Welcome to the software package for Transport-Based Morphometry (TBM). This README describes the content of this package, installation instructions, as well as sample execution commands.

**Contents of the package**

1. Codes for computing TBM forward transformation
2. Codes for generating the data matrix and dimensionality reduction
3. Codes for regression and classification analyses
4. Codes for visualizing the morphology associated with statistical differences

**Installation**

This package uses some C codes to speed up computations and there is a compilation step to generate these executables. First, make sure you have a valid C compiler by typing mex –setup in MATLAB.

Run the lines of code in the Installation.mat file to generate the necessary mex files.

**How to run TBM forward transformation**

Your files must be 3D .mat files. If they are .nii files, convert them to .mat files using SPM.

The procedure for optimal transport minimization is described in the paper Kundu S, Kolouri S, Erickson KI, Kramer AF, McAuley E, Rohde GK. “A Transport-Based Morphometry approach for detecting and visualizing structural changes from MRI.” *Submitted*.

The multVOT function generates the forward transformation as follows.

results = multVOT(double(I0\_b),double(I1\_b),,10^6,,numScales,0.25,)

The recommended parameters are, numScales = 3. The controls the penalization of the curl, and controls the smoothness of the image. The is the parameter that penalizes the mass transport. Making this larger may make it harder for the code to converge.

Now go to the Examples folder to make sure you can register two images using the appropriate section in the Main script. For multiple images, a for-loop can be written that reads, processes using multVOT, and saves each file. In practice, you can generate the template image by taking the Euclidean average of all images.

*Notes, tips and tricks*

* Generate pdf using the gen\_pdf function. For versions of MATLAB 2014 and older, go inside the function and comment out the appropriate blocks of code (instructions are in the function).
* If the multVOT step is too slow to converge, downsample your images using SPM resample. The ideal size is 128x128x128.
* The multVOT will have a hard time producing deformations at the edges of the image if they are too close to the border. Zero pad the images sufficiently to avoid problems with matching the edges. The tradeoff of low step size is time it takes to converge.
* The default level of convergence is , which is indicated by the variable cutoff the VOT3D function. This can be changed if the match of the images is not satisfactory based on visual appearance.
* The ringing in the descent of curl and MSE is due to a step size that is too high. Look for the variable step\_size and decrease it from 0.01 of a voxel to 0.005 of a voxel.

**How to generate data matrix**

By this point, you should have generated mappings are stored in the results generated from multVOT in [results.f1 results.f2 results.f3]. These are the components of the transport maps.

After concatenating mappings, run PCA analysis to generate a data matrix. Z is the reduced dimension data matrix in the TBM\_feats file.

The final data matrix is contained in the 3DTBM\_all folder, in 8 parts that should be concatenated as F = [TBM\_F1 TBM\_F2 TBM\_F3 TBM\_F4 TBM\_F5 TBM\_F6 TBM\_F7 TBM\_F8];