cvBMA Toolkit Manual

This manual describes the use of MATLAB code associated with the following publication:

> Soch J, Meyer AP, Haynes JD, Allefeld C (2017): "How to improve parameter estimates in GLM-based fMRI data analysis: cross-validated Bayesian model averaging", NeuroImage, in review.

If you use this method in published research, please refer to it as "cross-validated Bayesian model averaging (cvBMA)" or simply "the cvBMA approach" and cite the paper mentioned above. The cvBMA toolkit is compatible with SPM8 and SPM12 and can be obtained from the GitHub profile of the corresponding author as a ZIP file called "cvBMA-master.zip". Internally, this version is referred to as "MACS V13b", "MACS V0.9b" and "Release cvBMA".

Structure of the Toolkit

After download, the ZIP file should be unpacked and added to the MATLAB path. Generally, functions in the toolkit folder are organized into two groups with three sub-groups each:

- *interface functions* – allowing the user to perform desired operations:
  - `MA_*.m` : model assessment
  - `MC_*.m` : model comparison (not included in this limited release)
  - `MS_*.m` : model selection/averaging
- *mathematical functions* – allowing the toolkit to perform statistical analysis:
  - `ME_*.m` : model estimation
  - `MD_*.m` : many distributions (not included in this limited release)
  - `MF_*.m` : more functions

Users of the toolkit will only need to call interface functions, but developers are highly encouraged to have a look at and play around with the mathematical functions as well. The cvBMA toolkit requires SPM8 or later running on MATLAB 7.4 or later. No additional MATLAB toolboxes are needed.

Application of the Toolkit

The cvBMA methods are written for general linear models (GLMs) applied to functional magnetic resonance imaging (fMRI) data and directly build on "SPM.mat" files created using Statistical Parametric Mapping (SPM), Versions 8 or 12.

The overall goal of cvBMA is model averaging. Therefore, at least two models ($M \geq 2$) have to be fitted to at least one subject ($N \geq 1$). If this has been done, model inference proceeds in two steps: model assessment and model averaging.

Step 1: First-level model assessment

In the first step, each model in each subject (i.e. on the first level) is evaluated using the cross-validated log model evidence (cvLME) in each voxel (i.e. voxel-wise). To this end, the following commands have to be executed:

```
>> load [...]\SPM.mat      % loads an SPM.mat
>> MA_cvLME_multi(SPM)     % calculates the cvLME
```

This will produce a voxel-wise cvLME map in the folder where the first-level model corresponding to "SPM.mat" is located, here abbreviated as `[...]`. Note that for this function call, it does not matter

- whether this model is already *estimated* – in case it is not, this is done using `MA_GLM_AR_only`; however, the model must have been *specified* – or
- whether this model is a multi-session GLM or a single-session GLM – in the latter case, the toolkit automatically calls `MA_cvLME_single`.

Step 2: First-level model averaging

In the second step, cvLMEs for all models in each subject are transformed into posterior probabilities (PP) which are then used to combine parameter estimates from different models into averaged parameters using (voxel-wise) Bayesian model averaging (BMA). For this purpose, a MATLAB batch must be created using the SPM Batch Editor. After clicking on "Batch" in the SPM menu window (top left), simply select

- in SPM8: SPM >> Stats >> Bayesian Model Selection >> BMS: Maps; or
- in SPM12: SPM >> Stats >> Bayesian Model Selection >> BMS: Maps (Inference).

Here, it is sufficient to specify the subject-, session- and model-wise cvLME maps under "Data"; all other fields can be left as default or empty. Then, this batch has to be saved and the following commands have to be executed:

```
>> load [...]\batch.mat                  % loads the SPM batch
>> MS_BMA_group(matlabbatch,params)      % performs group BMA
```

In this call, `params` is an $M \times P$ matrix (M – number of models, P – number of parameters) indexing which regressor (column) in which GLM (row) belongs to which common model parameter. The order of the models has to follow the order in which the cvLME maps are entered into the processing batch. For example, in the case of three models ($M = 3$) and four parameters to be averaged ($P = 4$), this matrix could look like this:

```
>> params = [1, 2, 3, 4; 1, 3, 5, 7; 1, 4, 7, 10]

     1     2     3     4
     1     3     5     7
     1     4     7    10
```

Looking at the third column, this would indicate that the 3rd regressor in the 1st model, the 5th regressor in the 2nd model and the 7th regressor in the 3rd model are the same parameter, so that their estimates can be averaged. Once the parameter matrix is specified, group-level BMA analysis will produce averaged parameter maps for each subject, aggregating these different estimates of the same parameter using cvLME maps generated in the first step.

(Step 3: Second-level statistical analysis)

Once the averaged parameters or BMA estimates have been generated, they can be used for second-level statistical analysis. After clicking on "Specify 2nd-level" in the SPM menu window (top left), simply specify the appropriate model – with the only difference to include *averaged* parameter maps instead of parameter estimates from a *particular* first-level model.

Summary: The cvBMA toolkit pipeline

To improve usability of the cvBMA toolkit, the toolkit folder contains a MATLAB script called "cvBMA_Pipeline.m" as a template for cvBMA-style analysis of study data arranged in a subject-model folder hierarchy. For further advice on the cvBMA technique, have a look into the referenced paper or contact the corresponding author.

Hands-On Example

In this section, we describe a simple example for application of the cvBMA toolkit. Since we don't supply data with our toolkit, we will work with an SPM template data set that can be downloaded from the internet. Because this is a single-subject data set, there is no real second-level analysis. We will however perform an ad-hoc parameter estimate check in order to get an idea of how the cvBMA technique works.

Step 0: Pre-processing and model estimation

Please download the SPM8 Manual or SPM12 Manual and work through Chapter 29/31 until the end of Section 29.3/31.3. At this point, you will have pre-processed the data and estimated two first-level GLMs, a "categorical model" and a "parametric model". These models share several similarities and exhibit two differences:

- The categorical model uses two HRF derivatives (time derivative and dispersion derivative) while the parametric model doesn't use HRF derivatives.
- The parametric model includes two parametric modulators in quadratic expansion while the categorical model doesn't use parametric modulators.
- Both models have the same regressors for the four experimental conditions.

After these preliminary analyses, the categorical model is located in `DIR/categorical` and the parametric model is located in `DIR/parametric` where `DIR` is some folder on your computer.

Step 1: First-level model assessment

Model assessment for these two models can be carried out as follows:

```
>> load DIR/categorical/SPM.mat
>> MA_cvLME_single(SPM)
>> load DIR/parametric/SPM.mat
>> MA_cvLME_single(SPM)
```

After that, a file "MA_cvLME_Ky_11.nii" has been created in each model folder.

Step 2: First-level model averaging

Model averaging requires a MATLAB batch to be created using the SPM Batch Editor:

- Click “Batch” in the SPM menu window (top left).
- Select “SPM >> Stats >> Bayesian Model Selection >> BMS: Maps”.
- Highlight “Data”, click “New: Subject”. Highlight “Subject”, click “New: Session”.
- Highlight “Models”, click “Select Files” and select
  - `DIR/categorical/MA_cvLME_Ky_11.nii` as well as
  - `DIR/parametric/MA_cvLME_Ky_11.nii`.
- All other options will not be used and can be left as default.
- Save this batch as `DIR/BMA_design.mat`.

As this is a single-session data set, we only input two cvLME maps for two models. In the case of a multi-session data set, session-wise cvLME maps (ending on “_S1.nii”, “_S2.nii”, “_S3.nii” etc.) have to be entered, because cvBMA operates session-wise, i.e. within each session.

Before we actually perform BMA, find out about the order of regressors in the GLM:

- Click “Review” in the SPM menu window (top left).
- Select `DIR/categorical/SPM.mat`, click “Done”.
- Observe that the four conditions are the 1st, the 4th, the 7th and the 10th regressor.
- Click “Review” in the SPM menu window (top left).
- Select `DIR/parametric/SPM.mat`, click “Done”.
- Observe that the four conditions are the 1st, the 2nd, the 5th and the 6th regressor.

Based on these observations, type into the MATLAB command window:

```
>> params = [1, 4, 7, 10; 1, 2, 5, 6]

     1     4     7    10
     1     2     5     6
```

With this parameter matrix, model averaging can be carried out as follows:

```
>> load DIR/BMA_design.mat
>> MS_BMA_group(matlabbatch,params)
```

After that, a folder “MS_BMA_subject_as” with BMA estimates (named after the first model’s parameter indices) has been created in `DIR`.

(Step 3: Ad-hoc parameter estimate check)

In place of a second-level analysis, we will now perform a little check of the parameter estimates in order to understand what cvBMA exactly does. First, let's calculate the absolute difference between the first parameter estimates from the two models:

- Click "ImCalc" in the SPM menu window (top left).
- Highlight "Input Images", click "Select Files" and select
  - `DIR/categorical/beta_0001.img` as well as
  - `DIR/parametric/beta_0001.img`.
- Highlight "Output Filename", click "Edit Value" and enter `beta_0001_diff.nii`.
- Highlight "Output Directory", click "Select Files" and select `DIR/MS_BMA_subject_as`.
- Highlight "Expression", click "Edit Value" and enter `abs(i1-i2)`.
- Save this batch as `DIR/BMA_check.mat` and run it!

Next, we want to visualize our results in a convenient way:

- Type `MF_visualize` in the MATLAB command window.
  (This diagnostic function belongs to the cvBMA toolkit!)
- As input images, select
  - `DIR/categorical/beta_0001.img` and
  - `DIR/parametric/beta_0001.img` and
  - `DIR/MS_BMA_subject_as/beta_0001_BMA.img`.
- As overlay image, select
  - `DIR/MS_BMA_subject_as/beta_0001_diff.img`.
- As overlay threshold, enter ">1".
- As plot type, select "bar".
- As LineSpec, enter "b".
- As x-axis labels, enter `{'cat', 'para', 'avg'}`.
- As y-axis limits, enter `[]`.
- As plot title, enter "categorical vs. parametric vs. average".

In the SPM graphics window (right), you should now see a picture similar to that in Figure 1.

Summary: A cvBMA hands-on example

This hands-on example is summarized as "cvBMA_HandsOn.m" in the cvBMA toolkit folder.

Results: Categorical vs. parametric vs. averaged model

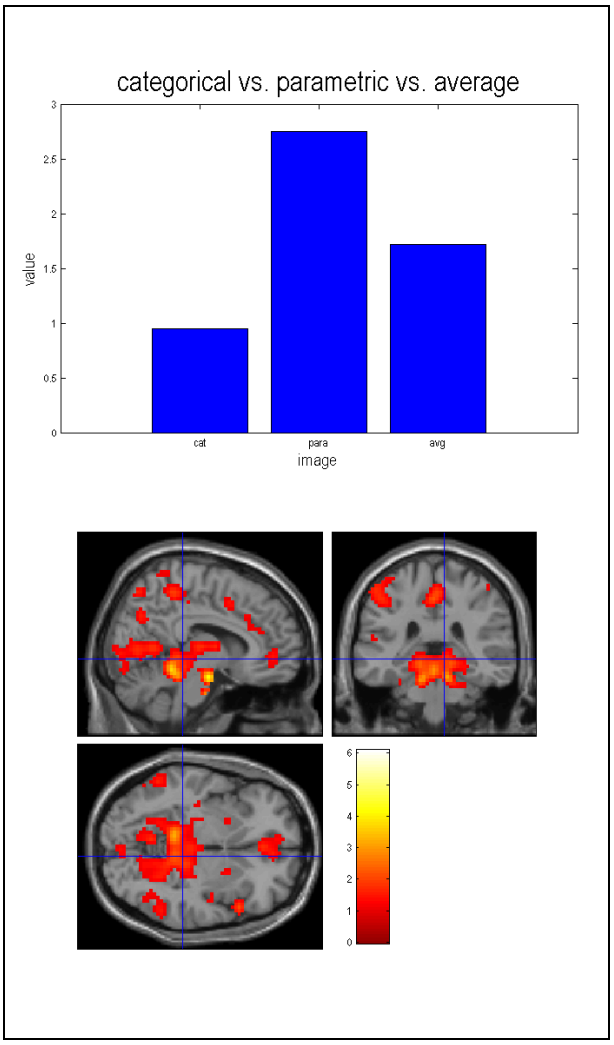

Figure 1. *Comparison of categorical model, parametric model and Bayesian model averaging (BMA).* The user can browse through the brain in the lower panel to observe what happens in the upper panel. We observe two things: First, there are many voxels in which there is considerable difference between parameter estimates. Colored voxels in the lower panel correspond to a difference of at least 1. An absolute value difference however does of course not imply a significant statistical difference! Second, the averaged parameter (right bar) is always between the parameter estimates of the categorical (left bar) and the parametric model (center bar). Often, the data are so decisive (and cvLME difference is so high) that the BMA estimate is practically equal to one of the model's estimates. Try to find a voxel in which the average is right in the middle between the individual model estimates (like in the figure)!

Closing Remarks

Maybe or maybe not, this little toolkit will be part of a greater toolbox on Model Assessment, Comparison and Selection (MACS) in the future. Internally, this version would then be referred to as "MACS V1.0" and "MACS R2017a".

Joram Soch, corresponding author and toolbox developer.

A. Meyer, **J.-D. Haynes** & **C. Allefeld**, co-authors.