DTIPrep

Building

1. Create an account with NITRC: http://www.nitrc.org/account/register.php
2. Get the source code

In a terminal (Ex. bash):

```
host$ mkdir ${source_dir}
host$ cd ${source_dir}
host$ svn checkout https://www.nitrc.org/svn/dtiprep/trunk DTIPrep
```

3. Run CMake

```
host$ mkdir ${build_dir} # (e.g. "DTIPrep-build")
host$ cd ${build_dir}
host$ ccmake ${source_dir}/DTIPrep
```

```
                                 Page 1 of 1
    BUILD_STYLE_UTILS              OFF
    BUILD_TESTING                  ON
    CMAKE_BUILD_TYPE               Release
    CMAKE_INSTALL_PREFIX           /usr/local
    CMAKE_OSX_ARCHITECTURES
    CMAKE_OSX_DEPLOYMENT_TARGET
    CMAKE_OSX_SYSROOT              /Applications/Xcode.app/Contents/Developer/Platfor
    EXTERNAL_PROJECT_BUILD_TYPE    Release
    FORCE_EXTERNAL_BUILDS          OFF
    ITK_VERSION_MAJOR              4
    QT_QMAKE_EXECUTABLE            /usr/bin/qmake
    SuperBuild_DTIPrepTools_BUILD_ ON
    SuperBuild_DTIPrepTools_USE_GI ON
    USE_DTIPrep                    ON
    USE_DTIProcess                 OFF
    USE_DTIReg                     OFF
    USE_DTI_Tract_Stat             OFF
    USE_FVLight                    OFF
    USE_NIRALUtilities             OFF
    USE_SYSTEM_DCMTK               OFF
    USE_SYSTEM_ITK                 OFF
    USE_SYSTEM_SlicerExecutionMode OFF
    USE_SYSTEM_VTK                 OFF
    VTK_GIT_TAG                    v5.10.0
    VTK_REPOSITORY                 git://vtk.org/VTK.git


 BUILD_STYLE_UTILS: Build uncrustify, cppcheck, & KWStyle
 Press [enter] to edit option                       CMake Version 2.8.9
 Press [c] to configure
 Press [h] for help              Press [q] to quit without generating
 Press [t] to toggle advanced mode (Currently Off)
```

Figure 1: Mac OSX example

4. Type 'c' to configure. When successful, you will see a new option to generate the makefiles, 'g'. Enter 'g' and CMake will exit.
5. Run Make. To run Make multi-threaded, use the "-j" flag:

```
host$ make #optinal: "-j ${NUMBER_OF_CORES}"
```
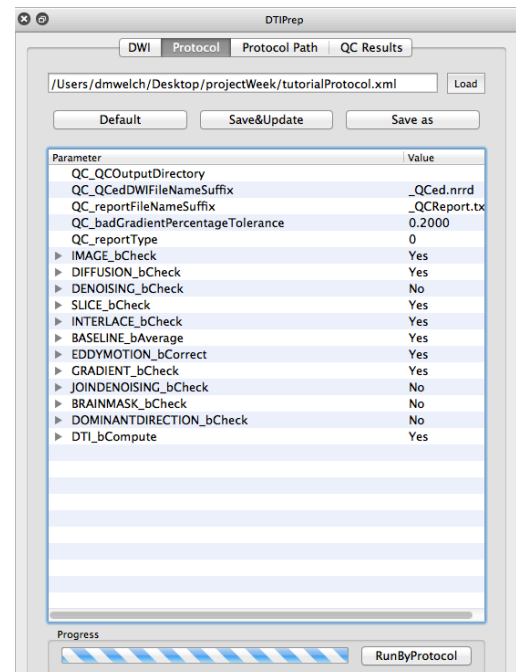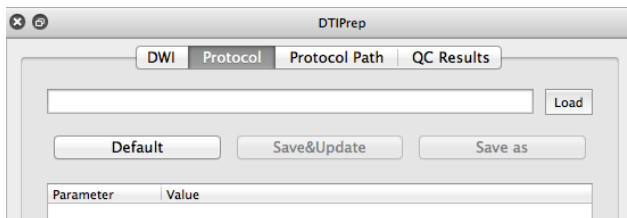
To update DTIPrep:

```
host$ cd ${source_dir}/DTIPrep
host$ svn update
host$ cd ${build_dir}
host$ make # -j ${NUMBER_OF_CORES}
```

Create a protocol file

Protocol files are XML files that set the attributes for DTIPrep and control the criteria with which DTIPrep fails during processing, among other things. Since DTIPrep is highly customizable, creating a custom protocol file is *highly* recommended. Protocol files are created using the graphical user interface (GUI), i.e. "window". To use the DTIPrep GUI, run the executable file for your particular operating system:

Operating System	GUI executable
Windows	${build_dir}/bin/DTIPrep
MacOS	${build_dir}/bin/DTIPrep.app/Contents/MacOS/DTIPrep
Linux	${build_dir}/bin/DTIPrep

1. Navigate to the executable file and double-click OR call it on the command line
2. Click on the "Open NRRD" button in the top left corner and load the file "DTI_30_1_Sept_2010.nhdr"
3. Once the images load, click on the "Protocol" tab. Load the default protocol by clicking the "Default" button.
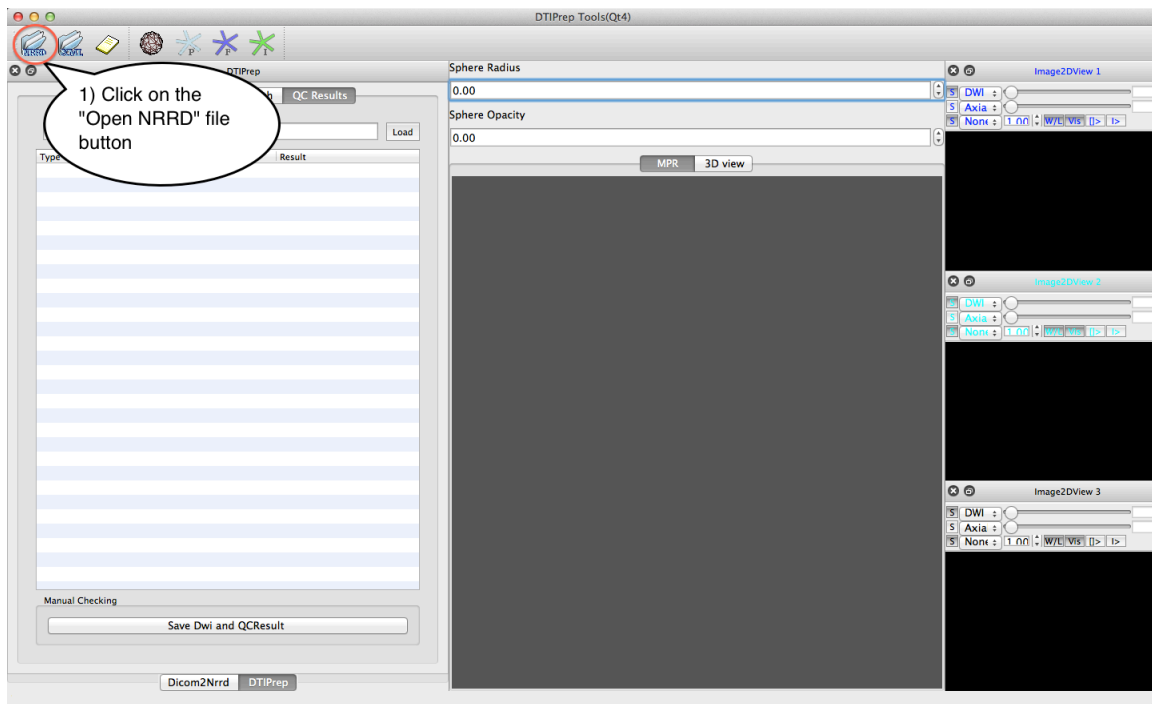
4. Change the default values for the test dataset:

Protocol Heading	Protocol Subheading	Value
IMAGE_bCheck	IMAGE_bCrop	No
	IMAGE_croppedDWIFileNameSuffix	(blank)
	IMAGE_bQuitOnCheckSpacingFailure	Yes
	IMAGE_bQuitOnCheckSizeFailure	No
DIFFUSION_bCheck	DIFFUSION_diffusionReplacedDWIFileNameSuffix	(blank)
	DIFFUSION_bQuitOnCheckFailure	Yes
SLICE_bCheck	SLICE_bSubregionalCheck	Yes
	SLICE_bQuitOnCheckFailure	Yes
INTERLACE_bCheck	INTERLACE_bQuitOnCheckFailure	Yes
GRADIENT_bCheck	GRADIENT_bQuitOnCheckFailure	Yes

5. Click "Save as" and name the file "tutorialProtocol.xml"
6. Close DTIPrep

Usage: GUI interaction

1. Navigate to the executable file and double-click OR call it on the command line
2. Click on the "Open NRRD" button in the top left corner and load the file "DTI_30__1_Sept_2010.nhdr"

3. Load the protocol file "tutorialProtocol.xml" and click "RunByProtocol"
   NOTE: Processing takes ~30 mins

Usage: Command line

1. You will need the full path to three files:
   a. input DWI file (${DWI})
   b. input protocol file (e.g. "tutorialProtocol.xml")
   c. output directory
2. Run the executable from the command line with the correct flags:

```
host$ ${build_dir}/bin/${executable_path} DTIPrep \
  --DWINrrdFile ${test_data_dir}/DTI_30__1_Sept_2010.nhdr \
  --xmlProtocol ${test_data_dir}/tutorialProtocol.xml \
  --check \
  --outputFolder ${output_dir}
```

3. Once complete, you will have three files in ${output_dir}:
   a. DTI_30__1_Sept_2010_QCed.nhdr
   b. DTI_30__1_Sept_2010_QCReport.txt
   c. DTI_30__1_Sept_2010_ XMLQCResult.xml

Choosing Protocol Parameters

Protocol parameters will depend on the quality of your data, the file organization you desire, the level of checking that you want to perform during the processing, etc., so experimentation is recommended. In this tutorial we will demonstrate a method to optimize the DTIPrep protocol using Slicer to explore the output data.

1. After running the dataset "DTI_30__1_Sept_2010.nhdr", load the original file and the cleaned file in Slicer:
2. Select the "Volumes" module
3. Set the **Active Volume** to "DTI_30__1_Sept_2010"
4. Set the **DWI Component** to 1. Examine the volume in the Slice Views. You will notice that there are intensity artifacts in the coronal plane.
   > FYI: This data isn't particularly noisy for DWI, but if your data has too much noise you may need to discard the whole set if DTIPrep can't distinguish artifacts from the background noise...
5. Now load the cleaned data. Set the **Active Volume** to "DTI_30__1_Sept_2010_QCed"
6. As you click through the **DTI Component** values and the slice planes, you *should* notice that the data has less noise and no artifacts.
7. If artifacts remain, go back to protocol file in DTIPrep and change the value of SLICE_correlationDeviationThresholdGradient from 3.5000 -> 3.0000. Save this protocol file and rerun.
8. Examine the new cleaned file. If it is ***still*** noisy, you can further lower the SLICE_correlationDeviationThresholdGradient to 2.5000. We have empirically found that any value lower than 2.5000 results in data is highly unreliable.

a. The reason is that DTIPrep puts the degree noise (measured in correlation between corresponding slices) of DWI data into a Gaussian distribution. The parameter SLICE_correlationDeviationThresholgradient sets how many standard deviations about the average correlation are considered acceptable. If a DWI file is very noisy overall, then data whose correlation values that are close to the average correlation value will still be noisy (even within 1 standard deviation!) In this case, you might as well discard the whole DWI file.