Last Update: August 14, 2020

Authors: Judit Ben Ami & Marina Khizgilov

Created during 2019-2020 as part of a final project towards BSc in biomedical engineering.

TCML lab at the Department of Biomedical Engineering, Technion - IIT, Haifa, Israel

Under the supervision of Dr. Moti Frieman

Special thanks for Elad Rotman for phantom\_simulation function.

**IvimParamsEstimLib**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

IvimParamsEstimLib is a multi-purpose Python library for IVIM parameters estimation.

The model used for IVIM–based biexponential analysis is:

Where D is the diffusion coefficient, which reflects tissue diffusivity; D\* is the pseudo-diffusion coefficient, which reflects microcapillary perfusion; and f is the perfusion fraction.

This library includes:

* 4 IVIM parameters estimation algorithms: SEGb, SEG, LSQ, BSP.
* Phantom creating function.
* Error calculation functions.

In addition, for efficiency and higher performance, the library uses vectorize programming, multiprocessing, assertions and exceptions.

# **IvimParamsEstimLib. dwmri\_images\_to\_ndarray**

**dwmri\_images\_to\_ndarray(file\_path, b\_value, slice\_num=0, plot\_flag=False, save\_plot\_flag=False, save\_plot\_prefix=’001’)**

Converts input DW-MRI images to a ndarray.

**Parameters: file\_path : List of PathLike**

Paths of the DW-MRI images to convert.

**b-values : ndarray**

Vector containing the B-values the DW-MRI image was taken at.

**slice\_num : integer, optional (0 by default)**

For 3D Images, choose a specific 2D slice.

**plot\_flag : boolean, optional (False by default)**

If True, plots the output figure.

**save\_plot\_flag : boolean, optional (False by default)**

If True, saves the output figure as a PNG file to current working directory.

**save\_plot\_prefix: str, optional (‘001’ by default)**

Prefix of the name of the file the figure will be saved as.

**Returns: slice: ndarray, shape [b\_num, y\_wid, x\_wid]**

Conversion result matrix.

Notes:

# **IvimParamsEstimLib.ivim\_params\_estim**

**ivim\_params\_estim(dw\_mri\_images, b, sim\_flag, estimator\_mode='ALL', image\_type\_mode='norm', b\_tresh=200, plot\_flag=True, save\_plot\_flag=False, save\_plot\_prefix="001", multi\_segs=1, env\_size=5, dbg\_features=False)**

Evaluates the IVIM parameters, D, D\* and f. Reconstructs the DW-MRI imaged using the estimated IVIM parameters.

**Parameters: dw\_mri\_images** **: ndarray, shape [b\_num,y\_wid, x\_wid]**

A 3D matrix consisting of a series of slices of DW-MRI images taken at b\_num different b-values.

**b : ndarray**

Vector containing the B-values the DW-MRI image was taken at.

**sim\_flag : boolean**

If True, the input images are from simulation meaning every pixel comes from IVIM bi-exponential model. Else (if False) the input images are clinical DW-MRI images.

**estimator\_mode** **‘SEGb’, ‘SEG’, ‘LSQ’ or ‘BSP’ ,’ALL’ optional (‘ALL’ by default)**

Type of IVIM parameters estimation algorithm. Choose ‘ALL’ to get estimation results using all the modes: ‘SEGb’, ‘SEG’, ‘LSQ’ or ‘BSP’.

**image\_type\_mode** **: ‘norm’ or ‘absolute’, optional (‘absolute’’ by**

**default)**

DW-MRI image mode. ‘norm’ if image is normalized, ‘absolute’ if Image is not normalized.

**b\_tresh : integer, optional (200 by default)**

b-value threshold for SEG/SEGb estimations.

**plot\_flag : boolean, optional (False by default)**

If True, plots the output figures - maps of IVIM parameters and the reconstructed DW-MRI images.

**save\_plot\_flag : boolean, optional (False by default)**

If True, saves output figures as a PNG files to current working directory.

**save\_plot\_prefix: str, optional (‘001’ by default)**

Prefix of the name of the files the figures were saved as.

**multi\_segs: integer, optional (8 by default)**

For multi-processing implementation. Maximum number of multi-processes.

**env\_size: integer, optional (5 by default)**

For ‘BSP’ mode, number of elements in each direction around a pixel.

**Returns : <estimator\_mode>\_estim\_D : ndarray (or list for ‘ALL’ estimator)**

Estimated D map, using <estimator\_mode> algorithm.

**<estimator\_mode>\_estim\_D\_star : ndarray (or list for ‘ALL’ estimator)**

Estimated D\* map, using <estimator\_mode> algorithm.

**<estimator\_mode>\_estim\_f : ndarray (or list for ‘ALL’ estimator)**

Estimated f map, using <estimator\_mode> algorithm.

**<estimator\_mode>\_** **estim\_images : ndarray (or list for ‘ALL’ estimator)**

Reconstructed images.

Notes:

* For shorter running time, more than 1 multi-process is favoured.
* For ‘ALL’ estimator mode, each return is a list consists the results of: ‘SEGb’, ‘SEG’, ‘LSQ’, ‘BSP’ in that particular order.

# **IvimParamsEstimLib.Phantom\_Simulation\_DWMRI**

**phantom\_simulation(b\_val, D\_val, D\_star\_val, Fp\_val, B0\_val, x\_wid=64, y\_wid=64, rads=None, noise\_type='NaN', SNR=0.1, plot\_flag=False, save\_plot\_flag=False, save\_plot\_prefix=’001’)**

Creates IVIM parameters maps and simulates DW-MRI images. The phantoms created consists of N concentric circles.

**Parameters:**  **b-values : ndarray**

Vector containing the b-values for the simulated image.

**D\_val : ndarray** **, shape [N+1,]**

Vector containing the D values of the simulated image.

**D\_star\_val : ndarray, shape [N+1,]**

Vector containing the D\* values of the simulated image.

**Fp\_val : ndarray, shape [N+1,]**

Vector containing the f values of the simulated image.

**B0\_val : ndarray, shape [N+1,]**

Vector containing the B0 values of the simulated image.

**x\_wid : integer, optional (64 by default)**

Number of pixels at x-axis of simulated image.

**y\_wid : integer, optional (64 by default)**

Number of pixels at y-axis of simulated image.

**rads : ndarray, shape [N,], optional ([10,20,30] by default)**

Vector containing the radii of the circles of simulate image.

The radii are in ascending order of length, where the value of the first index of the vector is the radius of the inner circle, and

Value of the last index of the vector is the radius of the external circle.

**noise\_type : ‘NaN’, ‘gaussian’, ‘rayleigh’, ‘rice’ or ‘non\_centralized\_chi2’, optional (‘NaN’ by default)**

Type of noise to be added to the simulated DW-MRI images.

**SNR : float, optional (0.1 by default)**

SNR value for gaussian noise.

**plot\_flags : boolean, optional (False by default)**

If True, plots the output figures – IVIM parameters maps and simulated DW-MRI images.

**save\_plot\_flag : boolean, optional (False by default)**

If True, saves the output figure as a PNG file to current working directory.

**save\_plot\_prefix: str, optional (‘001’ by default)**

Name of the file the figure was saved as.

**Returns: dwi\_images : ndarray, shape [b\_num,y\_wid, x\_wid]**

Simulated DW-MRI images.

**B0\_phantom :** **ndarray, shape [y\_wid, x\_wid]**

Initial simulated DW-MRI image taken at b=0.

**D\_phantom : ndarray, shape [y\_wid, x\_wid]**

Simulated D map.

**D\_star\_phantom: ndarray, shape [y\_wid, x\_wid]**

Simulated D\* map.

**Fp\_phantom : ndarray, shape [y\_wid, x\_wid]**

Simulated f map.

Notes:

* As mentioned, D\_val, D\_star\_val, Fp\_val, B0\_valvectors are the size of N+1. The elements of each vector are ordered according to circles locations:

The first element of each vector represents the inner circle and each element that comes afterwards represents the next circle. Last element of the vectors represents the background of the phantom.

# **IvimParamsEstimLib.error\_estim\_ivim\_maps**

**error\_estim\_ivim\_maps(dw\_mri\_images, b\_val, known\_D, known\_D\_star, known\_f, estim\_D, estim\_D\_star, estim\_f, error\_type='perc')**

Computes the error between simulated IVIM parameters maps and estimated IVIM parameters maps.

**Parameters: dw\_mri\_images: ndarray, shape [b\_num,y\_wid, x\_wid]**

A 3D matrix consisting of a series of slices of DW-MRI images taken at b\_num different b-values.

**b-val : ndarray**

Vector containing the B-values the DW-MRI image was taken at.

**known\_D : ndarray, shape [y\_wid, x\_wid]**

Simulated D map.

**known\_D\_star : ndarray, shape [y\_wid, x\_wid]**

Simulated D\* map.

**known\_f : ndarray, shape [y\_wid, x\_wid]**

Simulated f map.

**estim\_D: ndarray, shape [y\_wid, x\_wid]**

Estimated D map.

**estim\_D\_star: ndarray, shape [y\_wid, x\_wid]**

Estimated D\* map.

**estim\_f: ndarray, shape [y\_wid, x\_wid]**

Estimated f map.

**error\_type: ‘l1’, ‘l2’ or ‘perc’ (‘perc’ by default)**

Type of error calculation.

**Returns: D\_error : float**

Calculated error between known (e.g. simulated) D map and estimated D map.

**D\_star\_error : float**

Calculated error between known (e.g. simulated) D\* map and estimated D\* map.

**f\_error: float**

Calculated error between known (e.g. simulated) f map and estimated f map.

**mean\_error : float**

Mean of calculated IVIM parameters maps errors.

Notes:

# **IvimParamsEstimLib.error\_estim\_dw\_images**

**error\_estim\_dw\_images(orig\_dw\_mri\_images, rec\_dw\_mri\_images, b\_val, sim\_flag , error\_type='perc')**

Computes the error between original DW-MRI images and reconstructed DW-MRI images obtained from IVIM estimation function.

**Parameters: orig\_dw\_mri\_images : ndarray, shape [b\_num,y\_wid, x\_wid]**

A 3D matrix consisting of a series of slices of DW-MRI images taken at b\_num different b-values.

**rec\_dw\_mri\_images : ndarray, shape [b\_num,y\_wid, x\_wid]**

A 3D matrix consisting of a series of slices of reconstructed DW-MRI images taken at b\_num different b-values.

**b-val : ndarray**

Vector containing the B-values the DW-MRI image was taken at.

**sim\_flag : boolean**

If True, the input images are from simulation meaning every pixel comes from IVIM bi-exponential model. Else (if False) the input images are clinical DW-MRI images.

**error\_type: ‘l1’, ‘l2’ or ‘perc’ (‘perc’ by default)**

Type of error calculation.

**Returns: < error\_type >\_error : float**

Calculated error between known DW-MRI images and reconstructed DW-MRI images obtained from IVIM estimation function. ([Phantom\_Simulation\_DWMRI](#ivim_estim_func_bookmark))