Factorial Mass Univariate Toolbox
Documentation

Version Date: 28 June 2017
Author: Eric Fields
Contact: Eric.Fields@tufts.edu

If you use the toolbox to perform analyses or to produce figures used in a publication, please cite it. We are currently working on a paper explaining and motivating the tests implemented in the FMUT and providing simulation results of their Type I error rate and power. For now, you can cite the software directly:

Fields, E. C. (2017). Factorial Mass Univariate Toolbox [Computer software]. https://github.com/ericcfields/FMUT

Background

The Factorial 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, so it is highly recommended you read the MUT tutorial before reading this documentation.*

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 the References and Further Reading section at the end of this documentation.

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 folder. 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 an 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 (this fixes some issues with the way formatting is encoded).
3. Run `format_xls('filepath/your_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).
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]().

Simulation Results

Note: This simulation work is ongoing and these results should be considered preliminary. Results may be updated or augmented before publication.

Here I present some initial simulation results on the Type I error rate and power of the permutation-based mass univariate statistics implemented by the FMUT.[1]

These simulations were carried out by randomly selecting from a large set of real EEG noise trials (i.e., EEG epochs with the average subtracted; see Groppe, Urbach, & Kutas, 2011b). For simulations of power, real ERP effects from studies collected in our lab were added to the ERPs formed from randomly selected noise trials.

Note that simulations were performed for two large and widely distributed effects (the N400 and P300). Performance of these methods may be different for short-lived and spatially focus effects (see simulations in Groppe, Urbach, & Kutas, 2011b).

First, simulations show that when spatial and temporal assumptions are matched, the permutation-based mass univariate approach has similar or better power than the traditional mean time window and spatial ROI approach. When assumptions are relaxed, the permutation approaches—in particular the cluster mass approach—show only modest decreases in power.

Effect	Time Window	Electrodes	cluster mass power	F_{max} power	mean amplitude parametric power	cluster temporal extent	F_{max} temporal extent
N400	300 - 500	centroparietal ROI	0.82	0.77	0.76	144	73
N400	200 - 600	full 32	0.81	0.68	0.45	186	41
N400 (restricted)	200 - 600	full 32	0.79	0.68	0.29	159	39
N400	0 - 1000	full 32	0.69	0.57	0.15	216	34
P300	500 - 750	centroparietal ROI	0.39	0.34	0.34	144	45
P300	400 - 900	full 32	0.25	0.22	0.09	210	29
P300	0 - 1000	full 32	0.22	0.17	0.08	247	24

---

[1] Adapted from: Fields, E. C. (2017). *Event-related potential and functional MRI studies of emotion and self-relevance* (Doctoral Dissertation). Tufts University, Medford, MA.

Second, consistent with previous simulation work (Anderson & Ter Braak, 2003; Still & White, 1981; Winkler, Ridgway, Webster, Smith, & Nichols, 2014), our results show that the permutation of residuals method maintains the Type I error rate at acceptable levels with realistic ERP data and sample sizes (n=24).

Mult Comp Corr	Time Window	Type I Error Rate
cluster mass	0 - 300	0.054
cluster mass	500 - 900	0.053
F_{max}	0 - 300	0.052
F_{max}	500 - 900	0.058

References and Further Reading

Introduction and overview of mass univariate statistics and corrections for ERP data

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

Groppe, D. M., Urbach, T. P., & Kutas, M. (2011b). Mass univariate analysis of event-related brain potentials/fields II: Simulation studies. *Psychophysiology, 48*(12), 1726-1737.

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

The issue of multiple comparisons and biased statistics in ERP analysis

Kilner, J. M. (2013). Bias in a common EEG and MEG statistical analysis and how to avoid it. *Clinical Neurophysiology, 124*(10), 2062-2063.

Luck, S. J., & Gaspelin, N. (2017). How to get statistically significant effects in any ERP experiment (and why you shouldn't). *Psychophysiology, 54*(1), 146-157.

Permutation and resampling statistics

Good, P. I. (2005). *Permutation, parametric and bootstrap tests of hypotheses* (3rd ed.). New York, NY: Springer.

Manly, B. F. J. (1997). *Randomization, bootstrap, and Monte Carlo methods in biology* (2nd ed.). London, UK: Chapman & Hall.

General background on ANOVA and related issues

Kirk, R. E. (2013). *Experimental Design* (4th ed.). Thousand Oaks, CA: SAGE Publications.

Permutation tests for ANOVA, factorial designs, and multiple regression

Good general overviews:

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.

The original papers outlining the permutation of residuals method:

Freedman, D., & Lane, D. (1983). A nonstochastic interpretation of reported significance levels. *Journal of Business & Economic Statistics, 1*(4), 292-298.

Still, A. W., & White, A. P. (1981). The approximate randomization rest as an alternative to the F test in analysis of variance. *British Journal of Mathematical and Statistical Psychology, 34*(2), 243-252.

An initial paper on using the permutations of residuals method and F$_{max}$ method with ERP data:

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.

Permutation tests and multiple comparisons for multiple regression and factorial designs with fMRI data:

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.

The t_{max}/F_{max} procedure

Blair, R. C., & Karniski, W. (1993). An alternative method for significance testing of waveform difference potentials. *Psychophysiology, 30*(5), 518-524.

The cluster mass procedure

Bullmore, E. T., Suckling, J., Overmeyer, S., Rabe-Hesketh, S., Taylor, E., & Brammer, M. J. (1999). Global, voxel, and cluster tests, by theory and permutation, for a difference between two groups of structural MR images of the brain. *IEEE Transactions on Medical Imaging, 18*(1), 32-42.

Maris, E., & Oostenveld, R. (2007). Nonparametric statistical testing of EEG- and MEG-data. *Journal of Neuroscience Methods, 164*(1), 177-190.

False discovery rate

Benjamini, Y., & Hochberg, Y. (1995). Controlling the false discovery rate: A practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society Series B: Methodological, 57*(1), 289-300.

Benjamini, Y., Krieger, A. M., & Yekutieli, D. (2006). Adaptive linear step-up procedures that control the false discovery rate. *Biometrika, 93*(3), 491-507.

Benjamini, Y., & Yekutieli, D. (2001). The control of the false discovery rate in multiple testing under dependency. *Annals of Statistics, 29*(4), 1165-1188.