history log for data analysis

1. converting DICOMS to BIDS data-structure

- get the scanner-generated and time-ordered DICOM-files relating to a measured subject ("time-folder") from the server (USB-stick)
- create data folder ("raw_data_tnac") and within folder subdirectory (sub…)
- use the heudiconf docker-image and follow the instructions on http://nipy.org/heudiconv/ to transfer data into BIDS-structure (define a "unique heudiconf.py-file"; watch out, that subject folders are just named 01, 02,…; adjust paths to py-file, DICOM-images and output folder)
- use the online BIDS-validator to check, if your data is correctly formatted into BIDS-structure (http://incf.github.io/bids-validator/); use firefox and not opera as browser

2. defacing

- change read-write-execute rights of your BIDS-data (chmod... via bash)
- use the pydeface.py module to deface your T1 and T2 images (set up a python2 environment in bash; "source activate...deactivate")
- control via freeview or fsl
- rename the defaced images so that they still fit into BIDS and remove the non-defaced images

3. quality control

- use the mriqc docker-image and follow the instructions on http://mriqc.readthedocs.io/en/stable/docker.html to perform quality control on your BIDS
- visually inspect your images (html-output in reports), more explanations on https://mriqc.readthedocs.io/en/latest/reports/smri.html and https://mriqc.readthedocs.io/en/latest/reports/bold.html

4. anatomical "preprocessing" via FreeSurfer, ANTS, mindboggle

- use the mindboggle docker-image and follow the instructions on http://mindboggle.readthedocs.io/en/latest/ to process your anatomical data
- inspect output via fsl or freeview

5. "functional" preprocessing

- write preprocessing workflow in own jupyter notebook, not within docker images!
- (use as much SPM-interface-based nodes as possible for later DCM!!!)

- (segment to dissect grey matter from white matter, brainstem, cerebralfluid (use templates from the freesurfer "anatomical processing")

- unzip the functional T2-images and use the selectfiles-node for input-control

- (no slice time correction with short TR=1; no despike or TSNR, because we don’t know influences on time series needed for DCM; no scan drop, because scanner already performed shim-correction)
- motion-correct the functional images via MCFLIRT to a mean func image and use the parameters-file to visually inspect for motion-outliers via ArtifactDetect
- skullstrip the mean func image (BET)
- coregister the skullstripped mean func image to the skullstripped anatomical image (from the freesurfer-derived anatomical processing); use bbregister from freesurfer to create the “coregistration-matrix”
- create a transformation-matrix/”normalisation-matrix” (registration between subject's structural and MNI template) with ANTS-Registration
- Convert the BBRegister-transformation (out_fsl_file) to ANTS ITK-format for later merging of coregistration matrix with transformation matrix from antsreg
- put the 2 matrices into a list and perform coregistration and normalisation in one step; use AppliedTransform via ANTs for normalisation (before 1st level, because DCM!); Apply Transformation - applies the normalization matrix to contrast images (functional)
- for more information about normalisation via ANTs see http://miykael.github.io/nipype-beginner-s-guide/normalize.html (complete transformation)
- smoothing with FWHM=6mm of the realigned and normalized functionals