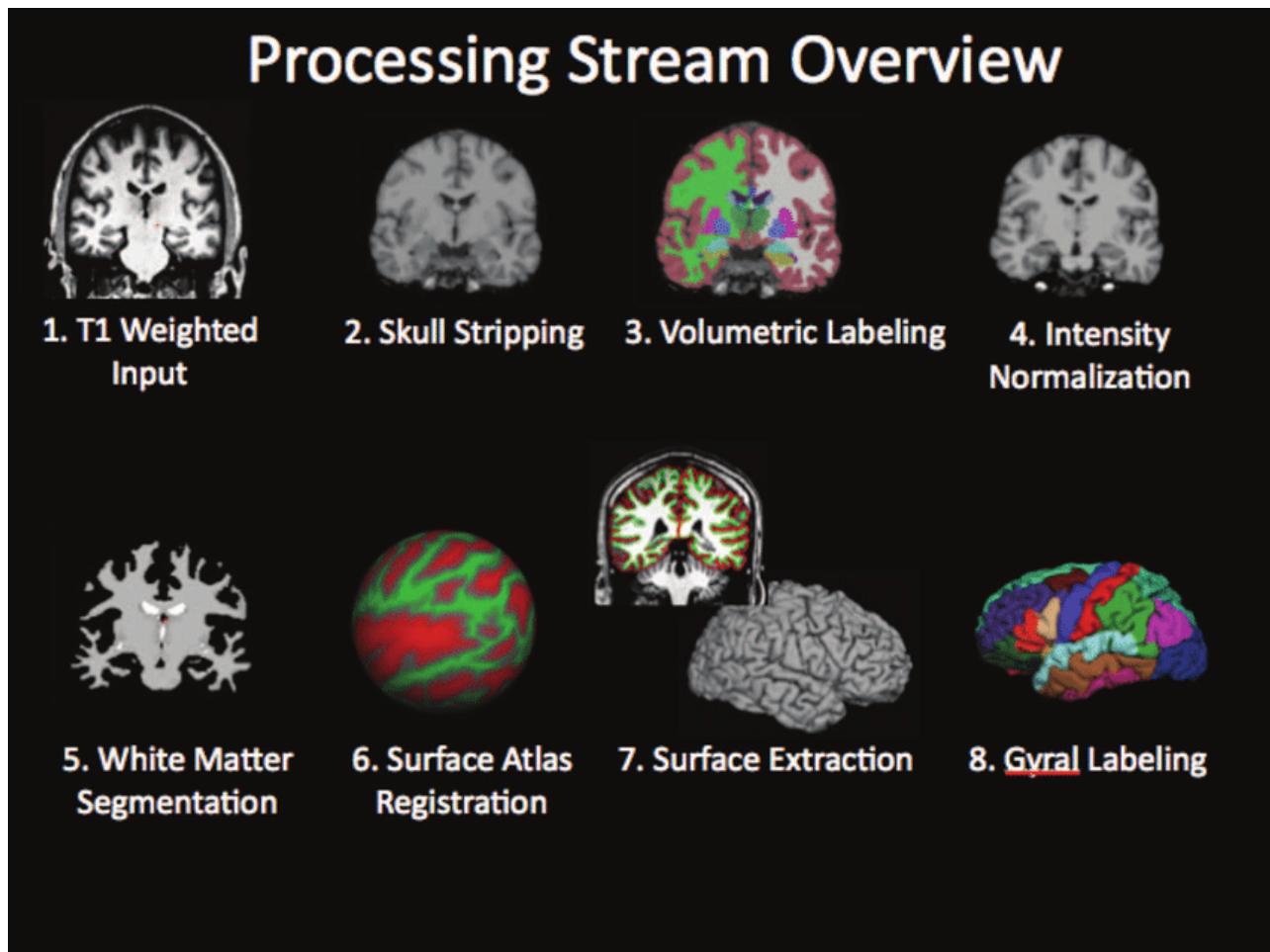
```
In [3]: # Let's check out our new nifty nifti image
bash_command('fsleyes ./sub-05_run-01_dwi.nii')
```

```
Out[3]: b''
```

Preprocessing

Preprocessing - MRI Anatomical Images

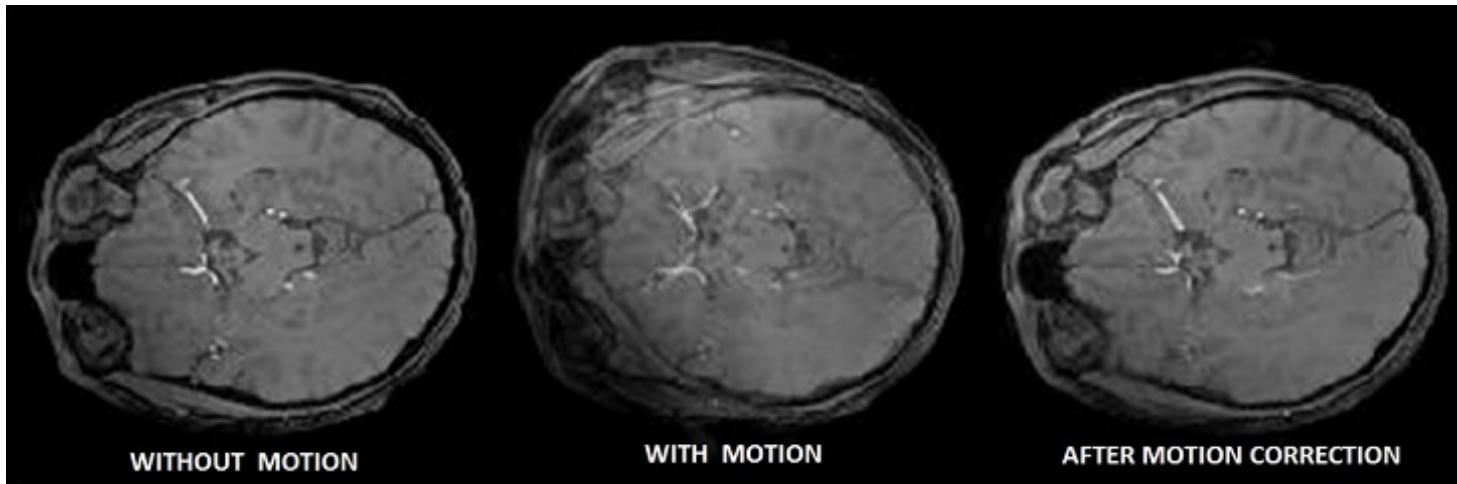
Freesurfer (Harvard, MGH)



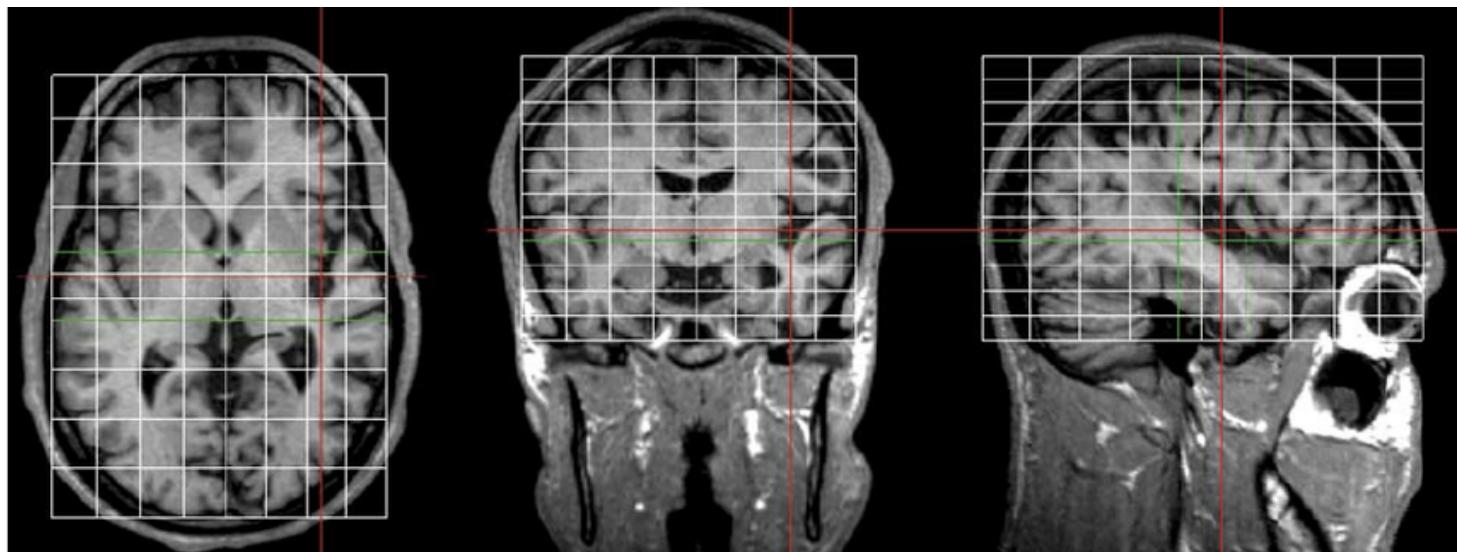
```
In [4]: #
bash_command('fsleyes sub-09_run-01_T1w.nii')
```

```
Out[4]: b''
```

```
#-----  
#@# MotionCor Sat Nov 3 12:02:41 EDT 2018
```

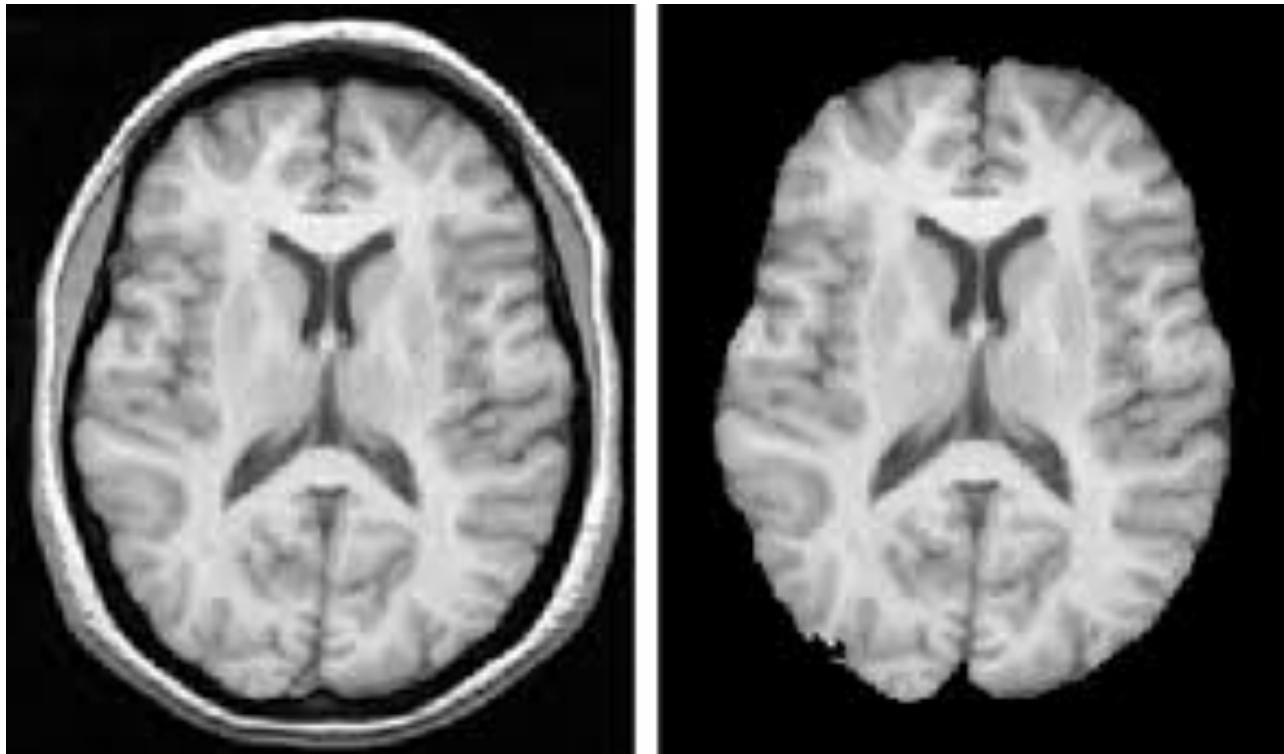


```
#-----  
#@# Talairach Sat Nov 3 12:02:51 EDT 2018
```



```
#-----
#@# Nu Intensity Correction Sat Nov 3 12:04:16 EDT 2018
mri_binarize --i ./tmp.mri_nu_correct.mni.14123/nu2.mnc --min -1 --o ./tmp
.mri_nu_correct.mni.14123/ones.mgz
#-----
#@# Intensity Normalization Sat Nov 3 12:06:05 EDT 2018
-----
3d normalization pass 1 of 2
white matter peak found at 110
white matter peak found at 105
gm peak at 56 (56), valley at 26 (26)
csf peak at 29, setting threshold to 47
building Voronoi diagram...
performing soap bubble smoothing, sigma = 8...
-----
```

@# Skull Stripping Sat Nov 3 12:08:11 EDT 2018

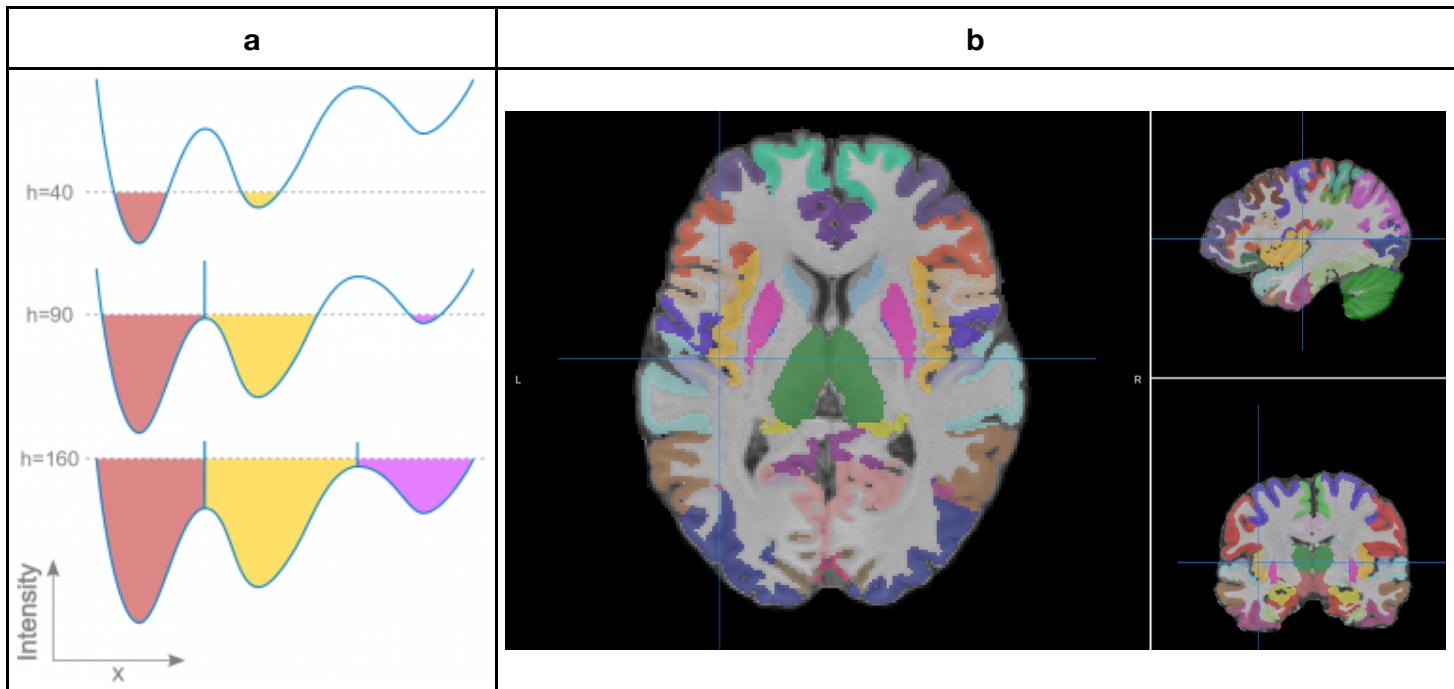


```
In [5]: # Checkpoint: Lets check the progress  
bash_command('fsleyes sub-09_run-01_fs/mri/T1.mgz sub-09_run-01_fs/mri/  
/brainmask.mgz')  
  
# and look at brain.mgz in MRICro for surface viz
```

```
Out[5]: b''
```

```
In [ ]: ****
The input file is T1.mgz
The output file is brainmask.auto.mgz
Weighting the input with atlas information before watershed
*****WATERSHED*****
Sorting...
    first estimation of the COG coord: x=122 y=124 z=105 r=95
    first estimation of the main basin volume: 3617151 voxels
    Looking for seedpoints
        2 found in the cerebellum
        15 found in the rest of the brain
    global maximum in x=146, y=113, z=80, Imax=255
    CSF=15, WM_intensity=110, WM_VARIANCE=5
    WM_MIN=110, WM_HALF_MIN=110, WM_HALF_MAX=110, WM_MAX=110
    preflooding height equal to 10 percent
done.
Analyze...

    main basin size=2383667146866273 voxels, voxel volume =1.000
                = 2383667146866273 mmm3 = 2383667120832.512 cm3
done.
PostAnalyze...Basin Prior
    141 basins merged thanks to atlas
        ***** 0 basin(s) merged in 1 iteration(s)
        ***** 0 voxel(s) added to the main basin
done.
Weighting the input with prior template
```

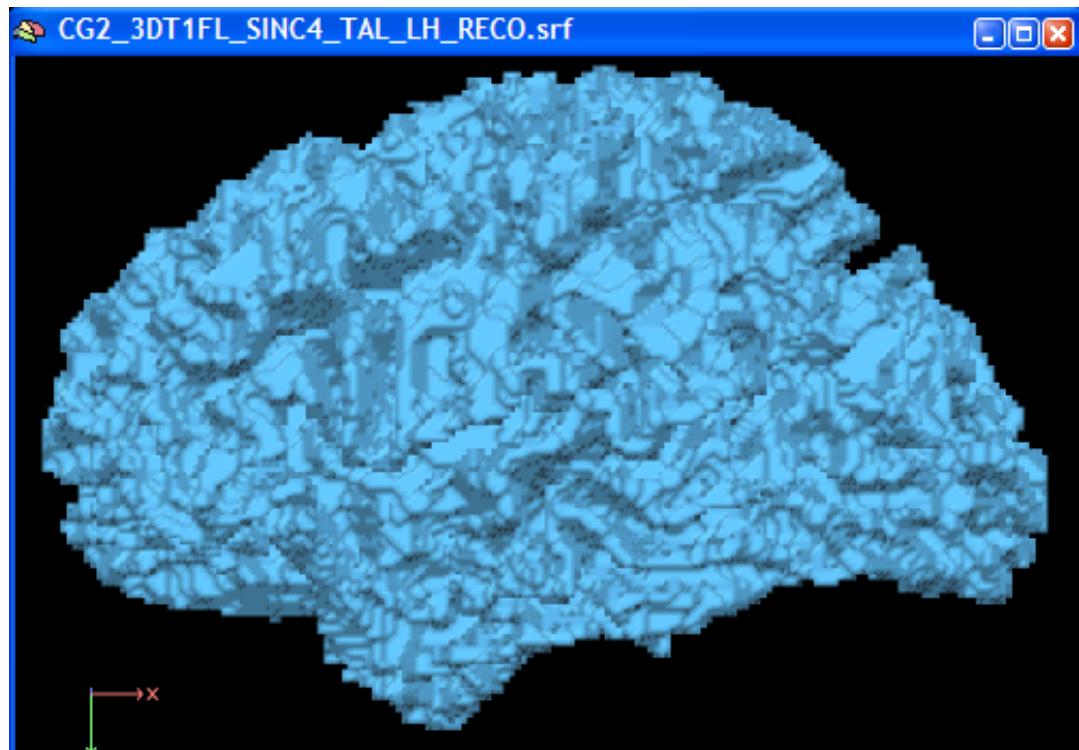


mri_strip_skull: done peeling brain

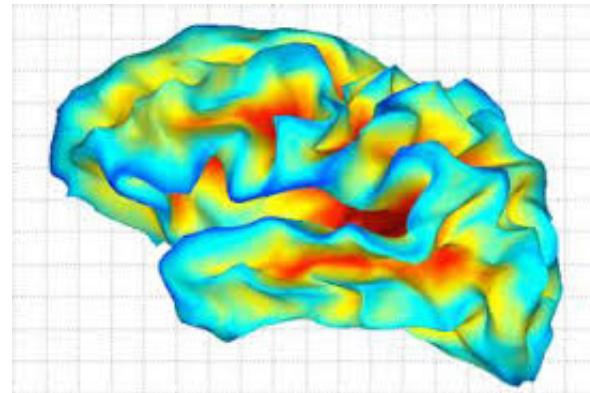
```
In [6]: # Checkpoint  
bash_command('freeview -v aseg.presurf.mgz:colormap=lut:opacity=1.0')  
# pretty good, but edges are not quite right
```

```
Out[6]: b''
```

```
#-----  
#@# SubCort Seg Sat Nov  3 14:31:51 EDT 2018  
#-----  
#@# WM Segmentation Sat Nov  3 15:23:59 EDT 2018  
#-----  
#@# Tessellate lh Sat Nov  3 15:26:37 EDT 2018  
#-----
```



```
#@# Smooth1 rh Sat Nov 3 15:26:55 EDT 2018
#-----
#@# Inflation1 lh Sat Nov 3 15:27:02 EDT 2018
#-----
#@# Fix Topology Copy 1h Sat Nov 3 15:33:20 EDT 2018
topology fixing took 103.9 minutes
```



In []: # see `wm.seg.mgz` in `MRIcro`!

<pre>#@# Make White Surf lh Sat Nov 3 23:57:50 EDT 2018 #@# Surf Reg rh Sun Nov 4 01:51:36 EDT 2018 #@# Jacobian white lh Sun Nov 4 01:45:45 EST 2018 #@# AvgCurv lh Sun Nov 4 01:45:48 EST 2018 #@# Cortical Parc lh Sun Nov 4 01:45:51 EST 2018 #@# Make Pial Surf lh Sun Nov 4 01:46:16 EST 2018 #@# Surf Volume lh Sun Nov 4 02:10:26 EST 2018 #@# Cortical ribbon mask Sun Nov 4 02:10:32 EST 2018</pre>	
---	--

```
In [7]: bash_command('fsleyes rh.ribbon.mgz')
```

```
Out[7]: b''
```

This takes a long time!

```
Started at Sat Nov 3 12:02:31 EDT 2018
Ended   at Sun Nov 4 02:55:07 EST 2018
#@#%# recon-all-run-time-hours 15.877
recon-all -s sub-02_run_01_fs finished without error at Sun Nov 4 02:55:07 EST 2018
done
```

```
In [8]: # Bust yields a nice result: Let's take a look!
```

```
bash_command('freeview -v sub-09_run-01_fs/mri/T1.mgz sub-09_run-01_fs/mri/wm.mgz sub-09_run-01_fs/mri/brainmask.mgz sub-09_run-01_fs/mri/aseg.mgz:colormap=lut:opacity=0.5 -f sub-09_run-01_fs/surf/lh.white:edgecolor=blue sub-09_run-01_fs/surf/lh.pial:edgecolor=red sub-09_run-01_fs/surf/rh.white:edgecolor=blue sub-09_run-01_fs/surf/rh.pial:edgecolor=red')
```

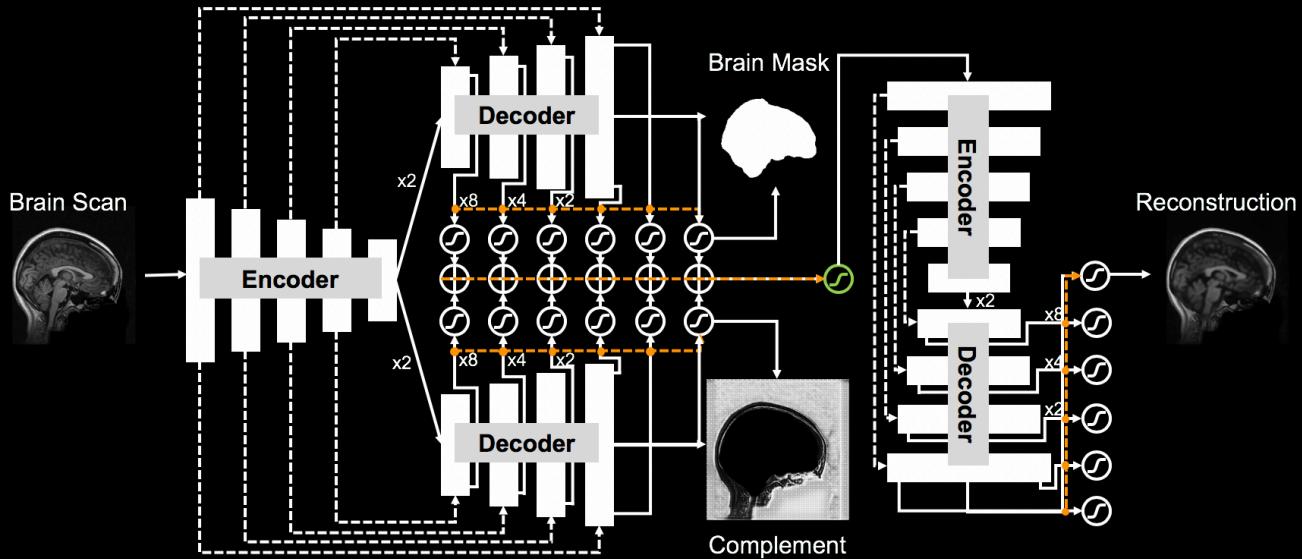
```
Out[8]: b''
```

The time requirement and limitations of traditional registration/segmentation of MRI data has led to deep learning-based approaches...

Deep Learning tackles Segmentation:

Slide borrowed from Dr Yi Hong

CompNet: Complementary Segmentation Network



[Dey and Hong, Accepted to MICCAI 2018]

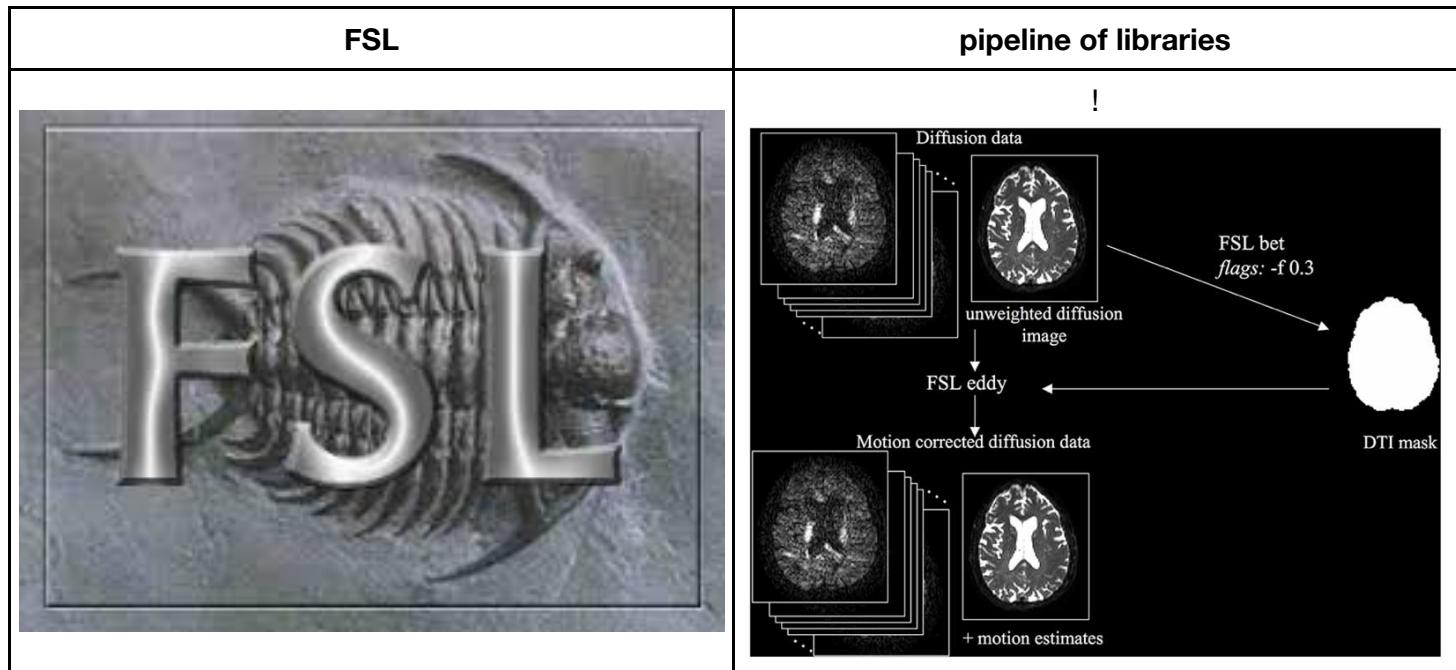
---> Concatenation

Ⓐ Sigmoid

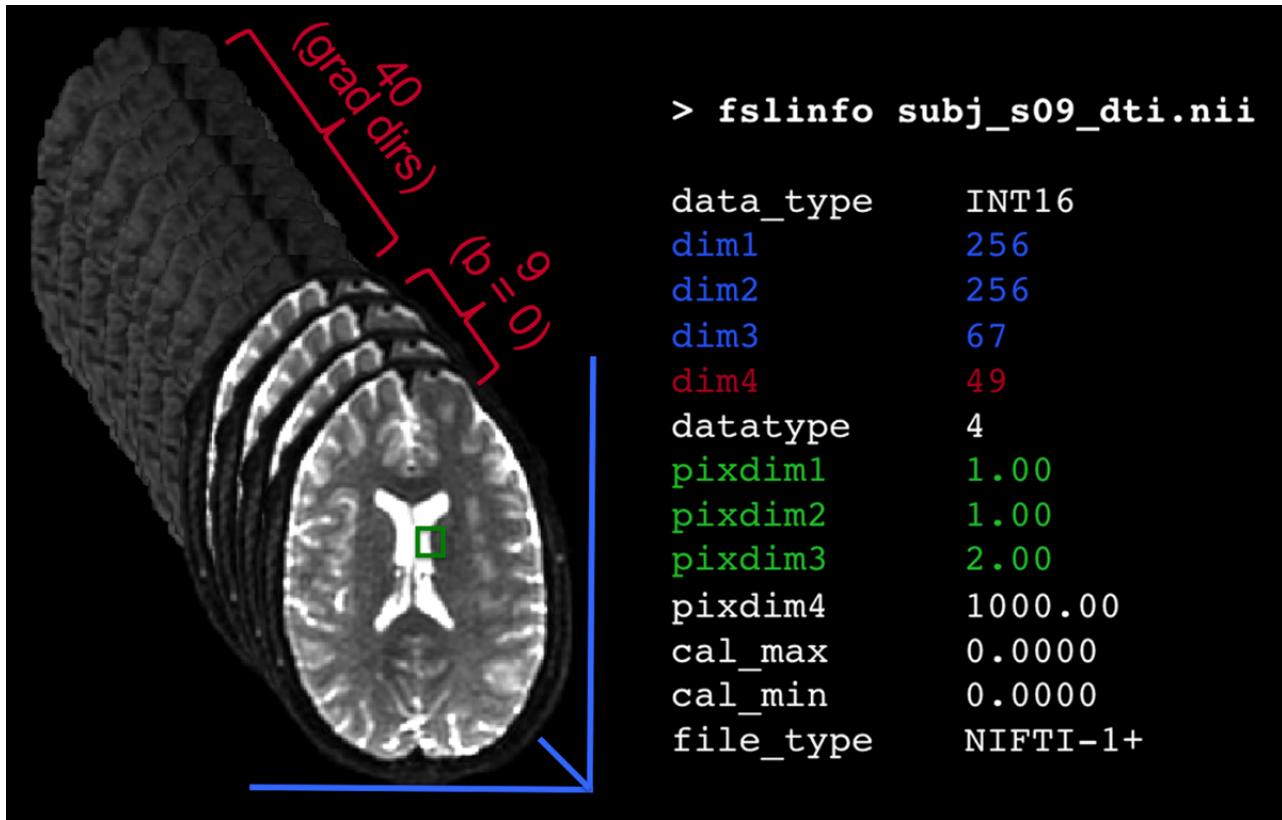
23/35

Preprocessing - DWI (Connectivity)

FMRIB Software Library - FSL (University of Oxford)



The key to dwi preprocessing: the b0



```
In [9]: # Find the b0 image in our dwi nifti file
bash_command('fsleyes ./sub-05_run-01_dwi.nii')
```

```
Out[9]: b''
```

```
In [10]: # Produce b0's for 2 or more acquisitions
bash_command('fslroi sub-05_run-01_dwi sub-05_run-01_b0 0 1')
bash_command('fslroi sub-05_run-02_dwi sub-05_run-02_b0 0 1')
bash_command('fslroi sub-05_run-03_dwi sub-05_run-03_b0 0 1')

# merge the b0's
bash_command('fslmerge -t sub-05_run-01-02-03_b0 sub-05_run-01_b0 sub-
05_run-02_b0 sub-05_run-03_b0')
bash_command('fsleyes ./sub-05_run-01-02-03_b0.nii.gz')
```

```
Out[10]: b''
```

The acquisition parameters:

- Each acquisition (image) has a phase encode direction and a readout time (image acquisition)

```
"TotalReadoutTime": 0.0419756,  
"PhaseEncodingDirection": "j-",
```

- Need at least two files that have opposing phase encode directions (along same axis) or different readout times

```
In [ ]: # Write file containing acquisition parameters  
bash_command('printf "0 -1 0 0.0419756\\n0 -1 0 0.0419756\\n 0 -1 0 0.04  
08244" > acq.txt')
```

FSL - topup

Susceptibility Artifacts

External magnetic fields induce local fields!

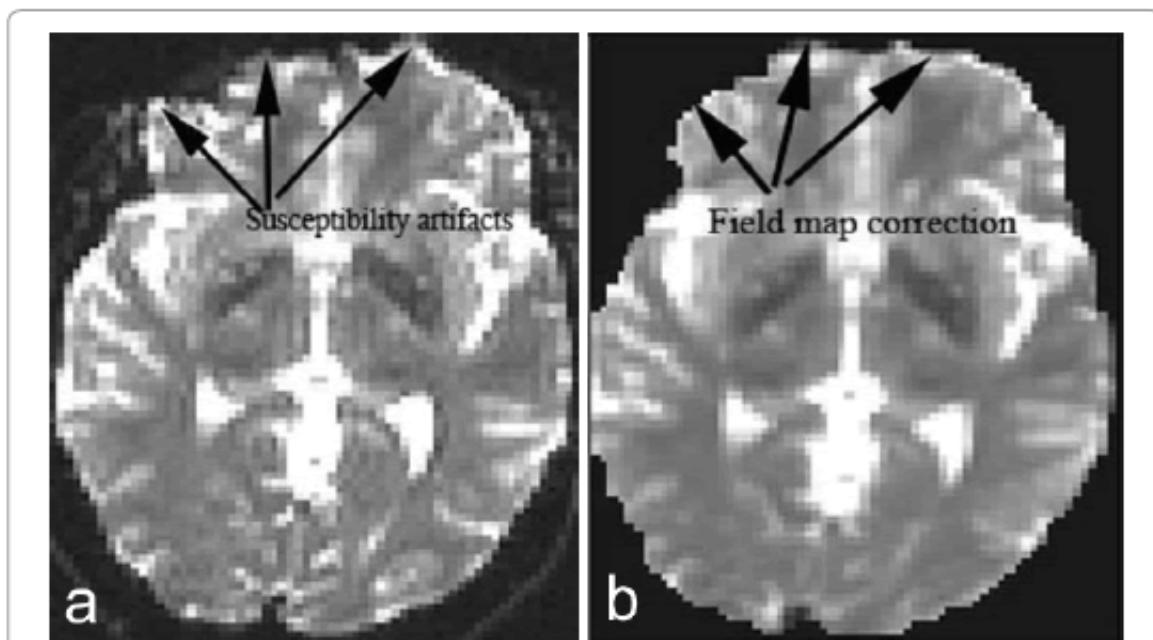


Figure 3: a) Shows susceptibility artifact in DTI data collected from one of our amyotrophic lateral sclerosis (ALS) patients, b) arrows indicate mislocalized voxels that were restored using field map correction procedure.

```
In [ ]: # run topup
bash_command('topup --imain=sub-05_run-01-02-03_b0 --datain=acq.txt --
config=b02b0.cnf --out=sub-05_run-01-02-03_topup --iout=sub-05_run-01-
02-03_b0_topup')
```

```
In [10]: bash_command('fsleyes sub-05_run-01-02-03_b0.nii.gz sub-05_run-01-02-0
3_b0_topup.nii.gz')
```

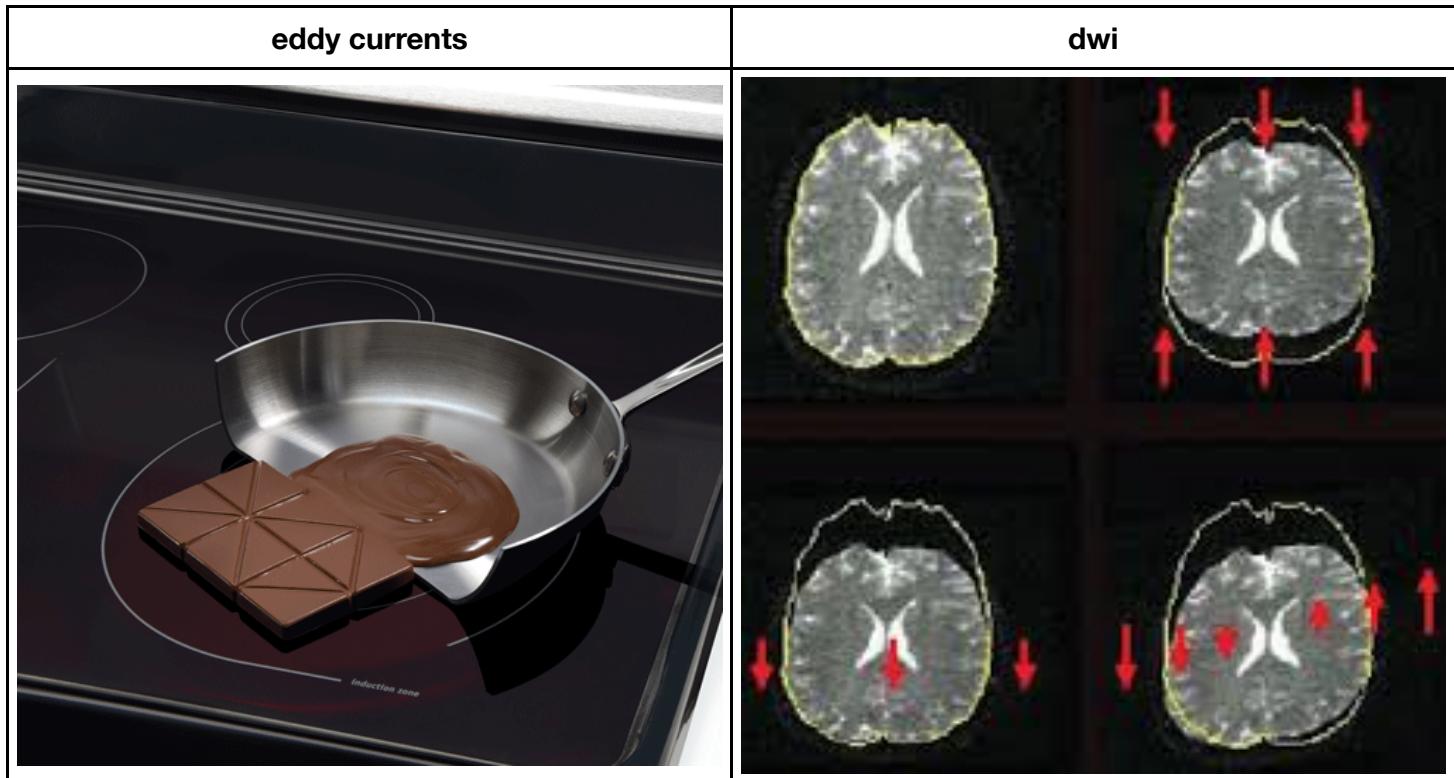
```
Out[10]: b''
```

```
In [12]: # create mask
bash_command('fslmaths sub-05_run-01-02-03_b0_topup -Tmean sub-05_run-
01-02-03_b0_topup')
bash_command('bet sub-05_run-01-02-03_b0_topup sub-05_run-01-02-03_b0_
brain -m')
bash_command('fsleyes sub-05_run-01-02-03_b0_brain.nii.gz sub-05_run-0
1-02-03_b0_brain_mask.nii.gz sub-05_run-01-02-03_b0_topup.nii.gz')
```

```
Out[12]: b''
```

```
In [ ]: # instruct fsl which slices belong to which acquisition parameters
bash_command('indx="")'
bash_command('for ((i=1; i<=65; i+=1)); do indx="$indx 1"; done')
bash_command('echo $indx > index.txt')
```

FSL - eddy



```
# run eddy
```

```
eddy --imain=sub-05_run-01_dwi --mask=sub-05_run-01-02-03_b0_brain_mask\
--acqp=acq.txt --index=index.txt --bvecs=sub-05_run-01_dwi.bvec\
--bvals=sub-05_run-01_dwi.bval --topup=sub-05_run-01-02-03_topup\
--out=sub-05_run-01_eddycorrected
```

eddy_cuda:

- basic eddy takes quite a long time to run
- can be run on a GPU, but uses CUDA version 7.5

```
In [11]: bash_command('fsleyes sub-05_run-01_eddycorrected.nii.gz sub-05_run-01-02-03_b0_brain.nii.gz')
```

```
Out[11]: b''
```

Now that our Diffusion-weighted images are cleaned,

Let's track those fibers!!

Now that our Diffusion-weighted images are cleaned,

Let's track those fibers!!

There are many different methods, we'll stick with our trusted Freesurfer Software....

DWI Image Processing - Tractography

TRActs Constrained by UnderLying Anatomy (TRACULA) by Freesurfer

Tracula – Free Surfer Wiki

surfer.nmr.mgh.harvard.edu/fswiki/Tracula

Most Visited Getting Started Latest Headlines Apple Yahoo! Google Maps YouTube Wikipedia

Login

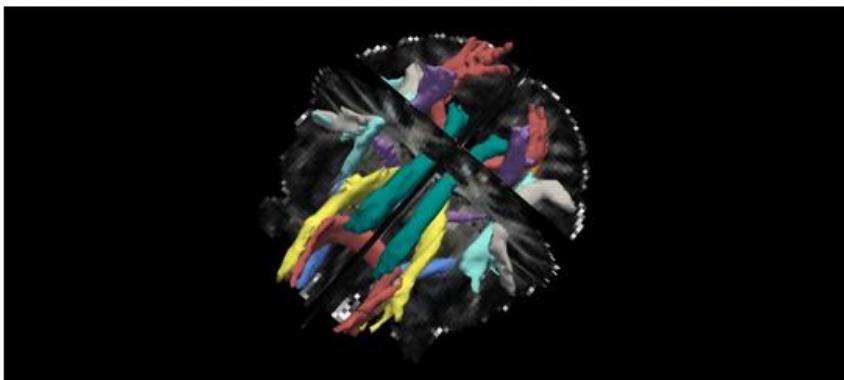
Tracula

FreeSurferWiki RecentChanges FindPage HelpContents **Tracula**

Immutable Page Discussion Info Attachments More Actions:

TRACULA: TRActs Constrained by UnderLying Anatomy

TRACULA is a tool for automatic reconstruction of a set of major white-matter pathways from diffusion-weighted MR images. It uses global probabilistic tractography with all structures of each pathway are derived from an atlas and combined with the FreeSurfer cortical parcellation and subcortical segmentation of the subject that is being analyzed for user interaction, e.g., to draw ROIs manually or to set thresholds on path angle and length, and thus automates the application of tractography to large datasets.



Updates

TRACULA in FreeSurfer 5.3:

First, TRACULA runs some quality assessment of our images e.g., although head motion can be corrected in the anatomical images via registration, this cannot be corrected in dwi images..

```
In [14]: # So let's check the output from the head motion assessment:
bash_command("cat dwi_motion.txt")
```

```
Out[14]: b'AvgTranslation AvgRotation PercentBadSlices AvgDropoutScore\n0.566
605 0.0028432 0 1\n'
```

"Our subject was pretty well-behaved!

These motion measures can be used to ensure that groups of subjects are matched with respect to head motion, or to introduce head motion as a nuisance regressor in group analyses" (Tracula Wiki)

```
In [12]: # Let's look at the tensorfit results:
```

```
# FA in native space:  
bash_command('freeview ./dtifit_FA.nii.gz') # use NIH color scheme  
# Follow the paths of high FA!
```

```
Out[12]: b''
```

```
In [14]: # FA in MNI space
```

```
bash_command('freeview -v ./MNI152_T1_1mm_brain.nii.gz ./dtifit_FA.bbr  
.nii.gz') # use NIH color scheme  
# Follow the paths of high FA!  
# Adjust the opacity of each image to see the quality of registration
```

```
Out[14]: b''
```

The registration on this image actually looks pretty bad! :(

There are many more options:

e.g., Comparison of nine tractography algorithms for detecting abnormal structural brain networks in Alzheimer's disease (Zhan et al)

<ul style="list-style-type: none"> + Tensor-based deterministic approaches: <ul style="list-style-type: none"> + Fiber Assignment by Continuous Tracking (FACT) + the 2nd-order Runge-Kutta (RK2) + Interpolated streamline (SL) + tensorline (TL) + Orientation Distribution Function (ODF)-based deterministic approach <ul style="list-style-type: none"> + ODF-RK2 + ODF-based probabilistic approach <ul style="list-style-type: none"> + Hugh voting 	<p style="text-align: center;">Tractography</p>  <ul style="list-style-type: none"> • Use local diffusion orientation at each voxel to determine pathway between distant brain regions • Local orientation comes from diffusion model fit (tensor, ball-and-stick, etc.) • Deterministic vs. probabilistic tractography: <ul style="list-style-type: none"> – Deterministic assumes a single orientation at each voxel – Probabilistic assumes a distribution of orientations • Local vs. global tractography: <ul style="list-style-type: none"> – Local fits the pathway to the data one step at a time – Global fits the entire pathway at once <p style="text-align: center;">04/04/12 TRACULA 3/21</p>

Next Steps:

Re: reproducing the results from Multi-View Graph Convolutional Network and Its Applications on Neuroimage Analysis for Parkinson's Disease (Zhang et al)

- After preprocessing, construct the graphs
- Brain Geometry Graph (BGG):
 - Constructed from MRI images
 - Each vertex = the center voxel of each region
 - The edges are formed as a KNN graph
- Brain Connectivity Graph (BCG):
 - Constructed from the DTI tractography outputs
 - the ROI's are used as vertices
 - The edges are determined and weighted by the fiber tracts

My contributions:

- add another modality: fMRI
- pretrain the network using RBM
 - re: # Restricted Boltzmann machines for neuroimaging: An application in identifying intrinsic networks (Hjelm et al)
- explore differences related to the disease progression re: subtyping

