

Tutorial: Prepare Date for DTI-MRtrix-pipeline

This tutorial shows how to prepare the data for the DTI-MRtrix-pipeline

CONTENTS

PART-1: BASICS+REGISTRATION

- 1) Prepare Study
 - GO TO STUDY-FOLDER
 - UPDATE ANT-TOOLBOX
 - CREATE A PROJECT-FILE:
 - Open ANT-gui & CREATE AN ANTx-PROJECT-FILE:
- 2) IMPORT BRUKER-DATA
- 3) VISUALIZE FILES AND FOLDERS
- 4) SELECTION OF ANIMALS
- 5) RENAME FILES
- 6) Determine Orientation-Type
- 7) REGISTER “t2.nii” TO TEMPLATE SPACE (STANDARD-SPACE)
- 8) Examine the Processing Report

PART-2: DTI-preprocessing/prepare data for DTI-MRtrix-pipeline

- 9) A special DTI-atlas is needed
- 10) DTI-preprocessing
 - 10.1) IMPORT B-tables [get b-table(s)]
 - 10.2) Specify DTI/DWI-files [get DTIfile(s)]
 - 10.3) specify the DTI-atlas [getDTItemplate]
 - 10.4) Check file-assignment and order of b-tables and DTI-files
 - 10.5) Run the DTIprep-tasks
 - 11) CHECK registration with 1st DWI-file

PART-1: BASICS+REGISTRATION

1) Prepare Study

The folder "F:\data5\multishell_test" contains two folders:

- 1) "raw"-folder containing two Bruker-raw data sets
- 2) "atl_auditsys_08dec20_v1" containing a special DTI-atlas (more info later)

GO TO STUDY-FOLDER

In Matlab set current working directory to "F:\data5\multishell_test" or create an empty study folder (here "multishell_test"). The study-folder is the folder where the registration of several animals of a study is performed.

```
cd F:\data5\multishell_test
```

UPDATE ANT-TOOLBOX

This is not mandatory... just type **updateantx(2)** to update the toolbox, i.e. obtain the latest version from GitHub. For more info type help updateantx.

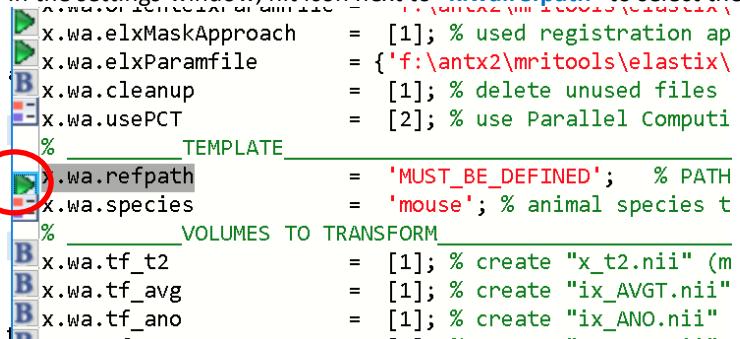
```
updateantx(2)
```

Open ANT-gui & CREATE AN ANT-PROJECT-FILE

Type "ant" to open the GUI.

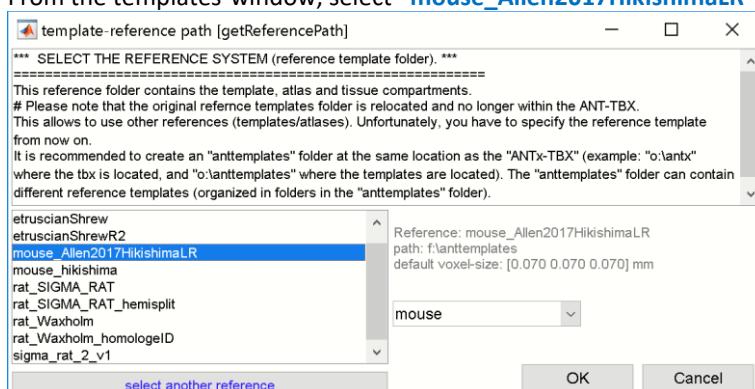
From the ANT-gui menu go to: **Main/New Project**

In the settings-window, hit icon next to "**x.wa.refpath**" to select the reference template



```
x.wa.elxMaskApproach = [1]; % used registration ap
x.wa.elxParamfile = {'f:\antx2\mrifiles\elastix\
x.wa.cleanup = [1]; % delete unused files
x.wa.usePCT = [2]; % use Parallel Computi
% TEMPLATE
x.wa.refpath = 'MUST_BE_DEFINED'; % PATH
x.wa.species = 'mouse'; % animal species t
% VOLUMES TO TRANSFORM
x.wa.tf_t2 = [1]; % create "x_t2.nii" (m
x.wa.tf_avg = [1]; % create "ix_AVGT.nii"
x.wa.tf_ano = [1]; % create "ix_ANO.nii"
```

From the templates-window, select "**mouse_Allen2017HikshimaLR**"...hit [OK].



With this, the settings are defined (see fig below).

Note: a suitable template has to be downloaded from google-drive :

<https://drive.google.com/drive/folders/1q5XOOVLvUYLqYsQJLqNRF7OK8fNwYhI9>

The template should be unzipped and stored where it could be reached (do not save the downloaded template in the current study-folder!).

```

SETTINGS [antconfig.m]
%
% *** CONFIGURATION PARAMETERS ***
%
% =====
x.project = 'NEW_PROJECT'; % PROJECT NAME (arbitrary tag)
x.datpath = 'F:\data\multishell_test\dat'; % studie's datapath, MUST BE specified
x.voxsize = [.07 .07 .07]; % voxel size (default for Alien Mouse: [.07 .07 .07])
x.wa.BiasFieldCor = [0]; % perform initial bias field correction (only needed if initial skull
x.wa.usePriorSkullstrip = [1]; % use a priori skullstripping (used for automatic registration)
x.wa.fastSegment = [1]; % faster segmentation by cutting boundaries of t2.nii [0,1]
x.wa.orientType = [1]; % index from ReorientationTable (see: help findrotation2) to roughly m
x.wa.orientelxParamfile = 'f:\antx2\mitools\elastix\paramfiles\trafoeuler5.txt'; % single Paramete
x.wa.elxMaskApproach = [1]; % used registration approach..click icon for further information
x.wa.elxParamfile = {'f:\antx2\mitools\elastix\paramfiles\Par002saffine.txt' 'f:\antx2\mri
x.wa.cleanup = [1]; % delete unused files in folder
x.wa.usePCT = [2]; % use Parallel Computing toolbox (@n/:SPMD@:parfor)

%
% ===== TEMPLATE =====
x.wa.refpath = 'f:\anttemplates\mouse_Allen2017HikshimaLR'; % PATH of the used reference
x.wa.species = 'mouse'; % animal species to investigate (mouse or rat)
x.wa.tf_t2 = [1]; % create "x_t2.nii" (mouse-t2-image) in TEMPL
x.wa.tf_avg = [1]; % create "ix_AVGT.nii" (template-structural-image) in TEMPL
x.wa.tf_ano = [1]; % create "ix_ANO.nii" (template-label-image) in TEMPL
x.wa.tf_c1 = [0]; % create "x_c1t2.nii" (mouse-grayMatter-image) in TEMPL
x.wa.tf_c2 = [0]; % create "x_c2t2.nii" (mouse-whiteMatter-image) in TEMPL
x.wa.tf_c3 = [0]; % create "x_c3t2.nii" (mouse-CSF-image) in TEMPL
x.wa.tf_cic2mask = [0]; % create "x_c1c2mask.nii" (mouse-gray+whiteMatterMask-image) in TEMPL

```

"...hit [OK]. If asked, just save the project-file as '**proj.m**' in the study-folder (the study-folder here is "multishell_test"). The project "proj.m" is automatically loaded into the ANT-gui.

The study-folder "multishell_test" now contains:

- 1) The folders: "atl_auditsys_08dec20_v1", "raw" and an empty folder "dat" and
- 2) the project-file: "proj.m"

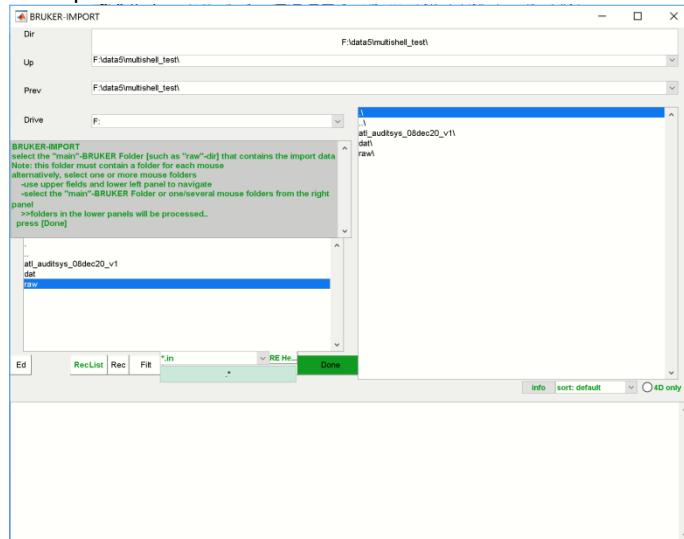
-The project-file "**proj.m**" defines the following: target voxel size is $0.07 \times 0.07 \times 0.07$ mm, the **animal template** is "**mouse Allen2017HikshimaLR**", with the species '**mouse**'.

Creating a project-file has to be done only once!

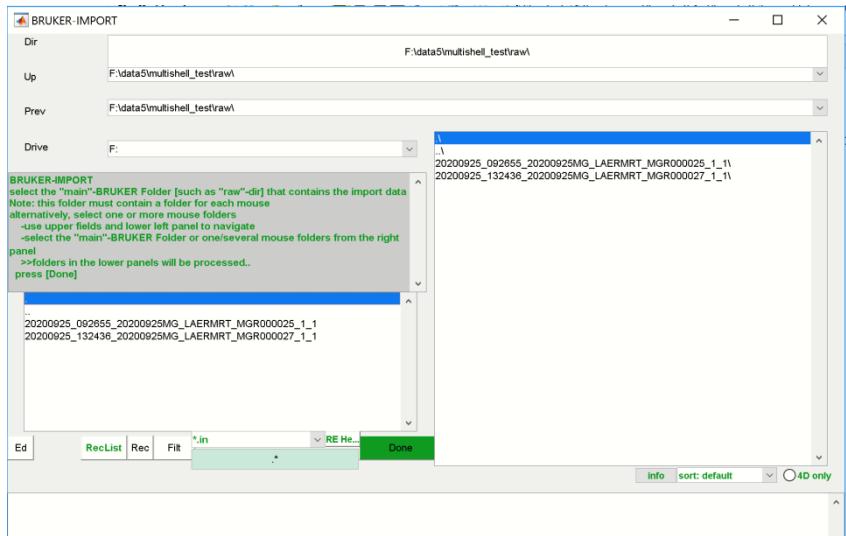
2) IMPORT BRUKER-DATA

From the ANT-gui menu go to: **Main/import Bruker data**

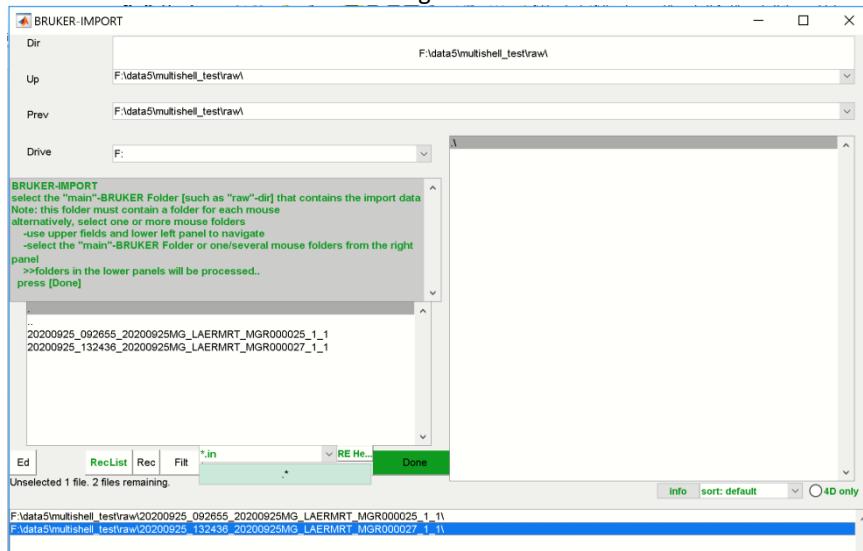
This opens the Bruker dataset-selector:



From the left listbox select the '**raw**'-folder..

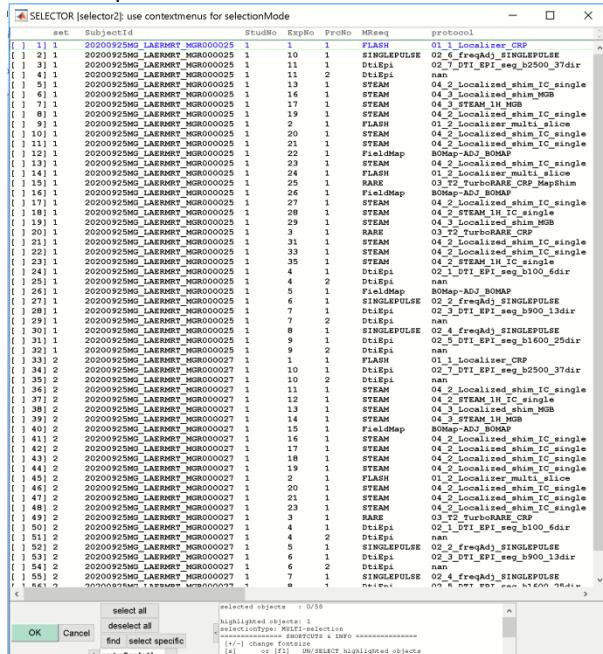


We now see the two datasets in the right listbox. Select the two datasets....

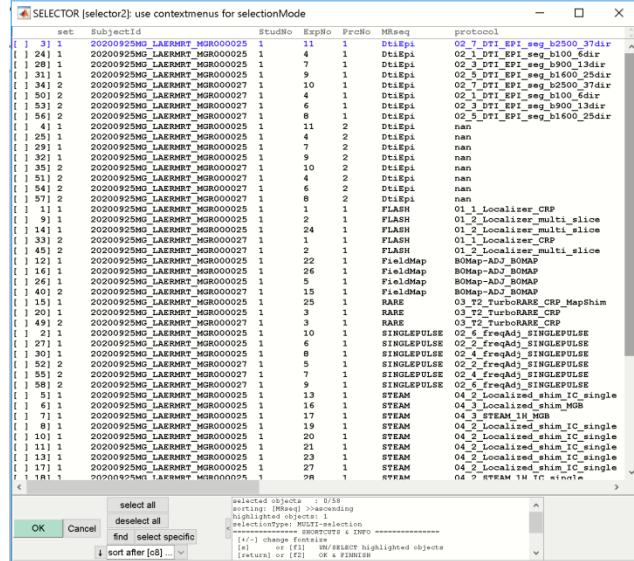


The two datasets now appear in the lower listbox. Hit [DONE].

This will open the file-selector window with all files of the two datasets:



In the “sort after” pulldown-menu select “sort after c8 [MRseq]”.

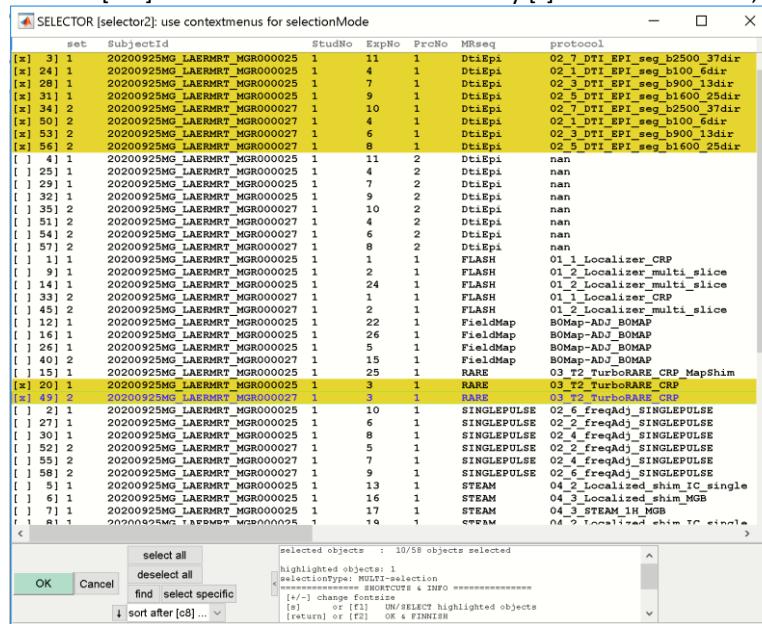


We are now interested in:

1) The DTI-EPI-files, which will be used for DTI-processing. Here, we use a multishell-approach with 4 DTI-EPI files for each dataset (*100*, *900*, *1600* and *2500*).

2) The T2-turborare-files, which will be used for registration to the Allen mouse brain template.

We select altogether 10 files (For selection: highlight the files: left mouse-button pressed while hovering over files or [ctrl]+left mouse-click → than hit key [s] to select the files; do deselect the files hit key [s] again).



Hit **[OK]**.

IN the parameter-Window, do nothing, just hit **[OK]**.

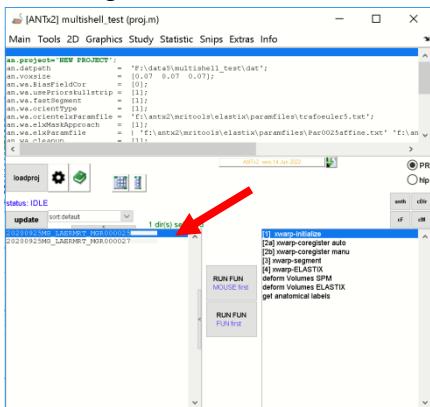
```

BrukerImport
  *** BrukerIMPORT ***
  -define names of output_dirs and imported filenames
  -the suffixes for ExperimentNumber (ExpNo_Dir/ExpNo_File) and ProcessingNumber (PrcNo_Dir/PrcNo_File) will critically determine the existence
  CRITICAL-1: if both ExpNo_Dir/ExpNo_File are not specified and the same protocol is used multiple times [ RUNS] in the same mouse (= 
  .., then the resulting file is overwritten by subsequent runs of the experiment (e.g., three subsequent MPAGE-runs --> [...]pdata\1\zd
  SOLUTION: (1) if there is only one RUNS/EXPERIMENTNUMBER for the same protocol: ExpNo_Dir/ExpNo_File is not necessary
  SOLUTION: (2) if there are multiple RUNS/EXPERIMENTNUMBERS for the same protocol: either ExpNo_Dir/ExpNo_File is necessary
  CRITICAL-2: if both PrcNo_Dir/PrcNo_File are not specified and different [RECONSTRUCTIONS] (i.e. processings) of the same protocol exist for 
  .. the resulting file is overwritten by subsequent reconstructions (e.g.: three processingsNumbers in experimentNumber=4 [...]pdata\
  SOLUTION: (1) if there is only one RECONSTRUCTION for the same protocol: PrcNo_Dir/PrcNo_File is not necessary
  SOLUTION: (2) if there are multiple RECONSTRUCTIONS for the same protocol: either PrcNo_Dir or PrcNo_File is necessary
  =====
  [1] SUFFIXES OF MOUSE DIRECTORY NAME (added to "SubjectId")
  =====
  x.StudNo_Dir = [0]; % VisuStudyNumber (bool)
  x.ExpNo_Dir = [0]; % VisuExperimentNumber (parent folder of "pdata"),(bool)
  x.PrcNo_Dir = [0]; % VisuProcessingNumber/ReconstructionNumber(subfolder of "pdata"),(bool)
  % arrangement of suffixes
  x.delimiter = ','; % delimiter between suffixes (cell); e.g: "s20141009_0isr_121" vs "s20141009_0isr_1_2_1"
  x.suffixLetter = [0]; % add first letter of suffix variable name prior to variable value (bool); e.g: "s20141009_0isr_sle2p1" vs "s20141009_0
  % =====
  [2] SUFFIXES OF FILENAMES (added to "protocol-name")
  =====
  x.ExpNo_File = [0]; % VisuExperimentNumber (parent folder of "pdata"),(bool)
  x.PrcNo_File = [1]; % VisuProcessingNumber/ReconstructionNumber(subfolder of "pdata"),(bool)
  x.renameFiles = ';;'; % rename files ...via GUI
  x.prefix = ';;'; % add prefix to filename
  x.suffix = ';;'; % add suffix to filename
  % =====
  [3] ADDITIONAL OPTIONS
  =====
  x.origin = 'brukerOrigin'; % define center(origin) of volume

```

This will import the DWI-files and the turborare-file for the two datasets.

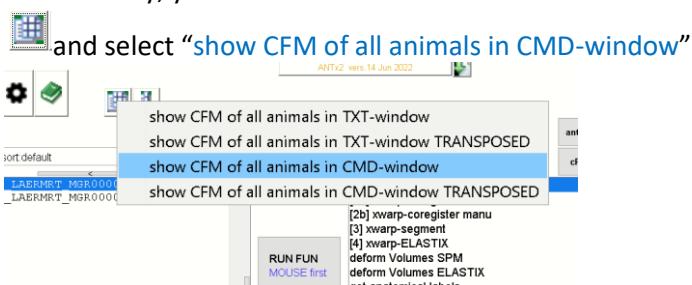
The ANT-gui should now contain two animals in the left listbox.



3) VISUALIZE FILES AND FOLDERS

-type **dispfiles** to see a file-by-folder matrix in the command-window (see dispfiles for further help):
dispfiles

Alternatively, you can select the context-menu of the button “[open Case-File-Matrix for all animals](#)”



This will display the case (animal)-file matrix in the CMD-window:

```

FILE X FOLDER
counts 20200925MG_LAERMRT_MGR000025 20200925MG_LAERMRT_MGR000027
=====
counts      5/5      5/5
=====
02_1_DTI_EPI_seg_b100_6dir_1.nii 2/2 + +
02_3_DTI_EPI_seg_b900_13dir_1.nii 2/2 + +
02_5_DTI_EPI_seg_b1600_25dir_1.nii 2/2 + +
02_7_DTI_EPI_seg_b2500_37dir_1.nii 2/2 + +
03_T2TurboRARE_CRP_1.nii       2/2 + +

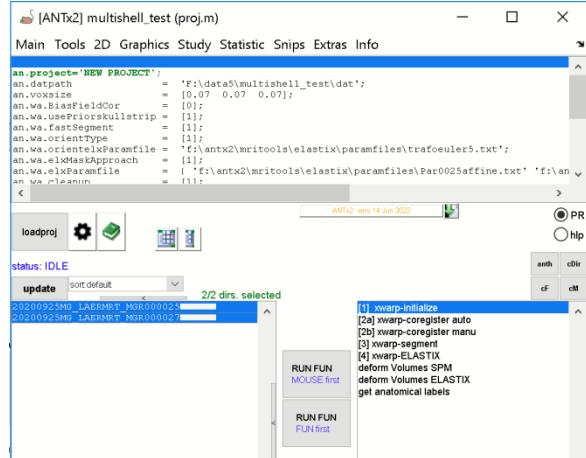
```

Here we see that the study's “dat”-folder now contains two animal-folders (“[20200925MG_LAERMRT_MGR000025](#)” and “[20200925MG_LAERMRT_MGR000027](#)”). Each Folder contains the imported files (4 DTIepi-files, 1 turborare file).

4) SELECTION OF ANIMALS

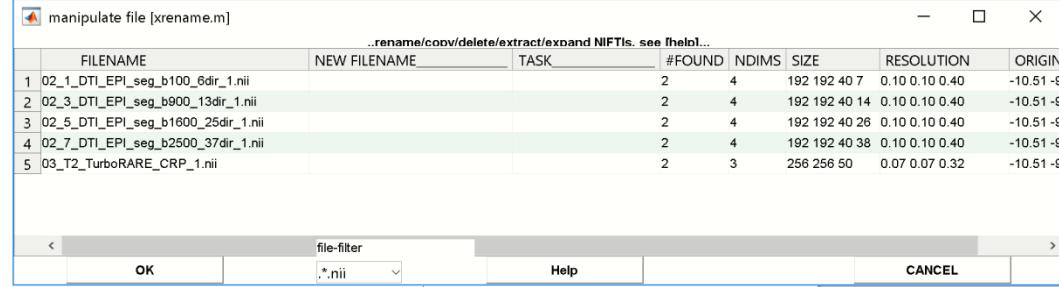
Before renaming the files let's first select the animals we want to process.

Here we will select all animals from the left listbox ([ctrl]-A):



5) RENAME FILES

From the ANT-gui menu go to: **Tools/manipulate files**



Basically we want to:

- 1) Rename the file '**03_T2_TurboRARE_CRP_1.nii**' to '**t2.nii**'. Note that the name convention of '**t2.nii**' is mandatory, because this file is used for registration to standard-space. I.e the registration relies on the name 't2.nii'!
- 2) Rename the DTI-EPI files to make the file-handling easier and more stringent

1) For the turborare-files we type "t2.nii" in the 2nd column ("New Filename") and a ":" (it's the copy symbol) in the 3rd column. Here, for safety reasons, we make a copy of the original file and rename the copied file. Note that copying and renaming of the copied version is defined via the colon-symbol (:) in the 3rd column "Task"). Alternatively, to rename the original files just keep the 3rd column empty.



- 2) For the DWI-files we insert "q<NUMBER>.nii" in the 2nd column, where NUMBER is the number from the filenames in the 1st column (100 ,900 ,1600 ,2500):

i.e. for:

- "02_1_DTI_EPI_seg_b100_6dir_1.nii" we insert "q100.nii" in 2nd column and ":" in the 3rd column
- "02_3_DTI_EPI_seg_b900_13dir_1.nii" we insert "q900.nii" in 2nd column and ":" in the 3rd column
- "02_5_DTI_EPI_seg_b1600_25dir_1.nii" we insert "q1600.nii" in 2nd column and ":" in the 3rd column
- "02_7_DTI_EPI_seg_b2500_37dir_1.nii" we insert "q2500.nii" in 2nd column and ":" in the 3rd column

manipulate file [xrename.m]

...rename/coov/delete/extract/expand NIFTIs. see [help]...									
	FILENAME	NEW FILENAME	TASK	# FOUND	NDIMS	SIZE	RESOLUTION	ORIGIN	
1	02_1_DTI_EPI_seg_b100_6dir_1.nii	q100.nii	:	2	4	192 192 40 7	0.10 0.10 0.40	-10.51 -9	
2	02_3_DTI_EPI_seg_b900_13dir_1.nii	q900.nii	:	2	4	192 192 40 14	0.10 0.10 0.40	-10.51 -9	
3	02_5_DTI_EPI_seg_b1600_25dir_1.nii	q1600.nii	:	2	4	192 192 40 26	0.10 0.10 0.40	-10.51 -9	
4	02_7_DTI_EPI_seg_b2500_37dir_1.nii	q2500.nii	:	2	4	192 192 40 38	0.10 0.10 0.40	-10.51 -9	
5	03_T2_TurboRARE_CRP_1.nii	t2.nii	:	2	3	256 256 50	0.07 0.07 0.32	-10.51 -9	

< file-filter >

OK Help CANCEL

Hit [OK].

NOTE: We make a copy of the original file and rename the copied file. Copying and renaming of the copied version is defined via the colon-symbol (:). Alternatively, to rename the original files just keep the 3rd column empty.

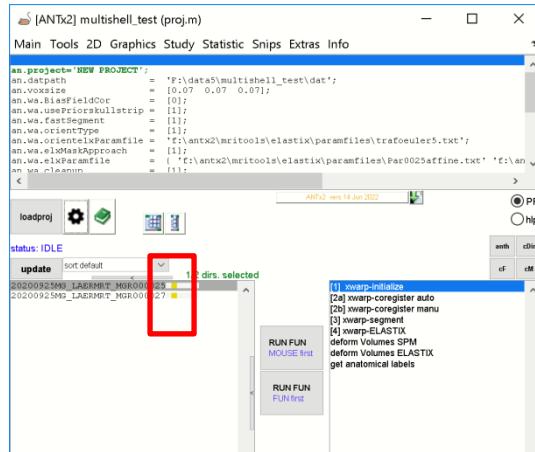
Now, we check whether the new files exist by typing “**disfiles**”

...which will return:

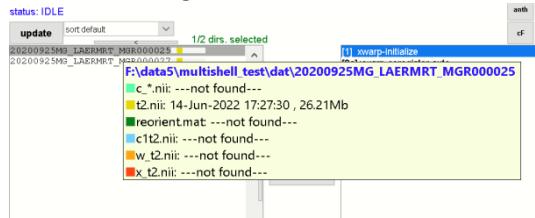
```
FILE & FOLDER
-----
counts 20200925MG_LAERMRT_MGR000025 20200925MG_LAERMRT_MGR000027
=====
counts ===== 10/10 10/10
=====
02_1_DTI_EPI_seg_b100_6dir_1.nii 2/2 + +
02_3_DTI_EPI_seg_b900_13dir_1.nii 2/2 + +
02_5_DTI_EPI_seg_b1600_25dir_1.nii 2/2 + +
02_7_DTI_EPI_seg_b2500_37dir_1.nii 2/2 + +
03_T2_TurboRARE_CRP_1.nii 2/2 + +
q100.nii 2/2 + +
q1600.nii 2/2 + +
q2500.nii 2/2 + +
q900.nii 2/2 + +
t2.nii 2/2 + +
```

Here we see that the files “q100.nii”, “q900.nii”, “q1600.nii”, “q2500.nii” and “t2.nii” were successfully created for all datasets.

We should also see that the two animals in the left listbox show a yellow-box.



When hovering over an animal we can see that the „t2.nii“-file exists for this animal.



6) Determine Orientation-Type

Before running the registration to standard-space we have to assure that the orientation of the animal is ok, otherwise we have to change the orientationType.

First, we have to import the template-data for this study.

For this, go to **ANT-menu: MAIN/create study templates**.

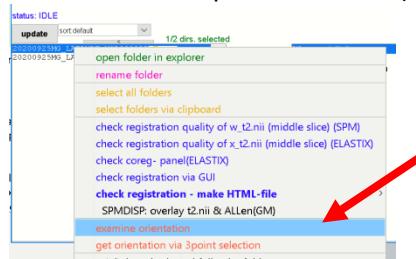
This step will copy the template-files from the reference-template (defined in variable **x.wa.refpath** in the “**proj.m**”-file) to the “templates”-folder with the target voxel-size.

The new “templates”-folder within this study-folder will now contain the following files:

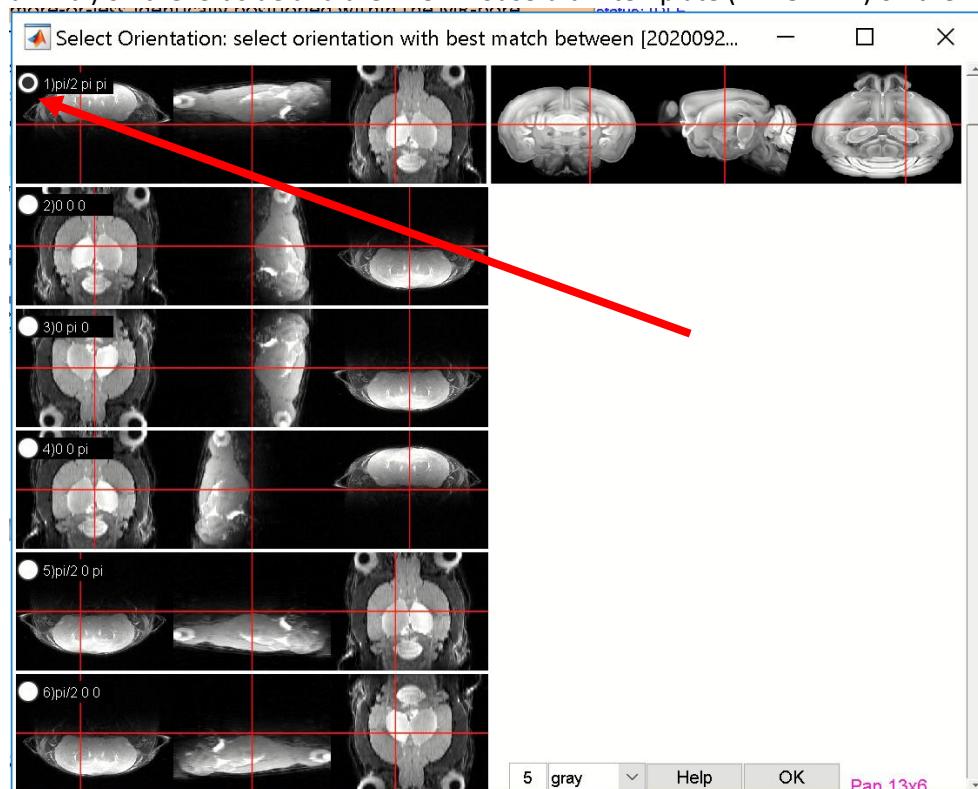
```
.
    AVGThemi.nii  parameter.m
...
ANO.nii      AVGTmask.nii  readme.txt
ANO.xlsx     _b1grey.nii
AVGT.nii     _b2white.nii
              _b3csf.nii
```

The orientation-type has to be determined only once per study assuming that all animals were more-or-less identically positioned within the MR-bore.

To determine the orientation select **one** animal from the left listbox and from the context-menu select “**examine orientation**”. (Alternatively you could also use the “**get orientation via 3 point selection**” as an alternative approach to obtain the orientation...-> see special tutorial for the “get orientation via 3 point selection”).



This will open a window with different orientation-types of the “t2.nii”-image (of the selected animal) on the left side and the Allen mouse brain template (“AVGT.nii”) on the right side.



Here we see that the 1st orientationType, i.e. 1st panel (rotation: pi/2 pi pi) shows the optimal orientation. We check the respective radio (**radio-1**) and hit [**OK**].

The CMD-window will show the following:

20200925MG_LAERMRT_MGR000025 rotTable-ID is [1] ..which is "pi/2 pi pi." [HELP]

This indicates that we selected the orientationType [1] (which is “...rotTable-ID is [1]”).

Now we have to set the orientation-type in the **parameter-settings (“proj.m”)**. For this we select the gear-icon from the ANT-gui (i.e. open the settings menu).

```

% ===== CONFIGURATION PARAMETERS =====%
%
x.project      = 'NEW PROJECT'; % PROJECT NAME (arbitrary tag)
x.datpath      = 'F:\data5\multishell_test\dat'; % studie's datapath, MUST BE specified, and named
x.voxsize      = [0.07 0.07 0.07]; % voxel size (default for Allen Mouse: [0.07 0.07 0.07])
%
x.wa.BiasFieldCor = [0]; % perform initial bias field correction (only needed if initial skullstripping failed)
x.wa.usePriorSkullstrip = [1]; % use a priori skullstripping (used for automatic registration)
x.wa.fastSegment = [1]; % faster segmentation by cutting boundaries of t2.nii [0,1]
x.wa.orientationType = [1]; % index from ReorientationTable (see: help findrotation2) to roughly match inputVol
x.wa.orientelexParamfile = 'f:\antx2\mrifitools\elastix\paramfiles\trafoeuler5.txt'; % single Parameter file for r
x.wa.elxMaskApproach = [1]; % used registration approach .click icon for further information
x.wa.elxParamfile = ('f:\antx2\mrifitools\elastix\paramfiles\Par0025affine.txt' 'f:\antx2\mrifitools\elastix\paramfiles\Par0025affine.txt')
x.wa.cleanup    = [1]; % delete unused files in folder
x.wa.usePCT     = [2]; % use Parallel Computing toolbox (@no:1:SPMD/2:parfor)
%
TEMPLATE
x.wa.refpath   = 'f:\anttemples\mouse_Allein2017HikshimaLR'; % PATH of the used reference system (
x.wa.species   = 'mouse'; % animal species to investigate {mouse or rat}
%
VOLUMES TO TRANSFORM
x.wa.tf_t2     = [1]; % create "x_t2.nii" (mouse-t2-image) in TEMPLATESPACE (from template)
x.wa.tf_avg    = [1]; % create "ix_AVGT.nii" (template-structural-image) in MOUSESSPACE
x.wa.tf_ano    = [1]; % create "ix_ANO.nii" (template-label-image) in MOUSESPACE
x.wa.tf_c1     = [0]; % create "x_c1t2.nii" (mouse-grayMatter-image) in TEMPLATESPACE (from template)
x.wa.tf_c2     = [0]; % create "x_c2t2.nii" (mouse-whiteMatter-image) in TEMPLATESPACE (from template)
x.wa.tf_c3     = [0]; % create "x_c3t2.nii" (mouse-CSF-image) in TEMPLATESPACE (from template)
x.wa.tf_cic2mask = [0]; % create "x_cic2mask.nii" (mouse-gray+whiteMatterMask-image) in TEMPLATESPACE (from template)

```

Now we have to change the **orientationType**: set “`x.wa.orientationType = [1]`”

Luckily, this `orientationType` is already the set (default). So we have to do nothing. Otherwise, set the respective numerical value here, then hit [OK], and if asked..than hit **yes overwrite current configfile** to update the current project-file.

7) REGISTER “t2.nii” TO TEMPLATE SPACE (STANDARD-SPACE)

Let's first select all animals from the left listbox.

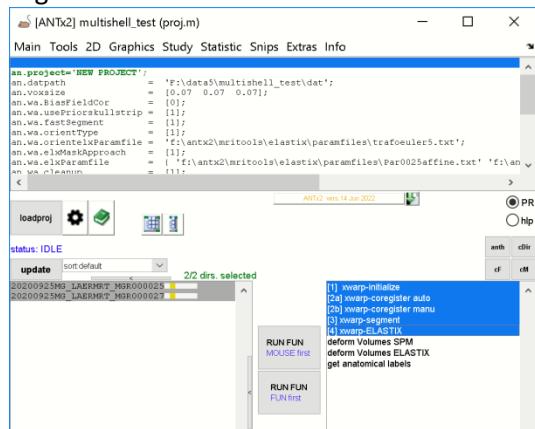
Registration of **“t2.nii”** to standard-space is done in 4 steps:

[1] initialization, [2] rigid registration, [3] segmentation and [4] warping. You can perform these steps ('task') isolated & sequentially or combined. Note however, that task '2' can be only performed when task '1' has been already performed and so on.

Step-[2] has two alternative steps: [2a] the automatic mode and [2b] the manual mode. In most cases you never need the manual mode (actually you don't need it!). Note also, if both mode are selected the automatic mode will override the manual mode (in effect, the manual mode is skipped from the task list).

Now, select the steps **[1],[2a],[2b],[3] and [4]** from the right listbox.

Again, step-[2a] the automatic registration mode is used here and will disable the mode [2b] manual registration... i.e. this selection is identical to selecting steps **[1],[2a],[3] and [4]**.



We can now run the registration to standard-space by hitting the **lower [RUN FUN]** button. If we want to process the data in **parallel** (assuming parallel processing toolbox is available and ready-to-work) we can also press the **upper [RUN FUN]**.

Lets press the **lower [RUN FUN]** here.

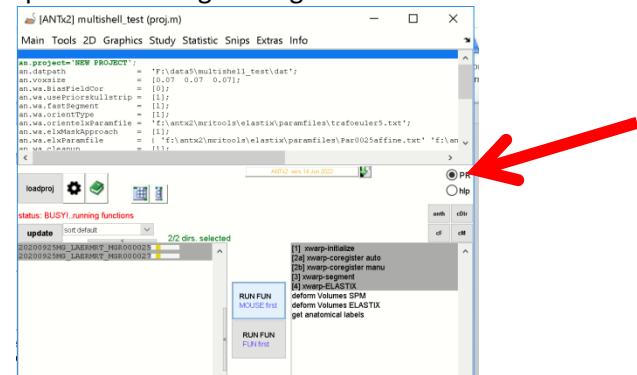
8) Examine the Processing Report

A web-browser should open when the registration to standard-space starts. The browser should contain a web-page with the progress of the registration. If the browser does not open, just open the

study-folder in the file explorer and select "[summary.html](#)" (this file is created and modified during the registration of animal data to standard-space).

Name	Änderungsdatum	Größe	Typ
raw	14.06.22 16:01		Dateiordner
atl_audit.sys_08dec2...	27.04.22 16:21		Dateiordner
dat	14.06.22 16:22		Dateiordner
proj.m	14.06.22 18:01	4 KB	M-Datei
templates	14.06.22 17:33		Dateiordner
summary_steps	14.06.22 18:31		Dateiordner
summary.html	14.06.22 18:33	5 KB	Chrome HTML Docu...
summaryLog.mat	14.06.22 18:31	1 KB	MATLAB Data

You may also verify that the **radio [PR]** in the ANT-gui is checked. Otherwise the webpage won't pop-up when starting the registration.



The current status of the registration looks as follows:

processing report

← → C Datei | F:/data5/multishell_test/summary.html

Processing Report

Project: F:/data5/multishell_test

50% processed █ waiting ## start time: 14-Jun-2022 18:31:01; ## end time: ; ## diff time: >7 mins, 57.0 secs; last performed case: "20200925MG_LAERMRT_MGR000027" <>> Coregistration

processing.. ● 20200925MG_LAERMRT_MGR000025 [inspect](#)
processing.. ● 20200925MG_LAERMRT_MGR000027 [inspect](#)

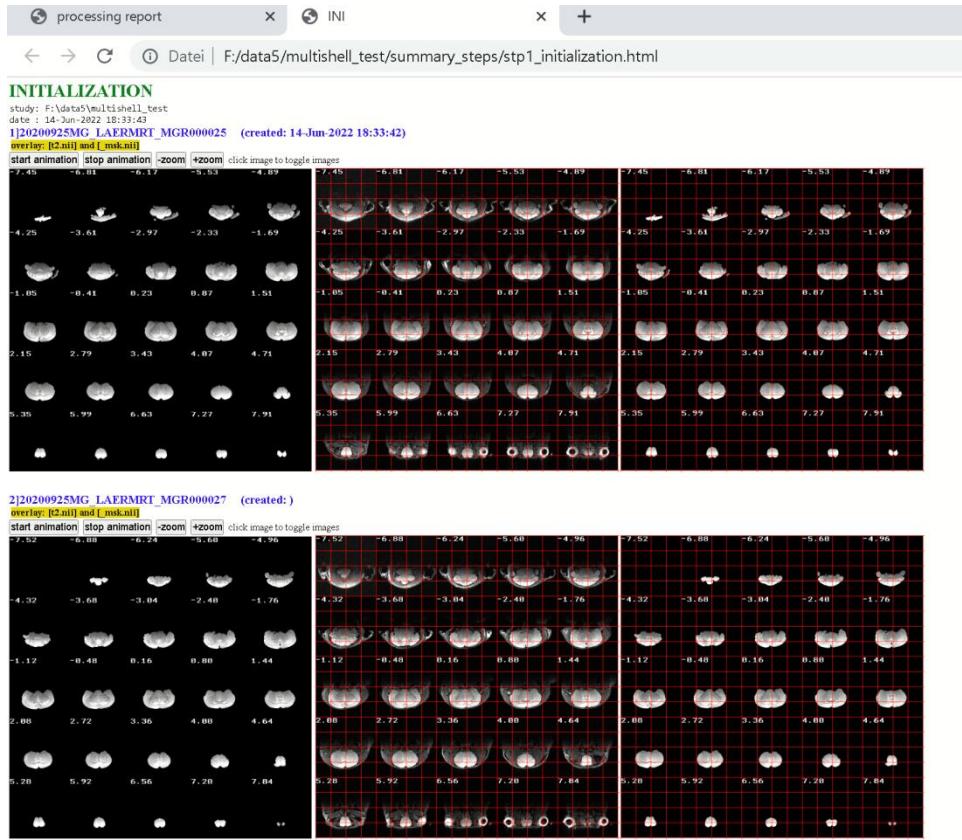
LEGEND:

- [dataset-x] : data processed in this session
- [dataset-y] : data not processed in this session
- ...selected tasks to process (marked yellow) {initialization, coregistration, segmentation, warping}
- ..task never performed for this data set
- ..task has been performed before (and might be overwritten)
- *..the currently running task
- finished task

[initialization](#) [coregistration](#) [segmentation](#) [warping](#)

We see that the first two steps (initialization and registration) are already performed (**green dot**) and the segmentation-step is currently running (**blue sun**)

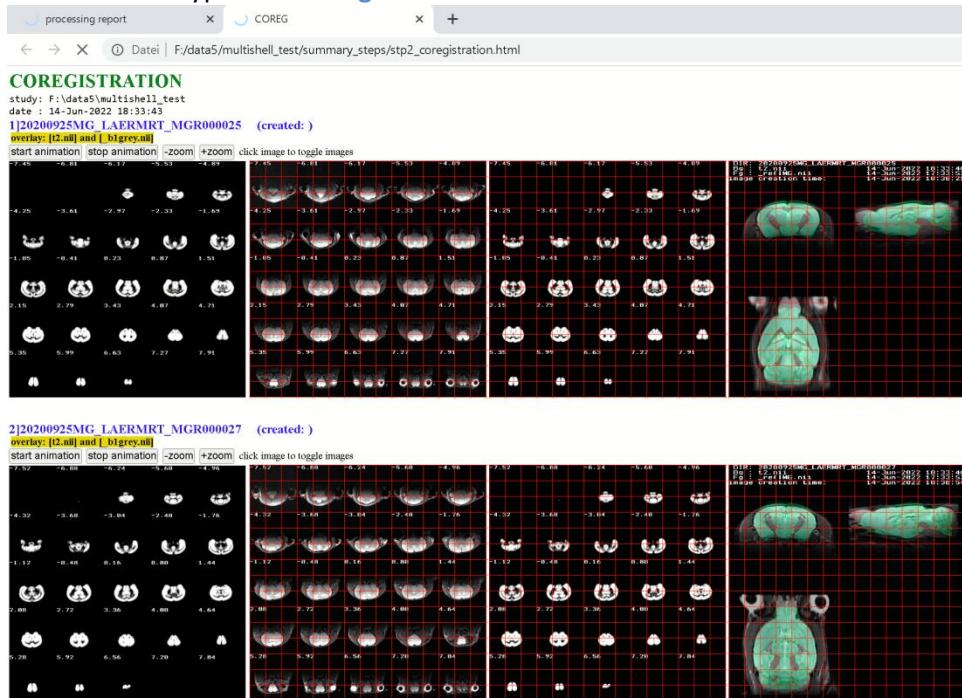
Let's click the hyperlink "[initialization](#)".



[\[top\]](#)

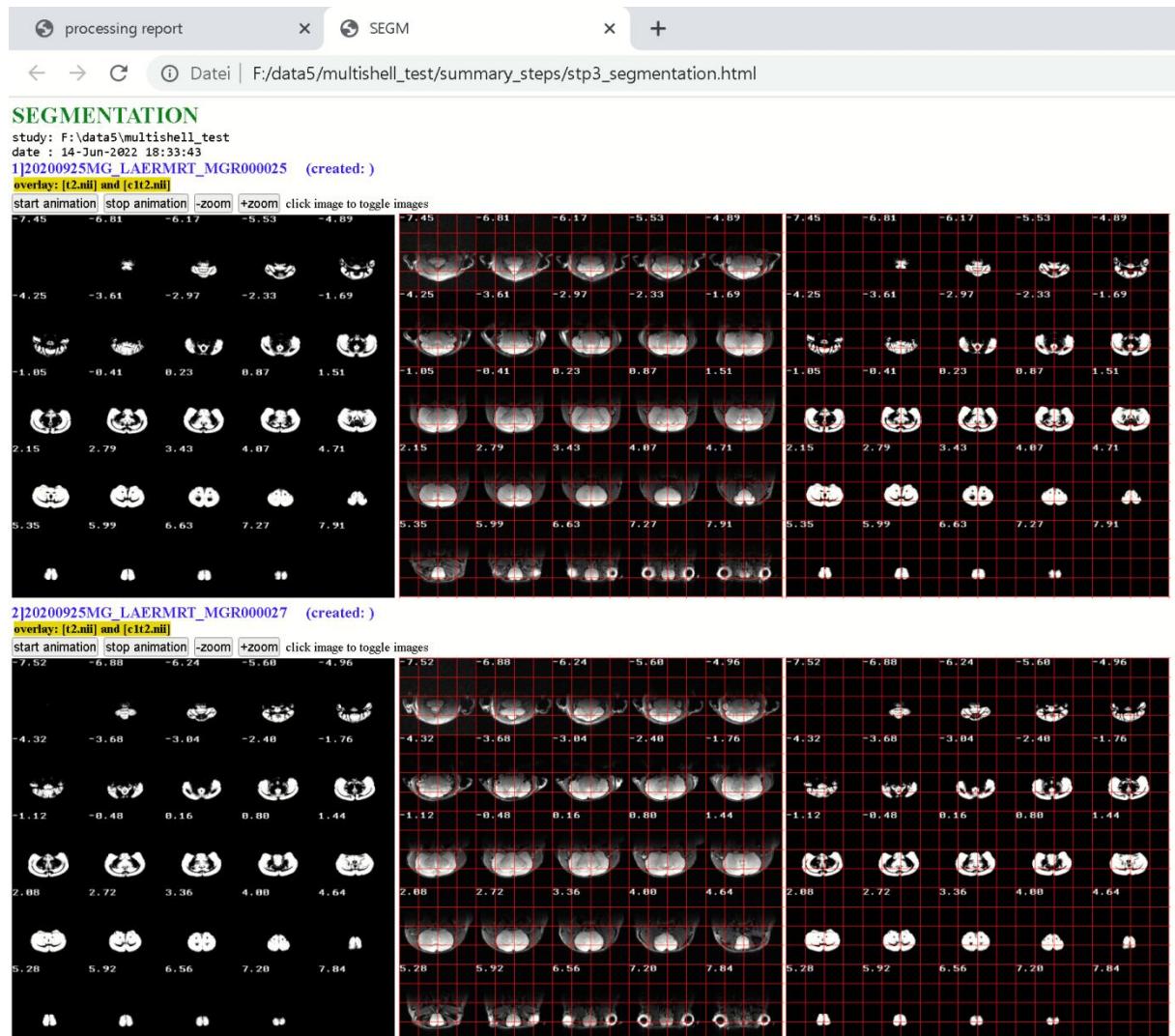
Here we see that the skullstripping has been performed successfully for the two animals. Hit left image to toggle between “t2.nii” and the skullstripped brain-mask.

Let's click the hyperlink “[coregistration](#)”:



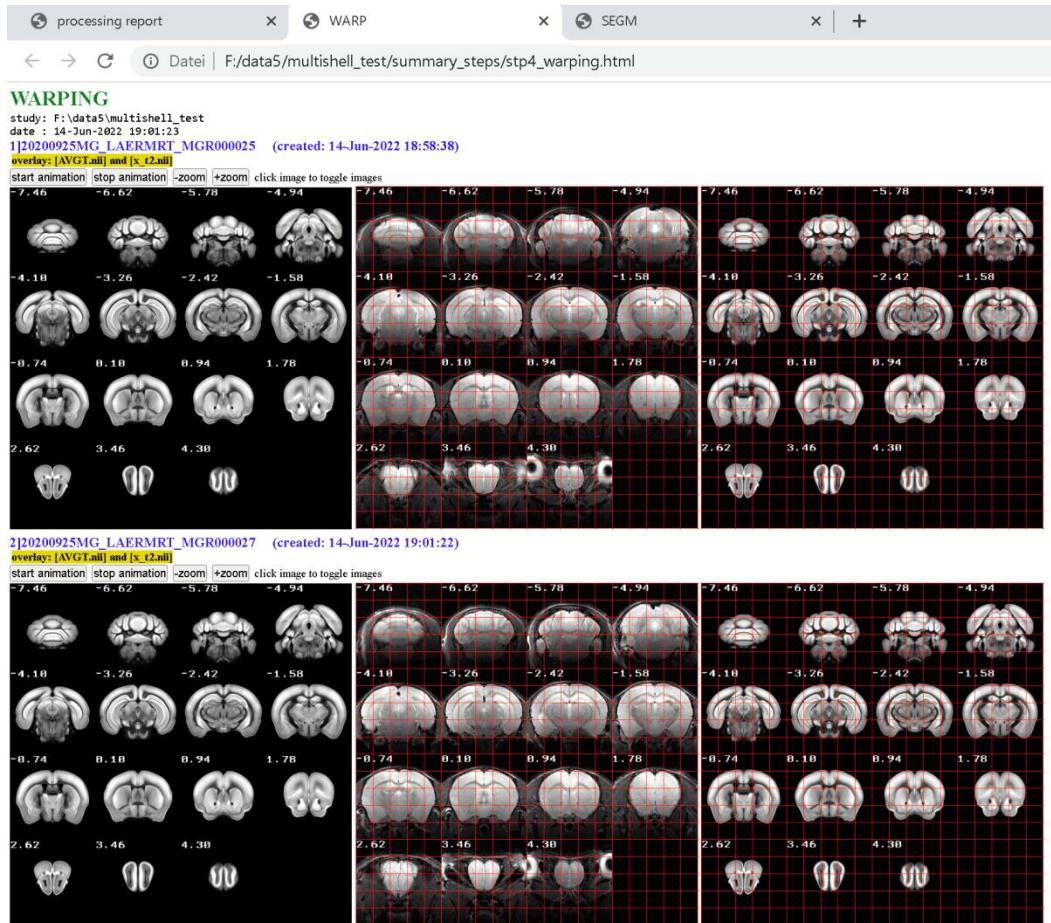
Here we see that the rigid registration was successful for the two animals. Hit the left image to toggle between the animal’s “t2.nii” and the Allen mouse-brain template. The right image is the most easiest ways to check whether the rigid registration worked.

Let's click the hyperlink “[segmentation](#)”:



Here we see that the tissue segmentation was successful for the two animals. Hit the left image to toggle between the animal's "t2.nii" and gray matter tissue image.

Let's click the hyperlink "[warping](#)":



Here we see that the warping to standard-space was successful for the two animals. Hit the left image to toggle between the animal's "t2.nii" and the Allen mouse template.

With the last step, checking the warping results via the progress-report, we can say that the registration to standard-space was successful.

The registration from standard-space to native-space (space of "t2.nii"), i.e. the inverse registration, is part of the registration-pipeline. Thus we can now use the calculated transformation parameters to transform images from standard-space to native-space, and vice versa.

Images transformed **from native-space to standard-space** will obtain the prefix "**x_**":

Examples:

- „t2.nii“ → is “**x_t2.nii**” in standard-space
- „masklesion.nii“ → is „**x_masklesion.nii**“ in standard-space (image does not exist in this study)
- „c1t2.nii“ → is “**x_c1t2.nii**” in standard-space (the grey matter image)

Conversely, images inversely transformed **from standard-space to native-space** obtain the prefix **“ix_”**:

Examples:

- „AVGT.nii“ → is “**ix_AVGT.nii**” in native-space, (Allen mouse brain template)
- „AVGThemI.nii“ → is “**ix_AVGThemI.nii**” in native-space (hemispheric mask)
- „ANO.nii“ → is “**ix_ANO.nii**” in native-space (Allen mouse brain atlas)

PART-2: DTI-preprocessing/prepare data for DTI-MRtrix-pipeline

Now the data needs to be prepared to run the DTI-MRtrix-pipeline.

The MRtrix pipeline is defined by a set of shellscripts and basically calculates the diffusion tensor metrics and performs the tractography. The DTI-processing is done using the MRtrix software package. Several steps have to be done before running the MRtrix-pipeline.

Note that a multishell-approach is used for this data. However for single-shell data (one instead of several DWI-files) the steps are more or less the same (viewer files)

Note that all necessary DWI-files are already imported (section: Import Bruker-raw data) and renamed to “q100.nii”, “q900.nii”, “q1600.nii” and “q2500.nii” (section: rename files)

9) A special DTI-atlas is needed

We need a special atlas with comparable fewer regions for DTI-processing (compared to the Allen mouse brain atlas, ABA). Otherwise the parcellation would be too fine-grained. For this, a NIFTI-file with regions of interest (regions partially merged from fine-grained regions from the ABA) was created (**‘atl_auditsys_08dec20.nii’**, here termed “DWI/DTI-filename”). Additionally a corresponding look-up-table was created (**‘atl_auditsys_08dec20_INFO.txt’**, here termed “lutfile”) which contains the region IDs in subsequent order and looks as follows:

#ID	Labelname	R	G	B	A
1	L_Supplemental_somatosensory_area_layer_6b_MODIF	24	128	100	255
2	L_Ventral_auditory_area_layer_6b_MODIF	1	147	153	255
3	L_Primary_auditory_area_layer_6b_MODIF	1	147	153	255
4	L_Postrhinal_area_layer_6b_MODIF	8	133	140	255
5	L_Anterior_cingulate_area_ventral_part_6b_MODIF	64	166	102	255
...					
36	R_Nucleus_y_MODIF	255	179	217	255
37	R_Flocculus_MODIF	255	252	145	255
38	R_Triangular_nucleus_of_septum_MODIF	150	167	211	255

Note that the study-folder already contained the folder ‘atl_auditsys_08dec20_v1’ containing the files **‘atl_auditsys_08dec20.nii’** and **‘atl_auditsys_08dec20_INFO.txt’**.

Important: A DTI-atlas and accompanied LUTfile are needed for DTI-processing using MRtrix.

To create your own Atlas you could use **ANT-Menu: Tools/”Make Atlas From Excelfile”** (xexcel2atlas.m). Before doing that you have to make a local copy the “ANO.xlsx”-file and add (somewhere) two columns:

- 1) A column with new IDs for those regions that should appear in the new atlas. The same new IDs can be given to regions that should be merged (i.e. incorporated into one larger region). Keep sequential order (i.e. do not use ID=20 if lower IDs are newer given in the new atlas)
- 2) A column that define the hemispheric set-up (such as to split the region into the left and right hemisphere) ... see help of **xexcel2atlas.m**.

10) DTI-preprocessing

The registration of “t2.nii” to standard-space is mandatory before performing the DTI-preprocessing step!

Luckily, we have already done this step.

The DTI-preprocessing steps contain a number of operations:

(1) Prepare data

- make “DTI”-folder & create the DTIprep-struct
- assign a sample Bruker-raw data set to extract the b-tables
- assign DWI/DTI-files (d.DTIfilename)
- assign DTI-atlas (d.DTITemplate & d.DTITemplateLUT) (those from the previous section)

(2) perform the following tasks:

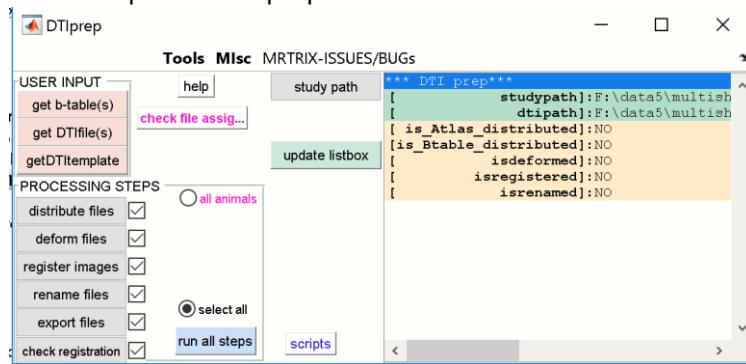
1. distribute files → copy DTI-atlas/lutfile & b-tables to animal-dirs

- 2. deform files → transform DTIatlas, brainmask etc. to native-space
- 3. register files → register "t2.nii" to DWI-file, than apply trafo to native-space files such as ("ix_" +DTIatlas, "ix" +brainmask etc.)
- 4. rename files → rename files (file names are fixed and are expected from the shellscripts)
- 5. export files → export files to the "DTI_export4mrtrix"-folder (OPTIONAL STEP!)
- 6. check registration-> create HTML-files with overlays of images to visualize the coregistration
 - This step is done using the data from the 'dat'-folder (not the export-folder!)
 - HTML-files are stored in the "checks"-folder within the study's folder
 - OPTIONAL STEP!

First, for safety reasons, let's first delete the "**DTI**"-folder within the study-folder. Note that the "**DTI**"-folder will be created in the next step and will contain basic parameters, the b-tables and the DTI-atlas & lutfile.

DTIprep('delete');

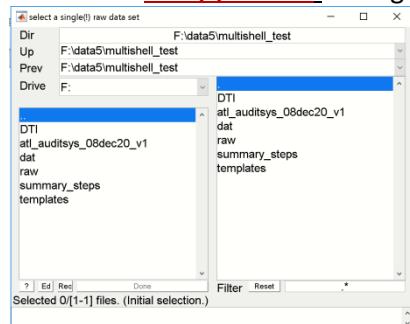
From ANT-MENU select: **STATISTIC/DTI-prep for MRtrix** (or type **DTIprep** in Matlab-CMD-window). This will open the DTIprep-window:



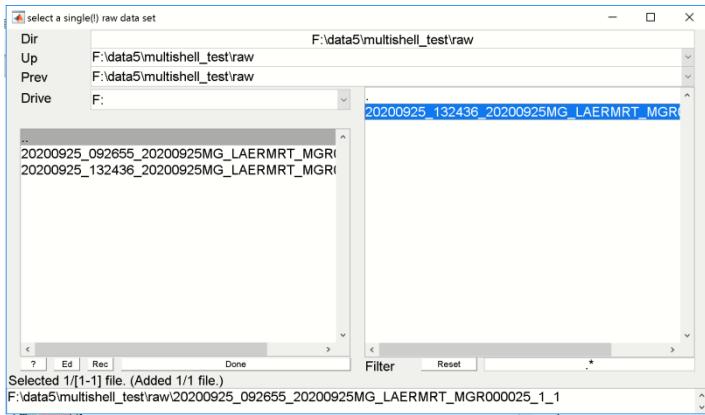
We now have to specify the three **USER INPUTs**: 1) [get b-table(s)], 2) [get DTIfile(s)] and 3) [getDTItemplate]

10.1) IMPORT B-tables [get b-table(s)]

Here we need to extract the b-tables (multishell) or the b-table (singleshell) from the Bruker raw-data set of one (!) animal. Let's go and hit the [get b-table(s)]-button.



From the left listbox select the "raw"-folder. This will list the two datasets within the raw-folder in the right listbox. We now select one dataset, which is appearing in the lower listbox.



Hit [DONE].

Note all USER-INPUTS have to be done ONLY ONCE!

ALSO, the b-table is assumed to be similar across animals of a study, thus we can select just the raw dataset of the first animal.

The CMD-window will display the following:

```
1b-table "100" stored as F:\data5\multishell_test\DTI\grad_b100.txt
  no directions : 7 (with 1 image(s) acquired without diffusion gradients)
  output file   : [grad_b100.txt]: Explorer or open >> F:\data5\multishell_test\DTI\grad_b100.txt
    ..input file: F:\data5\multishell_test\raw\20200925_132436_20200925MG_LAERMRT_MGR000027_1_1\4\method
    ..scan-No   : 18
2b-table "900" stored as F:\data5\multishell_test\DTI\grad_b900.txt
  no directions : 14 (with 1 image(s) acquired without diffusion gradients)
  output file   : [grad_b900.txt]: Explorer or open >> F:\data5\multishell_test\DTI\grad_b900.txt
    ..input file: F:\data5\multishell_test\raw\20200925_132436_20200925MG_LAERMRT_MGR000027_1_1\6\method
    ..scan-No   : 20
3b-table "1600" stored as F:\data5\multishell_test\DTI\grad_b1600.txt
  no directions : 26 (with 1 image(s) acquired without diffusion gradients)
  output file   : [grad_b1600.txt]: Explorer or open >> F:\data5\multishell_test\DTI\grad_b1600.txt
    ..input file: F:\data5\multishell_test\raw\20200925_132436_20200925MG_LAERMRT_MGR000027_1_1\8\method
    ..scan-No   : 22
4b-table "2500" stored as F:\data5\multishell_test\DTI\grad_b2500.txt
  no directions : 38 (with 1 image(s) acquired without diffusion gradients)
  output file   : [grad_b2500.txt]: Explorer or open >> F:\data5\multishell_test\DTI\grad_b2500.txt
    ..input file: F:\data5\multishell_test\raw\20200925_132436_20200925MG_LAERMRT_MGR000027_1_1\10\method
    ..scan-No   : 2
DONE!
```

And we can also see that the color of the button is changed...this user input is performed!



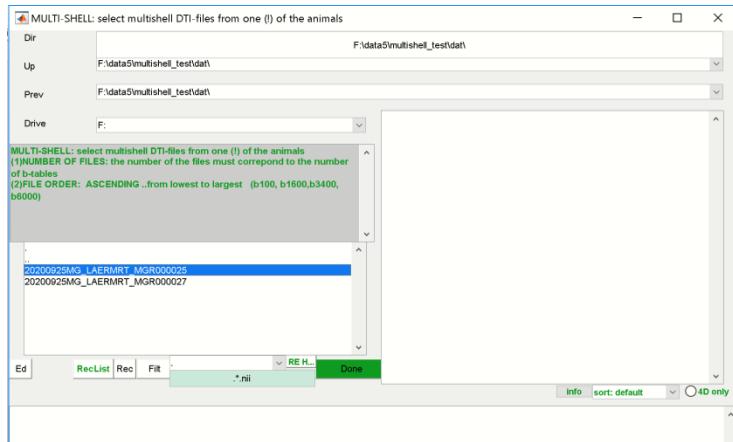
IMPORTANT: we can close the DTIprep window at any time and reopen the window later on. All necessary information is stored in the DTI-folder

10.2 Specify DTI/DWI-files [get DTIfile(s)]

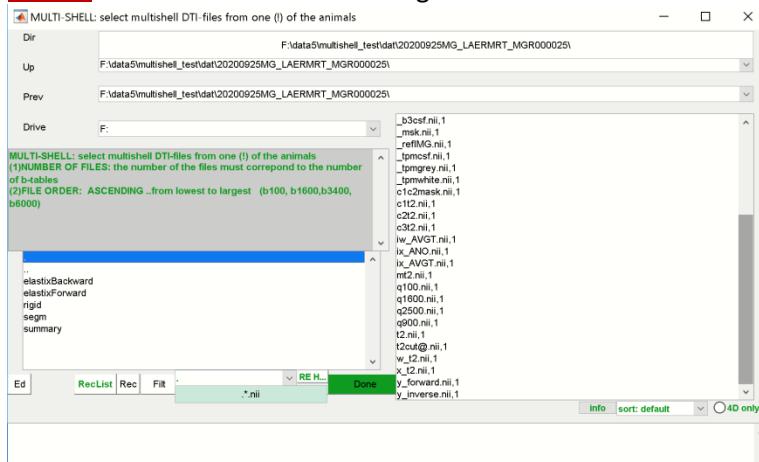
We next have to specify the names of the used DWI-files. Note that we assume that at this state all DWI-files across animals have the same names ("q100.nii", "q900.nii", "q1600.nii" and "q2500.nii").

Thus we specify the DTI/DWI-files from **the first animal in the dat-folder**.

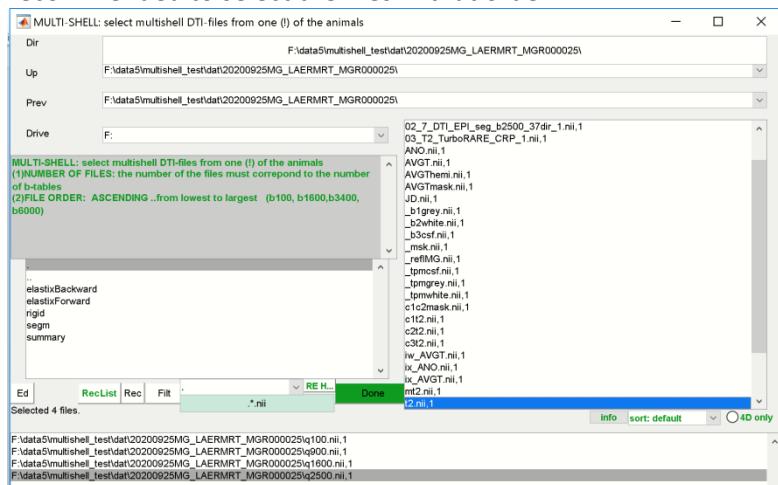
Let's go and hit the [get DTIfile(s)]-button.



The file-selector window shows the two animals of the “dat”-folder in the left listbox. Select **one!** animal from the left listbox. The right listbox will now show all Nifti-files from this animal.

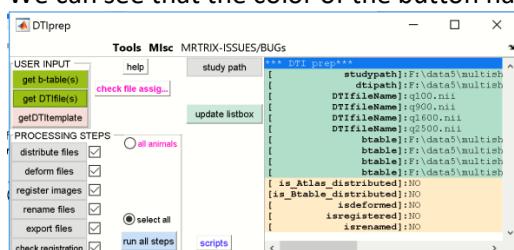


Now select the files “**q100.nii**”, “**q900.nii**”, “**q1600.nii**” and “**q2500.nii**” from the right listbox. It is recommended to select the files in that order.



Hit **[DONE]**.

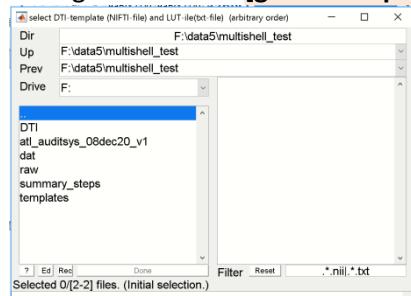
We can see that the color of the button has changed...this user input is performed!



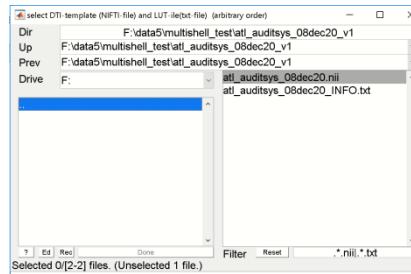
10.3) specify the DTI-atlas [getDTItemplate]

In the last step of the user-input we have to specify the DTI-atlas. Specifically we have to assign the **DTI-ATLAS** (NIFTI-file) and the **LUTFILE** (txt-file).

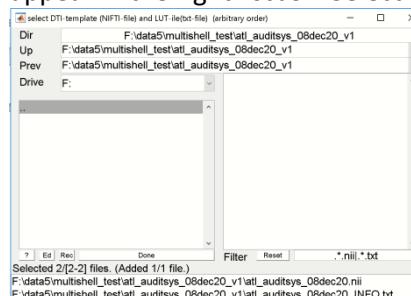
Let's go and hit the **[getDTItemplate]-button**.



From the left listbox select the "[atl_auditsys_08dec20_v1](#)"-folder...



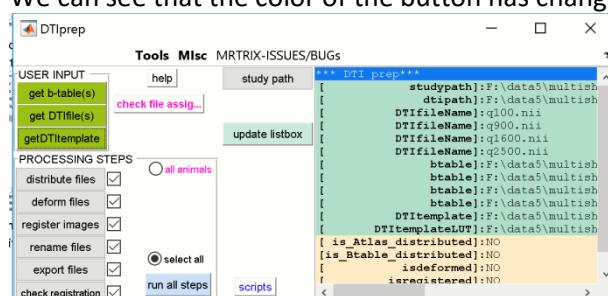
Now the DTI-Atlas ("[atl_auditsys_08dec20.nii](#)") and LUTfile ("[atl_auditsys_08dec20_INFO.txt](#)") appear in the right listbox. Select **both** files from the right listbox.



These files will appear in the lower listbox.

Hit **[DONE]!**

We can see that the color of the button has changed...this user input is performed!



10.4) Check file-assignment and order of b-tables and DTI-files

We now have to check the file-assignment to be sure that the b-tables and DWI/DTI-files are assigned in the right order. For this, Hit **[check file assignment]-button**.

b-table	INPUT-DWI	OUTPUT-DWI
grad_b100.txt	q100.nii	dwi_b100.nii
grad_b900.txt	q900.nii	dwi_b900.nii
grad_b1600.txt	q1600.nii	dwi_b1600.nii
grad_b2500.txt	q2500.nii	dwi_b2500.nii

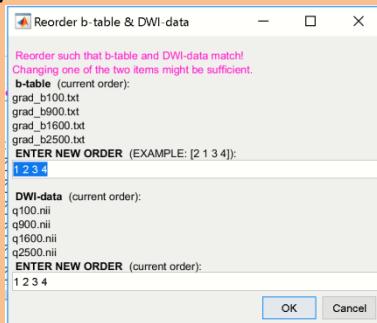
-Please check correspondence of b-tables and input-DWI-files!
to reorder files use contextmenu: reorder..
The table does not state that these files exist for each animal!

We see that:

- 1) The b-tables (1st column) are ordered from lowest to highest... which is good!
- 2) The DTI/DWI-files (2nd column) are ordered from lowest to highest and the numbers do match with the b-tables ...that's good!

IMPORTANT!

If the order of the b-tables or the order of DTI/DWI-files is not from lowest to highest number and both b-tables and DTI/DWI-files do not match than reorder the b-tables and/or DTI/DWI-files via DTI-prep GUI-MENU: **TOOLS/reorder b-tables/DWI-files (or via context-menu)** ... change the order accordingly:



Here, the order seems to be ok!

To check the assignment you could also type:

```
DTIprep('check');
```

Which returns:

```
>> DTIprep('check');
*** [DTIprep-info saved]: F:\data5\multishell_test\DTI\check.mat
-----[struct info]-----
    studypath: 'F:\data5\multishell_test'
    dipath: 'F:\data5\multishell_test\DTI'
    DTIfileName: {4x1 cell}
        btable: {4x1 cell}
        DTITemplate: 'F:\data5\multishell_test\atl_audit...'
        DTITemplateLUT: 'F:\data5\multishell_test\atl_audit...'
    is_Atlas_distributed: 'NO'
    is_Btable_distributed: 'NO'
    isdeformed: 'NO'
    isregistered: 'NO'
    isrenamed: 'NO'
-----[btable] -----
'F:\data5\multishell_test\DTI\grad_b100.txt'
'F:\data5\multishell_test\DTI\grad_b900.txt'
'F:\data5\multishell_test\DTI\grad_b1600.txt'
'F:\data5\multishell_test\DTI\grad_b2500.txt'
-----[DTIfileName] -----
'q100.nii'
'q900.nii'
'q1600.nii'
'q2500.nii'
```

Note that the order and the assignments match (see [btable] and [DTIfileName]).

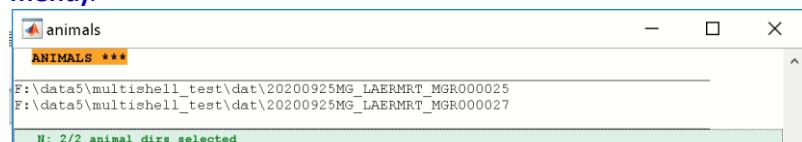
10.5) Run the DTIprep-tasks

We now want to run all 6 tasks. Just as a reminder, here are the tasks.

1. distribute files → copy DTI-atlas/lutfile & b-tables to animal-dirs
2. deform files → transform DTIatlas, brainmask etc. to native-space
3. register files → register "t2.nii" to DWI-file, than apply trafo to native-space files such as ("ix_" +DTIatlas, "ix"+brainmask etc.)
4. rename files → rename files (file names are fixed and are expected from the shellscripts)
5. export files → export files to the "DTI_export4mrtrix"-folder (OPTIONAL STEP!)
6. check registration → create HTML-files with overlays of images to visualize the coregistration
 -This step is done using the data from the 'dat'-folder (not the export-folder!)
 -HTML-files are stored in the "checks"-folder within the study's folder
 -OPTIONAL STEP!

But we first want to be sure that the 6 tasks will be performed for all our datasets.

From DTI-prep GUI-MENU select: **TOOLS/which animals are currently selected (or via context-menu).**



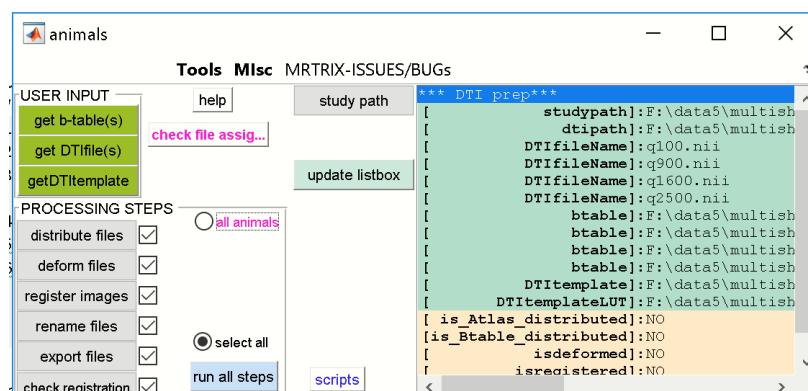
We see that two animals are selected. This corresponds to the animals selected in left listbox in the ANT-GUI.

IMPORTANT

The radio **[all animals]** in the DTIprep GUI deals with the animal selection and has two modes:

- DTIprep is done for **all** animals of the "dat"-folder
- DTIprep is done for **selected animals via ANT-gui animal-listbox**
- .. The default is []!

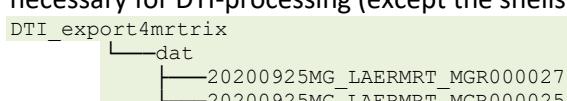
Be carefull!



Verify that the checkboxes of all 6 steps are checked.

Now hit the **[run all steps]**-button.

When finished, the study-folder should contain a new folder "**DTI_export4mrtrix**" with all data necessary for DTI-processing (except the shellscripts).



Each of the exported anima-folder contains the following files (multishell-approach):

'ANO_DTI.nii'	-DTI-atlas, back-transformed to native-space and then transformed to DWI-space, renamed
'ANO_DTI.txt'	-DTI-labels lookup table, renamed
'atlas_lut.txt'	-DTI-labels lookup table (copy of 'ANO_DTI.txt'), renamed

```

'c_t2.nii'           -t2.nii registered to DWI-file space
'rc_ix_AVGTmask.nii' -brain mask, back-transformed to native-space and
                      transformed+resliced to DWI-space
'rc_mt2.nii'         -bias-corrected t2.nii,transformed+resliced to DWI-space
'rc_t2.nii'          -t2.nii,transformed+resliced to DWI-space
'dwi_b100.nii'       -original DWI-file, copied & renamed
'dwi_b1600.nii'      ||
'dwi_b3400.nii'      ||
'dwi_b6000.nii'      ||
'grad_b100.txt'      b-table
'grad_b1600.txt'     ||
'grad_b3400.txt'     ||
'grad_b6000.txt'     ||

```

11) CHECK registration with 1st DWI-file

It is necessary to check that the registration of the images is in line with the DWI-file (1st 4D-DWI-volume: 'dwi_b100.nii'). Task-6 of the DTIprep-command did this, and HTML-files were created in the “checks”-folder. We visually inspect the registration quality and check whether the data are ready for DTIprocessing using MRtrix.

When open the “index.html” (located in the “checks”-folder) in a web-browser we can inspect the coregistration with the 1st DWI-file. Note that the registration of the other DWI-files is done within the MRtrix-pipeline. (Again, that’s the reason why B-tables and DWI/DTIfiles should be ordered from lowest to highest and should match):



OVERLAYS

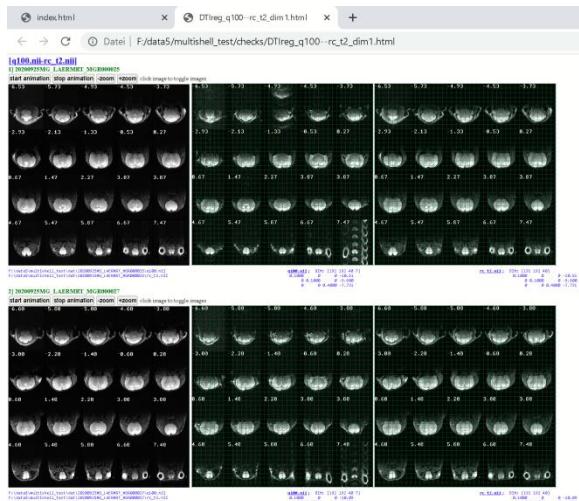
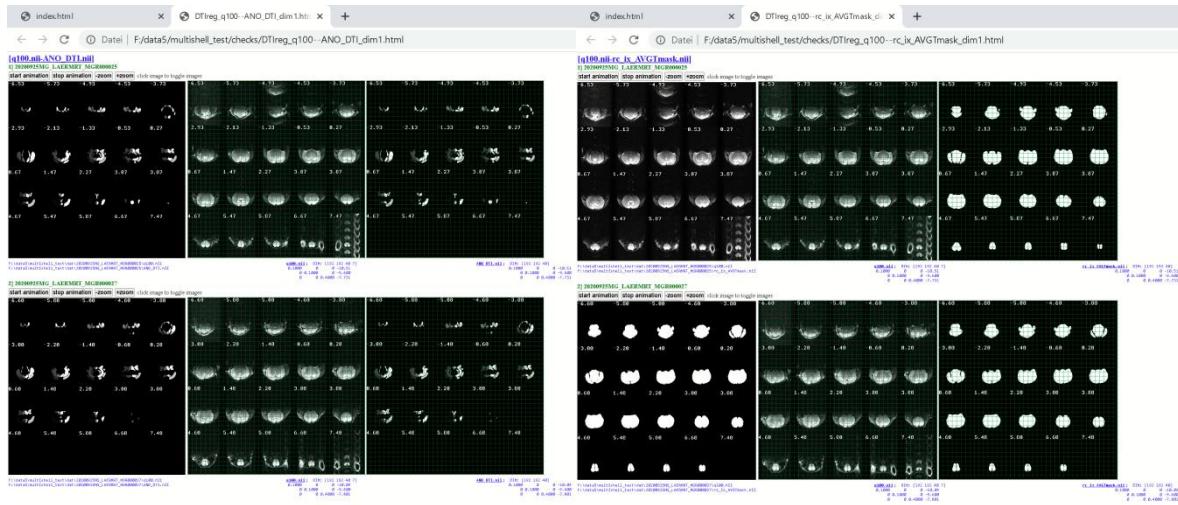
Path: F:\data5\multishell_test\checks

- [DTIreg_q100--ANO_DTI_dim1.html](#) Created: 14-Jun-2022 22:48:04
- [DTIreg_q100--rc_ix_AVGTmask_dim1.html](#) Created: 14-Jun-2022 22:48:02
- [DTIreg_q100--rc_t2_dim1.html](#) Created: 14-Jun-2022 22:48:00

Created: 14-Jun-2022 22:48:03

Click on one of the hyperlinks to check the registration.

Images below: overlay of the 1st DWI-file (“q_b100.nii”) with: “ANO.nii” (left), “rc_ixAVGTmask.nii” (right) and “rc_t2.nii” (bottom). In the browser: click left image to toggle between foreground and background images. Note that the HTML-files are created from the data of the “dat”-folder (not from the export-folder “DTI_export4mrtrix”, because exporting the data is optional). Note that the DWI-files are preserved (fixed image) while all other images (moving images) are registered to the 1st. DWI-file.



Data seems to be ok. DTI-processing via MRtrix can be done!

...End of this tutorial. More progress is hopefully coming soon.