Scripting with LIMO EEG

June 14

2013

I illustrate how to script with the LIMO EEG, keeping all the functionality of the toolbox. Rather than calling low level functions, I show which information are needed to create a LIMO.mat and call high level functions (or wrappers) which do all the hard work0

LIMO toolbox: LInear Modeling of EEG data.

‘Standard’ hierarchical modeling using SCRIPTING for mass univariate approach.

Cyril Pernet
cyril.pernet@ed.ac.uk

The LIMO toolbox is a set of Matlab functions primarily designed for hierarchical linear modeling, i.e. analyzes rely on statistical testing using the general linear model The 1st level analysis (per subject) relies on Ordinary Least Squares and 2nd level (group) analysis relies both on robust estimators like trimmed means to deal with outliers and on resampling techniques to ensure the validity of statistics even under non normality and heteroscedasticity of the parameter estimates, whilst keeping the type I error rate at the nominal level. [For now only 2nd level t-tests rely on trimmed means – other designs are being validated and will be available in version 1.5].

Contents

Before starting

LIMO Batch has been created to run your 1st level analysis effortlessly: it asks iteratively subjects folder, .set and variables and then run the analysis for all subjects. Even better, if the .set is in the subject folder and you use the same names across subjects for your categorical and continuous variables then you will be ask only once to select files. Now if you are not happy with limo_batch.m, you can also script.

Most of the LIMO functions take multiple inputs and often in 2 forms like (data, LIMO) or (data, parameter 1, parameter2, etc). The easiest way to scrip is (1) to create a LIMO.mat file containing the right information, which in turn will allow to call those functions easily and also to use the GUI at any time like e.g. to display results and (2) to use wrappers. Wrappers are meta-functions which call others. For instance, if you want, for 1 subject, to run a glm + bootstrap + tfce you could call each function (limo_glm1.m, limo_glm1_boot.m, limo_tfce.m), and each time create files to be saved on the disk. Alternatively you can create a LIMO.mat and call limo_eeg(4) which will do all of this. Below I describe how I script, i.e. using wrappers.

1st level analysis (for each subject)

A. Prepare the data

Create a structure LIMO

```
LIMO.Level                     = 1
LIMO.dir                       = directory for the analysis

LIMO.data.data_dir             = directory where to read them
LIMO.data.data                 = file name
LIMO.data.chanlocs             = import channel location information
LIMO.data.start                = when to start the analysis
LIMO.data.end                  = when to stop the analysis
LIMO.data.sampling_rate        = sampliong rate of the data
LIMO.data.Cat                  = Categorical variable(s)
LIMO.data.Cont                 = Continuous variable(s)
LIMO.data.neighbouring_matrix = matrix describing which electrodes
                                 are neighbourghs (not necessary
                                 unless bootstrap is used)
LIMO.design.fullfactorial      = 0/1 specify if interaction should
                                 be included
LIMO.design.zscore             = 0/1 zscoring of continuous regressors
LIMO.design.method             = 'OLS'
LIMO.design.type_of_analysis   = 'Mass-univariate'
LIMO.design.bootstrap          = 0/1 indicates if bootstrap should be
                                 performed or not (by default 0 for
                                 group studies)
LIMO.design.tfce               = 0/1 indicates to compute TFCE or not
```

By convention, we distinguish between two types of variables: the categorical variable and the continuous variables. The *categorical variables* are defined by a vector or matrix identifying groups of trials. Each column is considered as a factor whilst each value in each column is considered as a specific condition. The *continuous variables* are defined by a vector or matrix such as each column contains values characterizing each trials. The software handles from 1 to (N trials - Number of categorical + continuous - 1) predictors. Above this limit, there are more columns in X (design matrix) than rows in Y (trials) and no solution can be found.

B. Create the design matrix

To create your own design matrix see Appendix A, otherwise simply call limo_design_matrix to (1) create the design (2) update the LIMO.mat structure and (3) create new files and reorganize the data

```
[LIMO.design.X, LIMO.design.nb_conditions,
LIMO.design.nb_interactions, LIMO.design.nb_continuous] =
limo_design_matrix(Y, LIMO, 0);
```

At this stage LIMO is updated with

```
LIMO.design.X                = design matrix
LIMO.design.nb_conditions    = vector that returns the number of
                               conditions per factor e.g. [2 2 2]
LIMO.design.nb_interactions = vector that returns the number of
                               conditions perinteraction e.g. [4 4 4]
LIMO.design.nb_continuous    = scalar that returns the number of
                               continuous variables e.g. [3]
```

Also update manually LIMO with

```
LIMO.design.status           = 'to do'
LIMO.design.name             = name of the design (as you want)
```

Save LIMO.mat

C. Compute parameter estimates

Being in the right directory, simply call *limo_eeg(4)*

Analyzes are carried out such as the effect of experimental variables on the EEG time are tested for each electrode and time frames independently (in practice however the full time course is analyzed simultaneously electrode by electrode). Files previously created are updated. Depending on the design, files of the statistical results are also saved on the disk: Condition_effect_X (refers to a factor effect), Interaction_effect_X (refers to an interaction between factors), Covariate_effect_X (refers to the effect of a continuous regressor).

The LIMO structure is automatically updated

```
LIMO.design.weights         = matrix of trial weights (ones for OLS)
LIMO.model                  = information about the statistics
LIMO.model.conditions_df    = df [effect, error]
LIMO.model.interactions_df = df [effect, error]
LIMO.model.continuous_df    = df [effect, error]
LIMO.design.status          = 'done'
```

If you set LIMO.design.bootstrap =1, the bootstrap will also be computed
If LIMO.design.tfce = 1, TFCE will also be computed.

Note that if you already analyzed the data without bootstrap/tfce, you can simply update those fields by setting the relevant fields to 1 in the LIMO structure and recalling limo(4), as it will skip whatever has already been computed. For more details on the stats function called see Appendix B.

D. Get the results maps saved automatically

Simply call limo_eeg(5)

2nd level analysis (across subjects)

As for the 1st level, we have two steps to follow: (1) prepare the data and a LIMO.mat (i.e. do the job of *limo_random_select.m*) and (2) call the wrapper *limo_random_robust.m* (see Appendix C for low level functions used in this wrapper)

One-sample t-test

Organize the data with dimension electrode * time frames * subjects and save as Yr.mat. Prepare and save a LIMO.mat with that information:

```
LIMO.dir: where is the LIMO.mat and Yr
LIMO.data.chanlocs: chanel locations from the expected electrode file
LIMO.data.neighbouring_matrix: binary matrix of neighbourhood
LIMO.data.data: cells with the full name of the Betas or con files
LIMO.data.data_dir: cells with directories of the Betas or con files
LIMO.data.sampling_rate: should be the same across subjects
LIMO.data.trim1: 1st data point to analyze
LIMO.data.start: 1st data point to analyze in sec
LIMO.data.trim2: last data point to analyze
LIMO.data.end: last data point to analyze in sec
LIMO.design.bootstrap: nb of bootstrap to perform (0 if none)
LIMO.design.tfce: 0 or 1
LIMO.design.name: 'one sample t-test all electrodes'
LIMO.design.electrode: [] (or 1value or set of values for optimized
electrode analysis)
LIMO.design.X: []
LIMO.Level = 2;
```

Then call `limo_random_robust(1,Yr,parameter number,nboot,tfce)`
with
1 = a one-sample t-test
Yr = data (dim electrodes, frames, subjects)
parameter number = describe which parameters is analysed (e.g. 1 – used for naming only)
nboot = nb of resamples
tfce = set to 1 to compute TFCE

Two-samples t-test

Organize the data as two separate matrices with dimension electrode * time frames * subjects and save as Y1r.mat and Y2r.mat. Prepare and save a LIMO.mat with that information:

```
LIMO.dir: where is the LIMO.mat and Yr
LIMO.data.chanlocs: chanel locations from the expected electrode file
LIMO.data.neighbouring_matrix: binary matrix of neighbourhood
LIMO.data.data: 2 sets of cells e.g. {{1x10 cell}  {1x8 cell}} with
the full name of the Betas or con files
LIMO.data.data_dir: cells with directories of the Betas or con files
LIMO.data.sampling_rate: should be the same across subjects
LIMO.data.trim1: 1st data point to analyze
LIMO.data.start: 1st data point to analyze in sec
LIMO.data.trim2: last data point to analyze
LIMO.data.end: last data point to analyze in sec
LIMO.design.bootstrap: nb of bootstrap to perform (0 if none)
LIMO.design.tfce: 0 or 1
LIMO.design.name: 'one 'two samples t-test all electrodes'
LIMO.design.electrode: [] (or 1value or set of values for optimized
electrode analysis)
LIMO.design.X: []
LIMO.Level = 2;
```

Then call `limo_random_robust(2,y1,y2,parameter number,nboot,tfce)`
with
2 = two samples t-test
y1 = data (dim electrodes, frames, subjects)
y2 = data (dim electrodes, frames, subjects)
parameter number = describe which parameters is analysed (e.g. 33 – used for naming only)
nboot = nb of resamples
tfce = set to 1 to compute TFCE

Paired t-tests

Organize the data as two separate matrices with dimension electrode * time frames * subjects and save as Y1r.mat and Y2r.mat. Prepare and save a LIMO.mat with that information:

```
LIMO.dir: where is the LIMO.mat and Yr
LIMO.data.chanlocs: chanel locations from the expected electrode file
LIMO.data.neighbouring_matrix: binary matrix of neighbourhood
LIMO.data.data: cells with the full name of the Betas or con files
LIMO.data.data_dir: cells with directories of the Betas or con files
LIMO.data.sampling_rate: should be the same across subjects
LIMO.data.trim1: 1st data point to analyze
LIMO.data.start: 1st data point to analyze in sec
LIMO.data.trim2: last data point to analyze
LIMO.data.end: last data point to analyze in sec
LIMO.design.bootstrap: nb of bootstrap to perform (0 if none)
LIMO.design.tfce: 0 or 1
LIMO.design.name: 'paired t-test all electrodes'
LIMO.design.electrode: [] (or 1value or set of values for optimized
electrode analysis)
```

```
LIMO.design.X: []
LIMO.Level = 2;
```

Then call `limo_random_robust(3,y1,y2,parameter number,nboot,tfce)`
with
3 = paired t-test
y1 = data (dim electrodes, frames, subjects)
y2 = data (dim electrodes, frames, subjects)
parameter number = describe which parameters is analysed (e.g. 12 – used for naming only)
nboot = nb of resamples
tfce = set to 1 to compute TFCE

Regression

Prepare a LIMO.mat file

```
LIMO.dir: where is the LIMO.mat and Yr
LIMO.data.chanlocs: chanel locations from the expected electrode file
LIMO.data.neighbouring_matrix: binary matrix of neighbourhood
LIMO.data.data: cells with the full name of the Betas or con files
LIMO.data.data_dir: cells with directories of the Betas or con files
LIMO.data.sampling_rate: should be the same across subjects
LIMO.data.trim1: 1st data point to analyze
LIMO.data.start: 1st data point to analyze in sec
LIMO.data.trim2: last data point to analyze
LIMO.data.end: last data point to analyze in sec
LIMO.design.bootstrap: nb of bootstrap to perform (0 if none)
LIMO.design.tfce: 0 or 1
LIMO.design.name: 'regression analysis all electrodes'
LIMO.design.electrode: [] (or 1value or set of values for optimized
electrode analysis)
LIMO.design.X: []
LIMO.Level = 2;
```

Then call `limo_random_robust(4,y,X,parameter number,nboot,tfce)`
with
4 = regression analysis
y = data (dim electrodes, frames, subjects)
X = continuous regressor(s) (i.e. a matrix of continuous values)

Alternatively, to avoid some extra questions related to zscoring and running the analysis given the design matrix you can update the LIMO.mat file as follow

```
LIMO.design.type_of_analysis  = 'Mass-univariate';
LIMO.data.Cat                 = 0;
LIMO.data.Cont                = X;
LIMO.data.data_dir            = pwd;
LIMO.design.fullfactorial     = 0;
LIMO.design.zscore            = 1 (default) to zscore otherwise 0;
LIMO.design.status            = 'to do';
LIMO.design.method            = 'OLS';
```

Then update and create the design matrix

```
[LIMO.design.X, LIMO.design.nb_conditions,
LIMO.design.nb_interactions,LIMO.design.nb_continuous] =
limo_design_matrix(data, LIMO,1);
```

And call `limo_eeg(4)`

ANOVA/ANCOVA

Prepare the data with dimension electrodes * time frames * subjects and set a LIMO.mat file

```
LIMO.dir: where is the LIMO.mat and Yr
LIMO.data.chanlocs: chanel locations from the expected electrode file
LIMO.data.neighbouring_matrix: binary matrix of neighbourhood
LIMO.data.data: cells with the full name of the Betas or con files
LIMO.data.data_dir: N sets of cells with directories of the Betas or
con files for the N groups of subjects
LIMO.data.sampling_rate: should be the same across subjects
LIMO.data.trim1: 1st data point to analyze
LIMO.data.start: 1st data point to analyze in sec
LIMO.data.trim2: last data point to analyze
LIMO.data.end: last data point to analyze in sec
LIMO.design.bootstrap: nb of bootstrap to perform (0 if none)
LIMO.design.tfce: 0 or 1
LIMO.design.name: N-ways ANOVA all electrodes'
LIMO.design.electrode: [] (or 1value or set of values for optimized
electrode analysis)
LIMO.design.X: []
LIMO.Level = 2;
```

Then call `limo_random_robust(5,y,cat,cont,nboot,tfce)`
with

```
5     = N-way ANOVA/ANCOVA
y     = data (dim electrodes, frames, subjects)
cat   = categorical variable(s)
cont  = continuous regressors (covariates)
nboot = nb of resamples
tfce  = 1 run tcfe
```

Alternatively, to avoid an extra question related to running the analysis given the design matrix you can update the LIMO.mat file as follow

```
LIMO.design.type_of_analysis  = 'Mass-univariate';
LIMO.data.Cat                 = cat;
LIMO.data.Cont                = cont;
LIMO.data.data_dir            = pwd;
LIMO.design.fullfactorial     = 0 or 1;
LIMO.design.zscore            = 0 or 1;
LIMO.design.status            = 'to do';
LIMO.design.method            = 'OLS';
```

Then update and create the design matrix

```
[LIMO.design.X, LIMO.design.nb_conditions,
LIMO.design.nb_interactions, LIMO.design.nb_continuous] =
limo_design_matrix(data, LIMO,1);
```

And call `limo_eeg(4)`

Repeated Measures ANOVA

Prepare the data with dimension electrodes * time frames * subjects * repeated measures and set a LIMO.mat file

```
LIMO.dir: where is the LIMO.mat and Yr
LIMO.data.chanlocs: chanel locations from the expected electrode file
LIMO.data.neighbouring_matrix: binary matrix of neighbourhood
LIMO.data.data: cells with the full name of the Betas or con files
LIMO.data.data_dir: cells with directories of the Betas or con files
LIMO.data.sampling_rate: should be the same across subjects
LIMO.data.trim1: 1st data point to analyze
LIMO.data.start: 1st data point to analyze in sec
LIMO.data.trim2: last data point to analyze
LIMO.data.end: last data point to analyze in sec
LIMO.design.bootstrap: nb of bootstrap to perform (0 if none)
LIMO.design.tfce: 0 or 1
LIMO.design.name: 'Repeated measures ANOVA all electrodes'
LIMO.design.electrode: [] (or 1value or set of values for optimized
electrode analysis)
LIMO.design.X: []
LIMO.Level = 2;
```

Then call `limo_random_robust(6,y,gp,factor_levels,nboot,tfce)`
with

```
6 = Repeated measures ANOVA
y = data (dim electrodes, frames, subjects, measures)
gp = a vector defining gps like [111111222222333333]
factor_levels = a vector specifying the levels of each repeated
measures factor like [3 2]
nboot = nb of resamples
tfce = 1 run tcfe
```

Appendix

Appendix A. Create the design matrix

For some complicated types of design you might want to create your own design matrix. For this design matrix to be used with limo functions, a few things need to set.

Categorical variables are always on the left hand side of the matrix and are coded in LIMO.design.nb_conditions using a vector. For instance [2] means 2 columns, but [2 2] means 2 factors of 2 columns. The design matrix has thus 4 columns in this case, allowing testing the two main effects. LIMO.design.nb_interactions would be set to 0. However, it can also be set to 4 adding the interaction term. The design matrix has thus 8 columns (2 columns factor 1, 2 columns factor 2, 4 columns for interaction). See *limo_make_interactions.m* to automatically generate interaction columns from the main factors.

Continuous variables are on the right hand side of the design matrix and are coded as a scalar in LIMO.design.nb_continuous. For instance 3 means 3 variables (columns).

The constant term (ones) is coded in the last column.

Appendix B. Low Level functions used in 1st level analysis

limo_design_matrix: create design matrix and return outputs for limo_glm1
limo_make_interaction: from the data and a set of factors, create the interaction terms (used in limo_design_matrix)
limo_glm1: core of the stat computation
limo_glm1_boot: re-run the GLM under H0
limo_contrast: post-hoc tests from limo_glm1.m, allows
limo_tfce compute tfce scores for observed and bootstrapped data

Appendix C. Low Level functions used in 2st level analysis

limo_trimci one sample t-test using trimmed mean
limo_yuen_ttest: 2 samples t-test using trimmed means
limo_yuend_ttest: paired t-test using trimmed means
limo_glm1 core of the stat computation for regression/anova/ancova – called via limo_eeg(4)
limo_glm1_boot: re-run the GLM under H0 – called via limo_eeg(4)
limo_rep_anova Hotelling test for repeated measures ANOVA and gp x repeated measures