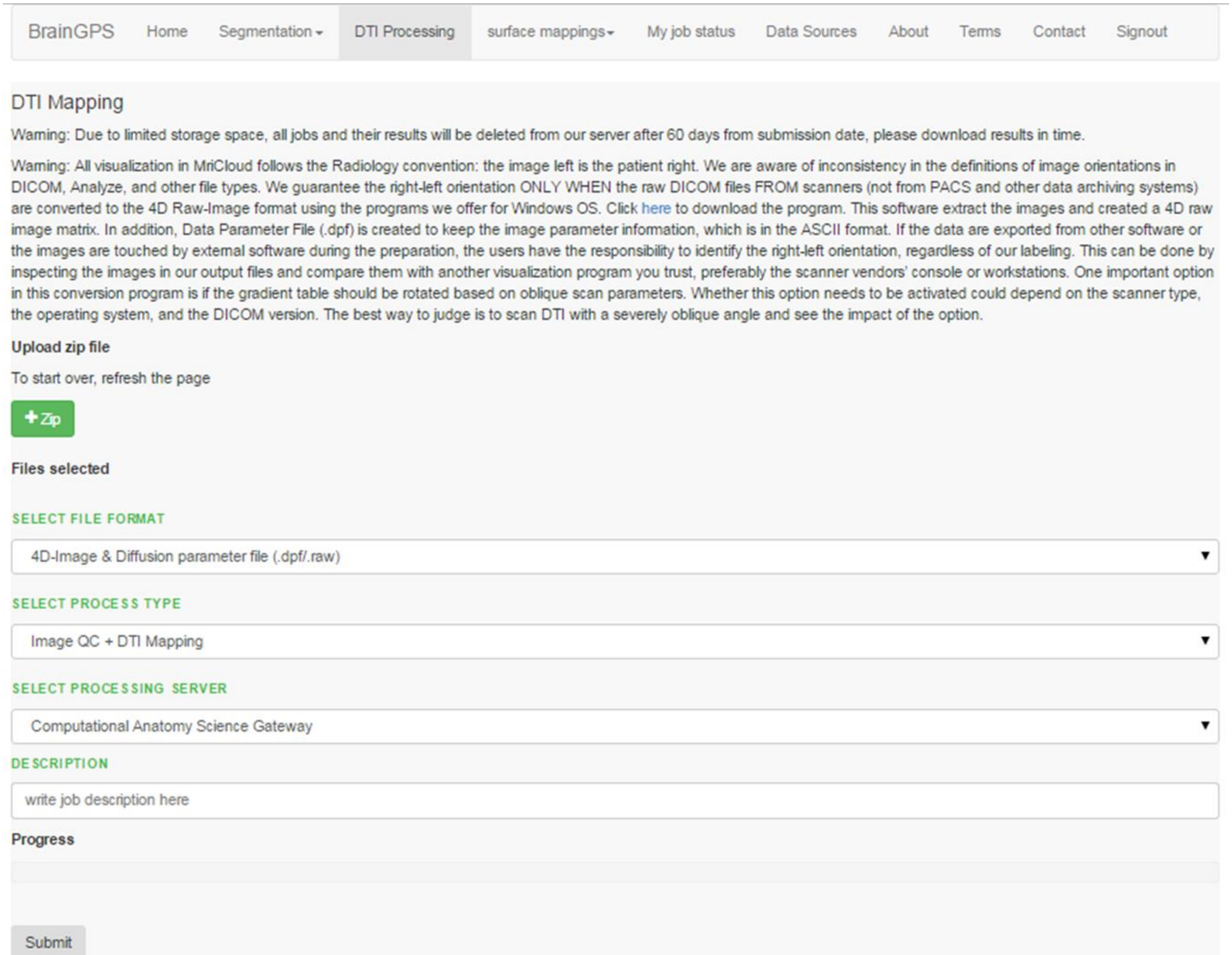
MriCloud DTI Processing Pipeline

1: Data Upload

1-1: Web interface

DTI processing can be initiated by choosing “DTI Processing” in the top menu bar.

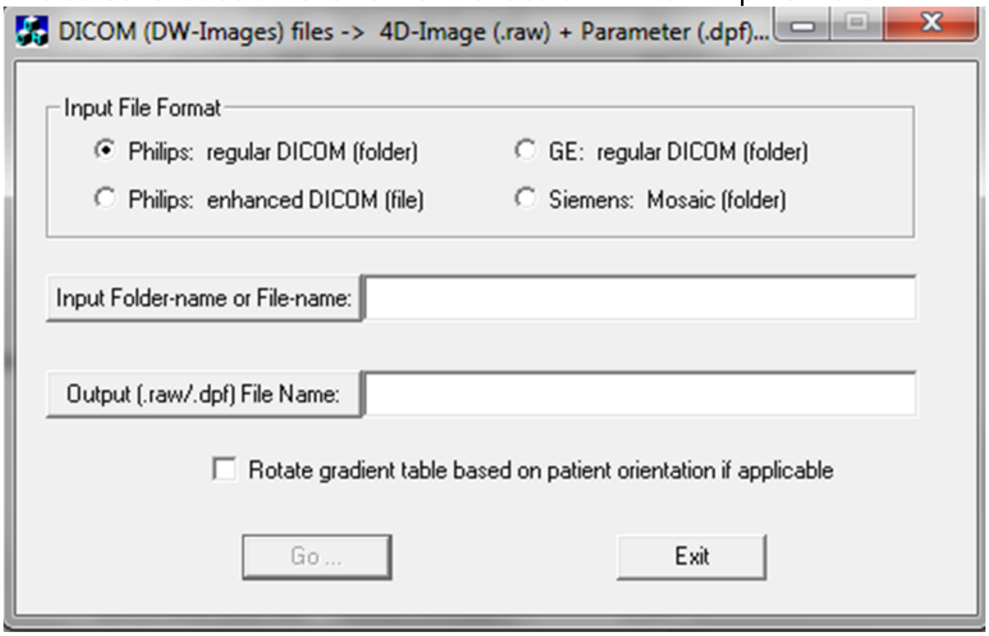


To avoid any HIPPA issues, data need to be first converted to a raw data format with a parameter file called Data Parameter File (.dpf). For this conversion, a local executable file can be downloaded from this web page by clicking “here” in the top description section. This executable is called “DwiDcm2DpfRaw.exe.” This conversion eliminates all personal IDs from the file. Also, this conversion works only for DICOM files directly from scanners. Many PACSs and other third-party software modify the DICOM format and thus they are not supported.

In this page, you can see that there are several option items for data format, data processing, and computation servers but at this moment, no selections are provided (only one option can be selected in each pull-down menu).

1-2: Data preparation in local computers

The screenshot below shows the interface of DwiDcm2DpfRaw.exe.



We assume that all DICOM files of raw diffusion-weighted images (DWIs) are stored in one directory. Only one of the DICOM files needs to be specified and all files within the same directory are automatically read. The output files are a raw image file (*.raw) and a parameter file (*.dpf).

If there are more than two repeated DTI data, stored in separate directories, they need to be combined. In any case, the data in each directory need to be converted separately. In an example below, two sets of DICOM files stored in two separate directories were converted to 601.raw/601.dpf and 701.raw/701.dpf files. To combine them, one of the dpf files needs to be manually modified. For example, the contents of the 601.dpf are (this is automatically generated by DwiDcm2DpfRaw.exe);

```
Begin
      ImageWidth: 256
      ImageHeight: 256
      ImageSlices: 70

      ProcSliceStart: 0
      ProcSliceEnd: 69

      FieldOfView(X): 212.0000
      FieldOfView(Y): 212.0000
      SliceThickness: 2.2000

      B-Value: 700

      SwapBytes: No
      ImgNoiseLevel: 10

      Begin_Of_Gradient_Table
            0: 0.000000, 0.000000, 0.000000
            1: -0.978747, 0.198562, -0.051255
            2: -0.201929, -0.976746, 0.072049
            3: -0.035757, 0.080867, 0.996083
            4: 0.100129, 0.023255, -0.994696
            5: -0.096230, 0.152618, -0.983535
```

```
        6: -0.160238, 0.127393, -0.978866
        7: -0.331720, -0.027696, -0.942915
        8: -0.229307, -0.401323, -0.886819
        9: -0.034654, -0.880897, -0.472052
        10: 0.179059, -0.363184, -0.914409
        11: 0.488783, -0.299203, -0.819537
        12: 0.168451, 0.547780, -0.819484
        13: 0.781784, -0.056217, -0.621032
        14: 0.700209, -0.509669, -0.500005
        15: 0.002314, -0.966216, -0.257892
        16: 0.771895, 0.229139, -0.592951
        17: -0.840693, -0.339238, -0.422081
        18: -0.870490, 0.254967, -0.421094
        19: -0.287928, 0.807178, -0.515388
        20: -0.809282, 0.325019, -0.489338
        21: -0.812719, -0.565379, -0.140791
        22: -0.365158, -0.741186, -0.563326
        23: -0.148155, -0.958531, 0.243470
        24: 0.360617, -0.932424, -0.025302
        25: 0.996607, -0.082461, 0.000004
        26: 0.962654, -0.270243, 0.018448
        27: 0.760943, -0.648408, -0.021841
        28: 0.045685, -0.940303, 0.337094
        29: -0.314395, -0.949154, 0.018486
        30: -0.515431, -0.720788, 0.463473
        31: -0.691326, -0.706981, 0.149337
        32: -0.984777, 0.139608, -0.104010
        33: 100.000000, 100.000000, 100.000000
    End_Of_Gradient_Table

    InputImgFile:
C:\Data\1001_140116_11_3_2014_17_39_19\1001_140116\SCANS\Image4D_601.raw

    InputImgOrder: Gradient_By_Gradient

    OutFileTensor: Tensor.dat
    OutFileRefB0: RefB0.dat
    OutFileMeanDwi: MeanDwi.dat

    OutFileEigenVal0: EgVal0.dat
    OutFileEigenVal1: EgVal1.dat
    OutFileEigenVal2: EgVal2.dat

    OutFileEigenVec0: EgVec0.dat
    OutFileEigenVec1: EgVec1.dat
    OutFileEigenVec2: EgVec2.dat

    OutFileFA: FaMap.dat
    OutFileRA: RaMap.dat
    OutFileVR: VrMap.dat

    OutFileColorMap0: ColorMap0.dat
    OutFileColorMap1: ColorMap1.dat
    OutFileColorMap2: ColorMap2.dat
End
```

To combine the 601.raw and 701.raw, a new line (indicated by a red color) has to be added;

```
Begin
	ImageWidth: 256
	ImageHeight: 256
	ImageSlices: 70

	ProcSliceStart: 0
	ProcSliceEnd: 69

	FieldOfView(X): 212.0000
	FieldOfView(Y): 212.0000
	SliceThickness: 2.2000

	B-Value: 700

	SwapBytes: No
	ImgNoiseLevel: 10

	Begin_Of_Gradient_Table
		0: 0.000000, 0.000000, 0.000000
		1: -0.978747, 0.198562, -0.051255
		2: -0.201929, -0.976746, 0.072049
		3: -0.035757, 0.080867, 0.996083
		4: 0.100129, 0.023255, -0.994696
		5: -0.096230, 0.152618, -0.983535
		6: -0.160238, 0.127393, -0.978866
		7: -0.331720, -0.027696, -0.942915
		8: -0.229307, -0.401323, -0.886819
		9: -0.034654, -0.880897, -0.472052
		10: 0.179059, -0.363184, -0.914409
		11: 0.488783, -0.299203, -0.819537
		12: 0.168451, 0.547780, -0.819484
		13: 0.781784, -0.056217, -0.621032
		14: 0.700209, -0.509669, -0.500005
		15: 0.002314, -0.966216, -0.257892
		16: 0.771895, 0.229139, -0.592951
		17: -0.840693, -0.339238, -0.422081
		18: -0.870490, 0.254967, -0.421094
		19: -0.287928, 0.807178, -0.515388
		20: -0.809282, 0.325019, -0.489338
		21: -0.812719, -0.565379, -0.140791
		22: -0.365158, -0.741186, -0.563326
		23: -0.148155, -0.958531, 0.243470
		24: 0.360617, -0.932424, -0.025302
		25: 0.996607, -0.082461, 0.000004
		26: 0.962654, -0.270243, 0.018448
		27: 0.760943, -0.648408, -0.021841
		28: 0.045685, -0.940303, 0.337094
		29: -0.314395, -0.949154, 0.018486
		30: -0.515431, -0.720788, 0.463473
		31: -0.691326, -0.706981, 0.149337
		32: -0.984777, 0.139608, -0.104010
		33: 100.000000, 100.000000, 100.000000
	End_Of_Gradient_Table
```

```
      InputImgFile:
C:\Data\1001_140116_11_3_2014_17_39_19\1001_140116\SCANS\Image4D_601.raw
C:\Data\1001_140116_11_3_2014_17_39_19\1001_140116\SCANS\Image4D_701.raw

      InputImgOrder: Gradient_By_Gradient

      OutFileTensor: Tensor.dat
      OutFileRefB0: RefB0.dat
      OutFileMeanDwi: MeanDwi.dat

      OutFileEigenVal0: EgVal0.dat
      OutFileEigenVal1: EgVal1.dat
      OutFileEigenVal2: EgVal2.dat

      OutFileEigenVec0: EgVec0.dat
      OutFileEigenVec1: EgVec1.dat
      OutFileEigenVec2: EgVec2.dat

      OutFileFA: FaMap.dat
      OutFileRA: RaMap.dat
      OutFileVR: VrMap.dat

      OutFileColorMap0: ColorMap0.dat
      OutFileColorMap1: ColorMap1.dat
      OutFileColorMap2: ColorMap2.dat
End
```

After this modification, 601.raw, 701.raw, and 601.dpf are zipped and ready for data upload.

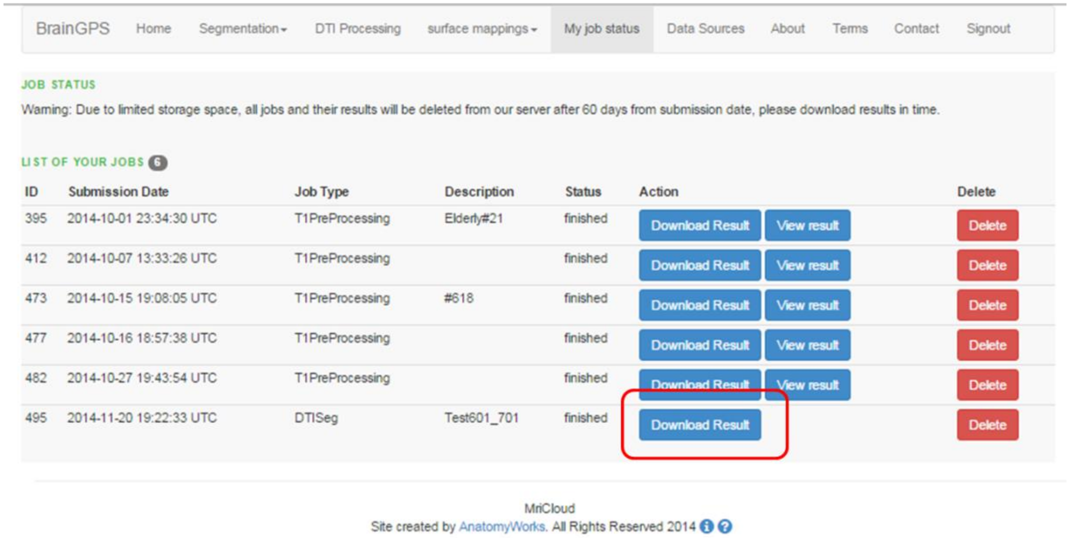
One important option is “Rotate gradient table.” It is very difficult to provide a definite answer about whether this option needs to be checked or not because it varies depending on the manufacturers, file formats, and versions. Our experience with Siemens and Philips so far suggested that they need to be checked. We advise to perform a test in which severe oblique angle is applied.

1-3: Data upload

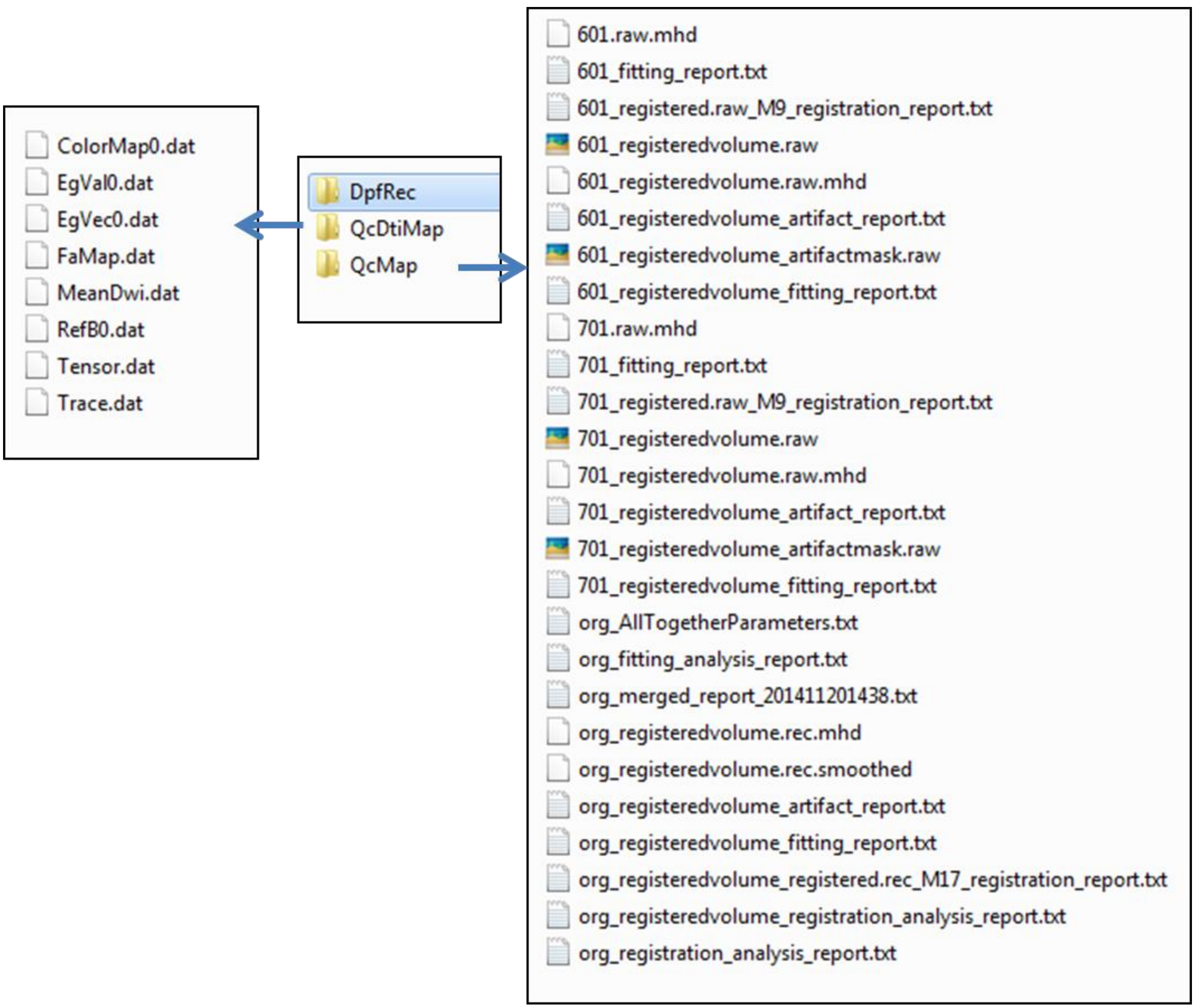
Data can be uploaded through “+Zip” button.

1-4: Data download

Once the job is completed, the results are ready for download through “MyJobStatus.”



The following figure shows the structure of downloaded files;



There are three directories;

- DpfRec: This contains the submitted data.

- QcDtiMap: This contains the calculated DTI-derived images.
- QcMap: This contains intermediate files and quality control reports.

From “Tensor.dat” file in QcDtiMap, various tensor-derived images such as FA, color, and vector maps can be created in local computers, while most widely used images are already included in QcDtiMap. To generate other types of measures such as radial and axial diffusivity, “Tensor.dat” can be read by DtiStudio and various images are generated and saved.

1-5: QC reports

QcMap contains “*_registeredvolume.raw”, which are images after registration. There are several important features in the QC pipeline.

Registration and registration error reports: The registration is based on 9-mode affine transformation, in which both 6-mode rigid motion and 3-mode gradient-dependent Eddy-current distortion are simultaneously fitted and estimated using the normalized mutual information as the cost function. The detail is provided in [1]. Based on this registration, motion (translation and rotation) and Eddy current (X, Y, Z, and B0 components) are estimated and reported. The files that report these errors are “*_registration_analysis_report.txt” In this processing, the gradient vector for each diffusion-weighted image is also rotated with the subject motion.

Voxel rejection: Voxels with outlier intensities during tensor fitting are automatically rejected. “*_registedvolume_artifactmask.raw” reports locations of all voxels that are rejected. From this file, a probabilistic map of voxel-by-voxel rejection rates can be calculated. Statistic summaries such as the slice-by-slice number of rejected voxels are reported in “*_registeredvolume_artifact_report”. In addition, if the number of rejected voxels reaches 5% of the size of the slice, the entire slice is rejected. The detail of outlier rejection is provided in [2].

Fitting quality report: The report of motion such as translation and rotation is accurate only when the post-processing registration works. In reality, when images are suffered from excessive motion, the image quality corrupts and the registration quality would be poor. In this case, the registration report is not reliable to judge the data quality. Therefore, we included report of tensor fitting quality before and after image registration. The table summary is provided in xxx_registeredvolume_fitting_report.txt and xxx_registeredvolume_fitting_report.txt.

1-6: Off-line population QC reports

As described above, each data carries a large amount of QC reports. In reality, the reported values such as motion (translation, rotation), eddy current, the number of rejected voxels, and fitting quality, are interpretable only after population data are acquired, from which outliers can be detected. We provide a MatLab code that read the QC reports from all subjects within a study and perform population-based analysis and outlier detection. This code is still being developed and your feedback would be appreciated.

[1] Penny W, et al., Statistical Parametric Mapping: The Analysis of Functional Brain Images, 1st Edition, Elsevier 2006.
[2] Li Y., et al., Image Corruption Detection in Diffusion Tensor Imaging for Post-Processing and Real-Time Monitoring PLOS ONE 2013; 8(10): e49764.