Data processing data from 3D cine gradient echo PC velocity encoding sequence under evoked motion using EMS or voluntary motion

Matlab 2017a or Matlab 2019a

Load data:

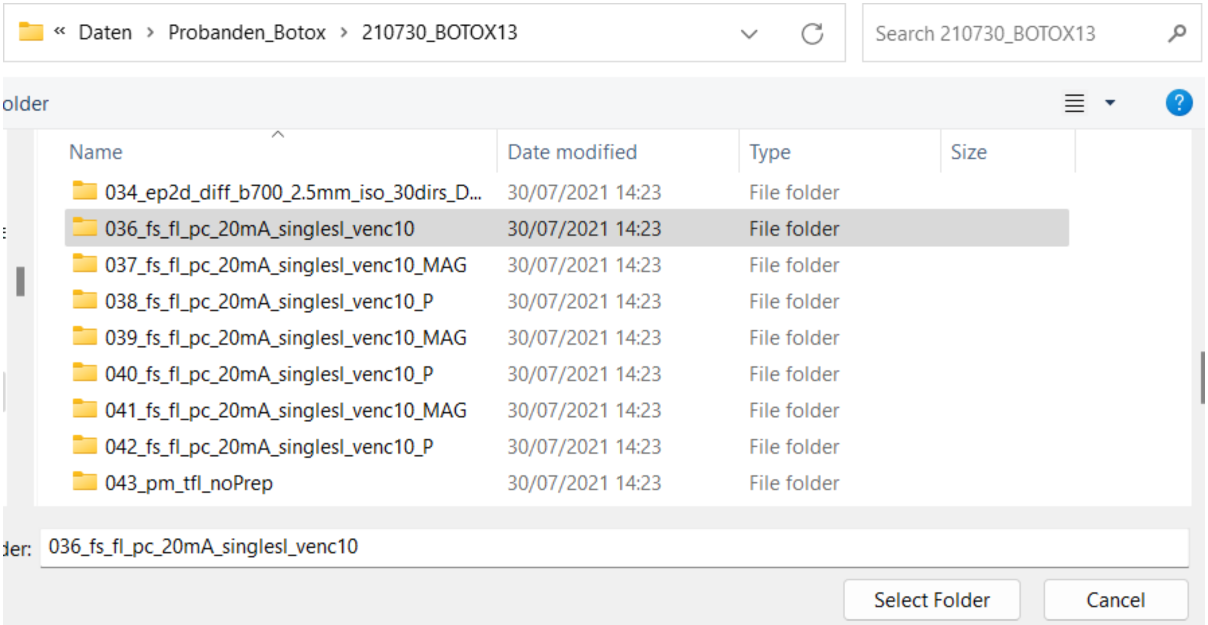
Here for single slice acquisition, for multi slice acquisition use std_velocity_visualization_multi (you will be asked which slice to choose for processing, processes only the chosen slice in each run)

>> std_velocity_visualization

insert head folder name: "E:\Daten\Probanden_Botox\"

select folder for subject containing all the DICOM files in window that pops up

and choose (in this folder) the first of the DICOM folders with the cine gradient echo PC data (in another window popping up)



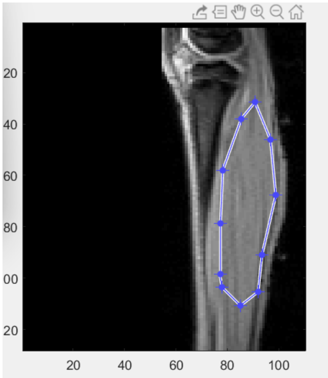
Loading Dicom: [....................] ETA: 0:00

Loading Dicom: [....................] ETA: 0:00

Loading Dicom: [....................] ETA: 0:00

Loading Dicom: [....................] ETA: 0:00

In a pop-up window you can select the ROI on your imaging slice



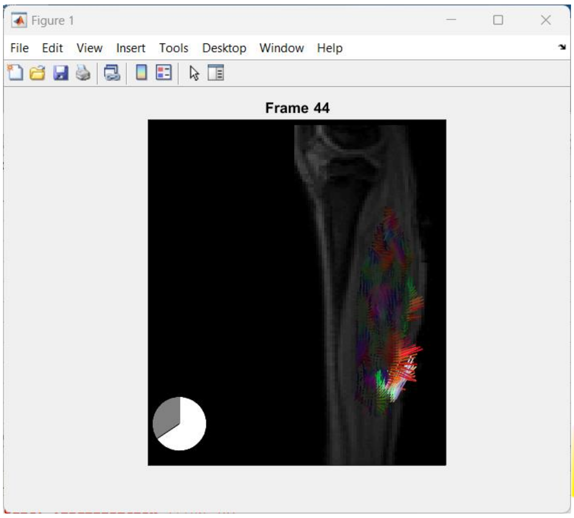
Give interpolation factor: 1

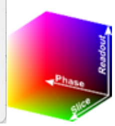
Visualize/ Save Velocity (Y/N=1/0): 1

If yes, then images with the velocity vectors for each temporal phase are displayed and stored in current directory (named velMap_001.jpg etc.)

R or L: 1/0: 0

Select where the pie chart with the temporal phase is displayed in the image (here 0 = left)





In the Matlab code of std_velocity_visualization.m you have to modify according to your data:

venc = 10;%cm/s sAngio.sFlowArray.asElm[0].nVelocity

scale_factor =16;

this is the length of the displayed velocity vectors (the colors represent the directions)

tDelay =5;%trigger delay sPhysioImaging.sPhysioExt.lTriggerDelay/1000

AcqTime=3500;

limit=29;

this is the number of letters following after the scan number 0%%_, in this example 036_fs_fl_pc_20mA_singlesl_venc10 => 29

when the program is finished save the workspace it contains

- ⇨ Variables mask (your drawn ROI, delete it if you want to rerun selecting a different ROI, otherwise the mask will be kept for the next run)
- ⇨ flow_x (flow in x direction in cm/s), here size 128x110x67 double for 67 temporal phases and 128x110 image matrix, flow_y, flow_z
- ⇨ flow3D: ROI median of the magnitude of the velocity vectors
- ⇨ time_ser (time for each temporal phase in ms), size 1x67

plot flow3D, this is your velocity time course in the ROI

\>> figure, plot(time_ser/1000,flow_3D);

Matlab-script to calculate

- ⇨ displacement in x-, y-, z-direction (dispVxi…),
- ⇨ 4D data of strain Eig_v with (size 128x110 x2x67); max./min. eigenvalues in :,:,1: (stretching component) and :,:,2,: (compression component)
- ⇨ 4D data of strain rate Eig_SR (size 128x110 x2x67); max./min. eigenvalues in :,:,1: (stretching component) and :,:,2,: (compression component)
- ⇨ e1_Line: time course of median of magnitude of strain stretch component in ROI (size 1x67 double) with max. value e1_max
- ⇨ e2_Line: time course of median of magnitude of strain compression component in ROI (size 1x67 double) with max. value e2_max
- ⇨ r_Srs: time course of median of magnitude of strain rate stretch component in ROI (size 1x67 double)
- ⇨ r_Srs2: time course of median of magnitude of strain rate compression component in ROI (size 1x67 double)

\>> script_Botoxprocessing

needs Matlab MaxPol package to calculate filter

calculates principal eigenvalues for strain and strain rate tensors, the principal eigenvalue with the largest absolute value represents the compression component

plots velocity time course (flow_3D, see above), strain and strain rate time course (e1_Line,e2_Line; rf_Srs,r_Srs2)

displays map of strain stretching component and compression component at middle of time course and strain rate stretching and compression component maps at frame with first velocity peak