Journal of Statistical Software

June 2015, Volume 65, Issue 9. *http://www.jstatsoft.org/*

Mann-Whitney Type Tests for Microarray Experiments: The R Package gMWT

Daniel Fischer
University of Tampere

Hannu Oja
University of Turku

Abstract

We present the R package **gMWT** which is designed for the comparison of several treatments (or groups) for a large number of variables. The comparisons are made using certain probabilistic indices (PI). The PIs computed here tell how often pairs or triples of observations coming from different groups appear in a specific order of magnitude. Classical two and several sample rank test statistics such as the Mann-Whitney-Wilcoxon, Kruskal-Wallis, or Jonckheere-Terpstra test statistics are simple functions of these PI. Also new test statistics for directional alternatives are provided. The package **gMWT** can be used to calculate the variable-wise PI estimates, to illustrate their multivariate distribution and mutual dependence with joint scatterplot matrices, and to construct several classical and new rank tests based on the PIs. The aim of the paper is first to briefly explain the theory that is necessary to understand the behavior of the estimated PIs and the rank tests based on them. Second, the use of the package is described and illustrated with simulated and real data examples. It is stressed that the package provides a new flexible toolbox to analyze large gene or microRNA expression data sets, collected on microarrays or by other high-throughput technologies. The testing procedures can be used in an eQTL analysis, for example, as implemented in the package **GeneticTools**.

Keywords: eQTL, Jonckheere-Terpstra test, Kruskal-Wallis test, Mann-Whitney test, permutation test, several samples, simultaneous testing, union-intersection test, U-statistic.

1. Introduction

We consider nonparametric tests used in the analysis of gene or microRNA expression data sets with several treatments (groups). For each separate expression variable, the null hypothesis to be tested is that there is no difference between the distributions of the expression in different groups. To avoid strong (parametric) distributional assumptions, the alternatives are formulated using probabilities that pairs or triples of observations coming from different

groups are in a specific order of magnitude. The interesting probabilities are called probabilistic indices (PI), see also Thas, De Neve, Clement, and Ottoy (2012). The test statistics are based on natural estimates of these PIs, that is, the corresponding two and several sample U-statistics. Classical several-sample rank test statistics such as the Kruskal-Wallis or Jonckheere-Terpstra test are special cases in this approach. Also, as the number of variables (microRNAs) is typically huge and the test statistics for different variables are dependent, we face a serious simultaneous testing problem. See Fischer, Oja, Sen, Schleutker, and Wahlfors (2014) for more details.

The package **gMWT** (Fischer and Oja 2015) provides nonparametric tools for the comparison of several groups/ treatments when the number of variables is large, and is available from the Comprehensive R Archive Network (CRAN) at `http://CRAN.R-project.org/package=gMWT`. The tools are the following.

(i) Computation of the PI estimates for the group comparisons. The probabilistic indices here are (a) the probability $P_{tt'}$ that a random observation from group t is smaller than a random observation from group t', and (b) the probability $P_{tt't''}$ that observations from groups t, t', t'' appear in this same order. The tools are also given to produce the plots of variable-wise PIs.

(ii) Computation of the p values of some classical and some new nonparametric tests for the comparison of several groups/treatments. The tests are based on the use of the probabilistic indices $P_{tt'}$ and $P_{tt't''}$. Classical Mann-Whitney-Wilcoxon, Kruskal-Wallis and Jonckheere-Terpstra tests are included.

(iii) Tools for the simultaneous testing problem. As the package is meant for the analysis of gene expression data for example, tools to control the family-wise error rate and/or the false discovery rate are provided as plots for expected versus observed rejected null hypotheses with the Simes (improved Bonferroni) and Benjamini-Hochberg rejection lines. A list of rejected null hypotheses may be obtained as well.

Some standard nonparametric methods such as the Mann-Whitney and Kruskal-Wallis tests have been implemented in the R **stats** (R Core Team 2014) package. Linear rank statistics for the two and several sample location problems with ordered and unordered alternatives have been implemented also in the **coin** (Hothorn, Hornik, van de Wiel, and Zeileis 2008) package. Exact and permutation versions of the Jonckheere-Terpstra test are given by the package **clinfun** (Seshan 2014); this function is used in our package as the second option in our implementation. One contribution of our package **gMWT** is that these and several other nonparametric tests are collected with the same syntax under the same roof with a simultaneous testing possibility for several variables. The interfaces of the functions are tailored for large datasets with many groups and several variables, so that the application and comparisons of competing testing procedures are easier. With scatterplot matrices for the relevant PIs, it is also possible to illustrate and understand the joint variable-wise behavior of the standard tests.

The structure of this paper is as follows. After a brief review of the theory in Section 2 we present some practical solutions in Section 3 for the computation of the PIs and the permutational p values of the corresponding tests. In Section 4 a general description of the package **gMWT** is given with a typical workflow for its use. We also discuss the calculation of

the PIs and their scatterplot matrices, and it is described how the tests are performed. Also, the tools for the multiple testing problem are described. In Section 5, the use of the package is illustrated with a simulated data set as well as with real genotype data. In the latter case, an expression quantitative trait locus (eQTL) analysis is performed with the packages **gMWT** and **GeneticTools** (Fischer 2014).

2. Statistical inference based on probabilistic indices

2.1. Null hypothesis and alternatives based on $\mathbf{P_{tt'}}$ and $\mathbf{P_{tt't''}}$

Consider first the univariate case and the comparison of T groups. Let $x_{t1}, \ldots, x_{tN_t}$ be a random sample from a distribution with cumulative distribution function F_t, $t = 1, \ldots, T$, and let the samples be independent. The total sample size is then $N = N_1 + \cdots + N_T$. We wish to test the null hypothesis

$$H_0 : F_1 = F_2 = \cdots = F_T.$$

The interesting alternatives are formulated using certain probabilistic indices. As ties may often be present, we write

$$\mathbb{I}(x, y) = \mathbb{I}(x < y) + \frac{1}{2}\mathbb{I}(x = y)$$

and

$$\mathbb{I}(x, y, z) = \mathbb{I}(x < y < z) + \frac{1}{2}\mathbb{I}(x = y < z) + \frac{1}{2}\mathbb{I}(x < y = z) + \frac{1}{6}\mathbb{I}(x = y = z),$$

with $\mathbb{I}(\cdot)$ being the indicator function, which is 1 if the argument $(\cdot)$ is true and 0 else. The interesting alternatives are then given in terms of the probabilities

$$P_{tt'} = \mathbb{E}(\mathbb{I}(x_t, x_{t'})) \text{ and } P_{tt't''} = \mathbb{E}(\mathbb{I}(x_t, x_{t'}, x_{t''})).$$

Note that, as

$$\mathbb{I}(x, y) = \mathbb{I}(x, y, z) + \mathbb{I}(x, z, y) + \mathbb{I}(z, x, y)$$

the probabilities satisfy

$$P_{tt'} = P_{tt't''} + P_{tt''t'} + P_{t''tt'}.$$

Under the null hypothesis $H_0 : F_1 = F_2 = \cdots = F_T$, for all t, t', t'',

$$P_{tt'} = \frac{1}{2} \quad \text{and} \quad P_{tt't''} = \frac{1}{6}.$$

We say that F_1 and F_2 are stochastically ordered and write $F_1 \preceq_{st} F_2$ if $F_1(x) \geq F_2(x)\ \forall x \in \mathbb{R}$. Then

$$F_t \preceq_{st} F_{t'} \quad \Rightarrow \quad P_{tt'} \geq \frac{1}{2}$$

and

$$F_t \preceq_{st} F_{t'} \preceq_{st} F_{t''} \quad \Rightarrow \quad P_{tt't''} \geq \frac{1}{6}$$

but the converse statements are not true.

In the comparison of $T = 3$ treatments interesting alternatives might then be formulated, for example, as

$$H_1: \quad P_{12} \neq \tfrac{1}{2} \text{ or } P_{13} \neq \tfrac{1}{2} \text{ or } P_{23} \neq \tfrac{1}{2},$$

or

$$H_1: \quad P_{12} \geq \tfrac{1}{2} \text{ or } P_{13} \geq \tfrac{1}{2} \text{ or } P_{23} \geq \tfrac{1}{2} \text{ with at least one strict inequality,}$$

or

$$H_1: \quad P_{13} \geq \tfrac{1}{2} \text{ or } P_{23} \geq \tfrac{1}{2} \text{ with at least one strict inequality,}$$

or

$$H_1: \quad P_{123} > \frac{1}{6}.$$

The tests will then be based on the estimates $\hat{P}_{12}, \hat{P}_{13}, \hat{P}_{23}$ and $\hat{P}_{123}$ and should be constructed keeping the interesting alternative in mind.

2.2. Estimation of $\mathbf{P_{tt'}}$ and $\mathbf{P_{tt't''}}$

The probabilities $P_{tt'}$ and $P_{tt't''}$ are naturally estimated by corresponding U-statistics

$$\hat{P}_{tt'} = \frac{1}{N_t N_{t'}} \sum_{i=1}^{N_t} \sum_{i'=1}^{N_{t'}} \mathbb{I}(x_{ti}, x_{t'i'})$$

and

$$\hat{P}_{tt't''} = \frac{1}{N_t N_{t'} N_{t''}} \sum_{i=1}^{N_t} \sum_{i'=1}^{N_{t'}} \sum_{i''=1}^{N_{t''}} \mathbb{I}(x_{ti}, x_{t'i'}, x_{t''i''}).$$

A natural statistic for the comparison between group t and other groups is

$$\hat{P}_t = \frac{1}{N - N_t} \sum_{t' \neq t} N_{t'} \hat{P}_{tt'}.$$

A general several-sample U-statistic theory can be used to find the (joint) limiting properties of $\hat{P}_t$, $\hat{P}_{tt'}$ and $\hat{P}_{tt't''}$ under the null hypothesis. See, e.g., Chapter 5 in Serfling (1980).

2.3. Tests based on estimates $\hat{\mathbf{P}}_{\mathbf{tt'}}$ and $\hat{\mathbf{P}}_{\mathbf{tt't''}}$

As seen before, we have a hierarchy

$$\left\{\hat{P}_{tt't''}\right\} \quad \rightarrow \quad \left\{\hat{P}_{tt'}\right\} \quad \rightarrow \quad \left\{\hat{P}_t\right\}$$

and one can construct test statistics at different levels of this hierarchy. Some choices are the following.

1. Use test statistics $\hat{P}_{tt't''}$ for $H_0 : F_t = F_{t'} = F_{t''}$ vs. $H_1 : P_{tt't''} \neq \frac{1}{6}$. Of course one-sided alternatives are possible as well.

2. Use the Mann-Whitney (MW) test statistics $\hat{P}_{tt'}$ for $H_0 : F_t = F_{t'}$ vs. $H_1 : P_{tt'} \neq \frac{1}{2}$ and the Jonckheere-Terpstra (JT) test statistics for $H_0 : F_1 = \cdots = F_T$ vs. $H_1 : P_{tt'} \geq \frac{1}{2}$ for all $t < t'$ with at least one strict inequality. Note that $F_1 \preceq_{st} F_2 \preceq_{st} \cdots \preceq_{st} F_T$ with at least one strict inequality implies the latter H_1. We have two versions of JT test statistic, namely,

$$JT = \sum_{t<t'} N_t N_{t'} \hat{P}_{tt'} \quad \text{and} \quad JT^* = \sum_{t<t'} \hat{P}_{tt'}.$$

3. For a fixed group t, use a Mann-Whitney test statistic $\hat{P}_t$ for $H_0 : F_1 = \cdots = F_T$ vs. $H_1 : F_1 = ... = F_{t-1} = F_{t+1} = ... = F_T \neq F_t$. Use the Kruskal-Wallis test statistic

$$KW = \frac{12}{N(N+1)} \sum_{t=1}^{T} \frac{(\hat{P}_t - N_t(N - N_t)/2)^2}{N_t}$$

for $H_0 : F_1 = \cdots = F_T$ vs. $H_1 : F_t \neq F_{t'}$ for at least one pair t, t'. The alternative then implies that $P_{tt'} \neq \frac{1}{2}$ for at least one pair t, t'.

4. Use a union-intersection test (UIT) to compare three groups t, t', and t''. The test statistic is a combination of statistics $\hat{P}_{tt''}$ and $\hat{P}_{t't''}$ and is meant for the alternative $\max(P_{tt''}, P_{t't''}) > \frac{1}{2}$. The test statistic can be found in Appendix A, see also Fischer *et al.* (2014) for the details.

3. Computational solutions

3.1. Fast computation of $\hat{\mathbf{P}}_{\mathbf{tt'}}$ and $\hat{\mathbf{P}}_{\mathbf{tt't''}}$

Consider the univariate case and write the N-vector

$$\boldsymbol{x} = (x_1, x_2, \ldots, x_N)^\top = (x_{11}, \ldots, x_{1N_1},\ x_{21}, \ldots, x_{2N_2}, \ldots,\ x_{T1}, \ldots, x_{TN_T})^\top$$

of observations coming from all T groups. The PIs are based on two $N \times N$ matrices $\mathbb{I}^{st} = \mathbb{I}^{st}(\boldsymbol{x})$ and $\mathbb{I}^{eq} = \mathbb{I}^{eq}(\boldsymbol{x})$ with the elements

$$(\mathbb{I}^{st}(\boldsymbol{x}))_{ij} = \mathbb{I}(x_i < x_j) \quad \text{and} \quad (\mathbb{I}^{eq}(\boldsymbol{x}))_{ij} = \mathbb{I}(x_i = x_j),$$

$i, j = 1, \ldots, N$. The matrices $\mathbb{I}^{st} = \mathbb{I}^{st}(\boldsymbol{x})$ and $\mathbb{I}^{eq} = \mathbb{I}^{eq}(\boldsymbol{x})$ can then be decomposed as

$$\mathbb{I}^{st} = \begin{pmatrix} \mathbb{I}^{st}_{11} & \mathbb{I}^{st}_{12} & \ldots & \mathbb{I}^{st}_{1T} \\ \mathbb{I}^{st}_{21} & \mathbb{I}^{st}_{22} & \ldots & \mathbb{I}^{st}_{2T} \\ \ldots & \ldots & \ldots & \ldots \\ \mathbb{I}^{st}_{T1} & \mathbb{I}^{st}_{T2} & \ldots & \mathbb{I}^{st}_{TT} \end{pmatrix} \quad \text{and} \quad \mathbb{I}^{eq} = \begin{pmatrix} \mathbb{I}^{eq}_{11} & \mathbb{I}^{eq}_{12} & \ldots & \mathbb{I}^{eq}_{1T} \\ \mathbb{I}^{eq}_{21} & \mathbb{I}^{eq}_{22} & \ldots & \mathbb{I}^{eq}_{2T} \\ \ldots & \ldots & \ldots & \ldots \\ \mathbb{I}^{eq}_{T1} & \mathbb{I}^{eq}_{T2} & \ldots & \mathbb{I}^{eq}_{TT} \end{pmatrix},$$

where the $N_i \times N_j$ submatrices $\mathbb{I}^{st}_{ij}$ and $\mathbb{I}^{eq}_{ij}$ compare treatments i and j, $i, j = 1, \ldots, T$. Then

$$\hat{\boldsymbol{P}}_{tt'} = \frac{1}{N_t N_{t'}} \mathbf{1}^\top_{N_t} \left(\mathbb{I}^{st}_{tt'} + \frac{1}{2} \mathbb{I}^{eq}_{tt'} \right) \mathbf{1}_{N_{t'}},$$

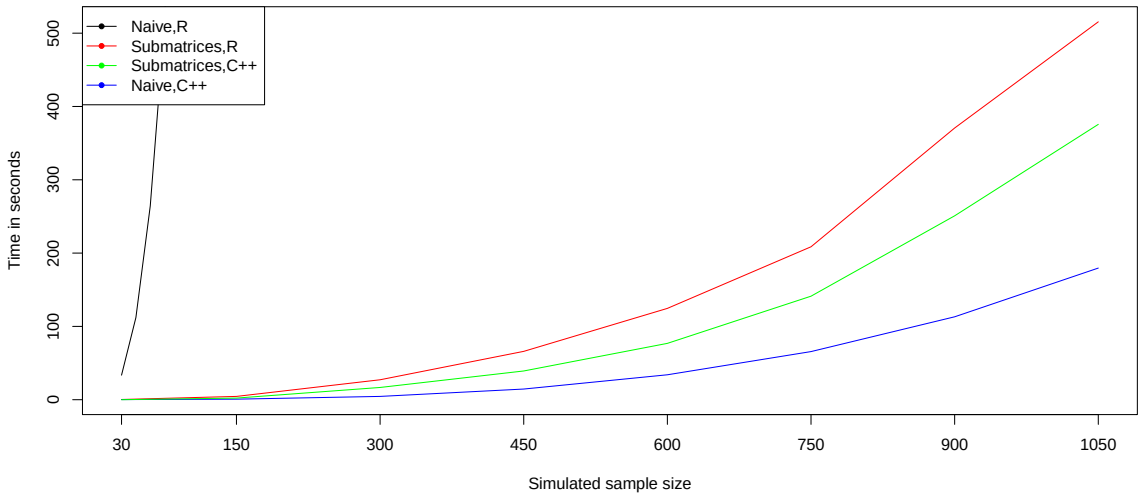

Figure 1: Computation times for the p values using the permutation test version with test statistic $\hat{\boldsymbol{P}}_{tt't''}$. Three groups with equal group sizes were used, and the number of permutations in each case was 2000. Naive and submatrix approaches implemented in R and C++ are compared.

where $\mathbf{1}_k$ is the notation for a k-vector full of ones. For the triples we get

$$\hat{\boldsymbol{P}}_{tt't''} = \frac{1}{N_t N_{t'} N_{t''}} \mathbf{1}_{N_t}^\top \left(\mathbb{I}_{tt'}^{st} \mathbb{I}_{t't''}^{st} + \frac{1}{2} \mathbb{I}_{tt'}^{eq} \mathbb{I}_{t't''}^{st} + \frac{1}{2} \mathbb{I}_{tt'}^{st} \mathbb{I}_{t't''}^{eq} + \frac{1}{6} \mathbb{I}_{tt'}^{eq} \mathbb{I}_{t't''}^{eq} \right) \mathbf{1}_{N_{t''}}.$$

In case that no ties are present, the matrix $\mathbb{I}^{eq}$ is simply a zero matrix.

Using a naive implementation, we would calculate the probabilities $\hat{\boldsymbol{P}}_{tt't''}$ one by one while going through all $N_t N_{t'} N_{t''}$ triple comparisons. In our submatrix approach we thus calculate the matrices $\mathbb{I}^{st}$ and $\mathbb{I}^{eq}$ only once and then use the submatrices to find the probabilities $\hat{\boldsymbol{P}}_{tt't''}$. This leads to improved calculation times especially for permutation versions of the tests. For the computation time comparisons in R and C++ (via **Rcpp**, Eddelbuettel and François 2011, and **RcppArmadillo**, Eddelbuettel and Sanderson 2014), see Figure 1.

3.2. Computation of p values

If the null hypothesis $H_0 : F_1 = \cdots = F_T$ is true, then $\mathbf{Px} \sim \mathbf{x}$ for all $N \times N$ permutation matrices $\mathbf{P}$. By $\mathbf{Px} \sim \mathbf{x}$ we mean that the distributions of $\mathbf{Px}$ and $\mathbf{x}$ are the same. Matrix $\mathbf{P}$ is an $N \times N$ permutation matrix if it is obtained from an identity matrix $\mathbf{I}_N$ by permuting its rows and/or columns. The number of distinct permutation matrices is $N!$.

Note that our test statistics are functions of the matrices $\mathbb{I}^{st}$ and $\mathbb{I}^{eq}$ and that

$$\mathbb{I}^{st}(\mathbf{Px}) = \mathbf{P}\mathbb{I}^{st}(\mathbf{x})\mathbf{P}^\top \quad \text{and} \quad \mathbb{I}^{eq}(\mathbf{Px}) = \mathbf{P}\mathbb{I}^{eq}(\mathbf{x})\mathbf{P}^\top.$$

The exact p value from a permutation test using a test statistic $Q = Q(\mathbb{I}^{st}, \mathbb{I}^{eq})$ is then

$$\mathbb{P}\left\{ Q(\mathbf{P}\mathbb{I}^{st}\mathbf{P}^\top, \mathbf{P}\mathbb{I}^{eq}\mathbf{P}^\top) \geq Q(\mathbb{I}^{st}, \mathbb{I}^{eq}) \right\},$$

where the probability is taken over $N!$ equally probable values of $\mathbf{P}$. The p value can in practice be estimated by

$$\frac{1}{M}\sum_{m=1}^{M} \mathbb{I}\left\{Q(\mathbf{P}_m \mathbb{I}^{st}\mathbf{P}_m^\top, \mathbf{P}_m \mathbb{I}^{eq}\mathbf{P}_m^\top) \geq Q(\mathbb{I}^{st}, \mathbb{I}^{eq})\right\},$$

where $\mathbf{P}_1, \ldots, \mathbf{P}_M$ is a random sample from a uniform distribution over the set of $N \times N$ permutation matrices. Naturally, the larger M, the better is the estimate of the exact p value.

Approximate p values may also be based on the limiting joint normality of the estimates (U-statistics) $\hat{P}_t$, $\hat{P}_{tt'}$, and $\hat{P}_{tt't''}$. If no ties are present, the test statistics based on the PIs are strictly distribution-free with limiting variances and covariances that are easily found.

4. The package gMWT

4.1. General features

The R package **gMWT** can be used to calculate the variable-wise probabilistic indices $\hat{P}_t$, $\hat{P}_{tt'}$, and $\hat{P}_{tt't''}$, to illustrate their joint distributions and dependence with scatterplot matrices, and to perform various rank tests based on $\hat{P}_t$, $\hat{P}_{tt'}$ and $\hat{P}_{tt't''}$ described in Section 2.3. See Figure 2 for possible workflows.

A practical application for the testing procedures is an eQTL analysis for a combined analysis of microarray and genotype data. In Section 5.2 we illustrate the use of the packages **GeneticTools** and **gMWT** with the directional triple test for testing for eQTL.

In the following, the input matrix `X` is a data matrix with observations as rows and variables as columns. The vector `g` indicates the group membership; its length is then the number of rows in `X`.

4.2. Computation of $\hat{\mathbf{P}}_\mathbf{t}$, $\hat{\mathbf{P}}_{\mathbf{tt'}}$ and $\hat{\mathbf{P}}_{\mathbf{tt't''}}$

The estimated PIs, $\hat{P}_t$, $\hat{P}_{tt'}$ and $\hat{P}_{tt't''}$, are calculated using the command

```
estPI(X, g, type = "pair", goi, mc = 1, order = TRUE)
```

The options `type = "single"`, `"pair"` or `"triple"` specify the PIs to be computed, that is, $\hat{P}_t$, $\hat{P}_{tt'}$ or $\hat{P}_{tt't''}$. The vector `goi` ("groups of interest") specifies the groups (values of `g`) to be used in the comparisons.

The option `order` specifies whether the PIs should be calculated for all possible pairs and triples (`order = FALSE`) or just for pairs and triples with increasing group labels. In the four group case, for example, an `estPI` call with the (default) options `(type = "pair", order = TRUE)` would calculate the estimated PIs $\hat{P}_{12}, \hat{P}_{13}, \hat{P}_{14}, \hat{P}_{23}, \hat{P}_{24}, \hat{P}_{34}$ and in case of using the parameters `(type = "triple", order = TRUE)` the estimated PIs $\hat{P}_{123}, \hat{P}_{124}, \hat{P}_{134}, \hat{P}_{234}$.

For matrix valued `X`, the option `mc` can be used to execute the parallel calculation on `mc`-many cores in order to speed up the calculation (available only for Linux systems).

The result of the function `estPI` is a list, containing a matrix `probs` with the PIs as rows and variables as columns. The other list items are the used parameters.

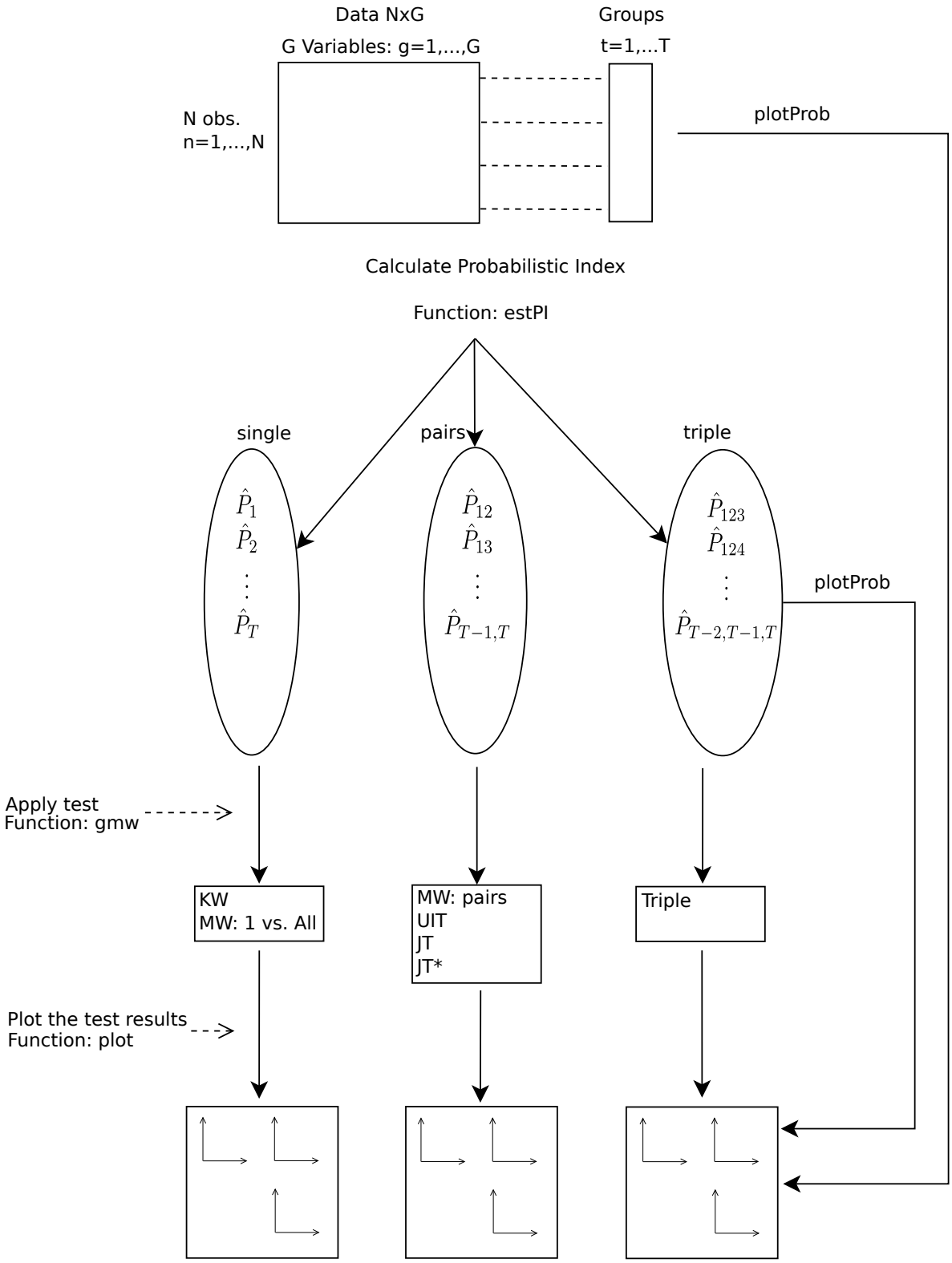

Figure 2: Possible calculation workflows.

4.3. Scatterplots for $\hat{\mathbf{P}}_{\mathbf{t}}$, $\hat{\mathbf{P}}_{\mathbf{tt'}}$ and $\hat{\mathbf{P}}_{\mathbf{tt't''}}$

Based on the probabilities calculated via `estPI` the package creates diagnostic scatterplot matrices for variable-wise PIs. The command is either

```
plotPI(X, g, col, zoom = FALSE, highlight = NULL, hlCol = "red")
```

if the diagnostic plots are produced directly from the data or

```
pe <- estPI(X, g, type = "pair", goi)
plot(pe, col, zoom = FALSE, highlight, hlCol)
```

if the probabilities are first calculated by `estPI`. The function `plotPI` naturally allows also all options used in the `estPI` call. The type of the plots depends on the selected PIs; the pairwise scatterplots are then for all possible pairs of PIs.

Additional plotting options are `highlight`, `hlCol` and `zoom`. Using the `highlight` option, indicated variables are plotted with a color specified in `hlCol`. If the Boolean flag `zoomed` is set, the plots are zoomed to the active area of the PIs. Without this flag, the bivariate plots are in $[0, 1] \times [0, 1]$.

4.4. Tests based on $\hat{\mathbf{P}}_{\mathbf{t}}$, $\hat{\mathbf{P}}_{\mathbf{tt'}}$ and $\hat{\mathbf{P}}_{\mathbf{tt't''}}$

The basic call for applying the testing procedure is

```
gmw(X, g, goi, test = "mw", type = "permutation", prob = "pair",
  nper = 2000, alternative = "two.sided", mc = 1, output = "min",
  keepPM = FALSE, mwAkw = FALSE)
```

where only input variables `X` and `g` are compulsory.

If `X` is a matrix then the chosen test is applied variable-wise to the data and the results are reported as a matrix of p values. Specifying the option `output = "full"` leads to a more detailed list output with the same length as the number of different alternatives are tested and each list item contains then again a list with as many columns as there are in `X`. Each entry in the encapsulated list is a test result of class '`htest`'.

The vector `g` gives the group numbers in a natural order. The groups used in the analysis can be specified via `goi`. If no `goi` is specified, all groups are used.

With the option `test`, the test (`"uit"`, `"triple"`, `"mw"`, `"kw"`, `"jt"`, `"jt*"`) can be specified. The option `prob` may be used only in the case of the Mann-Whitney test. For all pairwise group comparisons one uses (`prob = "pair"`) while option (`prob = "single"`) compares each group to the rest of the data. The option `type` specifies whether the permutation type test (`"permutation"`) or the asymptotical test version (`"asymptotical"`) is used. For some standard tests the procedures from base R are available, and the Jonckheere-Terpstra test is implemented in the **clinfun** package. The option `type = "external"` allows the use of these test versions. A permutation type of test is available for all tests but, as the package is still under active development, the asymptotical versions are not yet available for all cases. The number of permutations is selected with the option `nper`. Different alternatives are available whenever they are natural and are set with `alternative = "smaller"`, `"greater"` and `"two.sided"`.

If the Westfall & Young method, as proposed in Westfall and Young (1993), will be later applied for multiple testing, the permutation matrices used for the p value calculation must be stored for a later use. In that case the option `keepPM = TRUE` has to be set for the later maxT correction.

For the option `mwAkw = TRUE`, pairwise Mann-Whitney tests are performed after the global Kruskal-Wallis test for a more detailed analysis of the differences between the groups. Please notice that, to keep the overall significance level, the second step is allowed only after the rejection by the Kruskal-Wallis test.

Again, the additional option `mc` may be used to speed up the computation for a large number of variables if one works on a multiple core computer with Linux operating system. The amount of cores can be specified with the `mc` option. This option decreases the calculation time also on normal desktop computers drastically, since modern desktop computers usually have more than one core. However, using all available cores in the calculations can make the computer for the calculation time unusable for other tasks. Hence, for longer calculations we recommend to choose here `mc = detectCores() - 1` if other tasks are performed simultaneously on the same computer.

The `full` test output itself is a '`htest`' R object and has the standard output showing the used data and the grouping object:

```
R> library("gMWT")
R> set.seed(123456)
R> myData <- c(rnorm(50), rnorm(60), rnorm(40, 0.5, 1))
R> myGroups <- c(rep(1, 50), rep(2, 60), rep(3, 40))
R> gmw(myData, myGroups, test = "uit", type = "permutation",
+    alternative = "greater", output = "full")


$`H1: Max(P13,P23) > 0.5`

        ********* Union-Intersection Test *********

data:  Data:X, Groups:g, Order: max(P13,P23)
obs.value = 2.8081, p-value = 0.077
alternative hypothesis: greater
```

The attribute `obs.value` contains the value of the test statistic and the `p.value` contains the p value based on the selected test version. The minimal standard output looks like

```
R> gmw(myData, myGroups, test = "uit", type = "permutation",
+    alternative = "greater")


                        pValues
H1: Max(P13,P23) > 0.5    0.077
```

4.5. Multiple testing problem

As the testing procedures are applied simultaneously for a large number of dependent variables, we often face a severe multiple testing problem. Several attempts can be found in the literature to adjust the p values and/or the test levels so that the inference remains valid or the number of wrong decisions is as small as possible. The standard Bonferroni method and its improved version by Simes (1986) control the family-wise error rate (FWER). Benjamini and Hochberg (1995) suggested to control the false discovery rate (FDR) rather than the FWER. Their procedure, known as the Benjamini-Hochberg procedure, also controls the FDR at the same level and leads to the same practical rejection rule as the improved Bonferroni procedure. It can be shown that the procedures keep their control levels also under mild dependencies between the marginal test statistics. See, e.g., Fischer *et al.* (2014). For permutation type tests, a multiple testing procedure proposed by Westfall and Young controls the FWER and takes the dependence structure of the p values into account, see Westfall and Young (1993).

The package **gMWT** allows a visualization of the p values as a plot of expected versus observed proportions of rejected null hypotheses (cumulative distribution of the observed p-values) by

```
rejectionPlot(testResult, rejLine = "bh", alpha = 0.1, crit = NULL,
  xlim = c(0, 0.1))
```

The input `testResult` is a vector of variable-wise p values for a selected test statistic. It is also possible to pass a matrix `testResult` to the function. In that case each row is handled as an own test result and curves from different tests are shown in the same figure. The option `col` defines the colors for these curves. A solid line is the expected line under the null hypothesis. The FWER and FDR controlling lines are given by the option `rejLine`. Possible parameters are then `"bonferroni"`, `"bh"` and `"simes"`. The control level can be set with the option `alpha`. For the use of these rejection lines, see Fischer *et al.* (2014). For some examples, see Figures 6 and 7 in Section 5.1. Finally, the function `getSigTests` extracts the variables with rejected null hypotheses at a given FWER or FDR control level α. The function call `getSigTests(X, alpha = 0.05, method)` then provides a vector for the positions of these variables. Possible options for `method` are `"plain"`, `"bonferroni"`, `"bh"`, `"simes"`, `"maxT"`, `"csR"` and `"csD"`. Please keep in mind that in order to use the option `"maxT"` the permutation matrix of the test has to be available and that for this the option `keepPM = TRUE` has to be set when calling `gmw`.

5. Examples

5.1. Simulated data

We illustrate the use of the **gMWT** package first with a simulated data set. We created a dataset with 500 variables and 3 groups with sample sizes $N_1 = N_2 = N_3 = 50$. The 150×500 data matrix was obtained as follows.

1. Let z_{ij}, $i, j = 0, \pm 1, \pm 2, \ldots$ be independently identically distributed (iid) from $N(0, 1)$.

2. Let

$$x_{ij} = \sum_{k=-10}^{10} \psi_k z_{i,j+k}, \quad i = 1, \ldots, 150;\ j = 1, \ldots, 500,$$

where

$$\psi_k = 11 - |k|, \quad k = 0, \pm 1, \ldots, \pm 10.$$

This step thus makes the columns of the matrix $\mathbf{X} = (x_{ij})$ dependent while the rows are iid.

3. The final data matrix is then

$$\mathbf{X} + \begin{pmatrix} 0 \cdot \mathbf{1}_{50} \\ (1/3) \cdot \mathbf{1}_{50} \\ (2/3) \cdot \mathbf{1}_{50} \end{pmatrix} \mathbf{s}^\top$$

where $\mathbf{s}$ is a random vector with 25 ones and 475 zeros. Thus the null hypothesis is not true for 25 variables indicated by $\mathbf{s}$.

For our illustration, we first calculate the variable-wise PIs, $\hat{P}_t$, $\hat{P}_{tt'}$ and $\hat{P}_{tt't''}$ by

```
R> ep1 <- estPI(X, groups, type = "single")
R> ep2 <- estPI(X, groups)
R> ep3 <- estPI(X, groups, type = "triple", order = FALSE)
```

In the data set the 25 shifted variables are indicated by `pickGenes`. The scatterplot matrices in Figures 3, 4 and 5 are then obtained and the shifted observations highlighted as follows. As the same observations are used repeatedly for different PIs, the PIs are dependent as can easily be seen in the figures.

```
R> plot(ep1, highlight = pickGenes, zoom = TRUE)
R> plot(ep2, highlight = pickGenes, zoom = TRUE)
R> plot(ep3, highlight = pickGenes, zoom = TRUE)
```

Next, we simultaneously perform several tests for all the 500 variables.

```
R> kw.results <- gmw(X, groups, test = "kw")
R> jt.results <- gmw(X, groups, test = "jt")
R> uit.results <- gmw(X, groups, test = "uit", alternative = "greater")
R> triple.results <- gmw(X, groups, test = "triple", alternative = "greater")
```

The results from the testing procedures are summarized in rejection plots, see Figures 6 and 7. In Figure 7 a Benjamini-Hochberg rejection line at the FDR control level $\alpha = 0.1$ is provided as well. The null hypotheses with p values smaller than the (highest) crossing point of the straight rejection line and the curve are then rejected and at most 10% of the rejected hypotheses are then in fact true.

The figures were created by

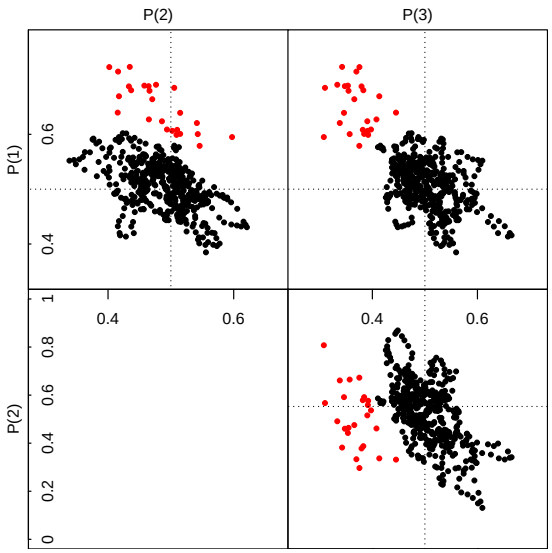

Figure 3: Scatterplot matrix for `ep1`.

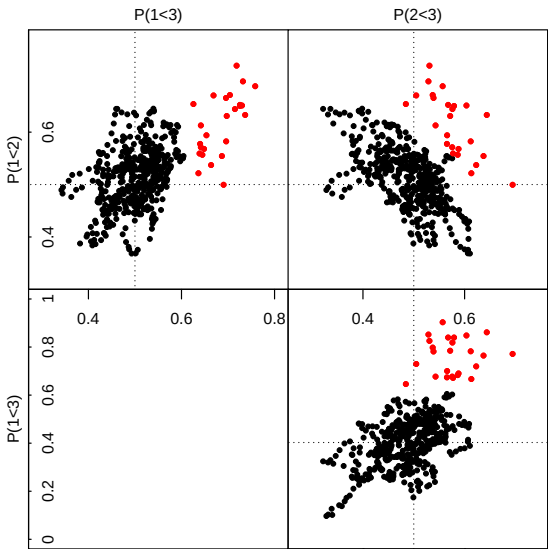

Figure 4: Scatterplot matrix for `ep2`.

```
R> tests <- rbind(kw.results$p.values, jt.results$p.values,
+    uit.results$p.values, triple.results$p.values)
R> rejectionPlot(tests, lCol = c("green", "red", "blue", "cyan"))
R> rejectionPlot(tests, lCol = c("green", "red", "blue", "cyan"),
+    xlim = c(0, 0.2), rejLine = "bh", alpha = 0.1)
```

The variables with rejected null hypotheses by a Kruskal-Wallis test at the FDR control level 0.10, for example, are obtained by

```
R> getSigTests(kw.results, method = "bh", alpha = 0.1)
```

For highlighting the results from one selected test, one can for example use the command

```
R> plot(ep3, highlight = which(triple.results < 0.01)))
```

This illustration shows that computing and looking at the scatterplots of the PIs may be a useful tool to understand the dependencies between marginal PIs and the corresponding test statistics as well as to detect hidden structures in the data. The rejection plot is a useful tool in the analysis of large datasets and in the multiple testing problem.

5.2. Application example: gMWT and eQTL analysis

The testing procedures implemented in the package **gMWT** can be used in an eQTL analysis. In the following, we combine the gene expression data with the genotype data in order to find important single nucleotide polymorphisms (SNPs) that are associated or influential to the expression level of certain genes. As a first step, using the gene expression data, we identify the genes which are differentially expressed between cases and controls. Interesting genes are for example identified with the Wilcoxon tests (two groups) or with a Kruskal-Wallis test (several groups). After the identification of an interesting gene, we consider all

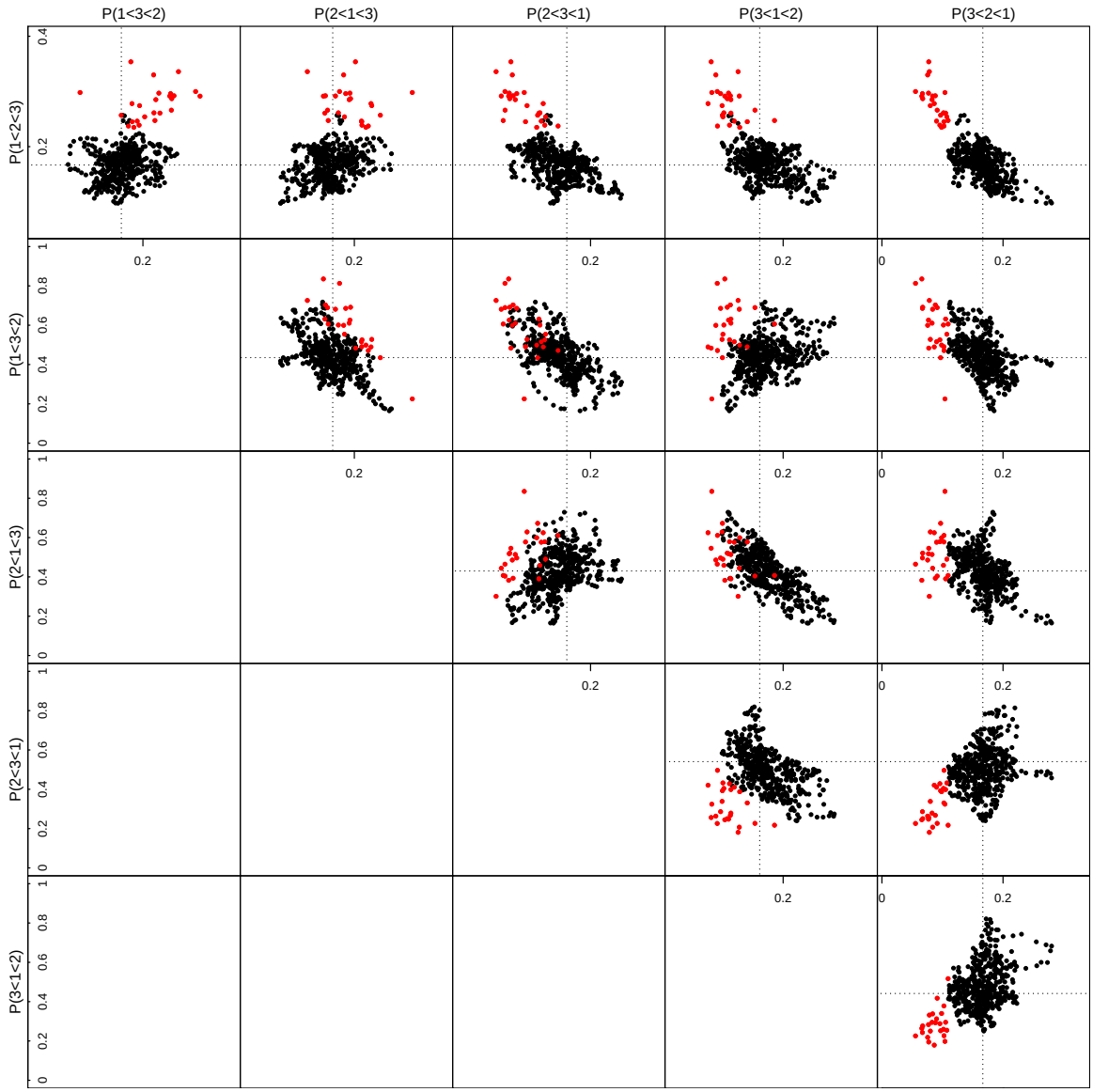

Figure 5: Scatterplot matrix for `ep3`.

SNPs located in its w-neighborhood (common values for w are 0.5MB or 1MB, where MB stands for megabases). The genotype information of these neighboring SNPs are then used to explain the expression of the respective center gene. For the SNPs, there are three different genotypes, the wild-type allele coded as 1 or (AA), the heterozygeous mutation 2 (AB) and the homozygeous mutation 3 (BB). Common platforms measure genotypes for 700k up to several million SNPs. See Figure 8 for a sketch on how gene expression values and genotype information are linked.

A popular but unrealistic method to consider SNP-wise associations between genotype and gene expression is to fit a linear model. We use the triple test with the PIs $\hat{P}_{123}$ and $\hat{P}_{321}$ (increasing or decreasing trend) to avoid strong assumptions of linear dependence. If p_1 and

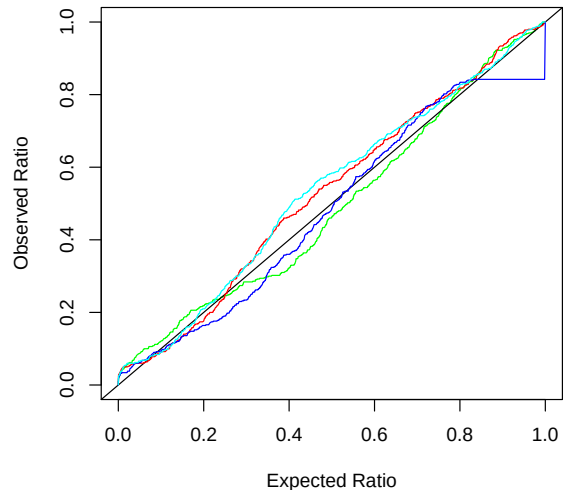

Figure 6: Rejection plot for $\alpha \in [0, 1]$.

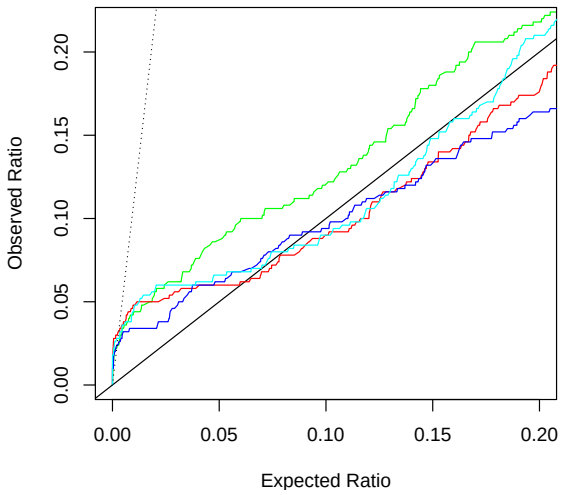

Figure 7: Rejection plot for $\alpha \in [0, 0.2]$.

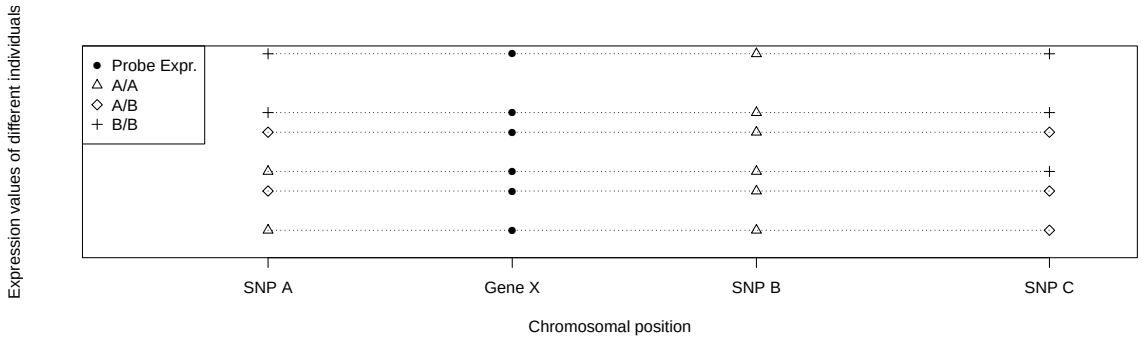

Figure 8: Sketch on how genotype information and gene expression are linked.

p_2 are the two p values from the corresponding directional tests, a two-sided p value for a SNP-wise test is obtained as $\min(2 \cdot \min(p_1, p_2), 1)$ and it is implemented in our other R package **GeneticTools**. The linear model approach is also an option there.

In order to perform an eQTL analysis with **GeneticTools** the genotype data has to be present in ped/map file format, which is the standard output format of many programs. In addition to that a gene expression matrix is required. The basic call to perform an eQTL is then

```
R> library("GeneticTools")
R> setwd(file.path("Data", "Genotypes"))
R> myEQTL <- eQTL(gex = geneEx, geno = "example", xAnnot = geneAnnotations,
+   windowSize = 0.5, method = "directional")
```

We assume here, that the ped/map file pair is stored in the Folder `file.paht("Data", "Genotypes")` and is called `example.ped` and `example.map`. The file name is given to the eQTL function with the `geno` option. In case the SNP data has been imported already previously using the package **snpStats** (Clayton 2014) and its function `read.pedfile()` this object can be passed to the `geno` option instead of a file name. The gene expressions are stored in the matrix `geneEx` and each column refers to a gene and each row is an individual.

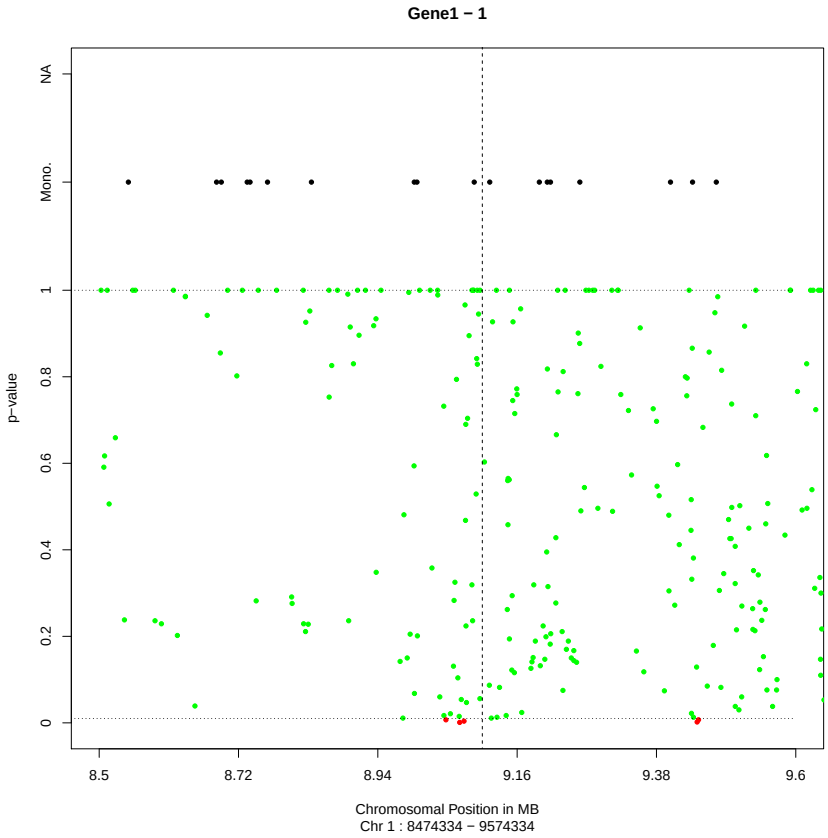

Figure 9: eQTL-plot. The dashed line symbolizes the location of the linked gene and each dot represents the test result for a SNP test. The dotted line refers to the chosen significance level. In case that all individuals have the same genotype no test is performed and this is marked as "Mono.". If no genotype information was available for a SNP, this is marked as NA.

The expression matrix is specified with the `gex` option.

The option `xAnnot` is important for the gene annotations and is a `list` where each list item refers to a gene from the columns of `geneEx`. The names have to match and the eQTL is only performed for matching pairs. In case that no annotation `xAnnot` is given to the function no window is used and all combinations are considered instead. Be aware that this might lead to a very long lasting calculation. Each list item in `geneAnnotations` is a matrix like

```
R> geneAnnotations

$Gene1
  Chr   Start     End
1   1 8974334 9074334

$Gene2
  Chr     Start       End
1  12 135633062 135738062
2  12 135735062 135838062
```

This takes into account that certain probes have multiple locations in the genome and our method will test for all those locations. In case that the labeling of individuals between the expression data and the genotype data differs, there is an option to give a new vector of labels, called `genoSamples`. Here can new labels for the individuals in the genotype data be specified, that match with the row names of `geneEx`.

It is important to note that the order of the rows and columns of all lists and matrices does not have to match. The function takes the smallest subsets of individuals and genes and takes then those SNPs, which are in a window around that gene. The window size can be specified with the option `windowSize` using the unit megabases (MB).

After the eQTL is performed we can visualize the results with

```
R> plot(myEQTL, which, file, sig)
```

The `which` option specifies for which genes from `geneEx` the plots shall be created. If no option is given, then all plots are created. Because this might lead to a vast number of pictures, there is also an option `file` to specify a file name such that the plots are saved in a file with this name. The output of a single plot can be seen in Figure 9 with a chosen significance level `sig = 0.1`.

This way we can see for interesting genes the behavior of the surrounding SNPs onto the gene expression. For small datasets it is also possible to check visually for interesting genes. For larger gene sets the function `extractEQTL` can be used to determine a set of interesting genes.

We applied this function on real data e.g., in Siltanen *et al.* (2013).

References

Benjamini Y, Hochberg Y (1995). "Controlling the False Discovery Rate: A Practical and Powerful Approach to Multiple Testing." *Journal of the Royal Statistical Society B*, **57**(1), 289–300.

Clayton D (2014). ***snpStats**: `SnpMatrix` and `XSnpMatrix` Classes and Methods*. R package version 1.16.0, URL http://www-gene.cimr.cam.ac.uk/clayton/.

Eddelbuettel D, François R (2011). "**Rcpp**: Seamless R and C++ Integration." *Journal of Statistical Software*, **40**(8), 1–18. URL http://www.jstatsoft.org/v40/i08/.

Eddelbuettel D, Sanderson C (2014). "**RcppArmadillo**: Accelerating R with High-Performance C++ Linear Algebra." *Computational Statistics & Data Analysis*, **71**, 1054–1063.

Fischer D (2014). ***GeneticTools**: Collection of Genetic Data Analysis Tools*. R package version 0.3, URL http://CRAN.R-project.org/package=GeneticTools.

Fischer D, Oja H (2015). ***gMWT**: Generalized Mann-Whitney Type Tests*. R package version 1.0, URL http://CRAN.R-project.org/package=gMWT.

Fischer D, Oja H, Sen PK, Schleutker J, Wahlfors T (2014). "Generalized Mann-Whitney Type Tests for Microarray Experiments." *Scandinavian Journal of Statistics*, **41**(3), 672–692.

Hothorn T, Hornik K, van de Wiel MA, Zeileis A (2008). "Implementing a Class of Permutation Tests: The **coin** Package." *Journal of Statistical Software*, **28**(8), 1–23. URL http://www.jstatsoft.org/v28/i08/.

Perlman MD (1969). "One-Sided Testing Problems in Multivariate Analysis." *The Annals of Mathematical Statistics*, **40**(2), 549–567.

R Core Team (2014). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. URL http://www.R-project.org/.

Serfling RJ (1980). *Approximation Theorems of Mathematical Statistics*. John Wiley & Sons.

Seshan VE (2014). ***clinfun**: Clinical Trial Design and Data Analysis Functions*. R package version 1.0.6, URL http://CRAN.R-project.org/package=clinfun.

Siltanen S, Fischer D, Rantapero T, Laitinen V, Mpindi JP, Kallioniemi O, Wahlfors T, Schleutker J (2013). "ARLTS1 and Prostate Cancer Risk – Analysis of Expression and Regulation." *PLoS ONE*, **8**(8), e72040.

Simes RJ (1986). "An Improved Bonferroni Procedure for Multiple Tests of Significance." *Biometrika*, **73**(3), 751–754.

Thas O, De Neve J, Clement L, Ottoy JP (2012). "Probabilistic Index Models." *Journal of the Royal Statistical Society B*, **74**(4), 623–671.

Westfall PH, Young SS (1993). *Resampling-Based Multiple Testing: Examples and Methods for p-Value Adjustment*. John Wiley & Sons.

A. Union-intersection test for P_{13} and P_{23}

We implemented two ways to calculate p values from the UIT, the first one is based on a permutation approach and the other one is based on asymptotical results. In both cases we first need to calculate the critical value c of the test statistic Q^*. For the test statistics $S_1 = \hat{P}_{13}$ and $S_2 = \hat{P}_{23}$, $\boldsymbol{S} = (S_1, S_2)^\top \sim N_2(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ approximately with

$$\boldsymbol{\Sigma} = \begin{pmatrix} 1 & \rho \\ \rho & 1 \end{pmatrix}.$$

Our UIT test statistic Q^* is then

$$Q^* = \mathbb{I}_0 \cdot 0 + \mathbb{I}_1 \cdot \frac{(S_2 - \rho S_1)^2}{1-\rho^2} + \mathbb{I}_2 \cdot \frac{(S_1 - \rho S_2)^2}{1-\rho^2} + \mathbb{I}_3 \cdot \boldsymbol{S}^\top \boldsymbol{\Sigma}^{-1} \boldsymbol{S},$$

with

$$\mathbb{I}_0 = \mathbb{I}(S_1 \leq \rho S_2, S_2 \leq \rho S_1), \qquad \mathbb{I}_1 = \mathbb{I}(S_1 < 0, S_2 > \rho S_1),$$
$$\mathbb{I}_2 = \mathbb{I}(S_1 > \rho S_2, S_2 < 0), \qquad \mathbb{I}_3 = \mathbb{I}(S_1 \geq 0, S_2 \geq 0).$$

As $\mathbb{I}_0 + \mathbb{I}_1 + \mathbb{I}_2 + \mathbb{I}_3 = 1$, at most one of the three test statistics in the sum contributes to Q^*. An approximate p value is then given by the approximation (Perlman 1969)

$$\mathbb{P}(Q^* > c) = \frac{1}{2}\mathbb{P}(\chi_1^2 > c) + \frac{\cos^{-1}\rho}{2\pi}\mathbb{P}(\chi_2^2 > c).$$

For a permutation test version, we permute the elements of the vector of the group variable M times with resulting values of test statistics $Q_1^*, Q_2^*, \ldots, Q_M^*$; the approximate p value is then

$$p = \frac{1}{M}\sum_{m=1}^{M} \mathbb{I}(Q_m^* \geq Q^*).$$

Affiliation:

Daniel Fischer
School of Health Sciences
University of Tampere
33014 Tampere, Finland
E-mail: Daniel.Fischer@luke.fi

Hannu Oja
Department of Mathematics and Statistics
University of Turku
20014 Turku, Finland

Journal of Statistical Software http://www.jstatsoft.org/
published by the American Statistical Association http://www.amstat.org/
Volume 65, Issue 9 *Submitted:* 2012-05-30
June 2015 *Accepted:* 2014-08-06