

rfxplot — User Manual

Jan Gläscher

contact: glascher@hss.caltech.edu

February 16, 2009

Contents

1	Preliminaries	3
1.1	Abbreviations	3
1.2	Acknowledgements	3
2	Overview and Features	3
3	Download and Installation	4
3.1	Supporting Toolboxes and Functions	5
3.1.1	Mandatory Toolboxes	5
3.1.2	Optional Toolboxes	5
3.2	Support	6
3.3	How to cite rfxplot in your publication	6
4	General Concepts of rfxplot	6
5	The Plot Configuration Dialogue	8
6	The Output of rfxplot	26
7	Some Background Information	28
7.1	Voxel Selection	29
7.2	Splitting of Parametric Modulators	31
7.3	Computation of event-related BOLD responses (PSTHs)	33

8	Non-interactive usage of rfxplot	35
8.1	Description of the <code>opts</code> struct	35
8.2	Howto setup rfxplot for non-interactive usage	37
9	Goddies and Known Issues (and their solutions)	39
9.1	AAL templates	39
9.2	Displaying semi-transparent error regions (Linux)	39
9.3	Exporting semi-transparent error regions (Linux/Windows)	39
9.4	How to speed things up	40

1 Preliminaries

1.1 Abbreviations

PM parametric modulator

PSTH peri-stimulus time histogram (aka event-related BOLD time course)

HRF hemodynamic response function

main SPM window the upper left SPM window

SPM interactive window the lower left SPM window

SPM Graphics window the right SPM window in which results are displayed.

1.2 Acknowledgements

Several other available tools have inspired some of the concepts of `rfxplot` among which are the plotting functions in **SPM** (`spm_graph`), `get_psth` by Darren Gitelman and the MarsBaR toolbox by Matthew Brett.

2 Overview and Features

`rfxplot` is intended as a versatile data visualization tool for second-level **SPM** analyses. It offers a range of different plot type and configuration options (splitting the sample into groups, exclusion of subjects, error bars, colors, x-axis positions etc.). `rfxplot` accesses data from subjects' first-level analyses and averages them rather than relying on the (rescaled) data from the second-level analysis. The plot configurations can be saved and the previously configuration can be reused. The extracted data is returned in the MATLAB workspace and can be used for further analyses.

Here is a list of the features of `rfxplot`:

- Plot types: bar graphs of effect sizes, fitted responses, event-related BOLD responses (aka time courses, peri-stimulus time histograms)
- 2-stage voxel selection process
 - (1) Define a search volume based on the random effects analysis (second level analysis)
 - (2) Select either all voxels in that search volume or find individual peak voxels in the RFX search volume (and define single-subject spheres around them)

- Select data from different image types (e.g. beta, con-images)
- Select images based on the regressor names, not the filename. This way `rfxplot` can cope with different first-level design matrices
- Split regressors into different bin based on
 - Trial number: e.g. split the effect sizes into first and second half)
 - Value or percentile ranges of parametric modulators
 and re-estimate the design
- Split the sample into different groups or exclude some (outlying) subjects
- Plot effect as bars, dots, or connected dots with error bars
- Configure error bars as standard deviation (sd), standard error of the mean (sem), or 90% confidence interval (ci)
- Many configuration options for filtering and adjusting the extracted time series (for PSTHs)
- For fitted responses and event-related BOLD time courses, plot errors as lines or as semi-transparent areas (see section 9 on page 39 for some information on troubleshooting)
- Plot the data in the SPM window or in another MATLAB figure (useful for creating compound figures for publications)
- Fit a simple regression model to the effect sizes extracted from the subjects (e.g. for testing and visualizing a linear increase in activation across different task conditions); currently available regression models: linear, quadratic, exponential, logarithmic

3 Download and Installation

`rfxplot` is hosted as a SourceForge project (<http://rfxplot.sourceforge.net>) and it can be downloaded there or via the SPM extensions (<http://www.fil.ion.ucl.ac.uk/spm/ext>).

`rfxplot` has been developed and tested under SPM5 and will be adapted for future versions of SPM. Support for SPM2 may be added on a demand basis. `rfxplot` will not run under SPM99 and earlier version and support for these outdated SPM versions is not intended.

The installation of `rfxplot` is quite simple: unzip the downloaded archive (`rfxplot.zip`) in your local SPM toolbox directory (`$SPMroot/toolbox`). It will automatically create its own folder in the toolbox directory.

`rfxplot` can be started via the Toolboxes menu in the main SPM window or by typing `rfxplot` at the MATLAB command prompt. This will start the configuration dialogue (see section 5 on page 8).

3.1 Supporting Toolboxes and Functions

rfxplot requires one mandatory, but freely available, and a couple of optional supporting toolboxes. Upon startup rfxplot will check for these supporting toolboxes and use them if they are available. rfxplot will also work without the optional toolboxes, albeit with somewhat limited functionality.

3.1.1 Mandatory Toolboxes

NaN-Suite A small collection of functions that will compute the mean, variance, standard deviation etc. on N-dimensional arrays while ignoring NaNs. The NaN-Suite can be downloaded at the MATLAB File Exchange¹, or at the SourceForge site (<http://rfxplot.sourceforge.net>). The folder `nansuite` containing the `nan`-function must be included in your MATLAB path.

3.1.2 Optional Toolboxes

Signal Processing Toolbox When creating plots of the time course of the BOLD response (peri-stimulus time histograms, PSTH) rfxplot offers the possibility to resample the time series to a fine resolution, which generally leads to a better estimation of the time course. This requires the function `resample` from the Signal Processing Toolbox which is commercially available from MathWorks.

Statistics Toolbox or stibox rfxplot offers the possibility to split PMs into different bins, thus providing a more informative presentation of the effect than simply the parameter estimate. One option of defining these bins is by a percentile range of the values of the PM which requires the function `prctile` from the Statistics Toolbox or `quantile` from the `stibox`. If you choose `stibox`, you have to make the toolbox available on your MATLAB path. Also, check the function names of the toolbox so that they do not interfere with some of your own functions. The Statistics Toolbox is commercially available from MathWorks, `stibox` is a collection of function of statistical distributions and basic descriptive statistics, freely available from <http://www.maths.lth.se/matstat/stibox/>.

¹<http://www.mathworks.com/matlabcentral/fileexchange/loadFile.do?objectId=6837&objectType=file>

3.2 Support

Support for `rfxplot` is provided via the mailing list `rfxplot-users@lists.sourceforge.net` and by directly email to the author. Announcements about updates and new releases are distributed via the list `rfxplot-announce@lists.sourceforge.net`. Link for subscribing to these mailing lists are provided at the project webpage <http://fxplot.sourceforge.net>.

When submitting a bug report, please include the following information in your email:

1. The full error message displayed at the MATLAB command prompt.
2. Type `dbstop if error` at the MATLAB command prompt and rerun `rfxplot` until you get the same error message. Then (still in DEBUG mode), type `whos` at the MATLAB command prompt and copy the result (the function workspace) into your email.

3.3 How to cite rfxplot in your publication

There is an accompanying paper under revision in Neuroinformatics:

Gläscher, J. (in press). Visualization of Group Inference Data in Functional Neuroimaging, Neuroinformatics.

The final manuscript can be downloaded at <http://rfxplot.sourceforge.net/credits>.

4 General Concepts of rfxplot

`rfxplot` is a data visualization toolbox for second level analyses in SPM, which are widely called random effects (RFX) analyses. It thus requires an RFX analysis as a start-off point. However, in extracting the data and creating plots it traverses through the underlying first level analyses, thus avoiding some scaling issues when plotting data based on the RFX parameter estimates. This, for instance, allows for a calculation of “percent signal change” as a measure of the observed effect. While this approach (getting the data from the single subject analyses) provides for more accurate plots, it comes with one major drawback — processing speed. Extracting image handles data from each individual `SPM.mat` requires considerable time of

loading this file into the workspace, especially since these files tend to be very large. In order to reduce this aspect of processing time, `rfxplot` will — upon first usage in a new RFX analysis — create various support files in the directory of the RFX analysis that contain image handles, design matrices, and others. These will be used upon subsequent usages and therefore reduce processing time. These files are

- `rfx_design.mat` holding a struct array `des` with the information from the first level analyses,
- `rfx_images.mat` holding a struct array `imgs` with the volume handles from all native SPM images (i.e., beta, con, ess, spmT, spmF, ResMS),
- `rfx_timeseries.mat` holding a struct array `ts` with the volume handles to all first level data (i.e., the BOLD time series), and
- `rfx_opts.mat` holding a struct array `opts` with saved plot configurations.

The variables in these support files are loaded into the global workspace and therefore speed up the data extraction upon subsequent invocations of `rfxplot`. In addition, if you have different RFX analyses that incorporate the *same* first level analyses you can simply copy these support files (`rfx_*.mat`) to the directory of the other RFX analysis and thus avoid creating them again.

Another aspect that requires considerable processing time is splitting up regressors into different bins and plotting event-related BOLD response (PSTHs). This entails that the original time series at the selected voxels has to be extracted which takes a considerable amount of time (depending on the number EPI images per subject and the number of subjects in the study).

Nevertheless, splitting a PM into different bins is one of the features of `rfxplot` that allows for creating more informative plots of the effects. The interpretation of the parameter estimate of a PM is straightforward if it simple encodes a linear increase across time or a variable such as reaction time that scales linearly. In these cases, the parameter estimate can be interpreted as the slope of the linear increase across time in a certain brain area. However, for more complex PMs, such as predictive values or prediction errors (concepts in learning theory) derived from computational models the parameter estimate only indicates the effect size, i.e., how good the model-derived variables fit to the BOLD response in a certain area. Thus, this cannot be interpreted as how large the BOLD response to these

model-derived variables actually is. `rfxplot` offers the possibility to split these PMs into several different bins (e.g., low, medium, and high values) and configure the events in these bins with simple canonical HRFs. The toolbox will then re-estimate the design matrix and plot the new parameter estimates, fitted responses, or PSTHs according to these bins.

The configuration of a plot requires user input on quite a number of configuration options. This information is acquired via graphical user interface (GUI) in an interactive dialogue (see section 5 for a detailed description of configuration options). Because re-entering all these configuration options over and over again can be tedious and rather annoying, `rfxplot` provides for two ways to circumvent this. First, the current plot configuration is saved in the workspace (in a struct named `opts`) and subsequent invocations of `rfxplot` will ask if the current configuration should be used again. This is particularly useful when using the same plot at different voxels. Second, each plot configuration can be saved to a `.mat` file (`rfx_opts.mat`) in the directory of the current RFX analysis for later usage. If `rfxplot` detect this `.mat` file it will ask if a previously saved plot configuration should be used. The names of the different plot configurations are then presented as a drop-down menu.

5 The Plot Configuration Dialogue

This section describes the plot configuration dialogue in detail. All configuration options are presented in the SPM interactive window. In order to start `rfxplot`, you have to open a second level analysis. `rfxplot` will check for this and terminate if it cannot find an RFX analysis. Otherwise it will start the following configuration dialogue. In this manual, the user input questions appear in a `box`.

Re-use previous plot configuration?

This question is only presented if `rfxplot` finds the struct variable `opts` in the MATLAB workspace. Choosing “yes” will circumvent all the following configuration dialogue and will use the configuration specified in `opts` struct. This is most likely the one that was used immediately before, although you can also load this variable from a `.mat` file. However, the current cross-hair position in the SPM Graphics window will be renewed upon every invocation of `rfxplot`. This allow

for an easy and comfortable re-usage of the same plot configuration at different voxels.

Use saved plot configuration?

Upon completing the configuration dialogue `rfxplot` asked the user if he/she wants to save the current plot configuration and if so, asks to specify a name. The plot configuration is then saved to the file `rfx_opts.mat` in the directory on the current RFX analysis. Using a previous saved plot configuration will cause `rfxplot` to load `rfx_opts.mat` and present the names of the different plot configuration in a drop-down menu for selection. Because one may not know if one of saved plot configurations is suitable for the current plot, you can always choose “None of the below” to quit the selection of previously saved configurations. This will then start the normal configuration dialogue described next.

Use this RFX analysis?

`rfxplot` will display the path of the current `SPM.mat`, which is the current RFX analysis. Selecting “yes” will cause `rfxplot` to use the current `SPM` struct to extract all necessary information. However, you can also select “no” and select a different `SPM.mat` file (should be an RFX analysis as well). Choosing a different `SPM.mat` can be useful, if you want to select a certain location from the current RFX analysis, but plot the effect of regressors from a different model at this location. One scenario would be, to open an RFX analysis of a functional localizer, select the peak of some functionally relevant voxels and then plot the effects from the `SPM.mat` of the main experimental task.

What do you want to do?

With this question the general type of plot is selected.

Plot effect size (e.g. parameter estimates) You can plot the data of some selected images (e.g. beta-images, con-images) in different formats. This option is the most versatile plot and can display the most amount of infor-

mation from a lot of different images. It is both suitable for exploring the results and producing figures for publications. This plot type gives the most configuration options and can therefore produce the greatest variety of plots.

Plot fitted event-related responses This is a variant of the first option, in that it will plot the product of the effect size with the current basis function. It is most useful when using more complex basis sets than the simple canonical HRF (e.g. HRF + derivatives or Fourier basis set). This plot type will take the basis function into account and can therefore produce different shapes of the fitted BOLD response. It is most useful when different shapes of responses are expected and explicitly modeled (e.g. when administering different drugs that can change the hemodynamic response).

Plot peri-stimulus time histograms (aka time courses) This plot will extract the time course from every subject and will estimate the mean event-related response for the events of a selected regressor. It uses an improved FIR approach (Finite Impulse Response) that will always yield the mean event-related response, even if the time windows overlap (e.g., due to a jittered inter-trial interval) (see below for details). In general, plot types 2 and 3 cannot display as many effects as type 1 without cluttering the plot so that it becomes incomprehensible. Therefore, fewer effects (2-3 max) should be selected for these two plots types.

Just gimme the data This option will not plot anything, but it will just extract the data (time courses) from all subjects at selected RFX voxels and save them in a struct named `data` in the MATLAB workspace (see section 6 on page 26 for detail). It uses a similar approach than the “Eigenvariate”-Button in SPM interactive window (formerly VOI-Button, `spm_regions`). This option is useful for extracting time courses from all subjects in order to run PPI or DCM analyses.

Enter file prefix (e.g. beta, con)

In this option you can specify the type of images from which the data should be extracted. This can be one of the native images produced in an SPM analysis (beta, con, `spmT`, `spmF`, ResMS) or a user-specific image type produced e.g. with the Image Calculator (`spm_imcalc`). Of the native SPM image, beta-images will

provide the most configuration options. If the selected plot type is time courses or data extraction, then the file prefix is automatically set to “beta”.

RFX search volume

rfxplot determines the voxels from which it will plot the data in a two-stage process. First, a search volume can be defined based on the RFX analysis. Later, individual peaks within this search volume can be selected. The different options are:

Voxel This option only selects the voxel at the cross-hair position (in the current RFX analysis). Although it shows a certain effect at the second level, a plot of the data based on the individual data may not show a convincing pattern, since it may be very noisy, because the single subjects peaks may not fall in this particular voxel.

Sphere In this option you can define a sphere with a certain radius (in millimeter) to serve as the RFX search volume. The current cross-hair location serves as the center of the sphere. Later you can either select individual peaks within this search volume or plot the mean effect of all voxels in the RFX search volume.

Box This option is similar to the Sphere except that the search volume has the shape of a box and the x, y, z dimensions (in mm) have to be specified. The current cross-hair location serves at the center of the box.

Image In this option you can specify a single mask image that defines the RFX search volume or you can specify several mask images (one for each subject in the current RFX analysis that defines a particular region of interest). The latter is useful if a certain anatomical region shows considerable variability between subjects (e.g. ventral striatum). Defining this area based on the individual T1 images by creating a mask for it can insure that only the BOLD signal from this anatomically confirmed location is analyzed (e.g. in MarsBaR) or plotted with rfxplot. If you specify only a single mask image (defining an RFX search volume) you can still select in individual peaks within that mask, but if you specify a mask for each subject, then the data from all voxels in each mask are extracted and averaged.

Limit to supra-threshold voxels?

This option is only available if you are plotting from the currently opened RFX analysis (Use this RFX analysis? “yes”) and only if you have chosen Sphere, Box, Image (or RFX mask) as the RFX search volume. If this is the case, the subsequent voxel selection will be limited to those voxels in the RFX analysis that survived the currently selected threshold and are within the spatial extent of the search volume. Limiting the RFX voxels to supra-threshold voxels makes most sense, if you will then select “all voxels in RFX search volume” in the next voxel selection. Instead if you want to select individual peak within the RFX search volume, it is better not to limit it to supra-threshold voxels.

Voxel selection ...

This option specifies which voxels should be selected from the previously defined RFX search volume.

Select single subject peak within RFX search volume With this option the voxel within the RFX search volume that shows the maximum effect in the current (RFX) contrast is selected. This is done for each subject independently and only the data from this selected peak voxel are extracted. The selection of the voxel with the maximum effect is therefore based on the current RFX analysis and is therefore independent of the images that are being plotted. That is, you can select the maximum effect of each subject in the current RFX contrast, but plot the data from completely different images. The algorithm for finding the maximum effect works like this:

1. Obtain the contrasts weights of the current RFX contrast
2. Extract the data from the images that each subject submitted to the RFX analysis for all voxels in the RFX search volume and multiply these with the weights under 1 to obtain the single-subject effect in each voxel.
3. Find the maximum effect within the voxels of the RFX search volume and save this for later data extraction.

Define sphere around single subject peak This option finds the individual peak as described under 5 and then defines a sphere around it. The user is then

prompted for a specification of the radius of the sphere in mm. Subsequently, the user can choose to limit the individual spheres to the voxels in the RFX search volume. This is useful in case of an anatomical (e.g. the amygdala), when you want to select the individual peak in the amygdala and define a sphere around it, you also want to make sure that only voxels within the amygdala will be extracted and plotted. Without limiting the individual spheres to the RFX search volume, a subject's peak voxel can be located at the edge of the amygdala and due to the sphere around the peak, voxels from outside the amygdala could get selected.

Select all voxels in RFX search volume This option will select either all voxels in the entire RFX search volume or will limit them to the supra-threshold voxels as specified in the previous configuration option.

Index number for effect/PSTH plot ...

In this input field you list all the index numbers of all images that you want to plot. The table is displayed in the SPM Graphics Window. These numbers must be separated by a single space. Trailing spaces are truncated.

`rfxplot` will now load the image handles of the image type specified with the image prefix into the global workspace and present a table of all images found in the single-subject analyses. The table has 3 columns: (a) Index, (b) Count, and (c) Description .

Index is a running number by which the images will be selected. Count gives the number of images with this particular description string found in all single-subject analyses. Description gives the name of the image itself. The count will normally show the number of subjects in the analysis indicating that an image with this name has been found for every subject. However, when you have different design matrices at the first level, for instance a regressor encoding only stimuli that have been forgotten by a subject, then the count column might show a smaller number if there are subjects that have remembered all stimuli (and hence do not have a "forgotten" regressor). While in general the order of the regressors is preserved in this table, some of the regressors may appear at the end of the table if they are not found in the first subject's design matrix.

`rfxplot` can deal with these situations *provided that regressors that encode the same event type have identical names for all subject*. `rfxplot` will select the images

for plotting based on the index number and its associated description string. This way it will select the correct images from every subject even if one or several subjects have a different number of regressors in their first-level design matrices (and therefore the filenames differ between them).

When the current first level analyses utilize a higher-order basis set (e.g. HRF and its derivatives, Fourier set), then only the description for the first basis function is shown in black in the table, all other basis function will be shown in light gray. The first basis function has the ending **bf (1)*, whereas all other basis function have a number from **bf (2)* to **bf (n)*. For instance, the canonical HRF only has one basis function, hence all the onset regressors have the suffix **bf (1)* and are printed in black. The HRF+derivatives have 3 basis functions and hence the HRF has the suffix **bf (1)* (printed in black), the temporal derivative has the suffix **bf (2)*, and the dispersion derivative has the suffix **bf (3)*.

It is still possible to choose any image from the table, but most commonly the regressors with the ending **bf (1)* will be most interesting, because this option automatically selects all other associated basis function if necessary. This occurs, for instance, when effects are plotted as percent signal change (see below) and the other basis function might affect the maximum response of a given modeled BOLD response. Alternatively, when plotting fitted responses, the user only needs to choose the regressors with the ending **bf (1)* in order to take advantage of the full basis set and its possibility to visualize different response *shapes*.

Although SPM treats additional basis function as parametric modulation, *rfxplot* does not follow this scheme. Rather, *rfxplot* only considers parametric modulator that have been specified by the user as such (see “Regressor Types” below). Additional basis functions are selected automatically if necessary; the user does not have to burden himself with the selection of all relevant basis function unless there is a special interest in them (e.g. in the temporal derivative as a way to distinguish early from late peaks in the BOLD response (see Henson, NIMG, 2002)).

Regressor type for index X

If “beta” has been selected as the filename prefix, then for every selected image you have to specify if the image of index X is an onset regressor (“reg”) or a parametric modulation (“pmod”).

Label for effect X (reg/pmod)

In this option you are prompted to give a short, but descriptive name for each of the effects (=images) that you have selected. This is the label that will be displayed in the data plot.

Are selected regressors replications?

If “yes” is selected a new question “Indicate replications” will appear. With this option, you can specify if some of the images are in fact encoding the same event type and therefore should be averaged for the plot. This can occur, if the experiment is scanned in several sessions, with the same event types presented in different sessions. Supposedly, these should elicit the same BOLD response and can therefore be averaged when plotting the data. (If you want to look at changes across session, you can simply response with “no” and every selected event type (=effect) will be presented on its own. When choosing “yes”, `rfxplot` prompts for a vector indicating the replications. The vector should have as many elements as the number of images that are selected. Images that are replications should receive the same number in the indicator vector. For instance, if you select 9 images with the indices 2 5 8 11 14 17 20 23 26 and they have been scanned in 3 different session, then you setup the indicator vector like this: 1 2 3 1 2 3 1 2 3. This will have the effect that the data from images with the indices 2 11 20 will be averaged, as well as those with the indices 5 14 23 and the indices 8 17 26.

Split regX/pmodX into n bins

This option allows you to split the onset regressor or parametric modulation into several n different bins. New onset regressors will then be recreated that will contain all events in a particular bin. The beta weights for these new regressors will then be re-estimated and averaged across all subjects (see section 7.2 on page 31 for more details). If you do *NOT* want to split the effect into different bins, you enter “1”.

Specify bin as ...

If you have selected more than 1 bin above, then you can define here how the bins for the effect should be defined.

Percentile range (lower/upper thresholds) This option is only available for PMs and only if you have installed either the Statistics Toolbox or the stibox (see section X on Installation). The bins are defined by different percentiles of the range of the parametric value in the PM. This is particularly useful if your PM does not have a prespecified value range (e.g., reaction times) or if you want to insure that each bin will have the same number of events. Subsequently, you are prompted for $2 \times \text{nbins}$ thresholds defining the lower and upper percentile thresholds. The percentile threshold must be specified between 0 and 1. For instance, if you want to split your PM containing reaction times into 2 even bins at the median, then you have to specify the thresholds 0 0.5 0.5 1. The first two numbers 0 0.5 define the lower and upper percentile threshold for bin 1, the latter two 0.5 1 for bin 2. Alternatively, if you want to split your PM values into 2 bins of the bottom and top quartile, then you would specify the thresholds 0 0.25 0.75 1. Finally, if you want to split your PM values into 3 evenly distributed bins, you would specify this as 0 0.33 0.33 0.66 0.66 1. In general, two numbers always form a lower/upper threshold pair and should be entered from low to high percentiles) (see also section 7.2 on page 31). The parametric values that are then compared against the percentile thresholds are read from the 1st level SPM structs (SPM.Sess(s).U(u).P(p).P), which are the original values that have been entered upon design specification.

Value range (lower/upper thresholds) This option is similar to the percentile ranges, except that you are providing absolute thresholds for the different bins. It is useful, when your PM values have a prespecified range. For instance, in a very simply Rescorla-Wagner learning model the value of the CSs are scaled between 0 and 1. If you are interested in splitting your PM containing the model-derived predictive values of the CSs into a “low” (≤ 0.25) and a “high” (< 0.75) bin, then you would specify the value thresholds as 0 0.25 0.75 1. However, instead of interpreting these numbers as percentiles, these numbers represent absolute thresholds for the values encoded in the PM. Thus, specifying absolute thresholds does NOT insure that the number of events in each bin is evenly split (for further

details, see section 7.2 on page 31). The parametric values that are then compared against the absolute thresholds are read from the 1st level SPM structs (SPM.Sess(s).U(u).P(p).P), which are the original values that have been entered upon design specification.

Trial range (vector of trial numbers) Although this option is available for onset regressors and parametric modulations alike, it is most useful for onset regressors. It provides the possibility to split a regressor according to some trial range. For instance, if you want to split an onset regressor with 80 events into 2 bins encoding the first half of the event and the second half of the event, then you would specify the trial ranges as `1 : 40 41 : 80`. The specification of trial ranges must be separated by a space or the character “|”. The latter is useful if you need to specify specific trial number (e.g. `1 5 10 15 20 25`) for one bin, which are themselves separated by spaces. Then the “|” can be used a delimiter to separate the different trial range specifications for each bin. Also the special word “end” can be used and will expand to the last event number. Hence, in the first example, you could also specify the trial ranges as `1 : 40 41 : end`, if you have forgotten how many event there actually are in a regressor or if there is a different number of events per subject (see also section 7.2 on page 31).

n bin names for regX/pmodX

With this option you can specify different names for each bin in a selected regressor or parametric modulation. The number of names must correspond to the number of bins. They can be separated by a single space or — in case the space is needed for the name — by the character “|”. These bin names are later displayed in the data plot.

Split sample into group?

Clicking the “yes” button will allow you to split the sample in different groups and the data will plotted for each group separately. This can be useful, for instance, if you want to plot the data for males and females separately or if you want to split the sample into younger and older subjects. If you answer “yes”, you will

be prompted for the number of groups and a list of subjects will be presented in the SPM Graphics window. You can then specify the subject numbers for each group. If you answer “no” you are prompted for the subjects to include in the plot of the data. This gives you the opportunity to explicitly exclude an outlier subject (which you might have identified earlier with the overlay individual data option, see below). The default is to include all subjects, but you can use the same list of subjects to find the subject you wish to exclude. This question is not presented during the group specification dialogue because by specifying which subjects belong to a group you can also exclude an outlier subject.

Apply SPM design filters?

This option is only shown if one or more of the regressors have been configured to be split into different bins, because in order to do this, the individual BOLD time courses have to be extracted and re-analyzed. This option allows for the application of different filters to these extracted time series. All time series are extracted using the image handles found in the individual `SPM.mat` files. Therefore, the scaling of the images is automatically applied (e.g. global scaling, global normalization etc.). In a normal fMRI study all EPI images within a session are scaled in a way that the global mean across the entire session is 100, but the differences within each session are preserved. With this option you can configure, whether each of the 2 SPM design filters (high pass filter (“hpf”), modeling of serial autocorrelation via “AR(1)”) should be applied to the time series (or “none” or “both”). When the new regressors that are created for each bin are re-estimated, the same filter options will be applied to the new design matrix. SPM normally applies both filters (high-pass filter and serial autocorrelation). Therefore it is recommended to also use both filters when creating a data plot with `rfxplot` because these plots will be then similar to the original SPM analysis. The option of applying none of only one of the filters is intended for expert users who want to work on the raw time series and create their own filters.

Adjust data for ...

This configuration option offers a range of different possibilities to remove the variance of unwanted effects from the extracted time series.

No adjustments This leaves the time series as it is extracted with possible filtering options applied. Note also that the global session mean is not removed from the time series either. A percent signal change cannot be accurately calculated in this case and therefore this option will not be offered later. The no adjustment option is intended for expert users who want to work on the raw time series. For the general user it is recommended to adjust at least for block (and possibly nuisance effects) (see the next two options).

Block effects (aka session constant) This removes the effect of the session constants.

Block and nuisance effects This removes the block effects as well as the effects of some nuisance variables, such as movement parameters. Internally, SPM divides the design matrix into 4 partitions: (a) effects of interest, (b) covariates, (c) nuisance variables, and (d) block effects. These are configured by specifying the column numbers for these partitions in `SPM.XX.iH` (effects of interest), `SPM.XX.iC` (covariates), `SPM.XX.iG` (nuisance), and `SPM.XX.iB` (block effects, session constants). If you include the 6 affine movement parameters in your first-level design then these are nuisance regressors. However, SPM has no way of knowing which regressors you specify are nuisance or not. Thus, if you want to make use of this partition scheme you then have to edit the `SPM.XX` struct manually. This *can* be important for the estimation of the initial (and internal) effects of interest F-contrast which determines which voxels are selected for later analyses.

If you have manually configured `SPM.XX.iG`, then `rfxplot` will — under this adjustment option — remove the effects of these nuisance regressors together with the effects of the session constants. However, the most likely inclusion of nuisance regressor are by far the 6 affine movement parameters. If you have included them via the “multiple regressors” feature when configuring a first-level design, then these regressors will have the names R1 through R6. Even if you have not configured `SPM.XX.iG` manually `rfxplot` will look for these 6 names, and if it finds them, it will remove their effect from the extracted time series under this configuration option.

All other regressors (except those selected) This option will remove the effects of all other regressors except those that were selected earlier.

Specify manually (only if all designs are exactly identical) This option will re-display the table of the names and indices of beta images and the user is

prompted to specify all the indices of the effects that should be removed from the extracted time series. This option will only yield valid results if all design matrices at the first level are exactly identical.

F-contrast (only if present in all 1st level analyses) With this option the variance due to a 1st level F-contrast is removed from the time series. This F-contrast has to be present in the first level analysis of every single subject. The F-contrast has to be configured via the "columns for reduced design" feature (in batch mode called *iX0*) and *not* by explicitly specifying the contrast matrix. The F-contrast must be present in all individual first level analyses under the identical name. When this adjustment option is selected a list of the available 1st level F-contrasts will appear in the SPM Graphics window.

PSTH window bounds (wrt ons=0)

This option is only available when PSTH is the plot type and it configures the time window of the average event-related BOLD response that should be displayed. The time window is specified in seconds and the onset of the event-related BOLD response is at 0 seconds. You can specify negative number in order show the BOLD signal a few seconds before the response occurs. For instance, if your trial lasts 12 sec, you can specify the PSTH window as $-1 \ 16$. This will average the event-related BOLD response from 1 second before stimulus onset until 16 seconds post-stimulus onset (in order to display the response at stimulus offset 12s the PSTH window should be longer to account for the delay in the BOLD response). Of course, there have to be some scans before the first onset in order to plot data from a pre-stimulus period.

If you want to keep the resolution of the event-related time course in the units of TR (no resampling, see below), then I would recommend that you choose the window bounds as multiple of a TR. For instance, if you have a TR of 2.5 sec, you could choose a windows of $-2.5 \ 15$. This way, the alignment of the time course with the time axis will fit without interpolation.

PSTH bin width (s)

This is the size of the PSTH bins. The default is the TR of the first-level design. You can specify a smaller bin width, but that will either entail a resampling of the time series or a scattered FIR basis set (see the next option and the section 7.3 on page 33 for details) if the onsets do not fall exactly onto the beginning of a TR (which they should not).

Resample data to X sec

This option is only available if you have the MATLAB Signal Processing Toolbox installed. Otherwise `rxfplot` will simply proceed without resampling the extracted time series. If installed, `rxfplot` will resample the time series to the specified time resolution. The default value is the bin width specified earlier. `rxfplot` makes use of the function `resample` which resamples the time series to the specified resolution while preserving the shape of the BOLD response very accurately. This resampling operation has the advantage of yielding more accurate event-related BOLD responses because the resolution of the PSTH window (the bin width) corresponds to the resolution of the time series. Therefore, each bin in the FIR set will correspond to only a single data point and every bin in the event-related BOLD response is displayed more evenly.

Rescale PSTH to 0 at onset?

This option applies a “mean-correction” to the average event-related BOLD response so that each response will be zero at the time of stimulus onset (in the example above this corresponds to bin 2). Hence, the original average data point in bin 2 will be subtracted from the entire PSTH.

Plot data as ...

This option is only available when plotting effects of onset regressors or PMs that have been split into different bins.

Percent Signal Change (PSC) appears to have become a de facto standard when reporting functional activations. `rxfplot` implements the MarsBaR approach

to computing PSC (see http://marsbar.sourceforge.net/faq.html#percent_signal for a description). Briefly, it computed as:

$$PSC = \frac{\text{beta}(\text{task}) \times \max(\text{HRF}) \times 100}{\text{beta}(\text{constant})} \quad (1)$$

where $\text{beta}(\text{task})$ refers to the parameter estimate of the effect of interest, $\max(\text{HRF})$ is the maximum of the a single event of the current duration convolved with the current basis function, and $\text{beta}(\text{constant})$ the parameter estimate of the current session constant.

This approach is superior to simply reporting the contrast estimate as percent signal change, because this actually is a measure of percent global signal change, i.e. how much does the BOLD response deviate from the global signal. `rfxplot` (and `MarsBaR`) on the other hand will correctly compute a percent local signal change, i.e. how much does the evoked BOLD response deviate from its voxel-wise baseline, which is what most people actually have in mind when they talk about percent signal change. This option is only available for plotting the effects of onset regressors or PMs that have been split into different bins.

Effect Size (arbitrary units) will simple plot the parameter estimate or contrast estimate as it is, without any rescaling.

Plot data where?

Selecting “spm” will plot the data in the bottom half of the SPM Graphics window. Selecting “other” will query for the axes command. The second option is useful when creating figures for publications with different panels (subplots) which show a plot of the data at a certain voxel. The axes command can be something like `figure(10); subplot(2,3,4)` which will plot the data in the MATLAB Figure 10 as a panel in a 6-part (2x3) multi-panel figure. It is recommended that you always include a `figure(X)` in your axes command, otherwise the data will plotted somewhere in the current figure and the current axes which is most likely the SPM Graphics windows (`figure(1)`).

Plot data as ...

This option is only available when plotting effects. The mean effect will be plotted as bar graphs (“bar”), as dots (“dot”), or as dots with a connecting line (“dotline”). Error bars are added to all data plots.

Error bars are ...

This configures the error bars for all plots (effects, PSTH, fitted responses). For PSTH and fitted responses the errors are displayed as a semi-transparent area around the event-related BOLD response or the fitted response. However, when plotting a lot of images as PSTH or fitted responses, the display can get very cluttered with error “areas”. Thus, it might be preferable to omit the errors by choosing “none”. When plotting effects as a “bar”, “dot”, or “dotline” the errors will be displayed as an error bar. For all plots the size of the error bar can be ± 1 standard deviation (“sd”), ± 1 standard error of the mean (“sem”), or a 90% confidence interval (“ci”) ($\text{sem} \times \text{spm_invNcdf}(0.95) = \text{sem} \times 1.6449$). These errors are all computed across subjects.

Plot errors as ...

For fitted responses and PSTHs the errors can be plotted as lines around the mean response or as a semi-transparent area. The latter option looks nice in figures for publications, but it has some display issues (see section 9 on page 39).

Error bar color

This is only available when plotting images are effects (not PSTH or fitted responses). The color of the error bar can be either the same as the bar/dot/dotline (“same”) or “black”. For PSTH and fitted responses the color of the transparent error bar is always the same as the main color.

Specify colors as ...

Here you can configure the colors in which the data should be plotted.

Effect will prompt you for as many color specifications as the number of images that have been selected (or alternatively, if there have been replications, the number of unique effects).

Bin will prompt for as many color specifications as the number of bins that have been specified when splitting a regressor into different bins. The different colors for each bin will be the same for all selected images (=effects).

Manual will prompt for as many color specifications as needed to plot all effects with all bins.

A color specification can be one of the 8 standard colors in MATLAB indicated by a single letter: (r)ed, (g)reen, (b)lue, (c)yan, (m)agenta, (y)ellow, blac(k), (w)hite. Alternatively, you can also provide your own color definitions as RGB triplets, e.g., 1 .5 0 as orange. Color specifications are separated by a single space, or — if spaces are needed for the color definition — by the character “|”. In the latter case you have to use “|” consistently as a separator for all color specs.

Color gradient for bins

This option is only available when you are specifying colors for effects. If you have split a regressor into different bins, these can be plotted in different colors according to a color gradient of the main color of that effect, or the different bins can be plotted in the same color as the main color for that effect. For instance, if you have split a PM into 3 different bins and this regressor should be plotted in red, the different bins can be colored from brown to dark red to red, or they can be uniformly plotted in red.

The final configuration options are only available when plotting effects (not PSTH or fitted responses).

Specify XTick positions

Here you can specify the positions on the x-axis where the different bars/dots should be plotted. The numbers do not have to be evenly spaced and therefore this

is an excellent way to achieve some visual grouping within the plot. For instance, if you are plotting the effects of 2 PMs, both split into 2 different bins, then you can specify the positions on the x-axis as 1 2 3 5 6 7. This will leave a small gap between the two different PMs and will thus make it easier to see which bar belongs to which condition.

Overlay individual data

This option will plot the individual data point of each subject as a small black dot. If the subject numbers are included (follow-up question), the dots and the corresponding subject numbers are plotted to the left and right of the average effect. This option is useful when exploring the results of a contrast for potentially outlying subjects (which can then be explicitly excluded in a subsequent plot, see group definitions above). If the subject numbers are not included, only the individual dots are plotted to the right of the the average effect.

Fit regression line to data

Finally, you have the option of fitting an additional regression to the data. This is useful when your experimental design incorporates some kind of parametric ordering of stimulus conditions (e.g. different degrees of motion coherence). When the data across different effects show a particular trend (e.g., linear) then this can be fitted with this option.

If “yes” is selected, a summary of all effects with potential bins along with the column numbers will appear in the SPM Interactive window. Then the user is prompted with the following configuration options:

How many regressions?

Specify the number of regressions you wish to fit.

Which columns?

Please refer to the summary to specify the column numbers

Type of regression

Here you can select the type of regression line that you want to fit. Linear and Quadratic are fitted using MATLAB's `polyfit` function. Exponential and Logarithmic are fitted using `fminsearch`. All are fitted using a least square criterion. This fitting procedure will be done for each subject individually to obtain individual parameters that can be then tested using other statistical tests. These individual parameters are collected in the struct array called `reg` and is exported into the MATLAB workspace (see section 6). However, the regression line that is displayed in the plot is derived from fitting the average data across all subjects (because the data are also plotted as the average across subjects). This may have unexpected consequences in the sense that parameters (a,b,c) based on individual fits are usually noisier than the parameters for the all subjects. Thus, while the plot of the data may show an obvious linear trend, a post-hoc one-sample t-test may not confirm that the slope parameter is significantly different from zero.

The different regression models are:

- Linear: $y = a + bx$
- Quadratic: $y = a + bx^2 + cx$
- Exponential: $y = a + b \times e^{cx}$
- Logarithmic: $y = a + b \times \log(x^c)$

Color of regression line?

Here you can specify the color of the regression line using the same color specifications as above.

6 The Output of `rfxplot`

In addition to plotting the data either in the SPM Graphics windows or some other MATLAB figure, `rfxplot` will return the data in the Matlab workspace. These are the structs/cell arrays that are returned:

`data` struct array with the data for each subject with the following fields

.rfxyz RFX coordinates (the location of the cross-hair)
 .rfxspace type of RFX search volume
 .rfximg type of image entered in the RFX analysis (usually 'con' or 'beta')
 .rfxcon the title of the RFX contrast
 .rfxdim dimension of the RFX search volume
 .xyz location of the individual peak (within the RFX search volume)
 .limtype method of voxels selection ('peakffx','sphereffx','allrfx')
 .ffxsphere dimension of the sphere around the individual peak
 .regressor names of the selected regressors (from SPM.xX.name)
 .maxeffect the maximum effect of each subject in the RFX contrast (for
 data.rfximg='con' the max effect is the T-value, for data.rfximg='beta'
 the max effect is the parameter estimate)
 .filter filtering option for data extraction ('none','hpf','ar(1)','both')
 .adjust adjustment option for extracted time series
 .scale 'percent signal change', 'effect size'
 .raw the raw extracted data (nScan×nVox)
 .filt the filtered time series (nScan×nVox)
 .adjust the (filtered and) adjusted time series (nScan×nVox)
 .tc (t)ime (c)ourse; 1st eigenvariate of the adjusted time series (nScan×1)
 .Y 1×nSess cell array with the data in .tc split according to the sessions
 .prefix e.g., beta extracted data from beta images (no data extraction neces-
 sary)
 .effect the data that are plotted, e.g. rescale to pcnt sig change if necessary
 .psth struct array for the event-related time course plot (nSelectedEffects×1)
 .psth.bin struct array for each bin if the regressors is split up
 .psth.bin.psth average time course that is displayed
 .psth.bin.sem s.e.m. (across events)
 .psth.bin.ci 90% confidence interval (across events)
 .psth.bin.pst the peri-stimulus time (e.g. x-axis of the middle of the PST
 bin (unused))

The fields .raw, .filt, .adjust, and .Y are only assigned if the plot required the extraction of the original time series in order split a regressor of PM. The field .prefix (e.g.,beta) is only assigned if data are plotted from images that do not require the extraction of the original time series (e.g. simple effects of an onset regressors or a non-splitted PM). The field .psth and its subfields are only assigned if event-related BOLD response (PSTHs) are plotted.

When plotting effects, a cell array **gdata** (nGroup×1) is also returned in the MATLAB workspace containing the data matrix for all subjects in each group

(e.g., `data.effect`). `gdata` is not returned when plotting PSTHs or fitted responses.

If effects are plotted and the data are then fitted with one of the simple regressions (linear, quadratic, exponential, logarithmic), then a struct array `reg` (`nRegression x 1`) is returned with the following fields

`reg` results from the regression on the plotted data

`.group` results for group data, struct with fields

`.type` type of regression

`.y` average group data that are fitted

`.param` the fitted group parameters of the regression

`.ssq` the residual sum of square of the regression

`.yhat` the fitted data

`.sub` struct array (`nSubjects x 1`) with the following fields

`.type` type of regression

`.y` average group data that are fitted

`.param` the fitted group parameters of the regression

`.ssq` the residual sum of square of the regression

`.yhat` the fitted data

When fitting the data of the plot with another regression (e.g. in order to detect a linear increase across parameters that corresponds to stimulus intensity), the regression uses a standard OLS estimation and is carried out for both the average group data and for each individual. The regression line plotted in the graph is that for the group data because it corresponds to the plotted group data. However, if one wants to apply a statistical test to one of the parameters of these regressions the individual data are more useful because they also show the variability of these parameters across subjects (which can then be statistically assessed via a one-sample t-test).

For a very crude model comparison (e.g., which regression fits better: linear or logarithmic?) the residual sum of a squares is also returned.

7 Some Background Information

This section provides some information about how `rfxplot` selects voxels, how the splitting of PMs is implemented and how PSTHs (e.g. event-related BOLD responses) are computed.

7.1 Voxel Selection

It is important to reiterate that the `rxfplot` has a 2-stage procedure of voxel selection:

1. Define a search volume for the RFX analysis
2. Select voxel from within that search volume

For stage 1, `rxfplot` offers the choice between defining a sphere, a box, or a single voxel at the current crosshair position. Alternatively, an RFX mask can also be specified and this is not bound to the current crosshair position. This RFX mask can be an anatomical mask that specifies a certain anatomical structure, or a mask of a previous contrast (e.g., from a functional localizer). The mask has to be specified as an image file with 1s at the voxels that define the mask and 0s elsewhere.

If you want to create a mask from a functional localizer in SPM, you would open that analysis and show the results at some suitable threshold. Then you press “save” in the SPM interactive window and specify a filename. SPM will then save the results: all voxels that exceed the threshold will have the T/F-value as their voxel value, and all other voxels have the voxel value of 0 (zero). In the next step, this saved image has to be binarized to create a mask. For this step, you can use the image calculator: Press “ImCalc” in the main SPM window (the upper left window), select the saved image as input, select an output filename, and use “ $i1 > 0$ ” as the logical expression.

Once an RFX search volume has been specified, you can choose to limit it to supra-threshold voxels. This as the consequence that only these voxels are available for the voxel selection at stage 2. It is most useful if you want to select all voxels in the RFX search volume and not look for individual peaks within the RFX search volume.

For stage 2, `rxfplot` offers the selection between all RFX voxels, the individual peak voxel with the RFX search volume, or (another) spherical search volume around that individual peak. Often it is argued that a search volume around an individual peak will show a more representative picture of the data. However, this argument is somewhat questionable, since the functional imaging data are usually smooth at the end of image preprocessing, and as a result, voxel intensities and the resulting statistical images are spatially correlated. Nevertheless, `rxfplot` also offers this option for voxel selection.

The algorithm for finding individual peaks within the RFX search volume goes like this:

1. Create a vector weights for all input images to the RFX analysis by mul-

tipling the RFX design matrix with the current RFX contrast. For conjunctions as the RFX contrast, a weight matrix is created with each column corresponding to the weight vector for each contrast entered in the conjunction.

2. If the input images to the RFX analysis are con-images, then the data for the voxels from the RFX search volume from the corresponding T-image for each subject's first level analysis are extracted. If the input images are beta-images, then the data from these images are extracted.
3. The extracted data are multiplied with the weight vector reflecting the current (RFX) contrast, and the voxel with the maximum "effect size" is selected. For conjunction analyses, data are multiplied with each column of the weight matrix (reflecting the different weights of the input contrasts) and the results are summed. Then the voxel with the maximum effect is selected.

This algorithm has the advantage of

- being able to create a measure of "effect size" for the current RFX contrast, even if it does not exist in the individual first-level analyses (e.g., when beta-images are the input for the RFX analysis).
- being applicable to a wide range of RFX designs, e.g. also in repeated measures designs (paired t-test, flexible factorial with a repeated measures factor) that also have subject constants in the design matrix.
- selecting the voxel with the maximum t-value if con-images are used as input for the RFX analysis. T-values (=effect size / noise) have the advantage that they also take the variability of the effect size into account. Thus, the individual peak voxels will balance effect size and its variability.

However, a word of caution is applicable for plotting the effects of RFX conjunctions. When plotting data from a conjunction analysis, the peak voxels from the contributing contrast may not be at the same voxels than the maximum of the RFX conjunction. To understand the problem, consider an RFX conjunction of 2 contrasts (A and B) which has the following effect sizes at 3 different voxels in the RFX search volume:

Voxel 1: effect size(A) = 1, effect size(B) = -1, sum = 0

Voxel 2: effect size(A) = 0.5, effect size(B) = 0.5, sum = 1

Voxel 3: effect size(B) = -1, effect size(B) = 1, sum = 0

Clearly, contrast A shows the biggest effect in voxel 1, whereas contrast B show the biggest effect in voxel 3. However, the RFX conjunction may yield its maximum in voxel 2.

When trying to plot these data with `rfxplot` with intention to plot the effect sizes for contrast A and B and their maxima in voxels 1 and 3 respectively, the voxel selection might actually miss these contrast-specific peaks, because the algorithm computes the (weighted) sum of the individual effect size. The consequence is that the plot will not show the maximal effect sizes for contrasts A and B. In the example above, the sum of the effect sizes for voxels 1 and 3 are 0, whereas the sum for voxel 2 is actually 1. Hence, the algorithm would select voxel 2 as the peak voxel, thus missing the contrast-specific peaks in voxels 1 and 3.

One way to work around this potential problem, is to open the RFX conjunction (without data plotting) and simply write down the coordinates of the local maximum in an area of interest. Then, open the RFX contrast A (the first one in the conjunction) and place the cross hair on the recorded coordinates (of the RFX conjunction) and start plotting the effects for contrast A with `rfxplot`. This time, `rfxplot` will detect voxel 1 as the peak voxel (provided it is within the RFX search volume defined on the recorded coordinates) and select it for data plotting. Finally, the same applies for contrast B: `rfxplot` will then detect voxel 3 as the peak voxel and plot the data from this location. This (admittedly complicated) procedure will insure that you will plot the data from the contrast-specific peak voxels in each contrast that contributes to an RFX conjunction.

7.2 Splitting of Parametric Modulators

Parametric modulations provide a way to test for specific correlations of the BOLD response for each event with an external variable. These external variables can be as simple as a linear or quadratic increase across time (i.e., over a session) or more complex such as predictive values or prediction errors derived from a computational model. In any case, the parameter estimate (beta) for the parametric modulator cannot be directly interpreted as the magnitude of the response; rather it is a measure of how well the BOLD responses fit with the external variable. In the case of theoretical signal derived from a computational model the parameter estimate can be seen as an indicator of how well the model fits to the data.

In the case of a simple linear increase across time, the beta for the PM can be directly interpreted as the slope of the increase, i.e. how much the magnitude of the BOLD response for each event increases across time. However, for more complex PMs, this straightforward interpretation is no longer viable and looking solely at the beta one can only talk about “effect size”, i.e., how well the model fits to the BOLD data.

`rfxplot` offers a way to translate the effect of these more complex PMs into simpler statements about the magnitude of the BOLD response. For instance, when a researcher finds a positive effect for a PM encoding the predictive value of a computational learning model, then this implicitly states that the BOLD response to trials of low value is smaller than the response to those trials with a high predictive value. `rfxplot` provides a way to visualize the effects of a PM in this way by splitting the PM into different bins. Each event in each bin is modeled as a simple onset regressor without a parametric modulation. These newly created regressors are then estimated and the parameter estimates (betas) for these regressors are then a measure of the average magnitude of the BOLD response to the event in this bin (e.g., all those trials that have a low predictive value according to a computational learning model).

`rfxplot` offers two ways of splitting PMs into different bins based on the parametric values that are entered in the PM:

1. Splitting according to a range of percentiles
2. Splitting according to absolute parametric values

The first case is appropriate if one does not know the absolute range of parametric values. For instance, if you want to plot those trials in the bottom quartile of the range of parametric values and those in the top quartile, you would enter 0 0.25 0.75 1 as lower/upper thresholds during the configuration dialogue. The bottom quartile is defined by the 0th percentile as the lower boundary and 25th percentile as the upper boundary. Likewise the top quartile is defined by 75th percentile as the low boundary and the 100th percentile as the upper boundary. All thresholds have to be specified as fractions between 0 and 1. You always have to specify two thresholds for each bin and the bins should be ordered from lower to upper. The definition of bins in terms of percentile ranges insures that for each subject, the appropriate trials are selected for each bin. It also certifies that the number of events in each bin will be (approximately) the same if the same percentile range (upper–lower threshold) is applied for each bin. This has advantages for the precision of the parameter estimates for these newly created regressors.

The second case can be used for splitting PMs in which you already know the possible range of parametric values, e.g. in a simple Rescorla-Wagner learning model of Pavlovian conditioning the possible range of predictive values lies between 0 and 1 (for theoretical reasons). If you are interested in the BOLD responses to those events for which the model predicts a value between 0 and 0.2, you can enter these absolute thresholds in this second option. Note however, that this will not insure the same number of events in each bin and in some subjects there may not be any events in that particular bin, which would lead to an error

message in `rfxplot`.

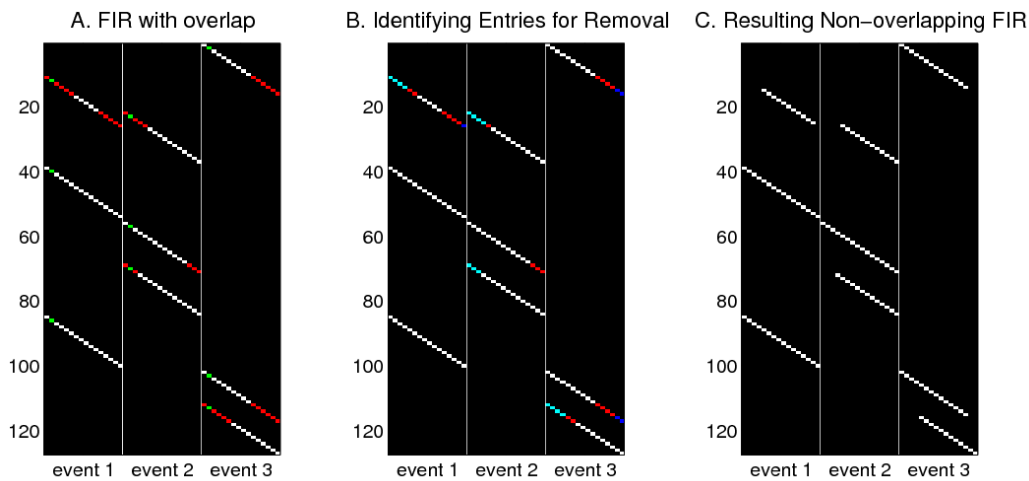
There is still a third option to split a PM, but it also applies to onset regressors without any parametric modulation attached to it. In this third option, you can also split any regressors according to the trial number. If you know the number of trials in each regressor, then this provides an easy way to plot the BOLD response to the first half and the second half of the trials (which - among other things - can be also used as a very crude test for learning related effect in the brain activation). The trial ranges for each bin are specified using the colon (`:`) operator in MATLAB or by supplying specific trial number (but then the trial ranges for each bin have to be separated by the `|` character. Also, the special word "end" is recognized as it evaluates as the last trial number (see section X for more details on how to specify trials ranges). For example, if you want to plot the average BOLD response to the first 40 trials and then to the remaining trials then you can specify the trial ranges as: `1:40 41:end`.

7.3 Computation of event-related BOLD responses (PSTHs)

`rfxplot` estimates the average event-related BOLD response via an FIR basis set (Finite Impulse Response). This is the same approach as taken in `spm_graph` (the "Plot"-Button in the SPM Interactive window). A series of new regressors is created (one for each PSTH bin) that will have a 1 at the time of the respective bin and 0 elsewhere. The entire FIR basis set of `nbins` regressors looks very much like a staircase for each event. A least square estimate of this basis set will yield `nbins` beta weights that are the average BOLD signal for that particular bin. These average BOLD signals will be plotted as the event-related BOLD response later on.

A particular problem can arise in the case of the specification of a rather long PSTH window (e.g. 16 seconds starting 1 sec before the stimulus onset until 14 sec post-stimulus onset). When these windows overlap between different events due to a jittered inter-trial interval, then the estimated beta weights are NOT the average BOLD signal for that bin. In order to avoid this problem, `rfxplot` will then successively remove some entries from the FIR set for an overlapping response starting from the beginning of the PSTH window until 2 seconds after stimulus onset.

The algorithm is demonstrated best in an example. Consider Figure 7.3 on the next page which shows the new regressors of the FIR basis set for 3 different event types with 3 trials for each. The stimulus onset (`=0`, the reference in the PSTH window specs, see Plot Configuration Dialogue) falls into the second bin



and is shown in green. The bins that overlap (i.e. are shared between different regressors) are shown in red. If this FIR design was estimated the beta weights for the columns with red entries would NOT be the average of the BOLD signal from all scans in that bin. In order to resolve this ambiguity, the overlap has to be resolved.

`rfxplot` will remove these overlapping entries starting from the beginning of an overlapping event until 2 sec after stimulus onset. The reason for this is that the BOLD response occurs with a delay of approximately 4-5 sec and thus the first few seconds of each event-related BOLD response are mostly not as important as the time when the response peaks. By removing these entries in the FIR design the rising edge of the BOLD response will show a greater variability, but the main time point of interest (when the BOLD response peaks) will be estimated with maximum precision. In the current example these entries corresponding to the first few seconds are shown in the second panel of Figure 7.3 in cyan. These are the candidates for being removed (= set to zero).

However, the example also shows that in some cases the overlap cannot be completely resolved by removing the entries marked in cyan. In that case, `rfxplot` will start removing entries from the tail of the PSTH window until the overlap is completely resolved. The reason for this being again, that the final tail of the BOLD response is most likely not as important as the time when the BOLD response peaks. In the example (Figure 1b) these entries at the tail of an FIR set are marked in blue. Thus, this two-stage removal of overlapping FIR entries in-

sure that the resulting beta weights will exactly reflect the average BOLD signal for that particular bin. For the fictitious example, the resulting FIR set with all overlapping entries removed is shown in Figure 7.3 on the previous page. This is the FIR design matrix that is estimated in order to obtain an average event-related BOLD response.

8 Non-interactive usage of rfxplot

rfxplot has the capability being completely scriptable thus circumventing the interactive GUI. In order to do so, one has to build the `opts` struct manually which holds all the configuration options. In addition, the 2nd level coordinates for the RFX search volume, the 2nd level SPM struct, and the xSPM struct which hold the information about the currently opened contrast have to be submitted as input arguments to `rfxplot`.

8.1 Description of the `opts` struct

In typographical conventions for the following description of the `opts` struct are:
`.fieldname` [data type] <possible field values> short description
of configuration option

The `opts` struct has the following subfields:

- `.rfxdir` [char] folder of the RFX analysis.
- `.xyz` [double] coordinates of the RFX voxel on which to build the RFX search volume.
- `.task` [char] <'effect', 'fitresp', 'psth', 'extract'> task type (i.e., plot effect, fitted response, PSTHs (aka time courses), extract the data.
- `.prefix` [char] filename prefix for 1st level analyses from which to extract and plot the data.
- `.space` [char] <'sphere', 'voxel', 'box', 'image'> type RFX search volume .
- `.dim` [double] dimension of the RFX search volume (in case of sphere, this denotes the radius).
- `.mask` [cell array] name(s) of the anatomical mask images defining the (group-wise or individual) search volumes.
- `.limit` [boolean] <0, 1> Limit voxel selection to supra-threshold voxels? (1 = yes, 0 = no).

.limtype [char] <'maskffx', 'peakffx', 'sphereffx', 'allrfx'>
how to select the individual voxels in the RFX search volume: 'maskffx' – anatomical masks (defined in opts.mask), 'peakffx' – individual peak voxel in RFX search volume, 'sphereffx' – individual sphere around peak voxel, 'allrfx' – all voxels in RFX search volume (confined to supra-threshold voxels if opts.limit = 1).

.ffxsphere [double] radius of the individual sphere around the peak voxel in RFX search volume.

.select [double] index numbers for the 1st level images from which to extract the data (see table in GUI version of rfxplot for a list of the indices).

.selname [cell array] image descriptions of the selected images (non-essential option).

.regtype [cell array] <'reg', 'pmod'> type of regressor for each select image (reg = onset regressor, pmod = parametric modulation).

.repl [double] vector indicating which images should be averaged (those with the same number in opts.repl).

.reglabel [cell array] short description of the image. This is used as a label in the plot. If some images are replications, then opts.reglabel should have max(opts.repl) elements.

.nbin [cell array] number of bins in which to split a regressor.

.binthresh [cell array] threshold for bins when splitting up regressors. The threshold of each image are themselves cell arrays with either a 2-element vector (lower/upper boundary) or a string specifying a trial range.

.binnames [cell array] names for the bins. For each regressors there is a cell array of nbin names. This is used as a label in the plot. If some images are replications (opts.repl=0), then opts.binnames should have max(opts.repl) elements.

.ngroup [double] number of group in which to split the sample

.group [cell array] subject numbers for each group. The numbers correspond to the order of the subject as they are entered in the RFX analysis (and found in SPM.xY.P)

.gname [cell array] names of the groups (empty for only a single group)

.filter [char] <'none', 'hpf', 'ar1', 'both'> filtering option for extracted time courses

.indiX0 [double] column numbers of regressors for which the time course should be adjusted (opts.adjust = 'user')

.adjust [char] <'none', 'block', 'nuis', 'all', 'user', 'con'> adjustment options for extracted time series

.winbound [double] boundaries of time window for plotting PSTHs
.winlength [double] length of PSTH window (computed automatically)
.binwidth [double] width of each PSTH bin (default 1 TR)
.resample [double] time resolution to which the extracted time course should be resampled
.rescale [boolean] <0,1> mean-correct PSTH at stimulus onset
.scale [char] <'pcnt', 'es'> scale of the plotted data ('pcnt' = percent signal change, 'es' = effect size)
.where [char] <'spm', axis command> where to plot the data
.plottype [char] <'bar', 'dot', 'dotline'> plot type for plotting effects
.overlay [boolean] <0,1> Overlay individual data?
.sublabel [boolean] <0,1> Include subject numbers?
.error [char] <'none', 'sd', 'sem', 'ci'> type of error bar
.errcol [char] <'s', 'k'> color of the error bar ('s' = same or
.errtype [char] <'bar', 'area'> type of error ('bar' = error bar, 'area' = error area (bounds around the PSTH or fitted response)) black'
.coltype [char] <'effect', 'bin', 'manual'> how to specify the colors for the plot
.colgrad [boolean] <0,1> Make a color gradient for different bins?
.xtick [double] XTick positions for the different effects
.fit cell array[] column number for which to fit an additional regression
.fitfun [cell array] <'lin', 'quad', 'exp', 'log'> type of regression to fit to the data
.fitcol [cell array] colors of the regression line
.name [char] name for the plot configuration (for saving)

8.2 Howto setup rfxplot for non-interactive usage

As mentioned above, in order to run rfxplot non-interactively, one has to submit 4 input arguments to rfxplot:

1. a manually created (or saved) `opts` struct (see above)
2. the RFX SPM struct
3. the xSPM struct holding the information about the current contrast
4. the RFX coordinates (in world space (mm)) on which to build the RFX search volume

The only tricky part is the xSPM struct which is usually create by (interactively) pressing the "Results" button. However, it can be also created non-interactively using various version of `spm_getSPM` that are set up for batch mode and can be found in postings to the SPM mailing list. One version is `results_nogui` which is a wrapper script that builds a `job` struct and then submits is to `spm_results_ui`, which in turn creates the xSPM struct.²

Thus, a non-interactive session of `rfxplot` could be scripted like this:

```
% 1. load the RFX SPM.mat
rfxdir = '/path/to/your/rfx/analysis/';
load(fullfile(rfxdir, 'SPM.mat'))

% 2. create the xSPM struct (e.g. for contrast 3
% at 0.001 uncorrected) [see help in results_nogui]
xSPM = results_nogui('val', rfxdir, 3);

% 3. define RFX coordinates
rfxxyz = [10 10 10];

% 4a. create a defaults opts struct
% and replace necessary fields
opts = rfx_setup_opts;
opts.rfxdir = rfxdir;
opts.xyz = rfxxyz;
...

% 4b. alternatively, load a previous saved
% opts struct (e.g. No. 3)
load(fullfile(rfxdir), 'rfx_opts.mat')
opts = opts(3);

% 5. call rfxplot non-interactively
rfxplot(rfxxyz, opts, SPM, xSPM);
```

²`results_nogui` has 2 different input formats for specifying the details of the RFX contrast: 1. parameter-value pairs or 2. contrast configuration values in a pre-specified order. See the help in the script for details. The script can be downloaded at [insertURLhere](#)

9 Goddies and Known Issues (and their solutions)

9.1 AAL templates

An automatic reference to the Automated Anatomical Labeling toolbox (Tzourio-Mazoyer, NIMG, 2002)) can be configured in `rfx_defaults.m`. In order to do so, you have to set `rfx.aal.display` to 1 and specify the full path to the AAL image and the text file with the anatomical labels. If you have MRICron (<http://www.sph.sc.edu/comd/rorden/mricron/>) installed, then you will find an image `aal.img` and the text file `aal.txt` in the templates folder of the distribution. Alternatively, if you download the original AAL toolbox from <http://www.cyceron.fr/freeware/> the files are called `ROI_MNI_V4.img` and `ROI_MNI_V4.txt`. You will have to specify the full path to either set of files in `rfx_defaults`. This will display the anatomical label of the current RFX coordinate (the one that you started `rfxplot` with) in the title of the plot.

9.2 Displaying semi-transparent error regions (Linux)

Semi-transparent error regions can only be displayed using the OpenGL rendering engine in MATLAB. Normally, MATLAB will switch to the appropriate rendering engine, but if you need to reset the renderer manually, you can do that with `set(figure_handle, 'renderer', 'opengl')`. `figure_handle` (figure handle) is usually 1 for the SPM Graphics window. `rfxplot` will set the rendering engine to OpenGL for PSTH and fitted response plots if the error regions are configured to be an `'area'`. The rendering engine can also be explicitly defined in `rfx_defaults`.

These links provide some information about the different rendering engines and when they are used:

<http://www.mathworks.com/support/tech-notes/1200/1206.html>

<http://www.mathworks.com/support/tech-notes/1200/1201.html>

9.3 Exporting semi-transparent error regions (Linux/Windows)

Printing/Exporting these semi-transparent error regions is another issue. MATLAB can export transparencies only into a bitmap graphics format, and since the SPM-Print button will only print into a Postscript file, the transparencies will appear opaque in this format. PNG (portable network graphics) is a format that is

capable of handling transparencies, but JPG and TIFF will also show the transparencies correctly, but naturally the fonts and lines are also converted to bitmaps and might appear pixelated. EPS also shows the transparencies, but rather than creating scalable Postscript, a bitmap is embedded in the EPS file and hence the lines and fonts look rather ugly.

However I have experienced that exporting graphs with transparencies sometimes results in a completely garbled image (regardless of the format). This is most likely an issue of the OpenGL engine that is used. Often the graphics card will provide OpenGL support, but on some laptops with an onboard graphics chip (like mine), this does not work.

One way to check which OpenGL engine is used is by using the MATLAB command `opengl info`. On my laptop this results in:

```
Version          = 1.4 Mesa 7.0.1
Vendor           = Tungsten Graphics, Inc
Renderer        = Mesa DRI Intel(R) 965GM 4.1.3002
MaxTextureSize  = 2048
Visual          = 0x25 (TrueColor, depth 24, RGB mask)
Software        = false
# of Extensions = 81
...
```

Using the Mesa DRI engine results in these garbled graphics files described earlier.

A workaround is to activate MATLAB's software OpenGL engine. This has to be done before the engine is initialized, i.e., before the first transparency plot and even before `opengl info`. The switch to the software implementation of OpenGL is done with the MATLAB command `opengl software`. You can even include this command in your startup scripts so that it will be always activated whenever you use MATLAB. On my laptop this solved the problem and I was able to export the plots with semi-transparent error regions into PNG, JPG, and TIFF files with reasonable quality of the fonts. However, when saving as Postscript or EPS, some font issues remain (pixelated fonts).

9.4 How to speed things up ...

By default `rfxplot` creates several `.mat` files in the RFX directory. These contain the single subject design matrices (`rfx_design.mat`), the SPM image handles (`rfx_images.mat`), and the image handles for the original EPI time series

(`rfx_timeseries.mat`). Especially the last `.mat` can get quite large for studies with lots of EPI images and subjects. Consequently, saving the time series to a `.mat` and loading it upon a future call to `rfxplot` might actually take longer than recompiling the struct with image handles. For this reason, the saving of these `.mat` files can be disabled in `rfx_defaults` (`defs.savemat.timeseries = false`). I would recommend leaving the other entries as `true`, because regenerating them causes `rfxplot` to reload all the individual `SPM.mat` files which usually takes quite a long time.