fMRI Design Explorer:
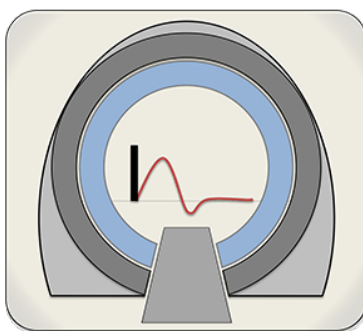
A program to design, optimize, and visualize task-based fMRI experimental designs

User Manual

Contents

List of Figures

Introduction

Complete introduction forthcoming

1.1 About the Program

1.2 About this Manual

This manual provides an interactive and comprehensive description of the functionality of the fMRI Design Explorer. When this PDF is viewed electronically, it offers a number of helpful internal and external hyperlinks. In addition, hovering over the hyperlink with a mouse should result in a preview of the link's content. These hyperlinks have been tested on both Mac OS X and Linux operating systems. However, individual operating system and PDF viewer idiosyncrasies may prevent functional hyperlinks. Please report all hyperlink issues to Dustin Moraczewski at dmoracze@umd.edu.

1.2.1 Internal Hyperlinks

Hyperlinks colored green refer to internal links. Clicking an internal link will navigate the PDF to the corresponding section or figure. For example, if the text refers to Figure 1, clicking on this link will navigate the PDF to Figure 1. In addition, the Table of Contents is fully interactive, however (as you probably noticed) the text color is not green. This decision was made for readability.

1.2.2 External Hyperlinks

Hyperlinks colored blue refer to external links. Clinking an external link will initiate behavior external to the PDF. There are three types of external links:

1. URL link - this will open a browser and navigate to a specific URL.

2. Email link - this will open an email client, if one is configured (currently does not work on the University of Maryland's GLUE network)

3. Download link - this will download relevant files (e.g. the Design Explorer installer). This link may also open a browser that navigates to Dropbox on certain operating systems.

For example, if the text refers to the AFNI package for fMRI analysis, clicking on the corresponding link will open a browser that navigates to the AFNI homepage.

1.3 Other Documentation

Installation

2.1   System Requirements

64-bit machines running Mac OS X & Linux have been tested.
It is likely that the program will run on others and may fail with some.

Windows operating system is not supported.

2.1.1   Dependencies

Mac OS X – XQuartz is required.
Linux – no dependencies.

2.2   Download

Please download the current Design Explorer installer here.

2.3   Install

1. Open Terminal console.

2. Navigate to the directory that contains the installer.

   ```
   > cd <path_to_installer>
   ```

3. Change installer to executable, changing the version number appropriately.

   ```
   > chmod 755 Install_DesignExplorer_XXXX-XX-XX.XXXX
   ```

4. Run the installer.

   ```
   > ./InstallDesignExplorer
   ```

5. Pay attention to the output in the terminal. If there is an error, you should report this text to the developers.

6. Once the installer is finished, an executable file called `DesignExplorer` will be deposited into your home (`~`) directory.

2.4   Run

1. To run the Design Explorer you can either double click on the `DesignExplorer` file OR you can run the program in the terminal:

   ```
   > ~/DesignExplorer
   ```

The executable file `DesignExplorer` is a bash script that points to a `~/.DesignExplorer` hidden directory that contains a python distribution, all dependencies, AFNI's 3dDeconvolve, and the default project.

2.5   Uninstall

To uninstall the Design Explorer, simply open the terminal and type:

```
> rm -rf ~/.DesignExplorer
```

Be aware, though, that this will also delete the default project as well.

Please email Dustin Moraczewski with any issues related to installation. Be sure to be as descriptive as possible and provide your system configuration, what you did, and what was the output.

Overview

To see the forest rather than jumping right to the trees, this section gives users a quick overview of the multiple large-scale features of the Design Explorer. In addition, this section provides tips on getting started, general information and an overview of the interface.

3.1   Getting Started

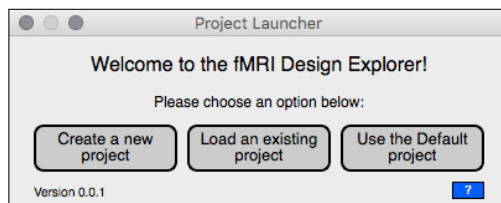Upon executing the Design Explorer, the Project Launcher window will appear:

Figure 1: Project Launcher

Users have a three choices regarding how they want to initialize a project:

1. Create a new project
2. Load an existing project
3. Use the Default project

3.1.1   Create a new project button

To create a new project within a user-defined location, choose this option. A file directory dialog window will appear and the user will need to choose a location for their new project. Typically, prior to the creation of a new project, users will create a unique directory for their project. This unique directory will ensure that all files relevant to the project are contained within a single location.

1. Choose the **Create a new project** button
2. Navigate to desired directory
3. Click **choose**
4. Enter project name
5. Click **ok**

The new project will be created and the main window will appear (Figure 2).

3.1.2   Load an existing project button

If a previous project had been created within a user-defined location (using the Create a new project button), users can choose to load this already existing project.

1. Choose the **Load an existing project button**
2. Navigate to project's directory
3. Click **choose**

The project will load where the user left off, including all variants.

3.1.3 Use the Default project button

The default project exists so that users can rapidly create and interact with experimental designs without cluttering up the file system. Upon initialization of the default project, the Design Explorer will deposit a directory called `default_project` within the `~/.DesignExplorer` directory and the user will be asked to name the project. If a default project already exists, clicking the **Use the Default project** button will open the previously saved default project.

To remove the default project, open a terminal and type:

```
> rm -rf ~/.DesignExplorer/default_project
```

3.2 General Information

3.2.1 Autosave

All changes within a valid design are autosaved, thus, there is no need to save manually. However, there is a caveat to the autosave feature: in order for changes to be saved, the changes must occur within a valid design and be valid with respect to the parameters of interest. For example, the autosave feature will not save changes under the following scenarios:

1. If events are defined and/or configured without first creating a design using the $\oplus$ button, the changes will be lost because they did not occur within the context of a design.

2. Entering a character instead of a number into the **TR** field would not be saved since this field expects a numeric value.

These scenarios are meant to represent the types of behaviors that would fail to autosave and do **not** encompass all possibilities.

3.3 Overview of Interface

This section gives a brief overview of the main interface that is common to all Tabs. Please see the Tabs section for functionality unique to each Tab. Figure 2 depicts the large-scale structure of the Design Explorer interface.

Consider the number annotations in Figure 2:

1. The Tab menu:
   This menu consists of the main functional divisions of the Design Explorer. The tabs consist of Design, Visualize, and Simulate.

2. The Design menu:
   This menu consists of the user-defined designs currently in the workspace. In Figure 2, this menu is currently blank which means that no designs have been created yet. A design must be created before defining or configuring events.

3. The New Design button $\oplus$:
   This button adds a new design to the Design Menu. When this button is clicked, users will be asked to enter a name for the new design. All designs must have unique names.

4. The Help Button ?
   This button is available in all three tabs.

5. The Plot window:
   This window is available in all tabs, however, the information presented within this window is unique to each tab. For example, the plot window in the design tab shows the event plots whereas this window in the visualize and simulate tabs show the ideal and simulated BOLD signal, respectively.
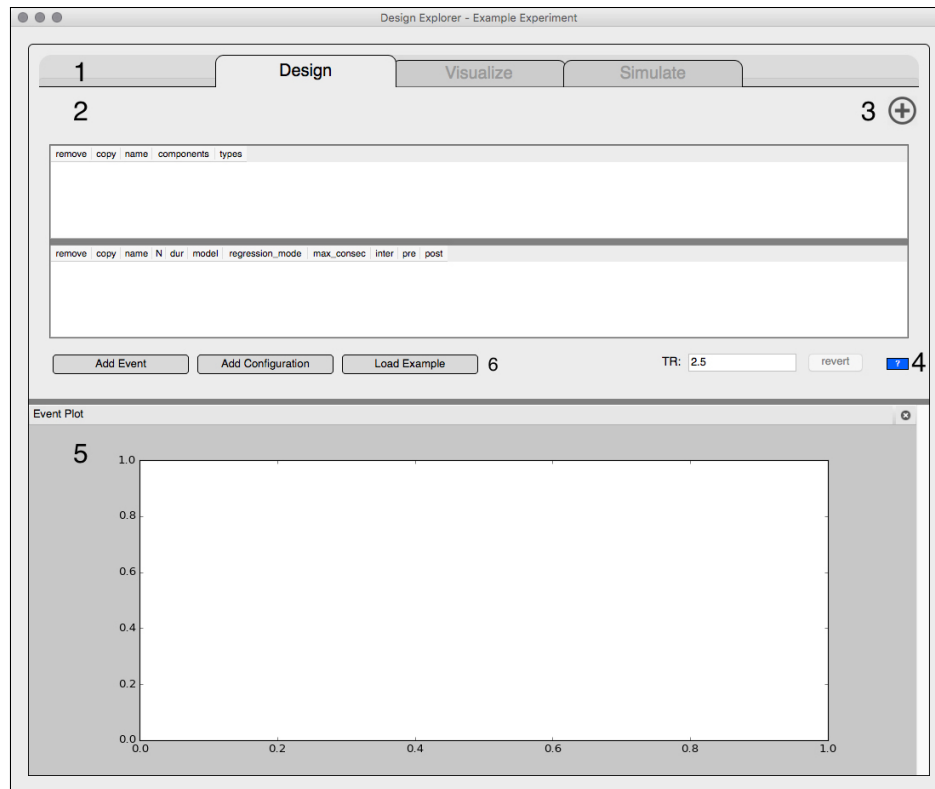
Figure 2: Overview of Interface

6. The Load Example button:
   Click this button to load specific examples from this workbook.

3.4   Tabs

This section gives a brief, big-picture overview of the functionality of each of the tabs. Please see the corresponding chapters for greater detail regarding the implementation of the features.

3.4.1   Design

The purpose of the Design tab is three-fold:

1. Define events into an event hierarchy

2. Configure the various parameters of the events

3. View experimental design as an event plot

There are multiple ways in which users can define their event hierarchy, see the Event Hierarchy chapter for more information. With regards to event configuration, users have complete control of most parameters: number of events, event duration, hemodynamic response models, maximum consecutive events of the same type, pre, inter, and post baseline periods. Both events convolved with a hemodynamic response and not are configurable.

3.4.2   Visualize

The purpose of the visualize tab is to view the idealized (no noise added) BOLD signal on your experiment. Users can also view event-specific signal to examine the relative contributions of each event to the overall BOLD signal. This signal is derived from the events that are configured to include a hemodynamic response

model. In addition, the Event Configuration table is also supplied so that users can make adjustments to the event parameters and see the changes in the BOLD signal in real time.

3.4.3 Simulate

The purpose of the Simulate tab is to use an experimental design to simulate BOLD signal according to user-defined event-specific β weights, event-specific noise, as well as global noise. Then, once new BOLD signal is simulated, users can examine the error in the recovered β estimates. The number of simulations is a user-defined parameter.

3.5 Help

3.5.1 The Load Example Button

The load example button is a quick and easy way to load the designs discussed in the units of the workbook. Thus, users can load an example quickly and begin interacting with the designs without spending time defining and configuring each individual event.

3.5.2 The Help Button

The help button , as seen in the lower right corner of Figure 1, will open this user manual to provide further assistance on the implications of these decisions. This button is available in the Project Launcher, Design tab, Visualize tab, and Simulate tab.

3.5.3 The Revert Button

The revert button will be available on all tabs. If you make a change that results in an error or misspecified design, the revert button will automatically enable. Clicking this button will revert your settings back to the most recent correctly-specified settings.

3.5.4 Debug Windows

In the current version, initializing the Design Explorer will also open two other windows: a terminal window and a debug window. Each of these windows provides different information about the background processes. Overall, these windows are not necessary to keep in view, however, if there are any errors or the program is not behaving as you expected it, the out put of these windows can help.

3.5.5 Further Assistance

If you find a bug, please Dustin Moraczewski with a description of what happened and any text you see in these windows. In the interest of fixing all bugs, the more information provided to the developers the better. A detailed description of what you did, screen shots, and debug and terminal window text would provide a helpful overview in order to diagnose what happened.

Prior Considerations

4.1   Establish Task

First and foremost, users must have a strong idea of their research question and design their task accordingly. While not all fMRI experiments can be appropriately optimized in the Design Explorer, the first step is for the researcher to know the general layout of their task and the motivation for these decisions. This knowledge will inform future steps such as constructing a minimum working hierarchy, defining experimental constrains, deciding which events to model with the hemodynamic response, and which events will be used to optimize the experimental design. The Design Explorer has been constructed to optimize fMRI experiments in which researchers are interested in convolving regularly occurring stimulus events with a hemodynamic response in order to apply a general linear model framework (GLM) to their statistical analysis. Thus, experiments that do not utilize this framework can not be appropriately optimized using this program. Such experimental designs include, but are not limited to, resting state, naturalistic viewing, irregular event presentations, or experiments that rely on participant response to determine event timing (as this cannot be predicted, though one could attempt to estimate based on behavioral pilots, however, events must be regularly occurring). Future effort will be made to expand the features of the Design Explorer to include more complex and diverse experimental designs.

4.2   Minimum Event Hierarchy

Once the task has been established, the next step is to identify a minimum event hierarchy (MEH). The MEH is a schematic of the task that encompasses all levels of the event hierarchy. While all levels of the hierarchy must be accounted for, reoccurring events (e.g. multiple trials within a run) is not necessary. Typically it is helpful to sketch this in a notebook or on a white board since there are many ways one could conceive of organizing their task. In addition, fMRI experiments usually also include periods of rest or baseline at the beginning and the end of each run. These events should also be included in the MEH.
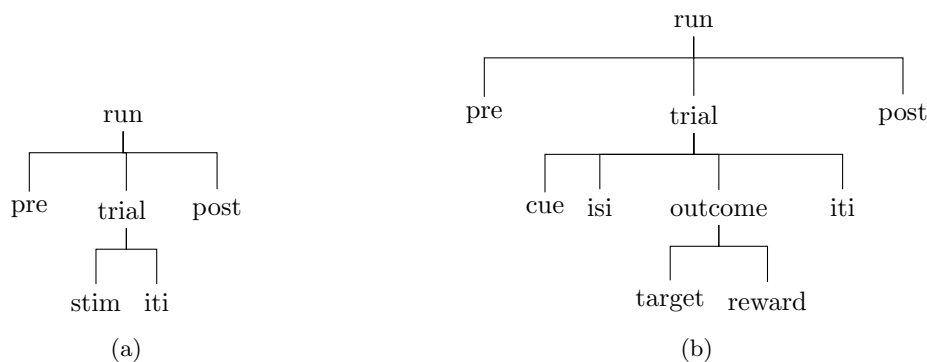
4.2.1   Examples

Figure 3: Example minimum event hierarchy

(a) This experiment consists of a simple stimulus presentation where each trial will be followed with an inter-trial interval (iti). In addition, there are baseline periods at the beginning and the end of each run (pre and post, respectively)

(b) This is an example MEH of a classic monetary incentive delay task. In addition to pre and post run baseline periods, the run will consist of trials. These trials will consist of cue, isi, outcome, and iti events. In addition, the outcome will further consist of target and reward events. One would combine the target and reward events as such if both events wanted to always be modeled together.

4.3   Define Experimental Constraints

Design Tab

5.1 Overview

As outlined in the previous chapter, the Design Tab will be used to define your event hierarchy and event-specific parameters (e.g. number of events, event duration, etc...). Please note that you must have an idea of the minimum event hierarchy prior to defining the events in the Design Explorer. First, this section will give an overview of the Design Tab interface:
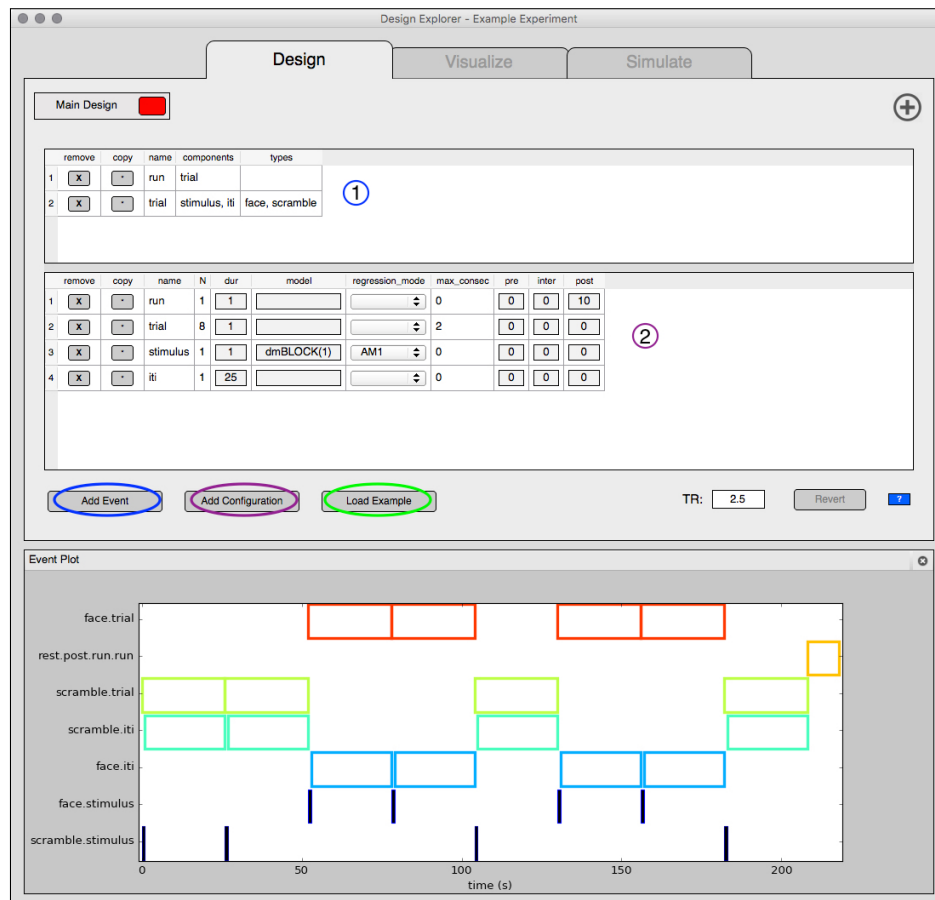

Figure 4: Design Tab interface

1. The Event Definition Table (①)

2. Add event to the Event Definition Table (blue circle)

3. The Event Configuration Table (②)

4. Add event to the Event Configuration Table (Purple circle)

5. **TR** entry: Change the repetition time (in seconds)

6. The Revert Button

7. The Help Button (?)

8. The Event Plot

5.2 Event Definition Table

The Event Definition Table is where you specify the event hierarchy of the experiment. This table has 5 columns:

1. **remove**

   This column consists of a button that, if clicked, will delete this row from the table.

2. **copy**

   This column consists of a button that, if clicked, will copy this row into a new row.

3. **name**

   This column consists of a text entry field that will take the user-defined name of an event. Event names should be short, but identifiable strings with no spaces. Each event row must have a unique name.

4. **components**

   This column consists of a text entry field that will take other event names that will be nested within the event defined in the **name** column. These names should also be short strings with no spaces. Multiple components should be separated by a comma. See below for examples.

5. **types**

5.2.1 Event Definition Examples

5.3 Event Configuration Table

5.4 The Event Plot

Visualize Tab

6.1 Overview of Interface

With a figure, briefly describing the functionality of each button

6.1.1 Events

6.1.2 Parameters

6.1.3 Options

6.1.4 Visualizer

6.2 Comparing Variants

Simulate Tab

7.1 Overview of Interface

With a figure, briefly describing the functionality of each button

7.1.1 Events

7.1.2 Parameters

7.1.3 Options

7.1.4 Visualizer

7.1.5 Simulation