

Week 6 Exercises

This week's exercises will introduce concepts related to analysis of functional data. You will work with a complete functional dataset collected on the BIAC 1.5T scanner. The dataset contains sessions from one subject who ran in two types of motor tasks, one with a blocked design and the other with an event-related design.

Exercise 1: Understanding the basic concepts of functional data

1.1 Data Organization

The data are organized into the following categories, from largest to smallest:

Examination: Each time that a session is started on the MR console, a unique examination number is generated. These numbers started from 0 on the 1.5T and from 40000 on the 4.0T. Note that one subject may have more than one examination number in rare cases, such as if the scanner computer crashed or we had to interrupt the session for some reason.

Series: Each time a different pulse sequence is run within an exam, it is given a new series number. So we may run one series of a sagittal SPGR scout, a second of an axial spin-echo anatomical, and a third of a spiral gradient echo functional.

Runs: If a single sequence is conducted more than once in succession, each time generates a new run. Mostly, this applies to functional series, where we will do perhaps 8 runs of the same task.

Volume: Each run may contain one or more volumes. For a functional run, there may be 300 volumes or more. Note that each volume corresponds to the data acquired in a single TR.

Slice: Within each volume, there will usually be more than one slice. Typical full-brain functional imaging protocols acquire 22-30 slices. Sometimes fewer slices will be acquired if a particular part of the brain is of interest and having a short TR is desirable.

Voxel: Within each slice, there are a large number of voxels (e.g., 4096 for a 64*64 matrix).

1.2. Data Files

All of the data that you will use in this exercise is in the FUNC_3 directory. **Change to this directory in Windows Explorer and look at the files.**

```
cd \\huxley\data\Class.01\Examples\FUNC_3\Run1_Block
```

Types of Data Files:

pfiles: The functional data comes from the scanner stored in "P files". Each P file contains the raw k -space data from an entire run. The files are usually very large (>100mb). The P files are not included in these directories; they are archived to tape immediately after being collected.

func: The P files are converted into functional data organized by runs. Each image is saved as a separate file with the extension "img".

anat: The anatomical images are stored in separate series; for this subject they are stored in the ANAT_3 subdirectory. We will not use the anatomical images in this exercise.

1.3 Details of the sample dataset.

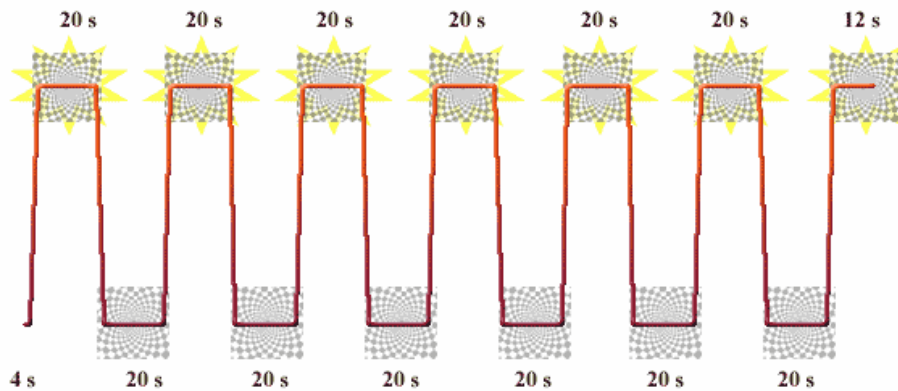
As mentioned above, this dataset consists of two tasks done with a single subject. The first run used a blocked design, while the second used an event-related design. Note that we have extracted seven slices spanning motor cortex from the original 28 slices to reduce the size of the data set.

Similar imaging parameters were used for both runs, **except for TR**:

Field Strength	1.5 T
Sequence	Gradient-Echo Spiral
Matrix size (x × y)	64 × 64
Slices (z)	7
Field of view	24 cm
Voxel size (x × y × z)	3.75 mm × 3.75 mm × 3.8 mm
Orientation	Axial
TR	2000 (blocked);1000 (event-related)
TE	40

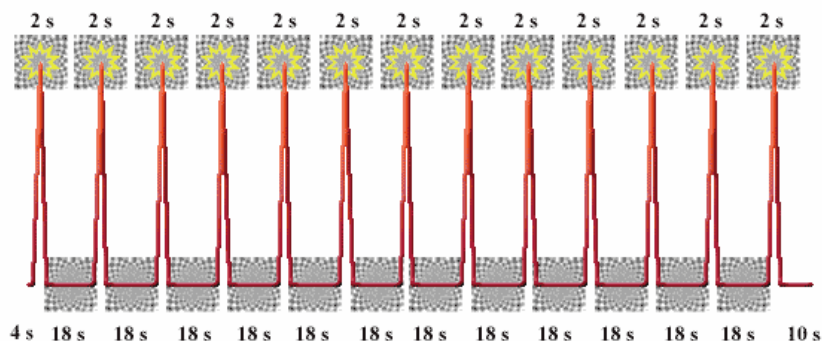
EXPERIMENTAL PARAMETERS — RUN 1 (Blocked Design)

A static checkerboard is displayed for the first two TRs (4 s). Alternating blocks begin with the third TR. The first block displays a flashing checkerboard for 10 TRs (20 s). The second block displays a static checkerboard for 10 TRs (20 s). This set of flashing/static blocks is repeated six times. After the final static block, another 6 TRs (12 s) of flashing checkerboard appear.



EXPERIMENTAL PARAMETERS — RUN 2 (Event Related)

A static checkerboard is displayed for the first 4 TRs (4 s). A flashing checkerboard is then displayed for 2 TR (2 s), after which a static checkerboard is displayed for 18 TRs (18 s). This pattern is repeated 12 times, after which a final static checkerboard is displayed for 10TRs (10 s).



Exercise 2: Using mrtest

In this exercise, you will use the analysis program `mrtest` to do a blocked-design analysis of these data. You need to create two files, a header and a paradigm, and then run `mrtest`. Create a directory under "students" to put these files for your group. Note that you will need separate subdirectories for the flashing and motion data if you analyze both.

2.1. Creating a header file

You should copy the "sample_mrtest_header.txt" file from <\\huxley\data\class.01\examples\control> to your group's personal directory and rename it. You will need to modify the file so that it has the right values for each variable.

2.2. Creating a paradigm file

You should create a paradigm file that consists of a series of time points that correspond to the points you want to include in each condition (e.g., flash vs. static). For each TR in the time series of volumes, you should either have a 1 (for task), a 2 (for non task), or a 0 for points that don't go into either category. It is probably easiest to create these using Excel, and then paste the results into a text file.

There are two things that you should consider when creating your paradigm file:

- 1) Introducing a delay from the onset of the block to the label of "task" activation
- 2) Inserting extra spacing between the blocks to account for transitions between them

2.3. Running `mrtest`

To run `mrtest`, simply start Matlab and type `mrtest`, and then select your header file. If MATLAB does not find `mrtest`, just add the appropriate directory to your path:

```
addpath P:\MATLAB\BIAC\MR
```

The output of `mrtest` is a file called `Map_1-2.T`; this file is a statistical map showing the t-statistic for each voxel based on your paradigm file. You can view this in `showsrs2`; use a single functional image from one of the runs as a base image.

Exercise 3: Using tstatprofile on event-related data

In this exercise, you will use the analysis program `tstatprofile` to do an event-related analysis of these data. You need to create three files, a header, a paradigm, and a template, and then run `tstatprofile`. Use the directory under "students" to put these files for your group. The output of `tstatprofile` will not overwrite that of `MRtest`.

There is an online help file for `tstatprofile` at: <http://www.biac.duke.edu/library/documentation/tstatprofile/>

3.1. Creating a header file

You should copy the "sample_tstat_header.txt" file to your directory and rename it. You will need to modify the file so that it has the right values for each variable.

3.2. Creating a paradigm file

You should create a paradigm file that consists of a series of time points that correspond to the points you want to include in each condition (e.g., flash vs. static). NOTE: THIS PARADIGM FILE HAS DIFFERENT PROPERTIES FROM THE ONE YOU CREATED FOR MRTEST. Here, you should indicate where events happened in the time series by putting a 1 in the paradigm file at the TR where each event happened.

3.3. Creating a template file

The template file is a series of weights that indicate your hypothesis about the possible hemodynamic response following each stimulus. Remember what we have talked about regarding the general form of the hemodynamic response when creating your template file.

3.4. Running *tstatprofile*

To run *tstatprofile*, simply start Matlab and type ***tstatprofile***. You will then need to select your header file. If MATLAB does not find *tstatprofile*, just add the appropriate directory to your path:
`addpath P:\MATLAB\BIAC\MR`

Tstatprofile provides a large number of different types of output. See the *tstatprofile* instructions in the education section of the website for details. You should look at the different types of output to see what they represent.

Exercise 4: Using *tstatprofile* on blocked data

Once you've run *tstatprofile* on the event-related data, you can run it on the blocked data. Note that you have to think about the blocked data as events. What would be good event in the experimental design for you to synchronize your analyses?

4.1. Determining the timing of activity change over blocks

Your goal should be to create an example of the changing BOLD time course over the experimental blocks. Remember to create a new header file with a different output directory, so that you do not overwrite your previous *tstatprofile* analysis.

Exercise 4: Comparisons of output results

Note that these tools give you very different output. You should be able to answer the following questions (use a separate sheet):

4.1. How does the *mrtest* output change depending upon changes in the paradigm file? If you shift the position of blocks in your paradigm file up or down (i.e., earlier or later in time), what happens?

4.2. What is the optimal hemodynamic delay?

4.3. What is the optimal spacing between blocks?

4.4. What different types of output does *tstatprofile* give?

4.5. Why would you use each of the different types of outputs?

4.6. What is the form of the best template file that you have created?

4.7. How do the activation maps compare between the analyses of the blocked and event-related runs? How do they compare between the two types of analysis of the blocked data?