**Diffusion Image Analysis 101**

Diffusion imaging is concerned with the movement of water molecules in the brain. You can read more about the details of Diffusion Tensor Image (DTI) [here](https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2041910/) and Diffusion Kurtosis Imaging (DKI) [here](http://mriquestions.com/diffusion-kurtosis.html). (Note: though you may work with other types of images such as structural T1 images, the primary MRI data analyzed in this lab is diffusion data.)

**There are numerous pieces of software that are essential and/or beneficial for image analysis**:

* [Filezilla](https://filezilla-project.org/) – An FTP software used to access CBIHome to obtain raw image data
* [Horos](https://horosproject.org/) – A program that allows for viewing dicom files
* [dicomsort](https://github.com/TheJaeger/dicomSort) – An in-house Matlab plugin for sorting dicoms into sequence folders
* [MRIcroGL](https://www.nitrc.org/projects/mricrogl) – An image viewing package that includes the necessary tool dcm2niix
* [PyDesigner](https://github.com/m-ama/PyDesigner) – An in-house software that is used to remove image artifacts from raw diffusion data and calculate parametric maps
* [FSL](https://fsl.fmrib.ox.ac.uk/fsl/fslwiki/FSL) – A toolbox that allow for a variety of image manipulations\*
* [MRtrix3](https://www.mrtrix.org/) - A toolbox that allow for a variety of image manipulations\*
* [FSLeyes](https://fsl.fmrib.ox.ac.uk/fsl/fslwiki/FSLeyes) – An image viewer included as a part of FSL; pointed out specifically due to its utility and uniqueness among other FSL tools
* [MRIcron](https://www.nitrc.org/projects/mricron) – Another image viewer with a slightly different interface than FSLeyes; optional but useful
* [Matlab](https://matlabacademy.mathworks.com/) – A programming environment used for certain types of analysis
* [SPM](https://www.fil.ion.ucl.ac.uk/spm/) – A Matlab-compatible software package that allows for a variety of types of image analysis

\*Note: FSL and MRTrix3 are very similar; however, they are both ubiquitous in this field and are often required so that other pieces of software will function. Despite their similarities, it is a very good idea to have both.

**A typical project lifecycle will go as follows**:

1. **Scan Data Acquisition** - You may need to access CBIHome in order to retrieve raw data from the main MRI server. I recommend Filezilla as it is straightforward and user-friendly.
   1. To log into the server, you will need the address, your NetID, and PW.
2. **File Organization** - It is recommended to begin every project by organizing your data.
   1. A general recommended folder/file organization structure can be seen in the folder **/01\_Basics/File\_Organization\_Example/**
   2. Keep a record of your subjects, any scan/study/demographic information you may need, and any processing steps you’ve completed in a spreadsheet or other organization tool of your choice.
      1. An example of this documentation can be seen here:
      2. **/01\_Basics/File\_Organization\_Example/04\_Summary/ProcessingLog.xlsx**
3. **Raw Data QC** - QCing raw data is a crucial step before doing any image manipulation. See Image\_QC.docx for more details.
4. **Dicom sorting and dicom to nifti conversion** 
   1. dicomsort is a tool that will sort your dicoms into more manageable folders based on MRI sequence.
      1. It is a Matlab plugin, so you will call it through Matlab by simply using the following command:
      2. dicomSort(/path/to/folder);
   2. dcm2niix is a tool that comes packaged with MRIcroGL. While it is not the only way to convert dicoms to niftis, it is an effective, simple, and user-friendly way of doing so.
      1. Once dcm2niix is set up on your system, it should only be necessary to run the following command in your terminal:
      2. dcm2niix -f %p [path/to/your/data]
5. **Preprocessing** - Preprocessing raw data is necessary to remove image articles. PyDesigner includes both artifact removal and tensor calculation to derive parametric maps. Preprocessed data should also be QC’d to ensure that preprocessing was completed as expected. See **Image\_QC.docx** for more details.
   1. Using PyDesigner is quite simple, though it does have some requirements and best practices:
      1. It is helpful to have your data organized consistently for each subject.
      2. PyDesigner requires a corresponding .json, .bvec, and .bval file for each .nii file you give it. These are generally created by dcm2niix.
      3. A typical file setup will look like this:
      4. A close up of text on a black background

         Description automatically generated
      5. Where the “nifti” folder contains all of your raw data and the “pydesigner” folder is the output folder where your processed data will go.
      6. A typical command using the folder explain above will look like this:
      7. pydesigner --denoise --degibbs --mask -w --force /desktop/user/PyDesigner-Example/Subj1/nifti/DKI.nii, /desktop/user/PyDesigner-Example/Subj1/nifti/B0.nii -o /desktop/user/PyDesigner-Example/Subj1/pydesigner
      8. (Notes: In this example we have a DKI sequence and an additional B0 sequence, which is common. Both sequences can be given to PyDesigner by just putting a comma between the file paths as you can see in the example command. Also note the “-o” command which specifies where files will be output. Other commands such as “--denoise” and “--degibbs” are described in the PyDesigner documentation linked at the top of this document.)
6. After preprocessing, next steps will depends on the PI’s desired analyses.
   1. Further reading on analysis techniques can be found throughout the rest of this documentation collective
7. Once all analyses are complete, it is a good idea to organize your project folder and store/backup/archive however your PI would like you to do so.
   1. See **03\_Post\_Project\_Data\_Storage.docx** for more info.