

# $\mu$ TOSS Quick Start Guide

Gilles Blanchard, Weierstrass Institute for Applied Analysis and Stochastics Berlin

Thorsten Dickhaus, Humboldt-University Berlin

Niklas Hack, Medical University of Vienna

Frank Konietzschke, Georg-August-University Göttingen

Kornelius Rohmeyer, Leibniz University Hannover

Jonathan Rosenblatt, Tel Aviv University

Marsel Scheer, German Diabetes Center

Wiebke Werft, German Cancer Research Center

December 5, 2011

## Abstract

$\mu$ TOSS is an R package providing an open source, easy-to-extend platform for multiple hypothesis testing (MHT), one of the most active research fields in statistics over the last 10-15 years. Its first motivation is to establish a common platform and standardization for MHT procedures at large. The  $\mu$ TOSS software has been designed and written in the framework of a “Harvest Programme” call of the PASCAL2 European research network. Basically, it consists of the two R packages **mutoss** and **mutossGUI**. For researchers, it features a convenient unification of interfaces for MHT procedures (including standardized functions to access existing specific MHT R packages such as **multtest** and **multcomp**, as well as recent MHT procedures that are not available elsewhere) and helper functions facilitating the setup of benchmark simulations for comparison of competing methods. For end users, a graphical user interface and an online user’s guide for finding appropriate methods for a given specification of the multiple testing problem is included. Ongoing maintenance and subsequent extensions will aim at establishing  $\mu$ TOSS as a state of the art in statistical computing for MHT.

## 1 Introduction

The  $\mu$ TOSS packages allow the user to discover, apply and compare multiple testing procedure and multiple interval estimation procedures.

The  $\mu$ TOSS packages include a corpus of functions implementing and integrating these procedures and a GUI. These are found in the **mutoss** and **mutossGUI** packages respectively.

## 2 $\mu$ TOSS Rationale

The rationale behind the  $\mu$ TOSS packages is two-fold.

It is aimed at allowing statisticians to discover, apply and compare standard and custom multiplicity controlling procedures. This is achieved by the `mutoss` package.

It is also aimed at the researcher wishing to Analyse new data or reproduce published results. This is accomplished by the `mutossGUI` package.

At the time of release, the package has only undergone basic testing. This being the case, we recommend new data to be analyzed with standard software alongside  $\mu$ TOSS. This is planned to change in future releases.

## 3 System Requirements

### 3.1 `mutoss` Package

The package will run on any machine running R with recommended version 2.8 and above.

### 3.2 `mutossGUI` package

On top of the `mutoss` package requirements, Java Run time Environment ver 5 and above is needed.

## 4 GUI Work flow

Download and install the `mutossGUI` package. The GUI should start automatically. Others wise load it with

```
>mutossGUI()
```

### 4.1 Testing of Hypotheses

If you have already a vector of p-values start at step (5).

1. Load the raw data (assumed to be a `data.frame` object) using the **Data** button.
2. Specify the model type and explanatory variables using the **Model** button. For linear contrasts choose **Single endpoint in k groups**. For applying the same model to many response variables choose **Multiple (linear) regression**.
3. Define model by choosing response and explanatory variables.
4. Define the hypotheses of interest by specifying the contrasts using the **Hypotheses** button.

5. Insert p-values using the **p-Value** button or calculate them following the previous steps.
6. Choose the error type to control using the **Error Rate** button.
7. Use the **Adjusted p-Values** to calculate the procedure specific adjusted p-values (you will be prompted for additional options when necessary) or choose **Rejected** to apply the procedure and reject hypotheses.
8. Visualize results by choosing the **Info** option in the **Adjusted p-Values** or **Rejected** buttons.
9. Save the output as an R object using the **File->Export MuToss Object to R** option.

Further analysis is now possible using the `compareMutoss` functions or other R functionality.

## 4.2 Interval Estimations

Steps 1-4 are identical to the hypothesis testing work flow.

1. Load the raw data (assumed to be a *data.frame* object) using the **Data** button.
2. Specify the model type and explanatory variables using the **Model** button. For linear contrasts choose **Single endpoint in k groups**. For applying the same model to many response variables choose **Multiple (linear) regression**.
3. Define model by choosing response and explanatory variables.
4. Define the contrasts of interest by specifying the contrasts using the **Hypotheses** button.
5. Choose the error type to control using the **Error Rate** button.
6. Use the **Confidence Intervals** to compute confidence intervals on parameters of interest.
7. Visualize results by choosing the **Info** option in the **Confidence Intervals** button.
8. Save the output as an R object using the **File->Export MuToss Object to R** option.

Further analysis is now possible using R functionality.

## 5 Command Line Work Flow

Download and install the *mutoss* package to access the different procedures in the package (note *mutossGUI* is not needed for this purpose). A list can be presented using

```
>help(package='mutoss')
```

To work with these elementary functions, just use them as any other R function. See inline help for further details.

To use these functions to read and write into Mutoss S4 class objects use the *mutoss.apply( )* function. See the inline help of the function for further details.

## 6 Glossary

**Hypotheses-Testing-Procedures** The corpus of procedures for testing multiple statistical hypotheses.

**Interval-Estimating-Procedures** The corpus of procedures for constructing interval estimates for multiple parameters.

**p-Value-Procedures** The corpus of (multiple) hypotheses testing procedures which rely on the marginal p-values of each hypothesis (and do not require the original data and model). These procedures might possibly require additional information such as logical relations between procedures, a qualitative description of the probabilistic relation between test statistics etc.

**Data-Procedures** The corpus of (multiple) testing procedures which require the original response variables, the explanatory variables (model) and the parameters of interest (contrasts).

These procedures can be seen as p-value-procedures with a specific relation between test-statistics which is derived from the model and the contrasts.

**Error-Type** The type of error a procedure aims to control. This can be a hypothesis testing error rate (FWER, FDR,...) or an interval estimation error rate (simultaneous coverage, false coverage rate,...).

**Error-Rate** The allowed rate of the *Error Type*.