MethodsCore
FirstLevel Script Documentation
09/05/2012

Introduction

The FirstLevel script lets you specify a first level design. The script is as flexible as the SPM GUI, but will save you a lot of time. The FirstLevel Script assumes you've already created a MasterData file that describes the trials of your experiment. Once that is done, there are four major sections to fill out:

1. Image and subject information
2. MasterData file and condition information
3. User-specified regressor information
4. Contrast information

Additionally, there is an Advanced Section to the script, which will not be covered in this documentation. There is a brief description of the advanced features in the script itself. If you want to use any of these features, we recommend you contact MethodsCoreHelp@umich.edu for information. Filling in these four sections in the script is mostly self-explanatory, and there is extensive help in the script itself. Below we'll walk you through the steps that require more explanation.

Visual Tutorials

There are a series of video tutorials available online. It may be useful to view these while following along in this document.

1 Introduction	http://youtu.be/FHWJgJFLMU8?hd=1
2 MasterData File	http://youtu.be/6OoV_SPkzSU?hd=1
3 Subjects & Runs	http://youtu.be/epfNW4u-z9c?hd=1
4 Conditions, Onsets, & Durations	http://youtu.be/xsLuvXqwZy4?hd=1
5 Parametric Regressors	http://youtu.be/SbXZ0CpxerE?hd=1
6 User-Specified Regressors	http://youtu.be/D5kw4RJY1GU?hd=1
7 Contrasts	http://youtu.be/EPSB2FObnyY?hd=1
8 Conclusion	http://youtu.be/CrRaQELaKNw?hd=1

Getting Started

Copy the FirstLevel_template script from your Methods Core directory into your local experiment directory. This is the file you'll be editing.

Part 1: MasterData Files

A MasterData file is a single CSV file that contains all the information about your experiment. All the information that you would normally specify in the SPM GUI subject-by-subject and condition-by-condition is now specified in a single file. This makes analysis easier, more flexible, and more transparent. You have some flexibility in setting up a MasterData file, but I will describe one way of doing this that you will probably want to follow.

	A	B	C	D	E	F	G	H	I
1	Subject	Run	Trial	Onsets	TrialType_12	TrialDur	TrialType_14	TrialDur_0	
2	1	2	3	4	5	6	7	8	
3	50011	1	1	0	1	0.7	1	0	
4	50011	1	2	5.496	1	0.776	1	0	
5	50011	1	3	10.991	2	0.872	3	0	
6	50011	1	4	16.5	2	0.833	2	0	
7	50011	1	5	21.996	2	0.884	4	0	
8	50011	1	6	24.497	2	0.75	2	0	
9	50011	1	7	27.997	1	0.811	1	0	
10	50011	1	8	30.498	1	0.802	4	0	
11	50011	1	9	33.998	1	0.645	4	0	
12	50011	1	10	37.497	2	0.981	3	0	
13	50011	1	11	40.997	2	1.257	2	0	
14	50011	1	12	47.491	2	0.916	2	0	
15	50011	1	13	48.994	1	0.62	1	0	
16	50011	1	14	55.501	1	0.711	1	0	

Figure 1

Look at Figure 1. I have two header rows here: Row 1 identifies the type of column and row 2 provides the column number (you'll see why this is useful a bit later). The remaining content of the MasterData file is a trial-by-trial exhaustive list of every trial in the experiment. In the above spreadsheet, I have listed for each trial: the Subject (col 1), Run (col 2), Trial Number (col 3). I also have information for each trial about its Onset, Condition, and Duration but I'll get to that a bit later.

The MasterData file is one long list. When I get to the end of a run, I start the next run and continue listing trials. The onsets restart because SPM wants to know the onsets relative to the start of a run.

98	50011	1	*new run*	96	439.495	*onsets*	2	1.228	2	0
99	50011	1		97	447.997	*restart*	1	0.789	1	0
100	50011	1		98	450.499		1	0.605	1	0
101	50011	1		99	459.002		2	0.825	2	0
102	50011	1		100	461.503		2	1.288	2	0
103	50011	2		101	0		2	1.483	2	0
104	50011	2		102	4.485		2	1.274	2	0
105	50011	2		103	7.985		2	0.949	2	0
106	50011	2		104	13.493		1	0.71	1	0
107	50011	2		105	15.995		2	1.752	2	0

If you start a new subject, then you list the new subject number, and restart run number and trial number (and onsets).

195	50011	2		193	463.992		1	0.638	1	0
196	50011	2		194	466.493	*new subject*	2	1.049	2	0
197	50011	2		195	470.991		2	1.186	2	0
198	50011	2		196	480.492		1	0.686	1	0
199	50011	2		197	485.987		1	0.518	1	0
200	50011	2		198	492.494		2	0.849	2	0
201	50011	2		199	494.995		2	0.899	2	0
202	50011	2		200	500.491		2	0.95	2	0
203	50012	1		1	0		1	0.517	1	0
204	50012	1		2	5.496		1	0.545	1	0
205	50012	1		3	10.991		2	0.976	3	0
206	50012	1		4	16.5		2	1.02	2	0
207	50012	1		5	21.996		2	1.154	4	0
208	50012	1		6	24.497		2	1.16	2	0
209	50012	1		7	27.997		2	1.075	3	0

run and trial number restart

One important note: Other than header rows, there can be only numbers, and no strings, in the MasterData csv file. So give your subjects a numerical ID, at least for MasterData purposes. This has to do with Matlab's weird way of parsing csv files, and we just have to live with this limitation. Now we can get around this just a bit. The script lets you have not only header rows but also header columns. You set these up by setting the variables below.

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%Number of rows and columns to skip when reading the MasterData csv file
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
MasterDataSkipRows = 2;
MasterDataSkipCols = 0;
```

set this to 1 to make the first column a header column

Another important note: Do your CSVs in excel, not Open Office as Open Office

Conditions, onsets, and durations

For each trial, we want to be able to assign to it a ***Condition***, an ***Onset***, and a ***Duration***. Recall that for an SPM analysis this information is required for every single trial. Assigning the ***Onset*** and ***Duration*** are easy – they are simply listed in Figure 1 in columns 4 and 6.

The ***Condition*** assignment is done with a numerical method. First, I assign each condition a number starting with 1. So let's say I have two conditions, Congruent and Incongruent. Then I would assign Congruent as 1 and Incongruent as 2. Next, for each trial, I assign it the appropriate condition using these numbers. For example, in Figure 1, column 5, I assigned every trial a 1 or a 2 based on whether it is Congruent or Incongruent. A bit later, we will 'point to' the relevant columns in the MasterData file as we set up our FirstLevel script.

Part 2: Setting up the FirstLevel_template Script

You specify a lot of experiment-related information in the FirstLevel script. Most of this should be self-explanatory and there is help in the script itself.

One important issue is setting up paths. You set up the paths to a lot of things (path to functional images, output path, MasterData file, regressor file if you have one, etc…). Setting up paths is done through a standardized 'Path Template' system. See the **Path Template Documentation** in the root directory of the Methods Core repository for help on this. We'll be using this method for setting up paths in pretty much all Methods Core software.

Setting up Subjects and Runs

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%% A list of run folders where the script can find the images to use
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
RunDir = {
    'run_05/';
    'run_06/';
};

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%% The list of subjects to process
%%% The format is 'subjectfolder',subject number in masterfile,[runs to include]
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
SubjDir = {
    '5001/Tx1',50011,[1 2];
    '5028/Tx1',50281,[2];
    '5029/Tx1',50291,[1 2];
};
```

runs to include

name subject folder as it appears on the disk

subject number as it appears in the MasterData file

RunDir lists the run names as they appear on the disk. SubjDir lists the subjects to include in the analysis. It has three columns that are explained above. Notice that 'runs to include' (i.e., column 3 of SubjDir) indexes the rows of RunDir. So in the above example, we are omitting run_05 for subject 5028 Tx1.

Setting up conditions, onsets, durations, etc..

In the FirstLevel_template Script, you 'point to' the relevant columns the MasterData file we set up earlier (see Figure 2 below). The script will then 'find' the information from the MasterData file. It will go to just the subjects, runs you tell it to, and will ignore the rest. Notice you also name your conditions, and you need to make sure the order of names corresponds to the order of numbers you used in the MasterData File.

But let's say I change my mind and want to model my data with four conditions: Congruent, Incongruent, Non-response, and Errors. Then I create a new condition column in the MasterData file with the numbers 1 through 4 that represent these conditions. This is column 7 in Figure 1. Then in the FirstLevel script (see Figure 2 below), I replace The CondColumn with 7, and list the new condition names in the ConditionName variable.

```
93      %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
94      %%%Column number in the MasterData file where subject numbers are located
95      %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
96 -    SubjColumn = [1];
97
98      %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
99      %%%Column number in the MasterData file where run numbers are located
100     %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
101 -    RunColumn = [2];
102
103     %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
104     %%% Column number(s) in the MasterData file where conditions numbers are
105     %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
106 -    CondColumn = [5];
107
108     %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
109     %%% Column number in the MasterData file where your Onset times are locat
110     %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
111 -    TimColumn = [4];
112
113     %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
114     %%% Column number in the MasterData file where your Durations are located
115     %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
116 -    DurColumn = [6];
117
118     %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
119     %%% List of conditions in your model
120     %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
121 -    ConditionName = {
122         'Congruent';          <-- name your conditions here
123         'Incongruent';
124     };
125
```

Figure 2

Parametric Regressors

A parametric regressor is a regressor that modulates activity on a trial-by-trial basis. Adding a parametric regressor is easy. Just add a new column in the MasterData file that specifies the value the parametric regressor takes for each trial.

	A	B	C	D	E	F	G	H	I	J
1	Subject	Run	Trial	Onsets	TrialType_12	TrialDur	TrialType_14	TrialDur_0	DistractionLevel	
2	1	2	3	4	5	6	7	8	9	
3	50011	1	1	0	1	0.7	1	0	10	
4	50011	1	2	5.496	1	0.776	1	0	5	
5	50011	1	3	10.991	2	0.872	3	0	6	
6	50011	1	4	16.5	2	0.833	2	0	3	
7	50011	1	5	21.996	2	0.884	4	0	5	
8	50011	1	6	24.497	2	0.75	2	0	2	
9	50011	1	7	27.997	1	0.811	1	0	10	

parametric regressor

Then you list the name of the parametric regressor and its associated column in the FirstLevel script.

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%% If you are including any Parametric regressors in your model
%%% syntax: 'Parameter Name', column number in MasterData file thatcontains
%%% the parameter value
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
ParList = { ...
        'DistractionLevel',9;
};
```

User-Specified Regressors

A user-specified regressor (the terminology is from SPM) is a regressor that modulates activity on a scan-by-scan basis. You have two methods for specifying this.

1. Master File Method. Create a new file just like your MasterData file. The difference is the number of rows per subject is not the number of *trials* in the experiment, but rather the number of *scans* in the experiment. Otherwise, it is the same idea. List the regressor values in these rows. In the FirstLevel script, you 'point to' the columns of the regressor_MasterData file, and assign names to the regressors, etc…
2. Subject-Specific Motion File Method: The most common scan-by-scan regressors are your motion regressors. So the FirstLevel script lets you specify scan-by-scan regressors by specifying a variable called RegFileTemplates. Each line of RegFileTemplates refers to a single regressor file, followed by a number indicating the number of columns you'd like to use from that file (use Inf to automatically use all the columns). This is followed by a RegDerivatives that lets you choose any or all of the rows of RegFileTemplates to also have first derivative terms calculated for the regressors and also included in your model.

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%  Location of user specified regressor files. To use these make sure
%%%  column 2 of RegOp is set to 1 above.
%%%
%%%  Variables you can use in your template are:
%%%       Exp         = path to your experiment directory
%%%       Subject     = folder name of current subject
%%%       Run         = folder name of current run
%%%        *          = wildcard (can only be placed in final part of template)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
RegFileTemplates = {
    '[Exp]/Subjects/[Subject]/func/[Run]/scripttest_ppi1v3.csv',Inf;
    '[Exp]/Subjects/[Subject]/func/[Run]/mcflirt*.dat',Inf;
    '[Exp]/Subjects/[Subject]/func/[Run]/fdOutliers.csv',Inf;
    '[Exp]/Subjects/[Subject]/func/[Run]/frameOutliers.csv',Inf;
};

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%% Indices from RegFileTemplates for which regressor files should have
%%% automatic first derivatives calculated and also included
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
RegDerivatives = [2];
```

You control which method to use by setting the RegOp variable. You can use either, both, or neither method.

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%% User Specified Regressors
%%% The value in the first column controls the master file method
%%% The value in the second column controls the subject-specific motion file method.
%%% 0 = don't use method
%%% 1 = use method
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
RegOp = [0 1];
```

The setting above says don't use the master file method, but do use the subject-specific motion file method.

Contrasts

Let's say you had an experiment with two conditions, Congruent and Incongruent, and 6 user-specified regressors. Here is what the contrast might look like:

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%% List of contrasts to add to the estimated model
%%% Format is 'Name of contrast' [Cond1 Param1...N]...[CondN Param1...N] [Reg1...RegN]
%%% You need to properly balance/weight your contrasts below as if it was just one run/session
%%% The script will handle balancing it across runs
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
ContrastList = {
    'C'            1  0   [0 0 0 0 0 0];
    'I'            0  1   [0 0 0 0 0 0];
    'C-I'          1 -1   [0 0 0 0 0 0];
    'I-C'         -1  1   [0 0 0 0 0 0];
    'AllTrials'   .5 .5   [0 0 0 0 0 0];

};
```

condition 1 *condition 2* *six scan-by-scan regressors*

Let's say you now have 2 conditions and a single parametric regressor, and 6 user-specified regressors. Here is what the contrast might look like.

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%% List of contrasts to add to the estimated model
%%% Format is 'Name of contrast' [Cond1 Param1...N]...[CondN Param1...N] [Reg1...RegN]
%%% You need to properly balance/weight your contrasts below as if it was just one run/session
%%% The script will handle balancing it across runs
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
ContrastList = {
    'C'            [1  0] [0  0] [0 0 0 0 0 0];
    'C_r'          [0  1] [0  0] [0 0 0 0 0 0];
    'I'            [0  0] [1  0] [0 0 0 0 0 0];
    'I_r'          [0  0] [0  1] [0 0 0 0 0 0];
    'C-I'          [1  0] [-1 0] [0 0 0 0 0 0];
    'C-I_r'        [0  1] [0 -1] [0 0 0 0 0 0];
    'I-C'          [-1 0] [1 0]  [0 0 0 0 0 0];
    'I-C_r'        [0 -1] [0 1]  [0 0 0 0 0 0];
    'AllTrials'    [.5 0] [.5 0] [0 0 0 0 0 0];
    'AllTrials_r'  [0 .5] [0 .5] [0 0 0 0 0 0];
};
```

_r refers to the contrast for the parametric regressor

condition 1 *condition 1 paramtric regressor* *condition 2* *condition 2 parametric regressor* *six scan-by-scan regressors*

Brackets (i.e., [and]) that are placed around the condition/parametric regressor pair and also around the six scan-by-scan regressors are not for cosmetic purposes. The script uses the brackets to parse your ContrastList so do not omit the brackets!

Some other rules to keep in mind are (these are for most standard analyses, and there are exceptions):

- Contrasts against an implicit baseline (e.g., 'C') – positive numbers sum to 1, no negative numbers
- Contrasts of two conditions against each other (e.g., 'I-C') - positive numbers sum to 1, negative numbers sum to 1

If you follow these rules, your estimated betas will correspond to percent signal change, which is a good thing. The script helps you follow these rules by balancing your numbers across runs. Here is how it works. You specify the contrasts as if there were only one run and you do not include any trailing zeros

for the SPM-created run-related regressor. The script will automatically expand/rescale the numbers for the number of runs in your experiment. For example, if there are three runs, all the numbers in the contrasts above would be multiplied by 1/3, and additional trailing zeros for the SPM-created run-related regressors (one per run) would be automatically be created.

To repeat, there is also an Advanced Section, with a LOT of other features. For example, you can have run specific contrasts. You reset SPM defaults. You can drop conditions that appear just a few times. And many others. If you want to use any of these features, we recommend you contact MethodsCoreHelp@umich.edu for information.