Readme

Feedback is much appreciated at mikkel.skaarup@regionh.dk ←Help can also be requested at this email address

This is version 1 of the code made public. It was developed specifically for my colleagues and I's needs. I have tired to make it more generally usable, but some more work on this is needed before it can run straight away for different users. For now, I have written a tutorial on how to set-up the code for your own data set.

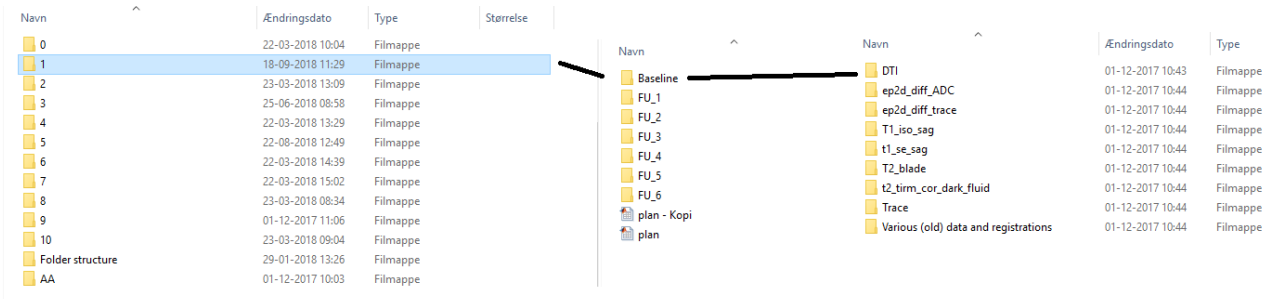
This is the code used in M. Skaarup et al. "A framework for voxel-based assessment of biological effect after proton radiotherapy in pediatric brain cancer patients using multi-modal imaging". (2021). We refer to this paper for an example of its use.

Note

Some of the data loading in the code is developed specifically to include dose distributions and Monte Carlo simulations of linear energy transfer (LET). I have not found a way to write the loading of this data more generally yet. Therefore, some modification of either the code or your data might be needed. See the document on "Pt data structure" for how it was setup for our study.

Data set

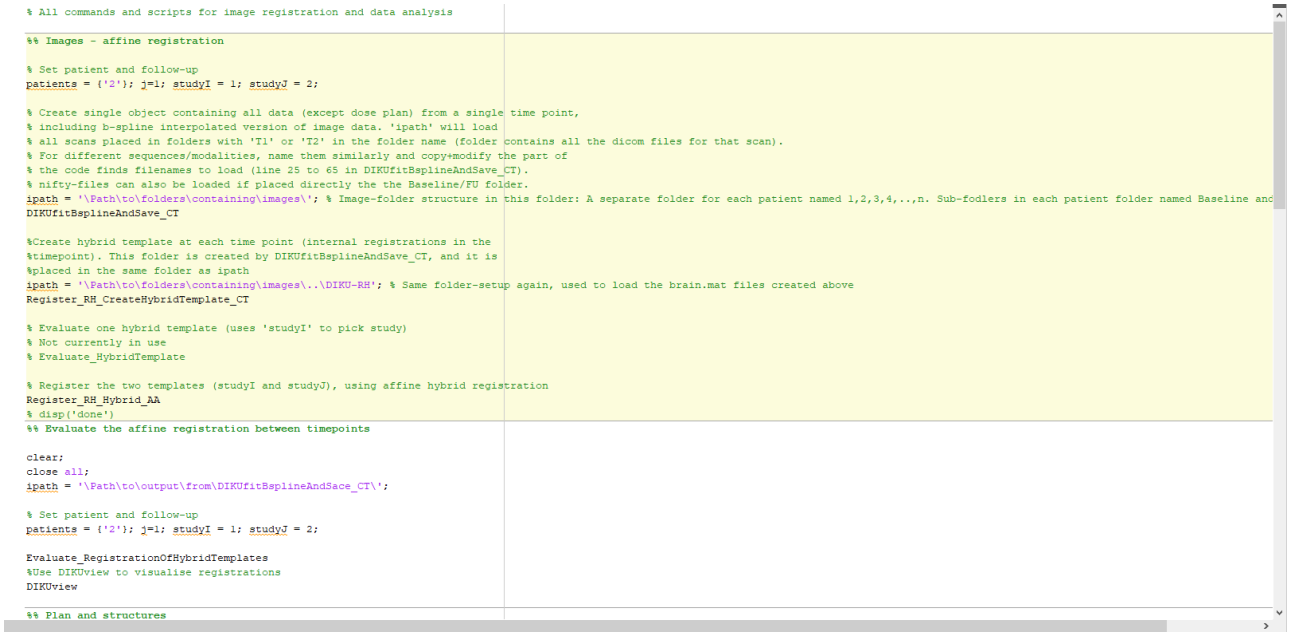
Before running the code, the data should be setup with folders and sub-folders as shown below. The code searches for folders containing "t1" or "t2", this can be changed (see the tutorial below). As of now, the code can load nifty files and dicom files.



Tutorial

Unpack the zip-file at a desired location.

Navigate to Reg_scripts\Reg_scripts\ in Matlab and open summary_of_scripts. This is the main script in the registration setup. At present, the code runs for one patient at the time, therefore change the numbers as needed between each patient and registration in line 6 as shown here (patient refers to the folder name at the top of the data folder-tree, StudyI and StudyJ defines which sessions will be registered where 1 = Baseline, 2=FU_1, 3=FU_2, and so on):



```
% All commands and scripts for image registration and data analysis

%% Images - affine registration

% Set patient and follow-up
patients = {'2'}; j=1; studyI = 1; studyJ = 2;

% Create single object containing all data (except dose plan) from a single time point,
% including b-spline interpolated version of image data. 'ipath' will load
% all scans placed in folders with 'T1' or 'T2' in the folder name (folder contains all the dicom files for that scan).
% For different sequences/modalities, name them similarly and copy+modify the part of
% the code finds filenames to load (line 25 to 65 in DIKUfitBsplineAndSave_CT).
% nifty-files can also be loaded if placed directly the the Baseline/FU folder.
ipath = '\Path\to\folders\containing\images\'; % Image-folder structure in this folder: A separate folder for each patient named 1,2,3,4,..,n. Sub-fodlers in each patient folder named Baseline and
DIKUfitBsplineAndSave_CT

%Create hybrid template at each time point (internal registrations in the
%timepoint). This folder is created by DIKUfitBsplineAndSave_CT, and it is
%placed in the same folder as ipath
ipath = '\Path\to\folders\containing\images\..\DIKU-RH'; % Same folder-setup again, used to load the brain.mat files created above
Register_RH_CreateHybridTemplate_CT

% Evaluate one hybrid template (uses 'studyI' to pick study)
% Not currently in use
% Evaluate_HybridTemplate

% Register the two templates (studyI and studyJ), using affine hybrid registration
Register_RH_Hybrid_AA
% disp('done')
%% Evaluate the affine registration between timepoints

clear;
close all;
ipath = '\Path\to\output\from\DIKUfitBsplineAndSace_CT\';

% Set patient and follow-up
patients = {'2'}; j=1; studyI = 1; studyJ = 2;

Evaluate_RegistrationOfHybridTemplates
%Use DIKUview to visualise registrations
DIKUview

%% Plan and structures
```

Change the folder path in line 14 to your data folder. The function “DIKUfitBsplineAndSave_CT” will load the data and prepare it for registrations. Of note, the code will create a new folder titled “DIKU-RH” where the registrations and parameters will be stored. The variable “ipath” in line 20 needs to be set to this folder, which should be equal to “ipath”, but up one level in the tree and there the DIKU-RH folder should be**. If you have not created your own plan.mat-file, you will need to modify line 28 and 155 to properly load the CT scan into the baseline template.**

The two remaining functions of this section should run without changes. These will perform affine registratsion internally in each template, and perfom a multimodal affine image registration between the templates. This registration is the baseline for the alter deformable image registration, and thus, an intermediate result.

Similarly, the next section (”Evaluate the affine registration between timepoints”) should run after just updating the “ipath” variable. This section will show the results of the intermediate hybrid template registrations.

In ”Plan and structures” the masks and delineations are imported from the plan.mat file (see the “Pt data structure” document) and resampled to the hybrid templates and saved. Here, the paths need to be upgraded, as well as the desired patient. This needs only be run once per patient as the contours are available from the baseline template. Thus, the parameter “studyJ” can be set to any of the follow-up sessions if that session has been through the intermediate hybrid template registration.

In the ”Register between timepoints using non-rigid registration” section, two hybrid templates are deformably registered and the result is visualized through the “Evaluate_...” function. Update the path to the DIKU-RH folder and update the desired patient and follow-up session to be registered.

In the "Dose" section, the dose is included into the registrations. **This section assumes the "Evaluate_..." from the previous section has been run with the desired patient and studyI/studyJ combination.** This section is heavily reliant on the pt.mat-file (see the "Pt data structure" document). Of note, update the "strBrain" and "str" functions. These select the contour/organ/structure of interest to save for further analysis in the R script or your own software/script.
An overview of this section can be seen here, and below it will be explained in greater detail:

```
%% Dose
%Load plan and masks
dpath = '\Path\to\plan.mat-file\';
%This section assumes Evaluate_RegistrationOfHybridTemplates_NR has been
%run in the previous section (NOTE: not necessarily Register_RH_Hybrid_NR as
%Evaluate_RegistrationOfHybridTemplates_NR can be run to simply load the
%saved registration parameters and apply them to the sessions in question)
j=1;
load(fullfile(dpath,patients{j},'plan.mat'),'pt');
load(fullfile(ipath,patients{j},studies{studyI},'masks'));

% If the CT is too large (>300 slices for example), it can be reduced here
% by un-commenting
% if patients{j} == '1'
%     reduceCT
% end

% Dose resampling in baseline hybrid frame
% Dose (total dose or some other dose distr)
% totDose = getTotalDose(pt);
[doseTot,doseTotCorr,dLET,doseOrig,nFr,nB,nPh] = correctDoseDistr(pt,0,2.1,2);
[dosePhase,dLETPhase,dLETDosePhase,nFr,nB,nPh] = PhaseDose(pt,0,2.1);

% Create dose distribution within mask only
% Set structure no (for brain: pt 1 = 18; pt 2 = 14; pt 3 = 15; pt 4 = 15; pt 5 = 15; pt 6 = 15;
strBrain = [18,14,15,15,15,15]; %for brain: pt 1 = 18; pt 2 = 14; pt 3 = 15; pt 4 = 15; pt 5 = 15; pt 6 = 15;
str = strBrain(str2num(patients{j}));

dose = doseTot;% doseTot / doseTotCorr / doseDiff. Change this, depending on the dose distribution to be displayed
DoseAndFAInMask;

%Use DIKUview to visualise registrations and dose/LET distributions
DIKUview
```

NOTE: If you wish to use a RBE corrected dose, you need to comment and uncomment in line 104 in correcDoseDistr and uncomment/comment line 107/108 in PhaseDose.

The "correctDoseDistr" function requires four inputs, the plan.mat file, the model of LET-dependent RBE correction by integers 0 (no model, RBE=1.1), 1 (Carabe et al. (2012)), 2 (Wedenberg et al. (2013)) or 3 (McNamara et al (2015)). The third input is the alpha/beta value for the tissue/structure of interest. The last input is a setting for the dose output, 1 for RBE 'standard' correction, and 2 for equivalent dose in 2 Gray fractions (this can be changed in line 79 of "correctDoseDistr"). The outputs are:
-doseTot: The total summed dose over phases and beams.
-doseTotCorr: the corrected dose according to the settings specified in the call of the function (input 2 and 4).
-dLET: the dose averaged linear energy transfer (LET) for each phase of the plan
-doseOrig: The dose, but kept as one distribution per beam and phase of the plan.
-nFr, nB, nPh: The number of fractions, beams and phases, respectively.

The "PhaseDose" function requires 3 inputs, the plan.mat file, the model of choice (RBE=1.1, Carabe, Wedenberg or McNamara, as above) and the alph/beta value for the structure/tissue of interest. This function outputs the following:

-dosePhase: The dose of each phase, summed per beam.
-dLETPhase: The dose averaged LET per phase
-dLETDosePhase: dose times the LET for each phase

In line 128 in summary_of_scripts, you choose which dose you wish to use for further analysis.

“DoseAndFAInMask” is called in line 129. This function resamples the dose to fit the deformable image registration by applying the baseline registrations applied to the CT scan that was used for dose-planning.
This function also sums the dLETDosePhase over all phases and divide it by the summed dose over all doses to create a single dose averaged LET over the entire volume. (this is not the correct way to use LET as it of course varies between beams, but this gives and estimate of where high LET and dose are present across the treatment. Modify this as needed for analysis
In line 16-25, a cutoff is found for FA values. This was specific to our study setup, so modify/remove as needed. It links with line 25 where the cutoff is applied to create a normal-appearing WM mask. **Comment/modify as needed here.**
Finally, the data is saved as .mat files in lines 29-37 in a folder called “Data”. Modify as needed