

Table of Contents

Region of interest analysis with MarsBaR.....	1
Goals.....	1
Prerequisites.....	1
Installing MarsBaR.....	1
Installing the AAL ROIs.....	1
Working your way round the help system.....	2
The example dataset.....	2
The first step: defining the ROI.....	2
Defining a functional ROI.....	2
Review the ROI.....	3
Refining the ROI.....	3
Defining a box ROI.....	4
Combining ROIs.....	4
Writing the ROI as an image.....	4
Running the ROI analysis.....	5
Stage 1: choosing the design.....	5
Stage 2: extracting the data.....	5
The optional stage: setting the autocorrelation.....	6
Stage 3: estimating the model.....	6
Assessing results in MarsBaR.....	6
Basic results: the statistic table.....	6
Comparing fast and slow events – the difference between run 1 and run 3.....	7
Using a structural ROI.....	7
Running an analysis using structural ROIs.....	8
Batch mode.....	8
Working with event time courses.....	9
Set and review the model.....	9
Extract data and estimate the model.....	9
Plot a fitted event time course for events in each session.....	9
Plot mean fitted event time course.....	11
Plot FIR estimate of the event time course.....	11
The end.....	12

Region of interest analysis with MarsBaR

Goals

1. Install MarsBaR
2. Learn to work with the online help
3. Define and combine ROIs
4. Run statistical analyses on ROIs
5. Run MarsBaR batch scripts
6. Extract and plot event-related time courses

Prerequisites

1. Standard installation of SPM
2. MarsBaR example dataset, with preprocessing already run.
3. An estimated three session model for the MarsBaR example dataset (see the `run_s3_model` batch script in the `batch` subdirectory).
4. A utility to unzip compressed archives (Windows Explorer in XP, WinZip, etc).

Installing MarsBaR

First download MarsBaR 0.37 from The Marsbar Sourceforge Page

Save it into a temporary folder.

Unpack it into this folder. You now have a new subdirectory called `marsbar-devel-0.37`.

Start matlab. Add the `marsbar-devel-0.37` directory to your matlab path:

```
addpath c:\Temporary_folder\marsbar-devel-0.37
```

Start SPM, and start MarsBaR

```
>> spm_fmri
>> marsbar
```

Installing the AAL ROIs

Download the AAL ROI archive into your temporary directory. Unpack the AAL library in the same directory; it will create a new directory, called something like `marsbar-aal-0.2`.

The library contains ROIs in MarsBaR format that were anatomically defined by hand on a single brain matched to the MNI / ICBM templates. The ROI definitions are described in:

Tzourio-Mazoyer N, Landeau B, Papathanassiou D, Crivello F, Etard O, Delcroix N, et al. (2002). Automated anatomical labelling of activations in SPM using a macroscopic anatomical parcellation of the MNI MRI single subject brain. *NeuroImage* 15: 273–289.

Working your way round the help system

Open a web browser (please don't let that be Internet Explorer. I bet it is).

Chose File -> Open (or whatever)

Navigate to `c:\Temporary_folder\marsbar-devel-0.37\doc`, and select `index.html`.

From that page, select the `marsbar` link, and from that page select `marsbar` again. This gives you the help documentation from the main MarsBaR program file. Note that you can click on links to find the help for all the MarsBaR functions. The same help is also available online for the latest version at:

<http://marsbar.sourceforge.net/doc-devel/latest/>

As always with matlab, for help with the `marsbar.m` function, you can also type

```
>> help marsbar
```

at the matlab prompt.

There is also a MarsBaR FAQ

The example dataset

The example data are taken from an experiment described in an HBM2003 conference abstract: Matthew Brett, Ian Nimmo-Smith, Katja Osswald, Ed Bullmore (2003) Model fitting and power in fast event related designs. *NeuroImage*, 19(2) Supplement 1, abstract 791

The data consist of three EPI runs, all from one subject. In each run the subject watched a computer screen, and pressed a button when they saw a flashing checker board. An “event” in this design is one presentation of the flashing checker board.

We did this experiment because we were interested to see if events at fast presentation rates give different activation levels from events that are more widely spaced. Each run has a different presentation rate. We randomized the times between events in each run to give an average rate of 1 event every second in run 1, 1 event every 3 seconds for run 2, and 1 event every 10 seconds for run 3.

The first step: defining the ROI

We (your kind organizers) have already run an SPM model for run 2 (and run 1 and run 3). Now we need to find an activation cluster in the visual cortex.

Go to the MarsBaR window, and click on ROI definition.

Defining a functional ROI

We are going to define the functional ROI using the SPM analysis for run 2. Select “Get SPM cluster(s)...”: from the menu. This runs the standard SPM results interface. Use the file selection window that SPM offers to navigate to the `sess2/SPM2_ana` directory. Select the `SPM.mat` file and click Done. Choose the `stim_hrf t` contrast

from the SPM contrast manager, click Done. Then accept all the default answers from the interface, like this:

Prompt	Response
mask with other contrasts:	no
title for comparison	stim_hrf
p value adjustment to control	none
threshold { T or p value }	0.05
& extent threshold { voxels }	0

Now you have run the Get SPM cluster(s) interface, you should have an SPM activation map in the graphics window:

Meanwhile, you may have noticed there is a new menu in the SPM input window for saving ROIs.

Another thing you may not have noticed is that the matlab working directory has now changed to the `sess2/SPM2_ana`. SPM2 does this to be able to keep track of where its results files are.

Move the red arrow in the SPM graphics window to the activation cluster in the visual cortex. You can do this by dragging the arrow, or right-clicking to the right of the axial view and choosing goto global maxima.

When the red arrow is in the main cluster, click on the Write ROI(s) menu in the SPM input window and select Write one cluster.

After you have selected Write one cluster, MarsBaR asks for details to save with the ROI, which are a description, and a label. Both provide information about the ROI for statistical output and display. The label should be 20 characters or so, the description can be longer. For the moment, accept the defaults, which derive from the coordinates of the voxel under the red arrow and the title of the contrast:

Prompt	Response
Description of ROI	stim_hrf cluster at [-9.0 -93.0 -15.0]
Label for ROI	stim_hrf_-9_-93_-15

After this, MarsBaR offers a dialog box to give a filename for the ROI. By default the offered filename will be `stim_hrf_-9_-93_-15_roi.mat` in the `sess2/SPM2_ana` directory. For simplicity, why not accept the default name and click Save to save the ROI.

Review the ROI

We can now review this ROI to check if it is a good definition of the visual cortex. Click on the ROI definition menu in the MarsBaR window, and select View.... Choose the ROI and click Done. Your ROI should be displayed in blue on an average structural image:

Refining the ROI

Now we have reviewed the ROI, we see that the cluster does include visual cortex, but there also seems to be some connected activation lateral and inferior to the primary visual cortex. Ideally we would like to restrict the ROI to voxels in the primary visual cortex.

We can do this by defining a box ROI that covers the area we are interested in, and combining this with the activation cluster.

Defining a box ROI

To decide on the box dimensions, click around the ROI in the view interface and note the coordinates of the cross-hairs that are shown at the top of the bottom left panel. This may suggest to you, as it did to us, that it would be good to restrict the ROI to between -20 and $+20$ mm in X, -66 to -106 mm in Y, and -20 to $+7$ mm in Z.

To define this box ROI, click on ROI definition, and choose Build..., . You will see a new menu in the SPM input window:

From the menu, select Box (ranges XYZ). Answer the prompts like this:

Prompt	Response
[2] Range in X (mm)	-20 20
[2] Range in Y (mm)	-66 -106
[2] Range in Z (mm)	-20 7
Description of ROI	box at -20,0>XYZ
Label for ROI	box_x_-20:20_y_-106:-66_z_-20:7
Filename	box_x_-20_20_y_-106_-66_z_-20_7_roi.mat

The last three values here are the defaults. To check this is as you want it, choose ROI Definition, View, select both of `box_x_-20_20_y_-106_-66_z_-20_7_roi.mat` and `stim_hrf_-9_-93_-12_roi.mat`, in that order, and click Done.

You should see the box in blue, with the activation cluster overlaid in red.

We now need to combine the two ROIs, to select only those voxels that are shared by the box and the activation cluster.

Combining ROIs

Choose ROI Definition, Transform... A new menu comes up in the SPM input window. Choose Combine ROIs; select both the box and the cluster ROIs, click on Done. The prompt now asks for a function with which to combine the ROIs. In this function, the first ROI you selected is `r1`, and the second ROI is `r2`. Here we want to get the overlap, and this is represented by the logical AND operator, which is `&` in Matlab. Enter the function `r1 & r2`

After this, accept the default description, set the label to something like “Trimmed stim run 2”, and save the ROI as `trim_stim_roi.mat`.

Writing the ROI as an image

Just for practice, let us write our new ROI as a binary image. You might want to do this so you can review the ROI using another program, such as the excellent MRICro (www.mricro.com). Click on ROI definition, then Export.... Select image from the new menu in the SPM input window, and choose the new `trim_stim_roi.mat` as the ROI to export. Another menu appears, asking for a Space for ROI image. The three options are Base space for ROIs, From image, or ROI native space.

In our case, the data are spatially normalized, and so are in the space of the MNI template. The MNI template space is the default base space for ROIs, so select Base space for ROIs, choose a directory to save the image, and accept the default filename for the image, which should be `trim_stim`. You can check this has worked, by finding the SPM buttons window, selecting Display, and choosing the new `trim_stim.img`.

Running the ROI analysis

First, let us estimate the activation within the ROI for the first run. There are three stages to the analysis.

1. Choosing the design
2. Extracting the data
3. Estimating the design model with the data

To this basic list I will also add:

- Setting the autocorrelation options

We have already processed the example data to create an SPM model for all three EPI runs, so we already have a design made for the first run. We are going to use this design and the `trim_stim` ROI to extract ROI data from the functional scans. Then we will use the design and the extracted data to estimate the model.

Stage 1: choosing the design

Click on the Design button in the MarsBaR window. Select the Set design from file option in the design menu and choose the `SPM.mat` file in the `sess1/SPM2_ana` directory. MarsBaR loads the design into memory and displays the design matrix in the SPM graphics window.

Stage 2: extracting the data

Before we can run the model, we need to extract the ROI data from the functional scans. This brings us to the data menu.

Choose Data – Extract ROI data(default); the GUI will ask you to select one or more ROIs files; select the `trim_stim_roi.mat` file. MarsBaR starts to whirr. As it whirrs, it will:

1. Take each image in the design (you had already set the default design from the design menu);
2. Extract all the data within the ROI for each image, to give voxel time courses for each voxel in the ROI.

When it has finished, MarsBaR will calculate a new summary time course for each ROI. The summary time course has one value per scan, per ROI; by default, this new time course is made up of the means of all the voxel values in the ROI. For example, if there are only 5 voxels in the ROI, the first value in the summary time series will be the mean of the 5 voxel values for scan 1, the second value will be the mean of the 5 voxel values for scan 2, and so on. You can change the method of summarizing voxel data using the Statistics, Data summary function item in the MarsBaR options interface.

As MarsBaR extracts the data you will see its progress printed to the matlab console. When the extraction is done, the data is kept in memory. You can save the data to disk if you want using the Save data to file option on the data menu. Now we have the design and the data we can estimate the model.

The optional stage: setting the autocorrelation

There are a few problems with the autocorrelation estimation in SPM2. For this reason MarsBaR includes the option of using the autocorrelation algorithm from Keith Worsley's fmristat program. This option is faster and more robust than the default in SPM2. To use the fmristat algorithm, choose add/edit filter from the the Design menu:

Prompt	Response
High pass filter?	specify
cutoff period (secs)	60
Correct for serial autocorrelations?...	fmristat AR(n)
fmristat AR model order	2

This sets an AR(2) model for the data.

Stage 3: estimating the model

Click on the Results menu in the MarsBaR window. Choose the first item on the menu, Estimate results. MarsBaR takes the default design and the extracted data, and runs the model. There are more progress reports to the matlab console; finally you see the suggestion that you use the results section for assessment.

Assessing results in MarsBaR

Basic results: the statistic table

Let us start the assessment by getting some t and F values for the effects in the design. Click on the Results button in the MarsBaR window:

We next need to specify a contrast. In our case the contrast is very simple: just a 1 in the column for the HRF regressor used to model the visual event. Usually the contrast will be more complicated, and you may have already entered it for a previous SPM or MarsBaR analysis. The Import contrasts option allows you to get contrasts from a previous analysis. To show how it works, click on this option. The SPM file selection window should appear. Navigate to the sess1/SPM2_ana directory, and select the SPM.mat file there. The SPM contrast manager comes up, showing all the F and t contrasts in the SPM.mat file. Select the stim_hrf t contrast, and click Done. MarsBaR will put this contrast into the current estimated design. Here we only selected one contrast, but you can select many contrasts by dragging the mouse, shift clicking etc. (depending on your system).

Now click on the Statistic table option in the MarsBaR results menu. Select the stim_hrf contrast and click Done. The results will print out in a rather ugly fashion in the Matlab window. You might want to enlarge your Matlab window to stop the text wrapping in an annoying way.

At the left you see the contrast name. Under this, and to the right, MarsBaR has printed the ROI label that you entered a while ago. The t statistic is self explanatory, and the uncorrected p value is just the one-tailed p value for this t statistic given the degrees of freedom for the analysis. The corrected p is the uncorrected p value, with a Bonferroni correction for the number of regions in the analysis. In this case, we only analyzed one region, so the corrected p value is the same as the uncorrected p value. MarsBaR (like SPM), will not attempt to correct the p value for the number of contrasts, because the contrasts may not be orthogonal, and this will make a Bonferroni correction too conservative.

There is also a column called Contrast value. For a t statistic, as here, this value is an effect size. Remember that a t statistic consists of an effect size, divided by the standard deviation of this effect. Here our contrast is very simple, containing only a single 1, so the contrast value is the same as the value of the first parameter in the model. The value of this parameter will be the best-fitting slope of the line relating the height of the HRF regressor to the fMRI signal. This effect size measure is the number that SPM stores for each voxel in the con_0001.img, con_0002.img ... series, and these are the values that are used for standard second level / random effect analyses. Just for practice, let us also run an F contrast. Click Statistic table again, choose the effects of interest contrast, click Done:

Now the Contrast value has become the Extra SS. This is a measure of the variance that would be added to a model that does not contain the effects in the contrast. The F statistic is this measure, adjusted for the number of effects, and divided by the residual variance for the whole model. There is no simple way of using this Extra SS value in second level analyses.

Comparing fast and slow events – the difference between run 1 and run 3

Our results so far show that there is indeed a highly significant effect of visual stimulation on the visual cortex, even for very frequent events. This is not a Nature paper so far. To make things a bit more interesting, we can compare this effect, from run 1, with the effect in run 3, for which the events were much less frequent.

Click on Design in the MarsBaR window, then Set design from file. Choose SPM.mat from sess3/SPM2_ana. Now we need to extract the data; select Extract ROI data (default) from the data menu. MarsBaR will ask you if you want to save the previous data. Why not say 'no' for the moment. Next choose trim_stim_roi.mat again. When that is done, run Estimate results from the Results menu. Again choose 'no' when asked if you want to save the previous estimated design. In the end, you get a new statistic table.

You can see that the contrast value – which is proportional to the change in signal for a single event – is greater for run 3 than for run 1. Despite this, the t statistic for run 3 is lower than for run 1. One explanation for this is that there are many more events in run 1, so the estimate of signal change per event is more reliable (has less variance).

Using a structural ROI

So far we have used a functional ROI. This has the advantage that it is usually well tuned to the subject we are analysing. The disadvantages are that we have had to use a whole run of data to define the ROI, which we would have preferred to be able to analyze, and that functional ROIs can be noisy, when the activation signal is not strong. An alternative is to use the anatomy of the brain to estimate the location of functional areas.

Using anatomical ROIs can work well for areas that are naturally defined by brain structure, such as the subcortical nuclei, or the primary sensory and motor cortices, where the functional areas are closely linked to the position of large and relatively invariant sulci. Outside these areas, it can be difficult to define functional areas using anatomy alone. The problems are compounded when anatomical ROIs are defined on one subject, and applied to another, because there is great variability between subject in sulcal anatomy.

In the example experiment, subjects responded with a key-press each time they saw the flashing checker board. We might therefore be interested to know the level of activation in the putamen. This would be a good candidate for an anatomical ROI, because the putamen can be accurately defined on a structural scan, and does not vary much between subjects after spatial normalization. The AAL ROI library contains a definition of the left and right putamen for a single subject after spatial normalization. The images from our subject have been spatially normalized, so the AAL definition of the putamen will probably give a reasonable approximation to the putamen

in our data.

Running an analysis using structural ROIs

is exactly the same as running the analysis with the functional ROI. Select Design from the MarsBaR menu, and Set design from file. Choose sess1/SPM2_ana/SPM.mat. Click on Data, Extract ROI data (default). When you are asked for ROI file, navigate to the AAL directory, select MNI_Putamen_L_roi.mat and click Done. When the data extraction is done, choose Results, Estimate results and wait till MarsBaR has done its thing. Select Results, Statistic table, enter the stim_hrf contrast, as shown in Figure 17, above. Repeat the same procedure, using the AAL MNI_Putamen_R_roi.mat ROI.

The subject responded with their right hand, so we expected that the right putamen would have less signal than the left.

Batch mode

MarsBaR has a powerful batch mode. It is also relatively simple. Here we will just rerun our first ROI analysis above using batch mode.

Go to the matlab (>>) prompt. Type or paste:

```
spm_mat = spm_get(1, 'SPM.mat', 'Select SPM.mat for run 2');
roi_mat = spm_get(1, '*roi.mat', 'Select trimmed ROI for run 2');
```

The SPM file window comes up twice, asking you for the SPM.mat and ROI files. The code above just gets the file names for the SPM.mat analysis file for run 2, and the ROI file. Type:

```
spm_mat
roi_mat
```

and you will see the paths for the two files.

Now we can batch. Type or paste the following into the matlab window:

```
D = mardo(spm_mat);
R = maroi(roi_mat);
Y = get_marsy(R, D, 'mean');
D = autocorr(D, 'fmrstat', 2);
E = estimate(D, Y);
```

What do these lines do? Do you recognize this from the beginning of the tutorial?

Next we set the simple HRF contrast:

```
[E contrast_no] = add_contrasts(E, 'stim_hrf', 'T', [1 0 0]);
```

and finally we evaluate the contrast:

```
stat_struct = compute_contrasts(E, contrast_no);
```

Type `stat_struct` in the matlab window. All the information from the contrast is available in this structure. Can you identify the contrast P value? Check the help for the `compute_contrast` function using the online help to

see what the `stat_struct` structure includes.

For more batch examples, see the MarsBaR FAQ.

There is also an example batch script in the example data set batch directory, called `run_tutorial.m`. You won't be surprised to hear that this is a batch script that runs most of the steps in this tutorial, as well as extracting and plotting reconstructed event time courses.

Working with event time courses

We will often want to estimate the shape of the event time courses within an ROI. For this we will use the three-session model provided by running the `run_s3_model` batch file from the MarsBaR example dataset. This is just a single model including all three of the experimental sessions.

Set and review the model

Choose:

1. Design – Set design from file.
2. Design – Add/edit filter for FMRI design

Prompt	Response
High pass filter?	specify
cutoff period (secs)	60
Correct for serial autocorrelations?...	fmrstat AR(n)
fmrstat AR model order	2

Then choose Design – Explore.

You should see the design in the SPM graphics window. Note that there is text at the bottom of the window reminding you of the filter you've chosen.

Extract data and estimate the model

1. Data – extract ROI data (default). Select the `trim_stim` roi.
2. Results – Estimate results

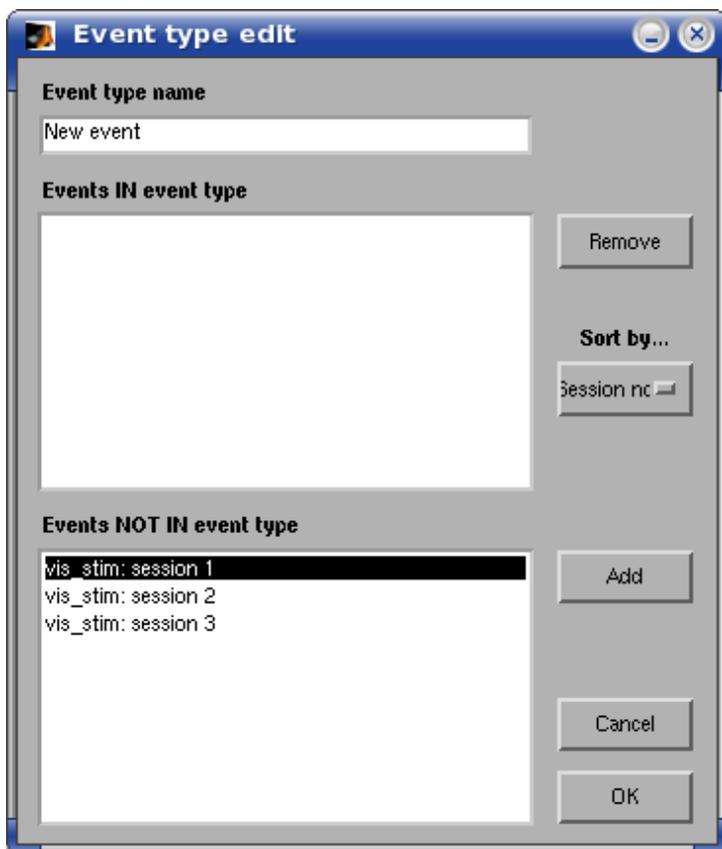
Plot a fitted event time course for events in each session

1. Results – Fitted event time course

Now you get the *event type management window*:



For now we have no event types. An event type is an event or combination of events in your model. To start, let us add an event type for each session, for the visual stimulation event. Click on the New button. You should see the *event type editing window*:



1. Click on the vis_stim: session 1 event in the bottom list box, and click Add.
2. Change the event type name to vis_stim_1
3. Click OK

Repeat these steps for session 2 and 3 (giving vis_stim_2 and vis_stim_3).

You now have 3 event types in the event type management window. Drag the mouse to select all 3 and click OK. Enter 0 for Event duration

You now have 3 event plots, one for each session. Note that the event shape varies a little from session to session. This is so because we have used the HRF and temporal derivative as our model of the event – and this allows some very constrained variation of the event shape, with different values of the temporal derivative parameter.

Plot mean fitted event time course

We may want to average across events in different sessions, or even across events in the same session. The event type interface allows us to do this.

1. Click Results – Fitted event time course
2. Click New in the event type manager
3. Drag select all 3 events (sessions 1 to 3) and click Add
4. Enter the name as `vis_stim_average`
5. Click OK
6. Click OK (the `vis_stim_average` event type is already selected)
7. Enter event duration of 0

You will see the average time course in the graphics window.

We could have achieved the same effect by using Results – Add event types by name. This finds all the events with the same names in the design, and makes event types which are the average of those events. Try it now:

1. Results – Add event types by name
2. Results – Fitted event time course

You will see a new event type called `vis_stim` in the event types management window. Click the edit button to see the events that have been included.

Plot FIR estimate of the event time course.

We have seen the fitted event time course, but we had a very constrained of the shape of the hemodynamic response for each event (the canonical hemodynamic response function and its temporal derivative). Now we would like to see a less constrained best estimate of the shape of the response following a single event. For this we use the FIR model. For more information on the FIR model, see the relevant section in the MarsBaR FAQ. In brief, the FIR is a way of doing event–related averaging that allows for overlap between events. In fact, if there is no overlap between events, the results of the FIR model are identical to event–related averaging.

1. Results – FIR event time course
2. Select `vis_stim_1` through `vis_stim_average` and click OK

Prompt	Response
Signal units...	Percent
Bin length (secs)	2.0273
No of bins	12

You will then get FIR plots for all three sessions, and the average, in the graphics window, in units of percent signal change.

Question: why don't the FIR plots begin at zero?

Again, see the MarsBaR FAQ for examples of how to batch the extraction of time courses.

The end

If you have time, you might want to have a look at the plot functions in the data menu. You might also want to look at the Frequencies – (event+data) option in the Design menu. This can be useful in deciding on the correct high-pass filter.

If not, then – all done.

-- MatthewBrett – 23 Mar 2005

I	Attachment 	Action	Size	Date	Who	Comment
	mars_event_type.png	manage	8.6 K	28 Mar 2005 – 21:48	MatthewBrett	MarsBaR event type management window
	mars_event_type_edit.png	manage	12.2 K	28 Mar 2005 – 21:50	MatthewBrett	MarsBaR event type edit window