

TECHNISCHE
UNIVERSITÄT
MÜNCHEN

FAKULTÄT
FÜR
PHYSIK

KLINIKUM RDI

RSLA Toolbox for Resting-State Lag Analysis of fMRI Data

-

Manual

Sascha Frölich

Munich, April 2018

Contents

1	Introduction	1
1.1	Getting Started	1
1.2	Performing Resting State Lag Analysis on Preprocessed Data	2
1.2.1	Global Signal Regression	2
1.2.2	TD Computation	2
1.3	Analysis of Results	3
1.3.1	Sum of Lag Threads	4
1.3.2	Panel “Plotting”	4
1.3.3	Convert matlab to nifti (Panel “.mat to .nii”)	5
1.3.4	Significance Test between Groups	6
1.3.5	Behavioral Correlates	6
1.4	Functions	8
1.4.1	create_mask_index	8
1.4.2	flipfile	9
1.4.3	readniftifileheader	9
1.4.4	write_nifti	9
1.4.5	combineCSV	9

Chapter 1

Introduction

This introduction will briefly explain how to use the RSLA toolbox for resting state lag analysis. This guide is by no means comprehensive, and the toolbox offers more information by hovering your mouse icon over fields and buttons, or by using the toolbox's console manual. This you can use by typing 'rsla' in the MATLAB console when in the toolbox directory, or after the toolbox directory is added to your MATLAB path after getting started (Sec. 1.1).

RSLA toolbox was tested under UBUNTU.

For bug reports do not hesitate to contact Sascha Frölich at [**sascha.froelich@gmail.com**](mailto:sascha.froelich@gmail.com).

1.1 Getting Started

In order to start using the toolbox, youj have to open MATLAB in the toolbox folder, or add the toolbox path to the MATLAB search path (`addpath(genpath('PathToToolbox/rsla'))`). Also, this toolbox uses the smoothing function of SPM, so make sure SPM is installed and added to the MATLAB path.

1.2 Performing Resting State Lag Analysis on Preprocessed Data

In order to perform resting state lag analysis on preprocessed data, you first need to compute the time delay matrix TD for each subject. You do this in the Initialization GUI. Type 'rsia init' to start that GUI.

1.2.1 Global Signal Regression

For this procedure, global signal regression (GSR) is advised. You can perform GSR on your data using the Initialization GUI's Batch Creator. Check the box 'Activate Batch Creator' on the upper right-hand side of the GUI, and choose 'Global Signal Regression'. Then, hit the button 'Load File(s)' and select all nifti files on which you want to perform GSR. All files need to be in the same directory. You then need to select one brainmask for each subject. The voxels over which the GSR shall be computed is defined by the non-zero voxels in the given GSR brainmasks. All brainmasks need to be in the same directory and will be automatically assigned to the nifti-files in alphabetical order. Last, choose a storage directory in the left-hand side of the GUI. This is where all GSR files will be stored. You can then hit the button 'Perform GSR' to perform GSR directly or 'Create Batch', which will put a batch-file in the storage directory. This file is scripted in bash and can be executed under UNIX-like systems.

The toolbox will create two files per subject:

GS_*: The Global Signal of the nifti file.

GSR*: The imaging file with the global signal regressed out. Use this file for further analysis.

1.2.2 TD Computation

Once you got all your preprocessing done, you can compute the subject-level TD matrices from your imaging data. In order to do this, select 'TD-Generation from nii-files' in

the Batch Creator of the Initialization GUI. In the left part of the GUI, hit 'Load File(s)' and choose the preprocessed nifti-files from which you wish to compute TD. Next, select 'Load Mask' and choose a brainmask. Type 'man maskfile' for more information. Next, choose the parameters "Lag", "r_min", "Lagged corr. p", and "TR (sec)". All but the last parameters can be freely chosen, while TR (repetition time) depends on your imaging data. Choose a storage directory. For the next step, you can either select 'Run Batch' or 'Create Batch Script'.

Run Batch

Upon hitting this button, TD computation will be performed on all selected nifti-files. The resulting TD matrices are stored in matlab-files in the previously chosen storage directory. Depending on the number of ROIs, this may take a long time (several hours).

Batch Script

This will create a bash file "Batchfile_*.sh" with a random sequence of numbers and characters at the end of the filename (i.e. a tag). The batch-file can be executed in bash on linux. The TD matrices are stored in matlab-files in the previously chosen storage directory. For each nifti file, a logfile (.txt) of the process of TD computation will be created in the storage directory. The logfile has the name of the original nifti file with the random ten character tag appended to it. In the process of TD computation, the script will generate and remove files from the storage directory, so do not interfere until the process is terminated. It will say so in the shell terminal.

1.3 Analysis of Results

For analysis of results, open the analysis GUI by typing 'rsia analyze'. In the top left corner of the GUI you can load matlab files containing TD. For most purposes, you will perform analyses on the second level, i.e. on the group level. In order to compute TD at the group level, load the first level files by selecting 'Load Files' in the top right corner of the

GUI in the 'Group Level TD' Panel. After selecting the first level matlab files containing the individual TDs, hit 'Create group TD mat-file'. This will create a matlab-file called "GROUPFILE.mat" which will be stored in the directory of the subject files which you just selected.

NB: The group-level file "GROUPFILE.mat" contains the field *pathfiles*. While you can move and rename the groupfile as you wish, from this point onwards removing the first level matlab files will require replacing the field *pathfiles* in the groupfile to the new directory.

After creating a group level file, load the file into the GUI by choosing 'Load File' (top left).

1.3.1 Sum of Lag Threads

Lag Threads are ambiguous. If vector v is a Lag Thread, so is vector $-v$. The recommended procedure is to create a weighted sum of Lag Threads and see what combination of Lag Threads correlates best with the Lag Projection of the same second level matrix TD. In order to do this, specify up to which Lag Thread you wish to compute the weighted sum by entering that number in the text field over the button.

1.3.2 Panel "Plotting"

The Panel 'Plotting' lets you plot your results. The buttons 'Plot Lagmap' and 'Plot 1D Lag Projection' let you plot those respective measures from the TD matrix in the matlab file which you loaded into the GUI. If PCA has already been performed on the file (i.e. Lag-Threads are already computed), you can also plot the respective Lag-Threads. The position of the plots are given in the 'i, j, k' textboxes and equivalent to the positioning system in fslview.

By selecting 'Plot Nifti' you can color plot a nifti file. You can create nifti files from matlab files in the panel '.mat to .nii' (Sec. 1.3.3).

NB: You can plot through a number of planes at the same time by using MATLAB notation "from:stepsize:to" in one of the fields i, j or k (for example: 10:5:70 to plot from slice 10 to slice 70 in steps of 5). **Do this only for one direction and uncheck the 'Plot' boxes in the other two directions..** You can save those plots by checking "Save Plots as png files".

NB: To obtain a group level lag projection,

1.3.3 Convert matlab to nifti (Panel ".mat to .nii")

For later analysis, it might be useful to have Lag Threads, Lag Maps and Lag Projection available as nifti files. In order to convert first level TD matrices from matlab to nifti, select via the radio buttons which measure you want to save as nifti file (Lag Thread, Lag Map or Lag Projection). Next, choose what kernel the nifti file(s) shall be smoothed with. The checkbox 'Create group-level nifti file' lets you decide whether you want the GUI to create a group level nifti file of the single subject nifti files you create. Then, select 'Create nifti'. The GUI will then guide you through your next steps.

For each matlab file, the toolbox will create two files:

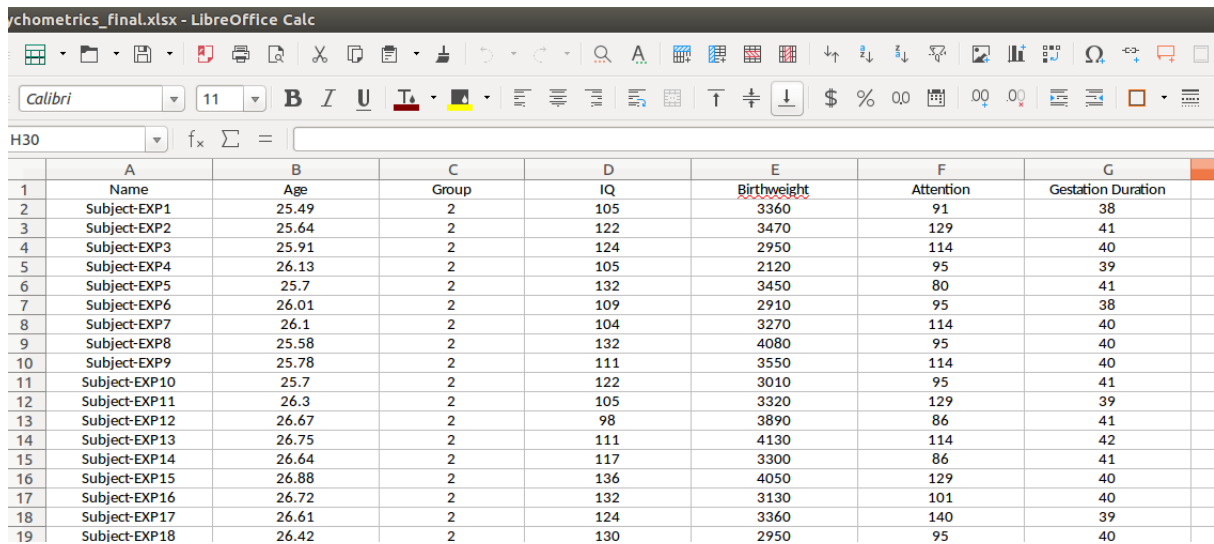
*_MeasureSpecification.nii: The nifti file of the chosen measure (indicated by Measure-Specification).

Sxsysz*_MeasureSpecification.nii: The smoothed nifti file of the chosen measure (indicated by MeasureSpecification). sx, sy and sz are the FWHM of the smoothing kernel in the three cardinal directions x, y and z.

1.3.4 Significance Test between Groups

In order to find differences in Lag Maps, Lag Threads and Lag Projections between groups, the advised method depends on the brainmask used to compute TD. If it is a continuous mask (like a gray matter mask), then it is recommended to convert your measure of interest (Lag Map, Lag Thread or Lag Projection) into nifti files using the '.mat to .nii' panel. You can then look for differences using the SPM toolbox. If the brainmask is discontinuous, SPM cannot be used for statistical significance testing. Use the options in the panel 'Significance Test' instead.

1.3.5 Behavioral Correlates



	A	B	C	D	E	F	G
	Name	Age	Group	IQ	Birthweight	Attention	Gestation Duration
1	Subject-EXP1	25.49	2	105	3360	91	38
2	Subject-EXP2	25.64	2	122	3470	129	41
3	Subject-EXP3	25.91	2	124	2950	114	40
4	Subject-EXP4	26.13	2	105	2120	95	39
5	Subject-EXP5	25.7	2	132	3450	80	41
6	Subject-EXP6	26.01	2	109	2910	95	38
7	Subject-EXP7	26.1	2	104	3270	114	40
8	Subject-EXP8	25.58	2	132	4080	95	40
9	Subject-EXP9	25.78	2	111	3550	114	40
10	Subject-EXP10	25.7	2	122	3010	95	41
11	Subject-EXP11	26.3	2	105	3320	129	39
12	Subject-EXP12	26.67	2	98	3890	86	41
13	Subject-EXP13	26.75	2	111	4130	114	42
14	Subject-EXP14	26.64	2	117	3300	86	41
15	Subject-EXP15	26.88	2	136	4050	129	40
16	Subject-EXP16	26.72	2	132	3130	101	40
17	Subject-EXP17	26.61	2	124	3360	140	39
18	Subject-EXP18	26.42	2	130	2950	95	40

Figure 1.1

RSLA lets you test for behavioral correlates of the latency in a given region. For this method, the measure of interest has to be converted into nifti files first. The behavioral correlates are read from a csv file, which can be created from a Microsoft Excel or LibreOffice Calc file. Here are example steps for the analysis of behavioral correlates.

Say you want to correlate the latency of the prefrontal cortex in the Lag Map of the poste-

rior insula with an IQ and an Age score. So far, you have computed TD for each subject, and you have one matlab file per subject.

1) Create one nifti file per subject. Each nifti file contains the Lag Map of the posterior insula. To do this, select 'Lag Map' from the radio buttons in the '.mnat to .nii' panel and enter the desired FWHM of the smoothing kernel. Select 'Create nifti' and choose all first level matlab-files. Next, choose a nifti file containing the seed region of the posterior insula. In that nifti file, all non-zero voxels are counted to the seed region. Last, choose the storage directory. In this directory, the new nifti files will be stored. After selecting the storage directory, conversion starts automatically.

2) For behavioral correlation of the prefrontal cortex with the IQ score, prepare a csv-file. This can be created from an excel file. The excel-file has to look like Fig. (1.1). The Subjects are listed in the leftmost column, starting from row 2. The names of the subjects in the first column need to be **identically present** in the nifti files' names. For example: The first subject in the excel file is called *Subject-EXPI*. Therefore, the name of the Lag Map nifti file can be *LagMap_Subject-EXPI_PosteriorInsula.nii*. This will enable the toolbox to match the row in the table with the corresponding nifti-file. The excel file needs then to be exported and stored as a csv-file (Use field delimiter ',' (comma)).

3) Now you can perform your analysis of behavioral correlates. Select the 'Psychometrics vs Lag' button in the 'Analysis' panel. Next, select the nifti files which you created in step 1. Then, select the csv-file which you created in step 2. In the next step, you will be able to choose one or more of the measures that are found in the csv-file. Choose 'IQ' and 'Age' for this example.

4) Select the seed region whose latency value you want to correlate with the chosen measures.

5) Choose your analysis method: Either **Two-Sample Pearson Correlation** or **Partial Correlation**.

Two-Sample Pearson Correlation This will compute the pearson correlation of the latency in the chosen seed region individually with all chosen measures.

Partial Correlation This will perform a partial correlation of the latency in the chosen seed region individually with all chosen measures, while correcting for all other chosen measures.

1.4 Functions

1.4.1 create_mask_index

This function lets you analyse the ROIs in your mask-file in nifti format.

`create_mask_index('PathToFile/maskfile.nii')` creates two files in the directory 'PathToFile' (the directory of the nifti-file):

maskfile.txt: This is a text-file which lists all ROIs in the selected nifti-file. It does this by interpreting voxels with the same values as one ROI. Therefore, ROIs might even be regionally disjoint. For each ROI it also gives the coordinates of one voxel within that ROI.

maskfile.mat: This is a mat-file containing a single vector. That vector contains all voxel values present in the maskfile.

1.4.2 flipfile

1.4.3 readniftifileheader

`readniftifileheader('PathToFile/Niftifile.nii')` returns a structure containing all header information of the nifti file as fields. This function only works for files with format nifti1. If your nifti files have nifti2 format, the function will let you know. More information about nifti file headers:

<https://brainder.org/2012/09/23/the-nifti-file-format/>
<https://brainder.org/2015/04/03/the-nifti-2-file-format/>

1.4.4 write_nifti

This function lets you easily create your own 3D and 4D nifti-files. It does this by copying the header information of a nifti file of your choice and then creating a new nifti-file with the same header. If the dimensions of the new data array or the voxel sizes differ from the file from which you copy the header, you have to specify those as arguments to this function. For more information, type "help write_nifti" in the MATLAB console and consult above links.

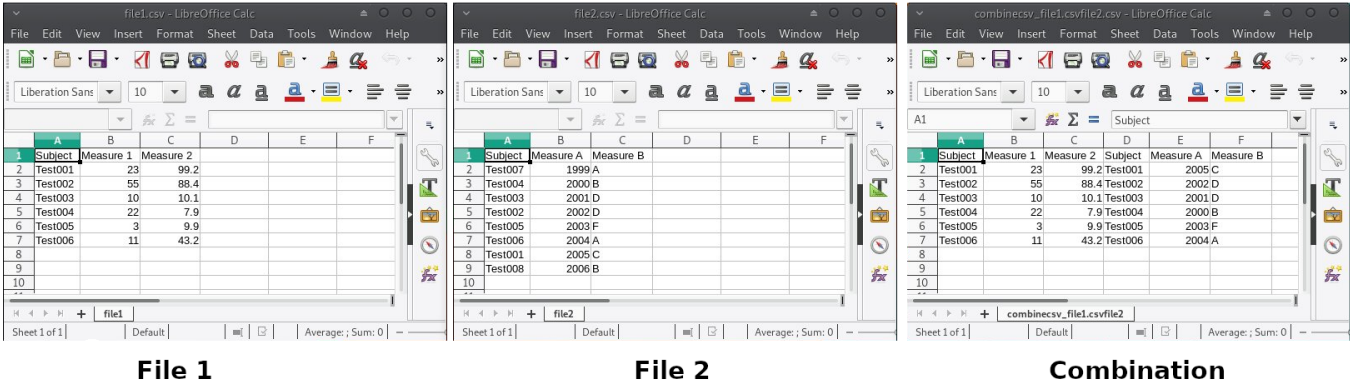
1.4.5 combineCSV

Inputs: None

Output: None

`combineCSV` merges two csv-files (LibreOffice Calc and Microsoft Excel both offer the possibility to export files as csv-files. Choose comma (",") as field delimiter). Upon running `combineCSV.m`, you will be asked to choose two csv-files (file 1 and file 2) and a storage directory for the combined csv-file. It will combine the csv file as follows:

Example 1



Example 2

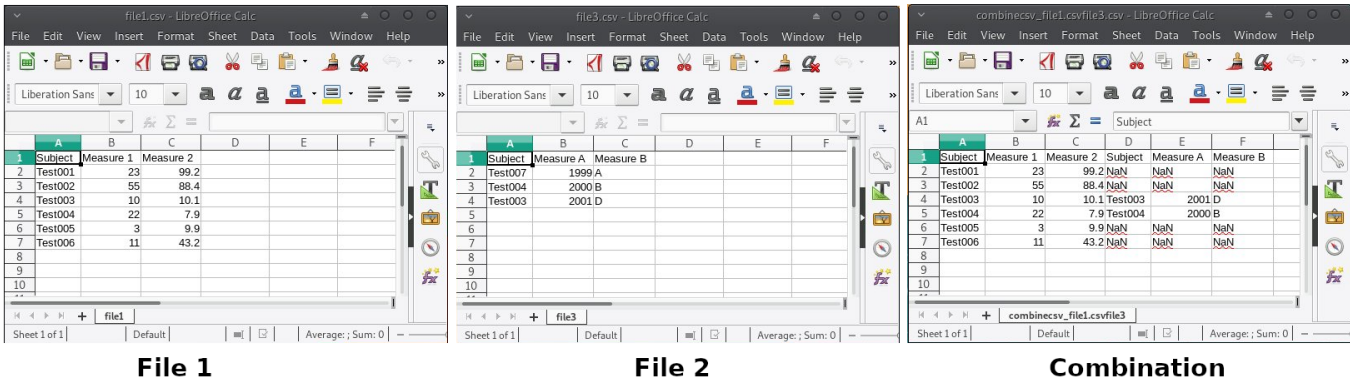


Figure 1.2: Two example results of function combineCSV.m.

For every row entry in the first column of file 1 (starting from row 2 as row 1 is usually used for column titles), it will look for a corresponding row entry in the first column of file 2. If it finds a corresponding entry, it will append the row from file 2 to the corresponding row in file 1. If it does not find a correspondence, then the row in file 1 is filled in with NaN entries. If file 2 contains rows that cannot be attributed to any rows in file 1, those rows will not be present in the combined csv-file.