Factorial Extension to the Mass Univariate Toolbox
Documentation

Version Date: 22 June 2017
Author: Eric Fields
Contact: Eric.Fields@tufts.edu

!!!
IMPORTANT NOTE: This is a beta version of this software. Do not assume the results are accurate without further independent testing. Please report any bugs, problems, issues, or suggestions to Eric.Fields@tufts.edu
!!

Background

The Factorial Extension to the Mass Univariate (FMUT) is an extension to David Groppe's Mass Univariate Toolbox (MUT). The MUT implements *t*-tests for several different mass univariate approaches to the analysis of ERP data. The FMUT adds to this by implementing one-way and factorial ANOVA versions of the same mass univariate approaches.

This documentation assumes you are familiar with the mass univariate approach and the various multiple comparisons corrections that can be used. It also assumes you are familiar with the Mass Univariate Toolbox and have some basic familiarity with MATLAB.

For reviews of the motivation, logic, and implementation of mass univariate statistics for ERP data, see:

Luck, S.J. (2014). The mass univariate approach and permutation statistics. https://mitpress.mit.edu/sites/default/files/Ch_13_Mass_Univariate_and_Permutations_0.pdf

Groppe, D. M., Urbach, T. P., & Kutas, M. (2011). Mass univariate analysis of event-related brain potentials/fields I: A critical tutorial review. *Psychophysiology, 48*(12), 1711-1725.

For introduction to the functionality and usage of the Mass Univariate Toolbox, see:
http://www.openwetware.org/wiki/Mass_Univariate_ERP_Toolbox

The FMUT uses the GND structure of the MUT. FMUT functions also use the same basic syntax and many of the same options as similar MUT functions. Information available in the MUT documentation will not be repeated below.

Permutation tests for factorial ANOVA

Permutation tests for multiple regression and factorial ANOVA aren't as straightforward as for simpler designs. Permutation tests for simple designs are exact tests: they maintain the Type I error rate at exactly α. For some factorial designs—namely when more than one factor has more than two levels—an exact test

is not possible. In these cases, there are several methods for implementing an approximate test that has a Type I error asymptotic to α as sample size increases and/or noise decreases.

For designs where an exact test is not possible, the FMUT uses the permutation of residuals method to conduct an approximate test. Simulation studies show that unless sample sizes are very small, this method generally maintains the Type I error rate at an acceptable level (Anderson & Ter Braak, 2003; Fields & Kuperberg, unpublished; Still & White, 1981; Winkler, Ridgway, Webster, Smith, & Nichols, 2014).

For background on permutation tests for multiple regression and factorial ANOVA, see:

Anderson, M. J. (2001). Permutation tests for univariate or multivariate analysis of variance and regression. *Canadian Journal of Fisheries and Aquatic Sciences, 58*(3), 626-639.

Anderson, M., & ter Braak, C. T. (2003). Permutation tests for multi-factorial analysis of variance. *Journal of Statistical Computation and Simulation, 73*(2), 85-113.

Wheldon, M. C., Anderson, M. J., & Johnson, B. W. (2007). Identifying treatment effects in multi-channel measurements in electroencephalographic studies: multivariate permutation tests and multiple comparisons. *Australian & New Zealand Journal of Statistics, 49*(4), 397-413.

Winkler, A. M., Ridgway, G. R., Webster, M. A., Smith, S. M., & Nichols, T. E. (2014). Permutation inference for the general linear model. *NeuroImage, 92*, 381-397.

FMUT Installation

Dependencies

For the FMUT to work, you need to have MATLAB and you must have EEGLAB and the Mass Univariate Toolbox installed. You may also need to have an instance of EEGLAB running to make sure that all EEGLAB functions are on the MATLAB search path.

Installing FMUT

To install the FMUT, simply unzip the folder, and place it somewhere on the MATLAB search path—probably the same location as the MUT functions. For further help, see:
http://www.openwetware.org/wiki/Mass_Univariate_ERP_Toolbox:_installing_the_Mass_Univariate_Toolbox

Statistical tests

The FMUT implements one-way and factorial within-subject ANOVAs. Between subjects factors are not currently supported, although there are plans to add them in future versions.

F_{max}, cluster mass, and several false discovery rate (FDR) corrections are supported. These are implemented as functions that closely mirror the `tmaxGND`, `clustGND`, and `tfdrGND` functions from the MUT. Most of the options are the same as for those functions, so I will not review them here. The documentation for each function with a full list of options can be seen by typing

```
>> help FUNCTION NAME HERE
```

at the MATLAB command line. Below I review how usage of these functions differs from the *t*-test versions in the MUT.

Specifying the model

The MUT implements all tests as one-sample *t*-tests. To test the difference between two conditions, you must first make a difference wave representing the effect you want to test.

For more complex factorial designs, it is often not possible to reduce effects to a single difference wave. The input to FMUT statistical functions is thus somewhat more complicated.

Instead of specifying a single bin, for FMUT functions you must provide the bins for all the conditions in your design. Let's say you are interested in the effects of priming and corpus frequency on the ERPs elicited by words. Your study uses a 3 (Frequency: low, medium, high) X 2 (Priming: primed, unprimed) design and you want to conduct and ANOVA that will test the two main effects as well as the two-way interaction. In your GND, the bins representing the six relevant conditions are as follows:

- Bin1: unprimed low frequency
- Bin2: unprimed medium frequency
- Bin3: unprimed high frequency
- Bin4: primed low frequency
- Bin5: primed medium frequency
- Bin6: primed high frequency

A cluster mass test of this design can be conducted with the following:

```
GND = FclustGND(GND, 'bins', 1:6, 'factor_names', {'Frequency', 'Priming'}, …
                factor_levels', [3, 2])
```

'bins' specifies that bins 1 through 6 contain the six conditions of the 3 x 2 design. For 'factor_names' and 'factor_levels', the factors are listed in fastest to slowest moving order within the bins specified. (Of course, you will likely want to specify other inputs as well, but these are the same or very similar to the MUT `clustGND` function.)

Output

The output to file option for FMUT functions is a little different than for MUT functions. Instead of outputting to an ASCII file, FMUT functions output to a spreadsheet. This allows for outputting the full results of factorial ANOVAs in a usable and readable fashion.

When using the 'output_file' option, you will need to specify the name of a .xlsx file. This spreadsheet will contain the full results including the design and options of the test, cluster statistics and ids, *F*-values, and *p*-values. It should also be formatted for convenient viewing (but see below).

You may find it easier to view and work with the results in this spreadsheet than in MATLAB. This spreadsheet is also intended be something that can be included as supplementary materials with publications to report full results.

If you would like to produce an output spreadsheet after a test has already been run and stored in the GND variable, you can use the `Ftest2xls` function.

If you would like to produce an FMUT-style output spreadsheet for *t*-test results from the MUT, you can use the `ttest2xls` function.

NOTE FOR MAC AND LINUX USERS:

Due to limitations with MATLAB's `xlswrite` function, FMUT is distributed with a third-party function (`xlwrite`) for writing to spreadsheets on non-Windows system. This function seems to work for basic output purposes, but I haven't tested it nearly as much as the Windows equivalent.

One issue with this solution is that the code that applies formatting to spreadsheet output has problems working with `xlwrite` created spreadsheets. As a result, spreadsheets are output without formatting applied by default. Here is a slightly annoying work-around if you want to add the formatting:

1. Create the spreadsheet without formatting (this is the default on non-Windows systems).
2. Open the spreadsheet in Excel and re-save it.
3. Run `format_xls('filepath/spreadsheet.xlsx')` at the MATLAB command line.

List of functions

For more information and a full list of features for each function type
`>> help FUNCTION NAME HERE`
at the command prompt in MATLAB.

MAIN FUNCTIONS:

FmaxGND	F_{max} test for one-way and factorial ANOVA. This is an ANOVA version of `tmaxGND` with very similar syntax.
FclustGND	Cluster mass test for one-way and factorial ANOVA. This is an ANOVA version of `clustGND` with very similar syntax.
FfdrGND	FDR based correction for factorial ANOVA. This is an ANOVA version of `tfdrGND` with very similar syntax.
ttest2xls	This function outputs *t*-test results (e.g., from tmaxGND or clustGND) to a spreadsheet in the style of FMUT functions.

SUB-FUNCTIONS

(these are generally not called directly)

calc_Fclust	Sub-function to calculate a single effect for cluster mass tests. Called by `FclustGND`.
calc_Fmax	Sub-function to calculate a single effect for F_{max} tests. Called by `FmaxGND`.
calc_param_ANOVA	Sub-function calculating a parametric ANOVA at each time point and electrode. Called by `FfdrGND`.
F_sig_raster	An F-test equivalent of the MUT `sig_raster` function.
fmut.py	Python module for FMUT. Provides a function to format spreadsheet output.
format_xls	MATLAB function for applying formatting to FMUT spreadsheet output. Calls fmut.py or py_fmut.exe (Windows) or py_fmut (Mac).
Ftest2xls	Output *F*-test results to Excel. Called by all *F*-test functions when the 'output_file' option is used.
get_effects	Sub-function that figures out all the effects (i.e., interactions and main effects) in a model and their labels based on the factor names. Called by several functions.
get_int_res	Calculate interaction residuals for approximate tests of interaction effects
perm_rbANOVA	Calculate permutation within-subjects ANOVA (*F*-observed and *F*-distribution). Used by the F_{max} and cluster mass tests.
py_addpath	General purpose function for updating the Python import path from MATLAB

py_fmut	“Compiled” version of fmut.py that will work without a Python interpreter for Mac.
py_fmut.exe	“Compiled” version of fmut.py that will work without a Python interpreter for Windows
reduce_data	Reduce data across irrelevant factors or reduce data for exact tests of interactions.

THIRD-PARTY FUNCTIONS & LIBRARIES

(All third party code and libraries are released under permissive free and open source licenses. Licenses are distributed with FMUT.)

Apache POI library	Java API for Microsoft Office documents. Called by `xlrwite`. See here.
xlwrite	Function for writing to spreadsheets on non-Winows systems. See here.