

## **Tutorial: Running ANTx on machines without graphic support “working without graphical user interfaces, GUIs”**

**This tutorial shows how to perform some basic steps without GUIs, for instance when running on a HPC-machine.**

### **CONTENTS**

- 1) **OPTIONAL:** How to set the paths of ELASTIX in UNIX (LINUX)-systems:
- 2) **OPTIONAL:** Open an interactive session on HPC and start Matlab
- 3) **BASICS**
  - ADD ANTx-PATHS
  - GO TO STUDY-FOLDER
  - UPDATE ANT-TOOLBOX
  - CREATE A PROJECT-FILE:
  - LOAD AN ANTx-PROJECT-FILE ("proj.m")
  - CHECK WHETHER THE PROJECT-FILE IS LOADED
- 4) **IMPORT BRUKER RAW-DATA**
- 5) **VISUALIZE FILES AND FOLDERS**
- 6) **SELECTION OF ANIMALS**
- 7) **RENAME FILES**
- 8) **REGISTER “t2.nii” TO TEMPLATE SPACE (STANDARD SPACE)**
- 9) **EXTRACT THE FIRST 3D-VOLUME FROM THE 4D-VOLUME 'dti\_b100.nii'**
- 10) **COREGISTER 'dti\_b100\_1stIMG.nii' TO ‘t2.nii’**
- 11) **TRANSFORM ANOTHER IMAGE TO STANDARD-SPACE**
- 12) **TRANSFORM ANOTHER IMAGE TO NATIVE-SPACE**
- 13) **CHECK REGISTRATION in STANDARD-SPACE - CREATE HTML-FILE**
- 14) **CHECK REGISTRATION in NATIVE-SPACE - CREATE HTML-FILE**
- 15) **REGIONWISE PARAMETER-EXTRACTION**

### **DTI-preprocessing**

- 16) **DTI-preprocessing: Import DTI/DWI-files from Bruker rawdata**
- 17) **DTI-preprocessing: rename DWI-files**
- 18) **DTI-preprocessing: A special DTI-atlas is needed**
- 19) **DTI-preprocessing: Perform DTI-preprocessing**

## 1) OPTIONAL: How to set the paths of ELASTIX in UNIX (LINUX)-systems:

### MAKE ELASTIX RUNNING ON UNIX/LINUX-Systems

```
=====
1) SET PATH OF ELASTIX IN .bashrc-FILE
=====
FOR INFORMATION:
see: Elastix-manual (section 3.2)
see: https://usermanual.wiki/Document/elastix490manual.1389615963/html#pf17

Linux: Add the following lines to your .bashrc file:
-----
export PATH=folder/bin:$PATHexport
LD_LIBRARY_PATH=folder/lib:$LD_LIBRARY_PATH
-----
...where "folder" is the path to the linux-Elastix-folder (which is located in the antx2-
folder): YOUR-DRIVE\antx2\mrtools\elastix\elastix_linux64_v4.7

EXAMPLE:
My linux-elastix-folder is "/sc-projects/sc-proj-agtiermrt/Daten-
2/ressources/antx2/mrtools/elastix/elastix_linux64_v4.7/". Thus, my bashrc-file is modified as
follows:
#-----[ELASTIX-PATH in bashrc]-----
export PATH=/sc-projects/sc-proj-agtiermrt/Daten-
2/ressources/antx2/mrtools/elastix/elastix_linux64_v4.7/bin:$PATH
export LD_LIBRARY_PATH=/sc-projects/sc-proj-agtiermrt/Daten-
2/ressources/antx2/mrtools/elastix/elastix_linux64_v4.7/lib:$LD_LIBRARY_PATH
#-----

=====
2) RELOAD .bashrc-FILE & TEST ELASTIX
=====
- save .bashrc-file, exit editor, then type the following to reload the .bashrc-file again:
source .bashrc

- check installation, by typing:
elastix
  - if successful, a message is displayed:
  Use "elastix --help" for information about elastix-usage.
```

## 2) OPTIONAL: Open interactive session on HPC and start Matlab

### OPEN INTERACTIVE JOB on HPC (optional)

```
srn --time 7-00 --mem=64G --ntasks=8 --pty bash -i
```

-Please check the parameters, here for 7 days, 64 Gb Ram, 8 cores

### LOAD MATLAB-MODULE AND START MATLAB (optional)

```
module load scientific/matlab/R2021b
```

```
matlab
```

-to access Matlab might be different on another machine!

## 3) BASICS

### FROM NOW ON type in the MATLAB CMD-WINDOW...

#### ADD ANTx-PATHS

Go to the ANTx-patd and link all necessary paths using "**antlink**"-command:

```
cd /sc-projects/sc-proj-agtiermrt/Daten-2/ressources/antx2/
```

antlink

## GO TO STUDY-FOLDER

Create an empty study folder (here “groeschel”). The study-folder is the folder where the registration of several animals of a study is performed.

```
cd /sc-projects/sc-proj-agtiermrt/Daten-2/mri/projects/groeschel/
```

## 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)
```

## CREATE AN ANT<sub>x</sub>-PROJECT-FILE:

```
makeproject('projectname',fullfile(pwd,'proj.m'),'voxsize',[.07 .07 .07],'wa_refpath','/sc-projects/sc-proj-agtiermrt/Daten-2/ressources/anttemplates/mouse_Allen2017HikishimaLR','wa_species','mouse');
```

-here the project-file “**proj.m**” is created using a target voxel size of 0.07 x 0.07 x 0.07 mm, the **animal template** is “**mouse Allen2017HikishimaLR**”, with the species ‘mouse’.

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

## LOAD THE PROJECT-FILE “proj.m”

Here we load the created project-file “**proj.m**”:

```
loadconfig(fullfile(pwd,'proj.m'));
```

## CHECK WHETHER THE PROJECT-FILE IS LOADED

```
global an;  
an
```

The global variable “an” (struct) contains the basic information for this study. In the CMD-window you should see the following:

```
project: 'NEW PROJECT'  
datapath: '/sc-projects/sc-proj-agtiermrt/Daten-2/mri/projects/groeschel/dat'  
voxsize: [0.0700 0.0700 0.0700]  
wa: [1x1 struct]
```

## 4) IMPORT BRUKER-DATA

Before doing this step, I just created the ‘raw’-folder within the study’s folder and copied the datasets of two animals into the ‘raw’-folder. First we just read the file-information of the Bruker raw-data from the “raw”-folder (fullfile(pwd,'raw')) which is located in the current study folder.

The file-information will be stored in the resulting w-struct.

```
w=xbrucker2nifti(fullfile(pwd,'raw'),0,[],[],'gui',0,'show',1); % first read all data and show it
```

The w-struct contains the file-information table “d” with header “hd”. This table is listed in the CMD-window when running this command. You could save & reload the struct and import some data later on (advantage: loading time is reduced). To show the table again just type:

```
w.showtable(w) ; %to show the table in CMD-window
```

If graphic is supported you could also visualize the table in an extra window via w.showtable2(w) .

Here, the raw-data folder contains two data-sets (i.e. the data from two animals):

BRUKER DATA									
set	SubjectId	StudNo	ExpNo	PrcNo	MRseq	protocol	sizeMB	date	file
1	20200925MG_LAERMRT_MGR000025	1	1	1	FLASH	01_1_Localizer_CRP	0.393216	20-Oct-2020 15:07:20	/sc-projects/sc-proj-agtiermrt/Daten-2/mri/proje
1	20200925MG_LAERMRT_MGR000025	1	10	1	SINGLEPULSE	02_6_freqAdj_SINGLEPULSE	0.004096	20-Oct-2020 15:08:32	/sc-projects/sc-proj-agtiermrt/Daten-2/mri/proje
1	20200925MG_LAERMRT_MGR000025	1	11	1	DtiEpi	02_7_DTI_EPI_seg_b2500_37dir	112.8666	20-Oct-2020 15:08:12	/sc-projects/sc-proj-agtiermrt/Daten-2/mri/proje
1	20200925MG_LAERMRT_MGR000025	1	11	2	DtiEpi	nan	129.7613	20-Oct-2020 15:06:59	/sc-projects/sc-proj-agtiermrt/Daten-2/mri/proje
1	20200925MG_LAERMRT_MGR000025	1	13	1	STEAM	04_2_Localized_shim_IC_single	0.004096	20-Oct-2020 15:07:11	/sc-projects/sc-proj-agtiermrt/Daten-2/mri/proje
1	20200925MG_LAERMRT_MGR000025	1	16	1	STEAM	04_3_Localized_shim_MGB	0.004096	20-Oct-2020 15:03:38	/sc-projects/sc-proj-agtiermrt/Daten-2/mri/proje
1	20200925MG_LAERMRT_MGR000025	1	17	1	STEAM	04_3_STEAM_1H_MGB	0.004096	20-Oct-2020 15:05:13	/sc-projects/sc-proj-agtiermrt/Daten-2/mri/proje
1	20200925MG_LAERMRT_MGR000025	1	19	1	STEAM	04_2_Localized_shim_IC_single	0.004096	20-Oct-2020 15:08:29	/sc-projects/sc-proj-agtiermrt/Daten-2/mri/proje
1	20200925MG_LAERMRT_MGR000025	1	2	1	FLASH	01_2_Localizer_multi_slice	1.10592	20-Oct-2020 15:09:05	/sc-projects/sc-proj-agtiermrt/Daten-2/mri/proje

```

1 20200925MG_LAERMRT_MGR000025 1 20 1 STEAM 04_2_Localized_shim_IC_single 0.004096 20-Oct-2020 15:05:33 /sc-projects/sc-proj-agtiermrt/Daten-2/mni/proje
1 20200925MG_LAERMRT_MGR000025 1 21 1 STEAM 04_2_Localized_shim_IC_single 0.004096 20-Oct-2020 15:04:55 /sc-projects/sc-proj-agtiermrt/Daten-2/mni/proje
1 20200925MG_LAERMRT_MGR000025 1 22 1 FieldMap B0Map-ADJ_B0MAP 1.048576 20-Oct-2020 15:05:31 /sc-projects/sc-proj-agtiermrt/Daten-2/mni/proje
1 20200925MG_LAERMRT_MGR000025 1 23 1 STEAM 04_2_Localized_shim_IC_single 0.004096 20-Oct-2020 15:05:32 /sc-projects/sc-proj-agtiermrt/Daten-2/mni/proje
1 20200925MG_LAERMRT_MGR000025 1 24 1 FLASH 01_2_Localizer_multi_slice 1.10592 20-Oct-2020 15:03:47 /sc-projects/sc-proj-agtiermrt/Daten-2/mni/proje
1 20200925MG_LAERMRT_MGR000025 1 25 1 RARE 03_T2_TurboRARE_CRP_MapShim 2.04932 20-Oct-2020 15:08:28 /sc-projects/sc-proj-agtiermrt/Daten-2/mni/proje
1 20200925MG_LAERMRT_MGR000025 1 26 1 FieldMap B0Map-ADJ_B0MAP 1.048576 20-Oct-2020 15:03:51 /sc-projects/sc-proj-agtiermrt/Daten-2/mni/proje
1 20200925MG_LAERMRT_MGR000025 1 27 1 STEAM 04_2_Localized_shim_IC_single 0.004096 20-Oct-2020 15:04:48 /sc-projects/sc-proj-agtiermrt/Daten-2/mni/proje
1 20200925MG_LAERMRT_MGR000025 1 28 1 STEAM 04_2_STEAM_IH_IC_single 0.004096 20-Oct-2020 15:05:25 /sc-projects/sc-proj-agtiermrt/Daten-2/mni/proje
1 20200925MG_LAERMRT_MGR000025 1 29 1 STEAM 04_3_Localized_shim_MGB 0.004096 20-Oct-2020 15:03:52 /sc-projects/sc-proj-agtiermrt/Daten-2/mni/proje
1 20200925MG_LAERMRT_MGR000025 1 3 1 RARE 03_T2_TurboRARE_CRP 6.5536 20-Oct-2020 15:05:38 /sc-projects/sc-proj-agtiermrt/Daten-2/mni/proje
1 20200925MG_LAERMRT_MGR000025 1 31 1 STEAM 04_2_Localized_shim_IC_single 0.004096 20-Oct-2020 15:05:39 /sc-projects/sc-proj-agtiermrt/Daten-2/mni/proje
1 20200925MG_LAERMRT_MGR000025 1 33 1 STEAM 04_2_Localized_shim_IC_single 0.004096 20-Oct-2020 15:08:23 /sc-projects/sc-proj-agtiermrt/Daten-2/mni/proje
1 20200925MG_LAERMRT_MGR000025 1 35 1 STEAM 04_2_STEAM_IH_IC_single 0.004096 20-Oct-2020 15:07:33 /sc-projects/sc-proj-agtiermrt/Daten-2/mni/proje
1 20200925MG_LAERMRT_MGR000025 1 4 1 DTIEPI 02_1_DTI_EPI_seg_b100_6dir 20.64384 20-Oct-2020 15:08:11 /sc-projects/sc-proj-agtiermrt/Daten-2/mni/proje
1 20200925MG_LAERMRT_MGR000025 1 4 2 DTIEPI nan 129.7613 20-Oct-2020 15:08:24 /sc-projects/sc-proj-agtiermrt/Daten-2/mni/proje
1 20200925MG_LAERMRT_MGR000025 1 5 1 FieldMap B0Map-ADJ_B0MAP 1.048576 20-Oct-2020 15:03:53 /sc-projects/sc-proj-agtiermrt/Daten-2/mni/proje
1 20200925MG_LAERMRT_MGR000025 1 6 1 SINGLEPULSE 02_2_freqAdj_SINGLEPULSE 0.004096 20-Oct-2020 15:05:00 /sc-projects/sc-proj-agtiermrt/Daten-2/mni/proje
1 20200925MG_LAERMRT_MGR000025 1 7 1 DTIEPI 02_3_DTI_EPI_seg_b900_13dir 41.28768 20-Oct-2020 15:08:40 /sc-projects/sc-proj-agtiermrt/Daten-2/mni/proje
1 20200925MG_LAERMRT_MGR000025 1 7 2 DTIEPI nan 129.7613 20-Oct-2020 15:09:00 /sc-projects/sc-proj-agtiermrt/Daten-2/mni/proje
1 20200925MG_LAERMRT_MGR000025 1 8 1 SINGLEPULSE 02_4_freqAdj_SINGLEPULSE 0.004096 20-Oct-2020 15:08:25 /sc-projects/sc-proj-agtiermrt/Daten-2/mni/proje
1 20200925MG_LAERMRT_MGR000025 1 9 1 DTIEPI 02_5_DTI_EPI_seg_b1600_25dir 76.67712 20-Oct-2020 15:04:13 /sc-projects/sc-proj-agtiermrt/Daten-2/mni/proje
1 20200925MG_LAERMRT_MGR000025 1 9 2 DTIEPI nan 129.7613 20-Oct-2020 15:04:41 /sc-projects/sc-proj-agtiermrt/Daten-2/mni/proje
1 20200925MG_LAERMRT_MGR000027 1 1 1 FLASH 01_1_Localizer_CRP 0.393216 20-Oct-2020 14:57:50 /sc-projects/sc-proj-agtiermrt/Daten-2/mni/proje
2 20200925MG_LAERMRT_MGR000027 1 10 1 DTIEPI 02_7_DTI_EPI_seg_b2500_37dir 112.8666 20-Oct-2020 14:56:04 /sc-projects/sc-proj-agtiermrt/Daten-2/mni/proje
2 20200925MG_LAERMRT_MGR000027 1 10 2 DTIEPI nan 129.7613 20-Oct-2020 14:56:29 /sc-projects/sc-proj-agtiermrt/Daten-2/mni/proje
2 20200925MG_LAERMRT_MGR000027 1 11 1 STEAM 04_2_Localized_shim_IC_single 0.004096 20-Oct-2020 14:57:36 /sc-projects/sc-proj-agtiermrt/Daten-2/mni/proje
2 20200925MG_LAERMRT_MGR000027 1 12 1 STEAM 04_2_STEAM_IH_IC_single 0.004096 20-Oct-2020 14:54:36 /sc-projects/sc-proj-agtiermrt/Daten-2/mni/proje
2 20200925MG_LAERMRT_MGR000027 1 13 1 STEAM 04_3_Localized_shim_MGB 0.004096 20-Oct-2020 14:57:43 /sc-projects/sc-proj-agtiermrt/Daten-2/mni/proje
2 20200925MG_LAERMRT_MGR000027 1 14 1 STEAM 04_3_STEAM_IH_MGB 0.004096 20-Oct-2020 14:57:16 /sc-projects/sc-proj-agtiermrt/Daten-2/mni/proje
2 20200925MG_LAERMRT_MGR000027 1 15 1 FieldMap B0Map-ADJ_B0MAP 1.048576 20-Oct-2020 14:59:45 /sc-projects/sc-proj-agtiermrt/Daten-2/mni/proje
2 20200925MG_LAERMRT_MGR000027 1 16 1 STEAM 04_2_Localized_shim_IC_single 0.004096 20-Oct-2020 14:54:32 /sc-projects/sc-proj-agtiermrt/Daten-2/mni/proje
2 20200925MG_LAERMRT_MGR000027 1 17 1 STEAM 04_2_Localized_shim_IC_single 0.004096 20-Oct-2020 14:55:28 /sc-projects/sc-proj-agtiermrt/Daten-2/mni/proje
2 20200925MG_LAERMRT_MGR000027 1 18 1 STEAM 04_2_Localized_shim_IC_single 0.004096 20-Oct-2020 14:56:48 /sc-projects/sc-proj-agtiermrt/Daten-2/mni/proje
2 20200925MG_LAERMRT_MGR000027 1 19 1 STEAM 04_2_Localized_shim_IC_single 0.004096 20-Oct-2020 14:59:15 /sc-projects/sc-proj-agtiermrt/Daten-2/mni/proje
2 20200925MG_LAERMRT_MGR000027 1 2 1 FLASH 01_2_Localizer_multi_slice 1.10592 20-Oct-2020 14:59:33 /sc-projects/sc-proj-agtiermrt/Daten-2/mni/proje
2 20200925MG_LAERMRT_MGR000027 1 20 1 STEAM 04_2_Localized_shim_IC_single 0.004096 20-Oct-2020 14:56:38 /sc-projects/sc-proj-agtiermrt/Daten-2/mni/proje
2 20200925MG_LAERMRT_MGR000027 1 21 1 STEAM 04_2_Localized_shim_IC_single 0.004096 20-Oct-2020 14:54:36 /sc-projects/sc-proj-agtiermrt/Daten-2/mni/proje
2 20200925MG_LAERMRT_MGR000027 1 23 1 STEAM 04_2_Localized_shim_IC_single 0.004096 20-Oct-2020 14:55:32 /sc-projects/sc-proj-agtiermrt/Daten-2/mni/proje
2 20200925MG_LAERMRT_MGR000027 1 3 1 RARE 03_T2_TurboRARE_CRP 6.5536 20-Oct-2020 14:57:06 /sc-projects/sc-proj-agtiermrt/Daten-2/mni/proje
2 20200925MG_LAERMRT_MGR000027 1 4 1 DTIEPI 02_1_DTI_EPI_seg_b100_6dir 20.64384 20-Oct-2020 14:58:07 /sc-projects/sc-proj-agtiermrt/Daten-2/mni/proje
2 20200925MG_LAERMRT_MGR000027 1 4 2 DTIEPI nan 129.7613 20-Oct-2020 14:58:30 /sc-projects/sc-proj-agtiermrt/Daten-2/mni/proje
2 20200925MG_LAERMRT_MGR000027 1 5 1 SINGLEPULSE 02_2_freqAdj_SINGLEPULSE 0.004096 20-Oct-2020 14:54:33 /sc-projects/sc-proj-agtiermrt/Daten-2/mni/proje
2 20200925MG_LAERMRT_MGR000027 1 6 1 DTIEPI 02_3_DTI_EPI_seg_b900_13dir 41.28768 20-Oct-2020 14:54:49 /sc-projects/sc-proj-agtiermrt/Daten-2/mni/proje
2 20200925MG_LAERMRT_MGR000027 1 6 2 DTIEPI nan 129.7613 20-Oct-2020 14:55:18 /sc-projects/sc-proj-agtiermrt/Daten-2/mni/proje
2 20200925MG_LAERMRT_MGR000027 1 7 1 SINGLEPULSE 02_4_freqAdj_SINGLEPULSE 0.004096 20-Oct-2020 14:59:22 /sc-projects/sc-proj-agtiermrt/Daten-2/mni/proje
2 20200925MG_LAERMRT_MGR000027 1 8 1 DTIEPI 02_5_DTI_EPI_seg_b1600_25dir 76.67712 20-Oct-2020 14:58:47 /sc-projects/sc-proj-agtiermrt/Daten-2/mni/proje
2 20200925MG_LAERMRT_MGR000027 1 8 2 DTIEPI nan 129.7613 20-Oct-2020 14:59:07 /sc-projects/sc-proj-agtiermrt/Daten-2/mni/proje
2 20200925MG_LAERMRT_MGR000027 1 9 1 SINGLEPULSE 02_6_freqAdj_SINGLEPULSE 0.004096 20-Oct-2020 14:54:33 /sc-projects/sc-proj-agtiermrt/Daten-2/mni/proje

```

Here we want to import the turboRARE-image (“**03 T2 TurboRARE CRP.nii**”) and the image “**DTI EPI seg b100 6dir**”.

To visualize the filtered table run the following command (note that the w-struct is used as 1<sup>st</sup> argument):

```
w2=xbrucker2nifti(w,0,[],[], 'gui',0,'show',1,'fit',{'pro','03_T2_TurboRARE_CRP|EPI_seg_b100'} );
```

the table now looks as follows:

BRUKER DATA									
set	SubjectId	StudNo	ExpNo	PrcNo	MRseq	protocol	sizeMB	date	file
1	20200925MG_LAERMRT_MGR000025	1	3	1	RARE	03_T2_TurboRARE_CRP	6.5536	20-Oct-2020 15:05:38	/sc-projects/sc-
1	20200925MG_LAERMRT_MGR000025	1	4	1	DTIEPI	02_1_DTI_EPI_seg_b100_6dir	20.64384	20-Oct-2020 15:08:11	/sc-projects/sc-
2	20200925MG_LAERMRT_MGR000027	1	5	1	RARE	03_T2_TurboRARE_CRP	6.5536	20-Oct-2020 14:57:06	/sc-projects/sc-
2	20200925MG_LAERMRT_MGR000027	1	4	1	DTIEPI	02_1_DTI_EPI_seg_b100_6dir	20.64384	20-Oct-2020 14:58:07	/sc-projects/sc-

Now let’s import these data. For this just set the ‘show’-parameter to 0:

```
w2=xbrucker2nifti(w,0,[],[], 'gui',0,'show',0,'fit',{'pro','03_T2_TurboRARE_CRP|EPI_seg_b100'} );
```

## 5) VISUALIZE FILES AND FOLDERS

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

dispfiles

The following is displayed in the CMD-window:

FILE x FOLDER			
	20200925MG_LAERMRT_MGR000025	20200925MG_LAERMRT_MGR000027	counts
02_1_DTI_EPI_seg_b100_6dir_1.nii	+	+	2/2
03_T2_TurboRARE_CRP_1.nii	+	+	2/2
counts	2/2	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 two imported files (“**02 1 DTI EPI seg b100 6dir 1.nii**” and

“**03 T2 TurboRARE CRP 1.nii**”).

## 6) SELECTION OF ANIMALS

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

Here we will select all animals:

```
mdirs=antcb('getallsubjects')
```

Note that the variable “**mdirs**” contains the full paths names of the selected animals:

```
{'/sc-projects/sc-proj-agtiermrt/Daten-2/mni/projects/groeschel/dat/20200925MG_LAERMRT_MGR000025'}
{'/sc-projects/sc-proj-agtiermrt/Daten-2/mni/projects/groeschel/dat/20200925MG_LAERMRT_MGR000027'}
```

Alternative selection of all animals:

```
mdirs =antcb('selectdirs',[1:2 ])
```

```
mdirs=antcb('selectdirs','all')
```

## 7) RENAME FILES

Here we 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'!

The renaming of the file '**03\_T2\_TurboRARE\_CRP\_1.nii**' is just because I hate long names.

Let's rename '03\_T2\_TurboRARE\_CRP\_1.nii' to t2.nii':

```
xrename(0,'03_T2_TurboRARE_CRP_1.nii','t2.nii',':', 'dirs',mdirs);
```

and rename '02\_1\_DTI\_EPI\_seg\_b100\_6dir\_1.nii' to 'dti\_b100.nii':

```
xrename(0,'02_1_DTI_EPI_seg_b100_6dir_1.nii','dti_b100.nii',':', 'dirs',mdirs);
```

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 (:); the 4<sup>th</sup> input arg).

Alternatively, to rename the original files just keep the 4<sup>th</sup> arg empty.

Now, we check whether the new files exist via dispfiles:

dispfiles

FILE	x	FOLDER			
-----					
		20200925MG_LAERMRT_MGR0000025	20200925MG_LAERMRT_MGR0000027	counts	
=====					
02_1_DTI_EPI_seg_b100_6dir_1.nii	+			2/2	
03_T2_TurboRARE_CRP_1.nii	+			2/2	
dti_b100.nii	+			2/2	
t2.nii	+			2/2	
counts		4/4	4/4		

## 8) REGISTER "t2.nii" TO TEMPLATE SPACE (STANDARD SPACE)

Registration of **"t2.nii"** to standard space is done in 4 steps: [1] initialization, [2] rough 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. Here the rough registration is done automatically (autoreg', 1). We also use parallel processing across animals ('parfor',1); Please check the memory and number of cores when using parallel processing:

```
xwarp3('batch','task',[1:4 ],'autoreg',1,'parfor',1, 'mdirs',mdirs(:) );
```

Now let's check whether the **"t2.nii"** is transformed to standard-space (new name is: **"x t2.nii"**) and the template (**"AVGT.nii"**) and atlas (**"ANO.nii"**) is back-transformed to native-space (new names:

**"ix ANO.nii"** & **"ix AVGT.nii"**) via:

```
dispfiles('flt','^x.*.nii|^ix.*.nii');
```

FILE	x	FOLDER			
-----					
		20200925MG_LAERMRT_MGR0000025	20200925MG_LAERMRT_MGR0000027	counts	
=====					
ix_ANO.nii	+			2/2	
ix_AVGT.nii	+			2/2	
x_t2.nii	+			2/2	
counts		3/3	3/3		

Here we see that all three files were created for each data set

## 9) Extract the first 3d-volume from the 4D-vlume 'dti\_b100.nii'

Use the following command to extract the 1<sup>st</sup> volume (4<sup>th</sup> input arg: 1) of **'dti\_b100.nii'** and save it as **'dti\_b100\_1stIMG.nii'** for all animals (mdirs):

```
xrename(0,'dti_b100.nii','dti_b100_1stIMG.nii','1','dirs', mdirs );
```

Again, check the existence of the new files via:

```
dispfiles('flt','^dt')
```

FILE	x	FOLDER			
-----					
		20200925MG_LAERMRT_MGR0000025	20200925MG_LAERMRT_MGR0000027	counts	
=====					
dti_b100.nii	+			2/2	
dti_b100_1stIMG.nii	+			2/2	
counts		2/2	2/2		

## 10) COREGISTER 'dti\_b100\_1stIMG.nii' to 't2.nii'

Now let's coregister the file **'dti\_b100\_1stIMG.nii'** (moving image) to the **'t2.nii'**-image (fixed image):

```
z=[];
```

```
z.TASK={ '[2]' };
```

```
;% flag 2 (or '2') signals to perform coregistration via SPM
```

```
z.targetImg1={ 't2.nii' };
```

```
;% TARGET-IMAGE
```

```

z.sourceImg1={ 'dti_b100_1stIMG.nii' };;% SOURCE-IMAGE
z.sourceImgNum1=[1];;% IN CASE OF 4D-vol use 1st 3d-volume of SOURCE
z.applyImg1= " ";;% HERE, THE TRAFO IS NOT APPLIED TO OTHER IMAGES
z.cost_fun='nmi';
z.sep=[7 2 1 0.5 0.1 0.05];
z.tol=[0.01 0.01 0.01 0.001 0.001 0.001];
z.fwhm=[4 4];
z.centering=[0];
z.reslicing=[1];;% OUTPUT-IMAGE IS RESLICED TO MATCH WITH TARGET
z.interpOrder='auto';
z.prefix='r3';;% OUTPUT FILE-PREFIX
z.warping=[0];;% WARPING IS "OFF"
z.isparallel=1;;% PARALLEL PROCESSING ENABLED
xcoreg(0,z,mdirs);

```

Again, check existence of the registered file:

```
dispfiles('flt','^r3')
```

```

FILE x FOLDER
-----
20200925MG_LAERMRT_MGR000025 20200925MG_LAERMRT_MGR000027 counts
=====
r3c_dti_b100_1stIMG.nii + + 2/2
r3c_t2.nii + + 2/2
counts 2/2 2/2
-----

```

Here we see that the file **'r3c dti b100 1stIMG.nii'** exists for each animal. The file **"r3c t2.nii"** is just a copy of the unchanged target-file ('t2.nii', fixed image).

## 11) TRANSFORM ANOTHER IMAGE TO STANDARD-SPACE

Now, we want to transform the image **"r3c dti b100 1stIMG.nii"** to standard-space (1<sup>st</sup> arg: 1) for all selected animals (mdirs), using b-spline interpolation (4<sup>th</sup> arg: 4), using the local reorientation information estimated from the rough rigid registration step (5<sup>th</sup> arg: 'local') and indicate that the input-file is located in the animal folder (6<sup>th</sup> arg: struct('source','intern')). This will create the file **"x r3c dti b100 1stIMG.nii"** in standard-space:

```
fis=doelastix(1,mdirs,{'r3c_dti_b100_1stIMG.nii'},4,'local',struct('source','intern'));
```

Let's check the existence of the file **"x r3c dti b100 1stIMG.nii"**:

```
dispfiles('flt','x_.*')
```

```

FILE x FOLDER
-----
20200925MG_LAERMRT_MGR000025 20200925MG_LAERMRT_MGR000027 counts
=====
ix_AND.nii + + 2/2
ix_AVGT.nii + + 2/2
ix_AVGThemi.nii + + 2/2
x_r3c_dti_b100_1stIMG.nii + + 2/2
x_t2.nii + + 2/2
counts 5/5 5/5
-----

```

## 12) TRANSFORM ANOTHER IMAGE TO NATIVE-SPACE

In the same way we could also transform an image from standard-space to the the template space (for instance the Atlas). Here we transform the template's hemispheric mask (**'AVGThemi.nii'**) to native space (new name: **'ix AVGThemi.nii'**). For transformation to native space the 1<sup>st</sup> arg is -1, we use NN-interpolation to preserve numbers/hemispheric-IDs (4<sup>th</sup> arg: 0):

```
fis2=doelastix(-1,mdirs,{'AVGThemi.nii'},0,'local',struct('source','intern'));
```

Let's check the existence of the file **'ix AVGThemi.nii'**:

```
dispfiles('flt','ix_.*')
```

```

FILE x FOLDER
-----
20200925MG_LAERMRT_MGR000025 20200925MG_LAERMRT_MGR000027 counts
=====
ix_AND.nii + + 2/2
ix_AVGT.nii + + 2/2
ix_AVGThemi.nii + + 2/2
counts 3/3 3/3
-----

```

## 13) CHECK REGISTRATION in STANDARD-SPACE - CREATE HTML-FILE



Next, we will create an HTML-file to visualize the overlay of the template **“AVGT.nii”** and **“x\_t2.nii”** (animal in standard-space). The 3<sup>rd</sup> arg defines the output-folder, the 4<sup>th</sup> input, defines the size of the images and the number of slices to visualize:

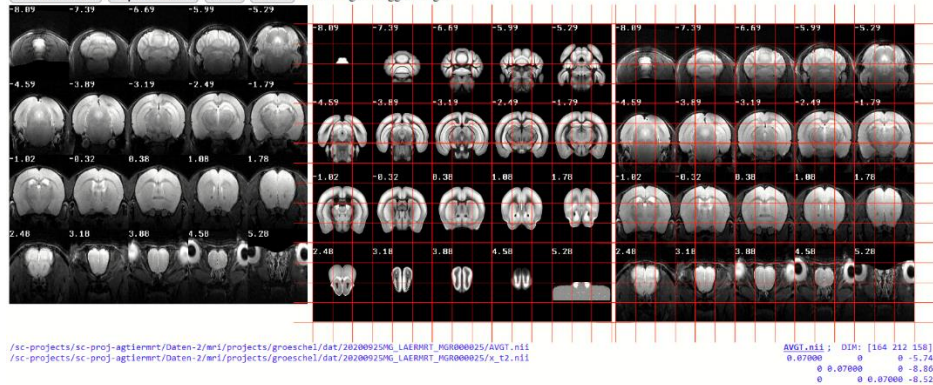
```
checkreghtml(mdirs,{'AVGT.nii','x_t2.nii'},fullfile(pwd,'checks'),struct('size',300,'slices','n20'))
```

Now you can visualize the registration with your Web-browser. The 1<sup>st</sup> image is an animated gif, showing the overlay of the **‘AVGT.nii’** and **‘x\_t2.nii’** images. Click onto the image to toggle between the two images, or hit ‘start animation’-button to see the animated gif. The 2<sup>nd</sup> (‘AVGT.nii’) and 3<sup>rd</sup> image (‘x\_t2.nii’) shows the two images side-by-side.

[\[AVGT.nii-x\\_t2.nii\]](#)

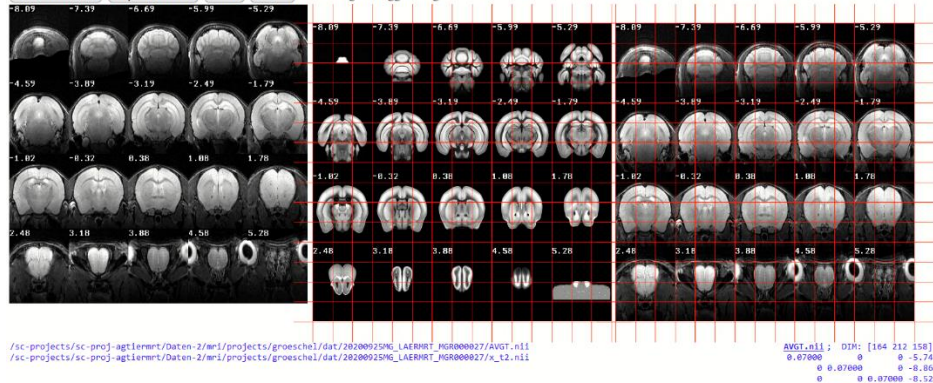
1] 20200925MG\_LAERMRT\_MGR000025

start animation stop animation -zoom +zoom click image to toggle images



2] 20200925MG\_LAERMRT\_MGR000027

start animation stop animation -zoom +zoom click image to toggle images

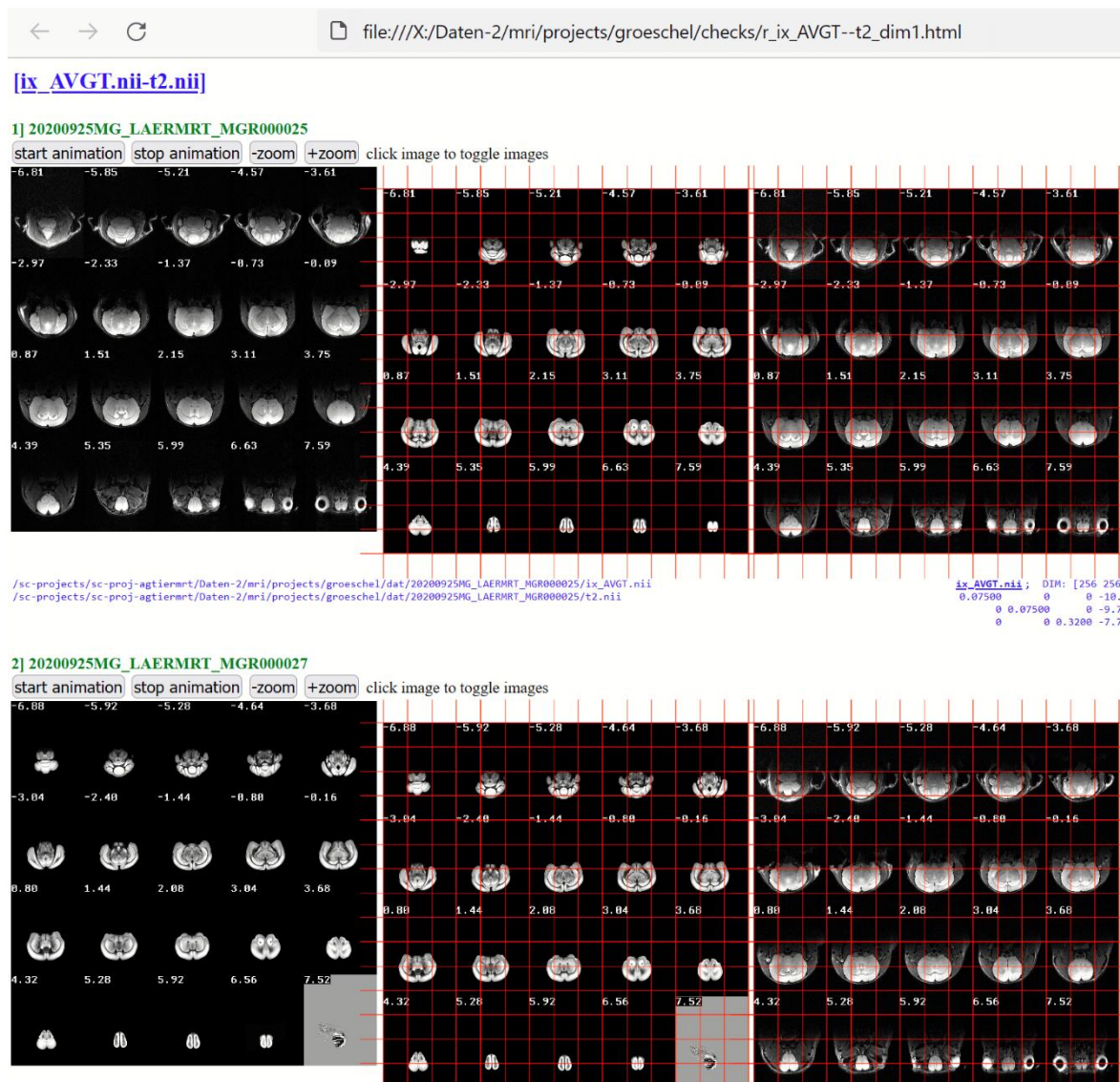


#### 14) CHECK REGISTRATION in NATIVE-SPACE - CREATE HTML-FILE

In the same, we can create an overlay of images in native-space. Here we compare the images **“t2.nii”** and **“ix\_AVGT.nii”** for all animals and save the output as HTML-file.

Note that the slicing is now done from the 1<sup>st</sup>-dimension (‘dim’: 1)

```
checkreghtml(mdirs,{'ix_AVGT.nii','t2.nii'},fullfile(pwd,'checks'),struct('size',300,'slices','n20','dim',1))
```



## 15) Regionwise parameter-extraction

We now extract parameters such as volume, mean, median etc. for each region using the image "t2.nii" and the standard atlas (z.atlas= 'ANO.nii'). Here, from the image in native space (z.space= 'native') the parameters are aggregated over the left and right hemisphere (z.hemisphere= 'both'). The resulting Excelfile 'regwise\_t2' is stored in the 'results'-folder (subfolder of the study-folder).

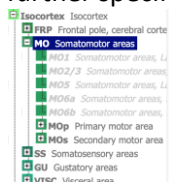
```
z=[];
z.files      = 't2.nii';      % file used for parameter extraction
z.atlas      = 'ANO.nii';     % selected atlas name, atlas has to be the standard space atlas name
z.space      = 'native';     % use images from "native" space
z.hemisphere = 'both';       % hemisphere used: [left,right or both]
z.fileNameOut = 'regwise_t2'; % <optional> name of the output-file.
xgetlabels4(0,z,mdirs);
```

The resulting excel-file contains a sheet for each parameter. Here see the volume-sheet ("vol") with region-wise volume in mm3 for each region (rows) and each animal (columns). See "info"-sheet & xgetlabels4.m for more information:



1	region	20200925MG_LAERMRT_MGR000025	20200925MG_LAERMRT_MGR000027
2	root	494.8288316	483.2800356
3	Basic cell groups and regions	439.7092504	430.7632534
4	Cerebrum	261.3599111	260.6777113
5	Cerebral cortex	209.5541287	207.7019293
6	Cortical plate	200.5649318	198.6353324
7	Isocortex	112.9841616	113.9579612
8	Frontal pole cerebral cortex	0.883799699	0.854999709
9	Frontal pole layer 1	0.131399955	0.17639994
10	Frontal pole layer 2/3	0.363599876	0.280799904
11	Frontal pole layer 5	0.313199893	0.320399891
12	Frontal pole layer 6a	0.075599974	0.077399974
13	Frontal pole layer 6b	0	0
14	Somatomotor areas	23.08859214	22.43699237
15	Somatomotor areas Layer 1	0	0
16	Somatomotor areas Layer 2/3	0	0
17	Somatomotor areas Layer 5	0	0
18	Somatomotor areas Layer 6a	0	0
19	Somatomotor areas Layer 6b	0	0
20	Primary motor area	10.50299643	10.2779965
21	Primary motor area Layer 1	1.349999541	1.274399566
22	Primary motor area Layer 2/3	3.648598759	3.639598762

Note: Zero volume entries as for the “somatomotor areas” is because the region-IDs are not defined/exist in the Allen brain Atlas (see also: <http://atlas.brain-map.org/atlas#atlas=1&plate=100960428&structure=500&x=5245.5&y=3833.5&zoom=-4&resolution=33.45&z=6>) Here we see that the layers of “somatomotor areas” are grayed, i.e. not further specified



Of course you could extract parameters from images in standard space as well:

Example: Just change the following from the above example:

z.files = 'x\_t2.nii'; i.e. the t2w-image in standard space is use  
z.space = 'standard'; this indicated that the image is in standard space  
z.fileNameOut = 'regwise\_x\_t2' just another name for the resulting Excelfile:

#### EXAMPLE: PARAMTER EXTRACTION FROM STANDARD SPACE

```
z=[];
z.files = 'x_t2.nii'; % file used for parameter extraction
z.atlas = 'ANO.nii'; % selected atlas name, atlas has to be the standard space atlas name
z.space = 'standard'; % use images from "standard" space
z.hemisphere = 'both'; % hemisphere used: [left,right or both]
z.fileNameOut = 'regwise_x_t2' ; % <optional> name of the output-file.
xgetlabels4(0,z,mdirs);
```

## DTI-preprocessing

In the upcoming steps we want to prepare the data for the DTI-pipeline, 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.

### 16) DTI-preprocessing: Import DTI/DWI-files from Bruker rawdata

For the preprocessing we first have to import the DWI-files.

First load the struct with the available files from the raw data-folder:

```
w=xbruker2nifti(fullfile(pwd,'raw'),0,[],[],'gui',0,'show',1);
```

BRUKER DATA							
set	SubjectId	StudNo	ExpNo	PrcNo	MRseq	protocol	size..
1..							
1	20200925MG_LAERMRT_MGR000025	1	7	1	DtiEpi	<b>02_3_DTI_EPI_seg_b900_13dir</b>	41...
1	20200925MG_LAERMRT_MGR000025	1	8	1	SINGLEPULSE	02_4_freqAdj_SINGLEPULSE	0.0..
1	20200925MG_LAERMRT_MGR000025	1	9	1	<b>DtiEpi</b>	<b>02_5_DTI_EPI_seg_b1600_25dir</b>	76...
..							
2	20200925MG_LAERMRT_MGR000027	1	5	1	SINGLEPULSE	02_2_freqAdj_SINGLEPULSE	0.0..
2	20200925MG_LAERMRT_MGR000027	1	6	1	DtiEpi	<b>02_3_DTI_EPI_seg_b900_13dir</b>	41...
2	20200925MG_LAERMRT_MGR000027	1	6	2	DtiEpi	nan	129..
..							
2	20200925MG_LAERMRT_MGR000027	1	8	1	<b>DtiEpi</b>	<b>02_5_DTI_EPI_seg_b1600_25dir</b>	76...
2	20200925MG_LAERMRT_MGR000027	1	9	1	SINGLEPULSE	02_6_freqAdj_SINGLEPULSE	0.0..
..							

The cmd-window displays all available files. Here we are interested in the files which protocol-name contains the string “DTI\_EPI\_seg\_” (bold) . To filter those files we use:

```
w2=xbruker2nifti(w,0,[],[],'gui',0,'show',1,'flt',{'pro','EPI_seg_'});
```

BRUKER DATA							
set	SubjectId	StudNo	ExpNo	PrcNo	MRseq	protocol	sizeMB
1	20200925MG_LAERMRT_MGR000025	1	11	1	DtiEpi	02_7_DTI_EPI_seg_b2500_37dir	112.0666
1	20200925MG_LAERMRT_MGR000025	1	4	1	DtiEpi	02_1_DTI_EPI_seg_b100_6dir	20.64384
1	20200925MG_LAERMRT_MGR000025	1	7	1	DtiEpi	02_3_DTI_EPI_seg_b900_13dir	41.28768
1	20200925MG_LAERMRT_MGR000025	1	9	1	DtiEpi	02_5_DTI_EPI_seg_b1600_25dir	76.67712
2	20200925MG_LAERMRT_MGR000027	1	10	1	DtiEpi	02_7_DTI_EPI_seg_b2500_37dir	112.0666
2	20200925MG_LAERMRT_MGR000027	1	4	1	DtiEpi	02_1_DTI_EPI_seg_b100_6dir	20.64384
2	20200925MG_LAERMRT_MGR000027	1	6	1	DtiEpi	02_3_DTI_EPI_seg_b900_13dir	41.28768
2	20200925MG_LAERMRT_MGR000027	1	8	1	DtiEpi	02_5_DTI_EPI_seg_b1600_25dir	76.67712

Now let's import those files by setting the 'show'-flag to 0:

```
w2=xbruker2nifti(w,0,[],[],'gui',0,'show',0,'flt',{'pro','EPI_seg_'});
```

Use dispfiles to check whether the data were imported for our two animals:

```
dispfiles('flt','EPI_seg');
```

FILE x FOLDER			
	counts	20200925MG_LAERMRT_MGR000025	20200925MG_LAERMRT_MGR0000
	=====	=====	=====
counts	4/4		4/4
	=====	=====	=====
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	+	+

Here we see that for the two animals 4 DWI-files were imported

### 17) DTI-preprocessing: rename DWI-files

First select all animal-dirs. Let's make a copy of the DWI-files files and renames them to make it easier. Note that the colon-sign (:) as 4<sup>th</sup> input arg of in **xrename** result in a copy of the files (If you want to rename the original file just keep the 4<sup>th</sup> arg empty).

```
mdirs=antcb('selectdirs','all')
xrename(0,'02_1_DTI_EPI_seg_b100_6dir_1.nii','q_b100.nii',':','dirs',mdirs);
xrename(0,'02_3_DTI_EPI_seg_b900_13dir_1.nii','q_b900.nii',':','dirs',mdirs);
xrename(0,'02_5_DTI_EPI_seg_b1600_25dir_1.nii','q_b1600.nii',':','dirs',mdirs);
xrename(0,'02_7_DTI_EPI_seg_b2500_37dir_1.nii','q_b2500.nii',':','dirs',mdirs);
```

Let's check files via dispfiles:

```
dispfiles('flt','q_b');
```

FILE x FOLDER	
	counts 20200925MG_LAERMRT_MGR000025 20200925MG_LAERMRT_MGR000027
	=====
counts	4/4 4/4
	=====
q_b100.nii 2/2	+
q_b1600.nii 2/2	+
q_b2500.nii 2/2	+
q_b900.nii 2/2	+

Here we see that for the two animals copies of the 4 DWI-files were created with simpler filenames.

### 18) DTI-preprocessing: 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

For sake of simplicity, a new folder 'atl\_auditsys\_08dec20\_v1' was created in the study-folder which contains these two files (Note that these files 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 splitted region for left and right hemisphere)
- ... see help of **xexcel2atlas.m**.

### 19) DTI-preprocessing: Perform DTI-preprocessing

The registration of "t2.nii" to standard-space is mandatory before performing the DTI-preprocessing step. Luckily, we have already done this in step-8 (REGISTER "t2.nii" TO TEMPLATE SPACE (STANDARD SPACE)). The DTI-preprocessing steps contains 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 DTI-atlas (d.DTItemplate & d.DTItemplateLUT) (.. those from above...:-)
- assign DWI/DTI-files (d.DTIfileName)

#### (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 (names are fixed and names are expected from shellscripts)
- 5 export files -> export files to "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');
```

Now we prepare the data and assign the following variables:

- (1) assign the fullpath name of one Bruker rawdata set. This is necessary to obtain the b-tables (**d.brukerdata**).
  - (2) assign the fullpath name of the DTI-atlas (NIFTI-file, **d.DTItemplate**)
  - (3) assign the fullpath name of the lutfile (textfile with region-IDs, **d.DTItemplateLUT**)
  - (4) assign the short names of the DWI-files (files should now exist in the animal-dirs (**d.DTIfileName**)).
- Please preserve the order of the DWI-files (100,900,1600,2500).

```
clear d;  
d.brukerdata = fullfile(pwd,'raw','20200925_092655_20200925MG_LAERMRT_MGR000025_1_1');  
d.DTItemplate = fullfile(pwd,'atl_auditsys_08dec20_v1','atl_auditsys_08dec20.nii');  
d.DTItemplateLUT =fullfile(pwd,'atl_auditsys_08dec20_v1','atl_auditsys_08dec20_INFO.txt');  
d.DTIfileName={'q_b100.nii' 'q_b900.nii' 'q_b1600.nii' 'q_b2500.nii'};
```

now run the initialization step:

```
DTIprep('ini',d); % initialize DTIprep-struct and fill requested parameters
```

The cmd-window displays the content of the DTIprep-struct:

```
*** [DTIprep-info saved]: /sc-projects/sc-proj-agttermrt/Daten-2/mri/projects/groeschel/DTI/check.mat
-----[struct info]-----
    studypath: '/sc-projects/sc-proj-agttermrt/Daten-2/mri/projects/groeschel/'
    dtipath: '/sc-projects/sc-proj-agttermrt/Daten-2/mri/projects/groeschel/DTI'
    brukerdata: '/sc-projects/sc-proj-agttermrt/Daten-2/mri/projects/groeschel/raw/20200925_092655_20200925MG_LAERMRT_MGR000025_1_1'
    btable: []
    DTIfileName: {4x1 cell}
    DTItemplate: '/sc-projects/sc-proj-agttermrt/Daten-2/mri/projects/groeschel/atl_auditsys_08dec20_v1/atl_auditsys_08dec20.nii'
    DTItemplateLUT: '/sc-projects/sc-proj-agttermrt/Daten-2/mri/projects/groeschel/atl_auditsys_08dec20_v1/atl_auditsys_08dec20_INFO.txt'
    INFO_ini: 'INI'
    is_Brukerdata_Defined: 'YES'
    is_DTItemplate_Defined: 'YES'
    is_DTItemplateLUT_Defined: 'YES'
    INFO_import: 'IMPORT'
    is_Btable_imported: 'NO'
    is_DTItemplate_imported: 'NO'
    INFO_task: 'TASK'
    is_Atlas_and_Btable_distributed: 'NO'
    isdeformed: 'NO'
    isregistered: 'NO'
    isrenamed: 'NO'
-----[btable]-----
...not imported yet!
-----[DTIfileName]-----
    {'q_b100.nii' }
    {'q_b900.nii' }
    {'q_b1600.nii' }
    {'q_b2500.nii' }
```

...with "YES" for all "initialization" steps.

Now, let's **import** the B-tables from the raw-data and the DTI-atlas (DTItemplate/lutfile)

```
DTIprep('import'); % import b-table and DTItemplate/DTItemplateLUT
```

The cmd-window should display the found B-tables (source/output/number of aqu. directions):

```
>> DTIprep('import'); % import b-table and DTItemplate/DTItemplateLUT
1b-table "100" stored as //sc-projects/sc-proj-agttermrt/Daten-2/mri/projects/groeschel/DTI/grad_b100.txt
no directions : 7 (with 1 image(s) acquired without diffusion gradients)
output file : [grad_b100.txt]:/sc-projects/sc-proj-agttermrt/Daten-2/mri/projects/groeschel/DTI/grad_b100.txt,>> /sc-projects/sc-proj-agttermrt/Daten-2/mri/projects/groeschel/raw/20200925_092655_20200925SMG_LAERMRT_MGR000025_1_1/4/method
..input file: /sc-projects/sc-proj-agttermrt/Daten-2/mri/projects/groeschel/raw/20200925_092655_20200925SMG_LAERMRT_MGR000025_1_1/4/method
..scan-No : 24
2b-table "900" stored as //sc-projects/sc-proj-agttermrt/Daten-2/mri/projects/groeschel/DTI/grad_b900.txt
no directions : 14 (with 1 image(s) acquired without diffusion gradients)
output file : [grad_b900.txt]:/sc-projects/sc-proj-agttermrt/Daten-2/mri/projects/groeschel/DTI/grad_b900.txt,>> /sc-projects/sc-proj-agttermrt/Daten-2/mri/projects/groeschel/raw/20200925_092655_20200925SMG_LAERMRT_MGR000025_1_1/7/method
..input file: /sc-projects/sc-proj-agttermrt/Daten-2/mri/projects/groeschel/raw/20200925_092655_20200925SMG_LAERMRT_MGR000025_1_1/7/method
..scan-No : 27
3b-table "1600" stored as //sc-projects/sc-proj-agttermrt/Daten-2/mri/projects/groeschel/DTI/grad_b1600.txt
no directions : 26 (with 1 image(s) acquired without diffusion gradients)
output file : [grad_b1600.txt]:/sc-projects/sc-proj-agttermrt/Daten-2/mri/projects/groeschel/DTI/grad_b1600.txt,>> /sc-projects/sc-proj-agttermrt/Daten-2/mri/projects/groeschel/raw/20200925_092655_20200925SMG_LAERMRT_MGR000025_1_1/9/method
..input file: /sc-projects/sc-proj-agttermrt/Daten-2/mri/projects/groeschel/raw/20200925_092655_20200925SMG_LAERMRT_MGR000025_1_1/9/method
..scan-No : 29
4b-table "2500" stored as //sc-projects/sc-proj-agttermrt/Daten-2/mri/projects/groeschel/DTI/grad_b2500.txt
no directions : 38 (with 1 image(s) acquired without diffusion gradients)
output file : [grad_b2500.txt]:/sc-projects/sc-proj-agttermrt/Daten-2/mri/projects/groeschel/DTI/grad_b2500.txt,>> /sc-projects/sc-proj-agttermrt/Daten-2/mri/projects/groeschel/raw/20200925_092655_20200925SMG_LAERMRT_MGR000025_1_1/11/method
..input file: /sc-projects/sc-proj-agttermrt/Daten-2/mri/projects/groeschel/raw/20200925_092655_20200925SMG_LAERMRT_MGR000025_1_1/11/method
..scan-No : 3
...DTItemplate imported
...DTItemplateLUT imported
```

The DTIprep-struct should now look as follows:

```
*** [DTIprep-info saved]: /sc-projects/sc-proj-agttermrt/Daten-2/mri/projects/groeschel/DTI/check.mat
-----[struct info]-----
    studypath: '/sc-projects/sc-proj-agttermrt/Daten-2/mri/projects/groeschel/'
    dtipath: '/sc-projects/sc-proj-agttermrt/Daten-2/mri/projects/groeschel/DTI/'
    brukerdata: '/sc-projects/sc-proj-agttermrt/Daten-2/mri/projects/groeschel/raw/20200925_092655_20200925SMG_LAERMRT_MGR000025_1_1'
    btable: {4x1 cell}
    DTIfileName: {4x1 cell}
    DTItemplate: '/sc-projects/sc-proj-agttermrt/Daten-2/mri/projects/groeschel/atlas_auditsys_08dec20_v1/atlas_auditsys_08dec20.nii'
    DTItemplateLUT: '/sc-projects/sc-proj-agttermrt/Daten-2/mri/projects/groeschel/atlas_auditsys_08dec20_v1/atlas_auditsys_08dec20_INFO.txt'
    INFO ini: 'INI'
    is_Brukerdata_Defined: 'YES'
    is_DTItemplate_Defined: 'YES'
    is_DTItemplateLUT_Defined: 'YES'
    INFO_import: 'IMPORT'
    is_Btable_imported: 'YES'
    is_DTItemplate_imported: 'YES'
    INFO_task: 'TASK'
    is_Atlas_and_Btable_distributed: 'NO'
    isdeformed: 'NO'
    isregistered: 'NO'
    isrenamed: 'NO'
-----[btable]-----
    {'/sc-projects/sc-proj-agttermrt/Daten-2/mri/projects/groeschel/DTI/grad_b100.txt'}
    {'/sc-projects/sc-proj-agttermrt/Daten-2/mri/projects/groeschel/DTI/grad_b900.txt'}
    {'/sc-projects/sc-proj-agttermrt/Daten-2/mri/projects/groeschel/DTI/grad_b1600.txt'}
    {'/sc-projects/sc-proj-agttermrt/Daten-2/mri/projects/groeschel/DTI/grad_b2500.txt'}
-----[DTIfileName]-----
    {'q_b100.nii'}
    {'q_b900.nii'}
    {'q_b1600.nii'}
    {'q_b2500.nii'}
```

...with "YES" for all "import"-steps.

To show the status of the DTIprep struct again just type :

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

Note that the information is stored in the DTIprep-struct and you can also close the session and resume later on (just type: DTIprep or DTIprep('check'));

Please note that the order of the B-tables and DTI/DWI-files should match and should start from lowest to highest number (100,900,1600,2500). If this is not the case use either DTIprep('reorder','b',[reorderIndex]) or DTIprep('reorder','d',[reorderIndex]) to reorder the B-tables and/or DWIfiles, respectively, example:

```
DTIprep('reorder','b',[1 4 3 2]) ; % to reorder files of 'DTIfileName' accord index 1,4,3,2
```

```
DTIprep('reorder','d',[1 2 4 3]) ; %reorder files of 'btable' accord index 1,2,4,3
```

After the initialization- and import-steps we next perform the following tasks:

(1) distribute files, (2) deform files, (3) register files, (4) rename files, (5) export files and (6) check registration. These tasks have to be performed in that order (except step 5 and 6; both tasks are optional). Note that the tasks can be performed sequentially; example: DTIprep('run',1:2,mdirs); % to run the step-1 and step-2 only.

In order to run all 6 tasks we can run DTIprep('run',1:6,mdirs) or simply DTIprep('run','all',mdirs):

```
DTIprep('run','all',mdirs);
```



When DTI-preprocessing is finished we can check the status of the DTIprep struct again:

**DTIprep('check');**

```
*** [DTIprep-info saved]: /sc-projects/sc-proj-agtiermrt/Daten-2/mri/projects/groeschel/DTI/check.mat
-----[struct info]-----
    studypath: '/sc-projects/sc-proj-agtiermrt/Daten-2/mri/projects/groeschel/'
    dtipath: '/sc-projects/sc-proj-agtiermrt/Daten-2/mri/projects/groeschel/DTI/'
    brukerdata: '/sc-projects/sc-proj-agtiermrt/Daten-2/mri/projects/groeschel/raw/20200925_092655_20200925MG_LAERMRT_MGR000025_1_1'
    btable: {4x1 cell}
    DTIfileName: {4x1 cell}
    DTItemplate: '/sc-projects/sc-proj-agtiermrt/Daten-2/mri/projects/groeschel/atlas_auditsys_08dec20_v1/atlas_auditsys_08dec20.nii'
    DTItemplateLUT: '/sc-projects/sc-proj-agtiermrt/Daten-2/mri/projects/groeschel/atlas_auditsys_08dec20_v1/atlas_auditsys_08dec20_INFO.txt'
    INFO_ini: 'INI'
    is_Brukerdata_Defined: 'YES'
    is_DTItemplate_Defined: 'YES'
    is_DTItemplateLUT_Defined: 'YES'
    INFO_import: 'IMPORT'
    is_Btable_imported: 'YES'
    is_DTItemplate_imported: 'YES'
    INFO_task: 'TASK'
    is_Atlas_and_Btable_distributed: 'YES'
    isdeformed: 'YES'
    isregistered: 'YES'
    isrenamed: 'YES'
-----[btable]-----
    {'/sc-projects/sc-proj-agtiermrt/Daten-2/mri/projects/groeschel/DTI/grad_b100.txt'}
    {'/sc-projects/sc-proj-agtiermrt/Daten-2/mri/projects/groeschel/DTI/grad_b900.txt'}
    {'/sc-projects/sc-proj-agtiermrt/Daten-2/mri/projects/groeschel/DTI/grad_b1600.txt'}
    {'/sc-projects/sc-proj-agtiermrt/Daten-2/mri/projects/groeschel/DTI/grad_b2500.txt'}
-----[DTIfileName]-----
    {'q_b100.nii'}
    {'q_b900.nii'}
    {'q_b1600.nii'}
    {'q_b2500.nii'}
```

Note that “YES” is displayed for all mandatory tasks.

The study folder should now contain a new folder “**DTI\_export4mrtrix**” with all data necessary for DTI-processing (except the shellscripts).

```
DTI_export4mrtrix
├── dat
│   ├── 20200925MG_LAERMRT_MGR000027
│   └── 20200925MG_LAERMRT_MGR000025
```

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

'ANO_DTI.nii'	-DTI-atlas, back-transformed to native space and than 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
'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'	

It is necessary to check that the registration of the images is in line with the DWI-file (1<sup>st</sup> 4D-DWI-volume: 'dwi\_b100.nii'). Task-6 of the DTIprep-command did this and HTML-files where 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” (in the “checks”-folder) in a browser we can inspect the coregistration with the 1<sup>st</sup> 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: /sc-projects/sc-proj-agtiermrt/Daten-2/mri/projects/groeschel/checks

- [DTIreg q b100--ANO DTI dim1.html](#) Created: 30-May-2022 13:28:21
- [DTIreg q b100--rc ix AVGTmask dim1.html](#) Created: 30-May-2022 13:28:20
- [DTIreg q b100--rc t2 dim1.html](#) Created: 30-May-2022 13:28:19
- [r\\_AVGT--x t2 dim2.html](#) Created: 13-May-2022 16:08:10
- [r\\_ix AVGT--t2 dim1.html](#) Created: 13-May-2022 16:08:51

Created: 30-May-2022 13:28:21

Images below: overlay of the 1<sup>st</sup> DWI-file ("q\_b100.nii") with "rc\_t2.nii" (left), "rc\_ixAVGTmask.nii" (right) and "ANO.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 1<sup>st</sup> DWI-file.



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

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