DTI Pipeline Manual

Zenan Tang

M: zenan.tang@mail.utoronto.ca

August 27th, 2021

Intro:

This document goes over the steps required to run a full structural connectivity DTI-based analysis including statistics and data-visualization, with brief explanations of scripts and script functions. All scripts can be found in the mabbottlab/temp_backup/Tommy/Github folder, with examples of previous files generated with the script on the Brain Matters Workspace (OneDrive), as well as on carbon.

Softwares/Packages/Languages Required:

- Bash/Shell
- Python3
- Matlab
- R (optional, can substitute with Matlab)
- Microsoft Excel (optional, can substitute with R/Matlab)
- BrainNetViewer (BNV; on Matlab)
- Network-Based-Statistics (NBS; on Matlab)
- MRtrix
- FSL
- Freesurfer

Order of Steps:

Acquire MRI scans -> MRTrix_Connectome_DTIpreprocessing.sh -> MRTrix_Connectome_DTIAnalysis.sh + mean_DTI_metric.py -> organize_stat_files.sh -> mean_DTI_metric_spreadsheet_script.m + a_2D_to_3D_matrix.m + average_connectomes.m -> Graph DTI metrics vs ROIs on Excel -> Run NBS -> Run stats, generate graphs -> Run BNV.

Note: For any scripts, modify file_paths as needed, particularly when experiencing error – subject file not found.

Acquire MRI scans

Acquire T1 and DWIs for every subject in sample.

MRTrix_Connectome_DTIpreprocessing.sh

This script is an optimized version of the preprocessing script from the BATMAN tutorial suited for our purposes, and without Gibbs denoising. Steps are explained within the script, but the overall process is: 1. Convert data to .mif and remove artificats -> 2. Generate fiber orientation

distributions for each tissue type. -> 3. Use 5-tissue-type segmentation to generate a GM/WM seed co-registered to subject DWI. 4. Construct cortical parcellations file from atlas using FreeSurfer -> 5. Generate tracts, filter, and reduce tracts to an arbitrary number (usually ~1M) using tcksift -> 6. Create whole-brain connectome.

MRTrix_Connectome_DTIAnalysis.sh + mean_DTI_metric.py

The MRTrix_Connectome_DTIAnalysis script generates masks, tracts, and statistics from only the output of the MRTrix_Connectome_DTIpreprocessing script. This means that you must adjust the file paths in this script if you change the file paths in the previous script. This script also calls mean_DTI_metric.py near the end, which generates a text file per DTI metric per ROI, which contains one value that represents the average of all tract stats of a particular DTI metric (ex. FA).

This script begins with identifying networks belonging to select overarching/encompassing networks (ex. PCC network is a part of the default mode network). Details of steps are in the script. In the current script, there are 14 networks belonging to two overarching networks: the default mode, and executive control networks. Each network is a combination of specific nodes extracted from the atlas parcellations file. We begin by making a mask file per node of interest for both left and right hemispheres of the brain. We then combine these node masks together to make our ROI masks. There are some conversions from nifti to mif for future use in FSL if needed. Once both left, right, and left-right ROI masks are made, we generate exclusion masks to exclude regions we are not interested in when generating tracts. Exclusion masks are made by adding all the nodes per hemisphere together, then subtracting the nodes that shouldn't be included (these nodes are the ones we are interested in using for tract generation). We then make two additional exclusion masks; one for the brainstem, and one for the cerebellum (these are optional depending on your analyses), and co-register them to each subject's b0 scan. We then generate all tracts that belong to each of the ROIs, create whole-brain DTI metric maps, and sample the tracts from the DTI metric maps. We also sample the number of streamlines as a connectivity strength measure, which is taken directly from the number of streamlines that belong to each tract file. Finally, we remove the node files used for generating the masks, as these are no longer necessary for analysis. This step frees up a considerable amount of space and is optional (Comment it out if undesired). For any documentation help related to MRTrix, lookup the MRTrix3 documentation online.

To modify this script to your analysis, identify the networks you are interested in, and replace the code one network at a time. I didn't have time to implement functions in this script, but this is also something that could be developed by future users. If there are errors, check for common typos in file names, as this is a very easy mistake to make with ~1200 lines of code that repeat networks!

Organize_stat_files.sh

This script takes all the files within the stats folder for each subject and reorganizes them into 3 folders (left, right, left-right). Within each of these 3 folders, there are 5 folders – one per DTI metric (rd, ad, md, fa, streamlines). Essentially, 15 folders per subject will be created, which will be the basis for any future work with matlab scripts, NBS, and BNV. For this script to work, the ROI naming convention in the analysis script must be adhered to.

mean_DTI_metric_spreadsheet_script.m + a_2D_to_3D_matrix.m + average_connectomes.m

The purpose of the mean_DTI_metric_spreadsheet_script is to get a quick glance if all the files were generated properly, without having to check each subject folder manually. This should be run first, and the tables generated should be viewed for any blank [] entries representing errors. This file should be modified according to where the ROI files are located (specifically what folders and subfolders within the subject stats file). Also note that in this script (also applies to the next script), for the variable generated (line 19) that contains a list of all subject files under Tommy/DTI-analysis/, there is a +n entry which changes depending on (ex. string(folders(k+5).name) what non-subject files are in the folder (ex. a script). This +n entry needs to reflect how many files are listed before the subject folders in bash when referencing i in the for loop. In the example, k+5 means that we are skipping 5 files before ST02 in DTI-analysis. Change the +n to reflect how many non-subject files appear (this is a very common source of errors when running this script, as well as the next).

The purpose of the a_2D_to_3D_matrix.m script is to generate the data matrices for all metrics and group comparisons of interest. The default outputs are saved as '.xlsx' in a stats folder, but for running NBS, I recommend saving it manually as a '.m' datatype in whatever folder you want. This also generates the spreadsheets needed for plotting line graphs. This script requires modification based on how the ROIs are listed in bash (alphabetical order) to change a 2D matrix (row=subject, column=ROIxROI) to a 3D matrix (row = ROI1, column = ROI2, 3rd dimension = subject).

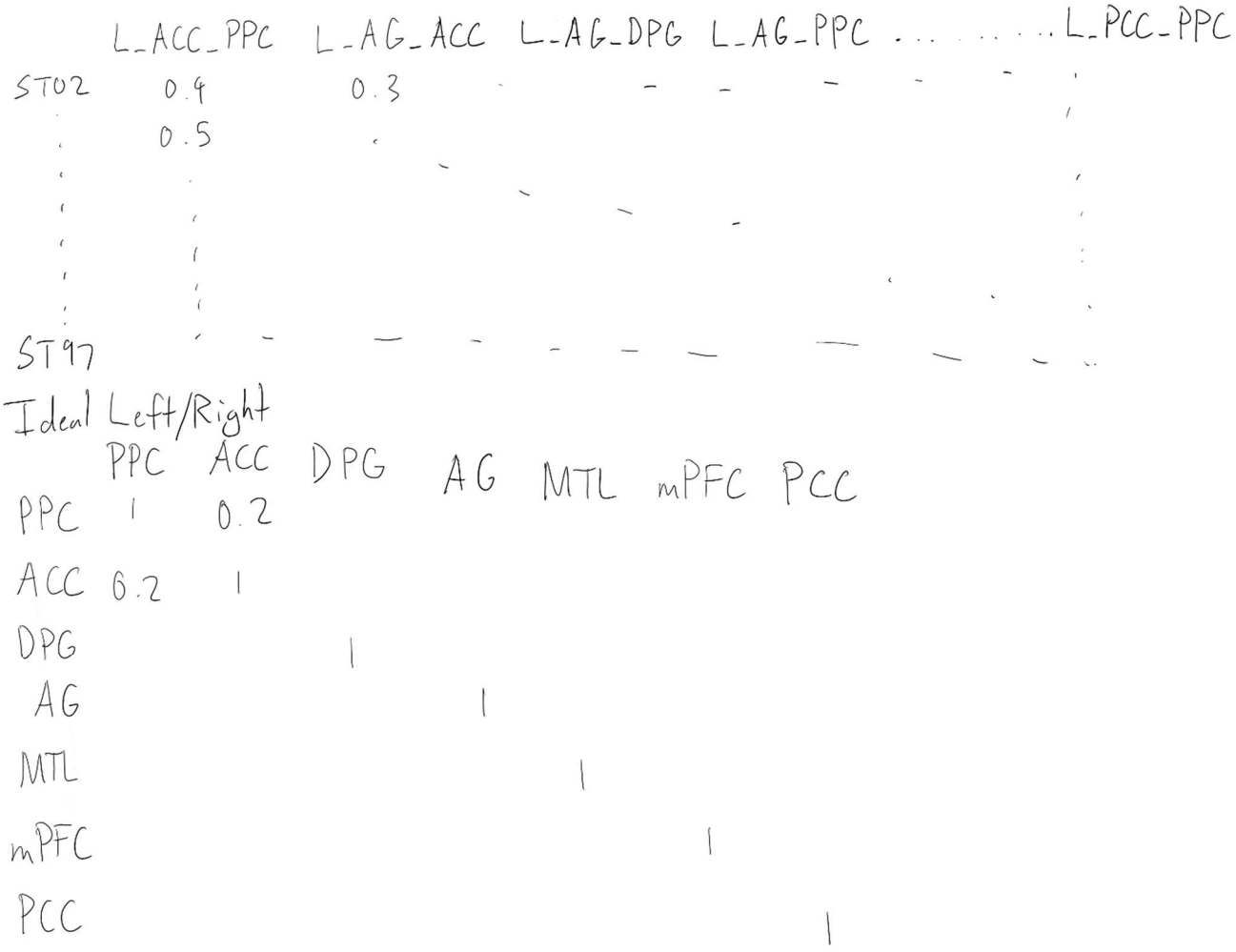

Figure 1. Transforming 2D matrix to 3D matrix. Top shows 2D matrix, bottom shows 3D matrix.

This script also requires modification depending on what groups you are running in NBS to generate different data matrices. For example, if you are running PBTS vs HC, you would include all your subjects in the script by letting the for loop run from start to finish. If you are running a sub-group, you will need to manually specify which subject files are included from the overall list of files in 'subject_stats_folder'. I recommend drawing the matrix out by hand for a reference when assigning values to each row and column in the script.

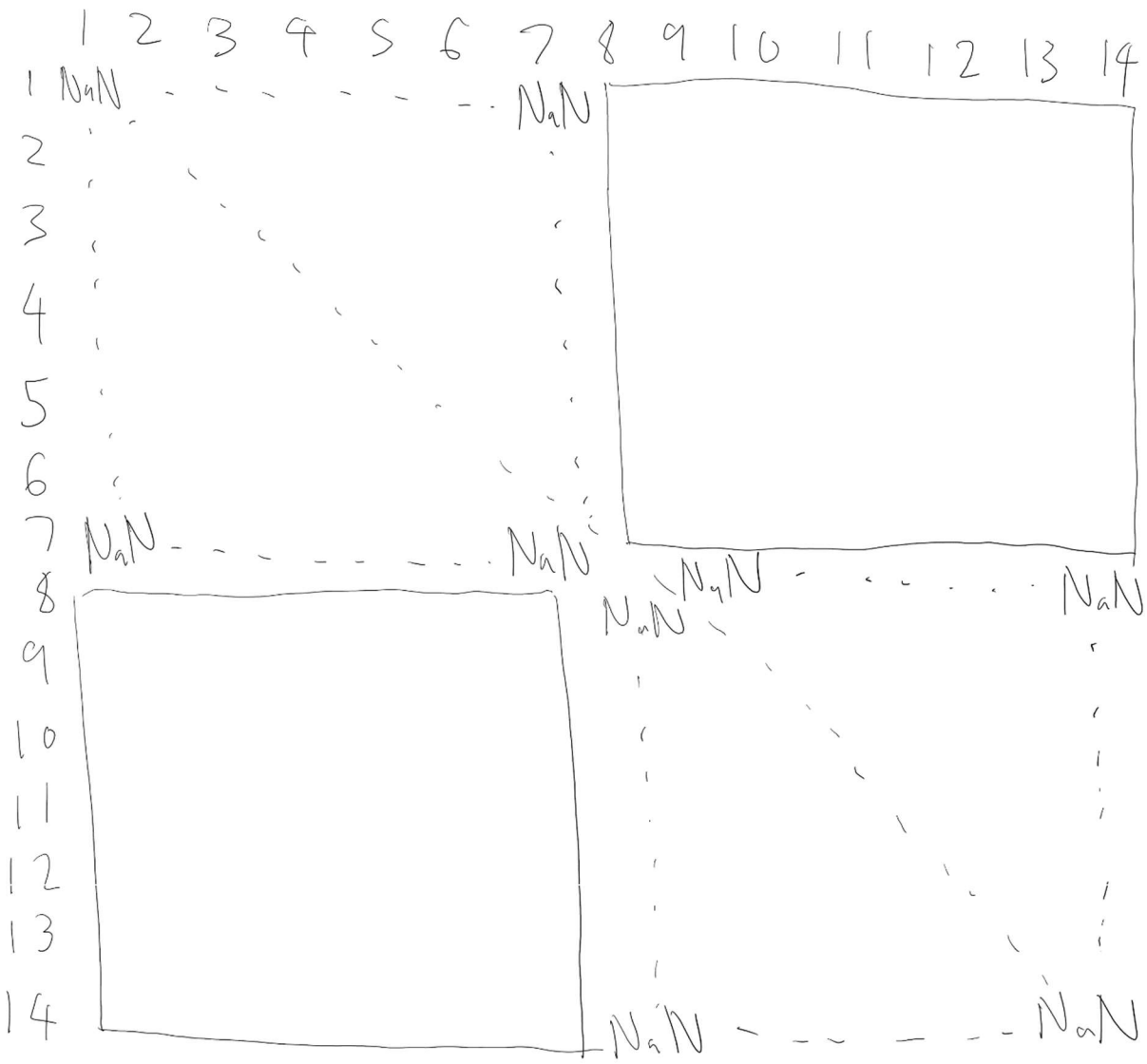

Figure 2. Structure of the connectivity/data matrix for left-right hemisphere connections. ROIs 1 to 7 are left hemisphere; ROIs 8 to 14 are right hemisphere. NaN values are present because left ROIs only connect to right ROIs and vice versa. NaN = not a number. Squares indicate where file values should be stored.

Lastly, the average_connectomes.m script averages the whole-brain connectomes generated from the preprocessing script, depending on the group specified. These connectomes can be visualized on matlab, with properties such as headings and colors modified using the GUI. You will have to fill in which subjects belong to which group in the script.

Graph DTI metrics vs ROIs on Excel

Using the excel spreadsheets generated from a_2D_to_3D_matrix.m, use the FA files to line graph the left, right, left-right ROIs against FA across your groups of interest (ex. PBTS vs HC). You will need to copy and paste your subject IDs and treatment group values from whatever database

you are working with into the spreadsheet and then sort them based on treatment group. This is because the matlab outputs do not come with subject identities and correspond to the entire list of subject files referenced from bash, which must correspond to the same order of subjects as the database. Once this is done, average the metrics per group and check that FA is consistent with literature/expected results. If so, proceed with NBS, otherwise check integrity of files.

Run NBS

Running NBS requires the following files:

STATISTICAL MODEL
1. Design Matrix - What subject belongs in what group
2. Contrast - How you're testing groups (ex. Group 1 > Group 2)

DATA
3. Connectivity Matrices - 3D matrix file of connections
4. Node Coordinates (MNI)
5. Node Labels (optional)

ADVANCED SETTINGS
6. Permutations - How many iterations NBS is run
7. Significance - p-value threshold to be considered significant
8. Method
9. Component Size

Use the help option on NBS to see examples of what file requirements are. Use NBS as method, set significance to 0.05, component size as extent, and start with threshold = 0.01 and ascend until significant p-values are reached, or the p-value maxes out at 1 (this is an indicator to stop running NBS). Design matrices will have to be manually made depending on your subjects and groups that are being analyzed. Contrast is best inputted as text directly into the NBS GUI instead of as a file because this is more convenient. Connectivity/data matrices are taken from the a_2D_to_3D_matrix.m outputs, which are the files you saved as .m into whichever folder you specified. Node coordinates and node labels depend on which networks you analyzed and will also have to be manually written as a text file. When statistically significant results are obtained, increase the threshold even more to see if a lower p-value can be obtained, without decreasing the number of significant node and edge differences. Save every significant result, as this will be used as an input in BNV.

Run stats, generate graphs

Use R or any statistical software to run t-tests or ANOVA + post-hoc test (ex. Tukey's HSD) on the significant results obtained from NBS. This streamlines your analysis so that stat testing is only done on the sig results from NBS, and not for every possible ROI combination per metric. Graph whatever is relevant for your project (ex. correlations with behavioural data, covariates).

Run BNV

Running BNV can generate both a visualization of significant connections on a semi-transparent brain, as well as a connectome for select ROIs. This requires an edge and node file. The node file is a text file that requires 6 columns, where the first 3 columns are MNI coordinates, the 4th and 5th columns are 1s, and the 6th column is the label (ex. R-PFC). The rows are the number of networks you have for analysis. This is very similar to the HCPCOG + HCPlabels used for NBS. For the edge file, use the saved significant results from NBS and load in Matlab. Specifically, select the nbs.NBS.test_stat file. For further information, go online and search for the BNV manual, which is contains a very thorough explanation of the inputs required to run BNV.

Concluding Remarks

With these instructions, hopefully you will be able to run a complete structural connectivity analysis! Or even better, improve my script! Having basic proficiency in bash, matlab, as well as the neuroimaging software listed previously would be beneficial in reducing the learning curve required to get through this pipeline. Lastly, if including me as a co-author, please use the name 'Zenan Tang' instead of 'Tommy Tang'. Best of luck!