

# Package ‘Het.Tree’

March 25, 2014

**Title** Het.Tree

**Version** 0.0.1

**Date** 2014-03-25

**Author** Kai Yu

**Description** A robust association test for detecting genetic variants with heterogeneous effects

**Maintainer** Bill Wheeler <wheelerb@imsweb.com>

**Depends** rpart, VGAM

**License** GPL-2

## R topics documented:

|          |   |
|----------|---|
| Het.Tree | 1 |
| het.tree | 2 |

|              |          |
|--------------|----------|
| <b>Index</b> | <b>5</b> |
|--------------|----------|

---

|          |                 |
|----------|-----------------|
| Het.Tree | <i>Het-Tree</i> |
|----------|-----------------|

---

## Description

A robust association test for detecting genetic variants with heterogeneous effects

## Details

One common strategy for detecting disease-associated genetic markers is to compare the genotype distributions between cases and controls, where cases have been diagnosed as having the disease condition. In a study of a complex disease with a heterogeneous etiology, the sampled case group most likely consists of people having different disease subtypes. If we conduct an association test by treating all cases as a single group, we maximize our chance of finding genetic risk factors with a homogeneous effect, regardless of the underlying disease etiology. However, this strategy might diminish the power for detecting risk factors whose effect size varies by disease subtype. We propose a robust statistical procedure to identify genetic risk factors that have either a uniform effect for all disease subtypes or heterogeneous effects across different subtypes, in situations where the subtypes are not predefined but can be characterized roughly by a set of clinical and/or pathologic markers.

The main function is [het.tree](#)

**Author(s)**

Kai Yu <yuka@mail.nih.gov>

**References**

Yu K, Zhang H, Wheeler W, Home HN, Chen J, Figueroa JD. A robust association test for detecting genetic variants with heterogeneous effects (submitted)

---

|          |                 |
|----------|-----------------|
| het.tree | <i>het.tree</i> |
|----------|-----------------|

---

**Description**

A robust association test for detecting genetic variants with heterogeneous effects

**Usage**

```
het.tree(data, cc.var, T.vars, X.vars, Z.vars, G.vars, wts, run.options,
         Robj.file, myseed)
```

**Arguments**

|             |   |
|-------------|---|
| data        | A data frame containing all data including an intercept column.         |
| cc.var      | The name of the case-control variable coded as 0-1 (0=control, 1=case). |
| T.vars      | A character vector of variables used to build the tree.                 |
| X.vars      | A character vector of variables with homogeneous effects.               |
| Z.vars      | A character vector of variables with heterogeneous effects.             |
| G.vars      | A character vector of genetic variables coded as 0-1-2.                 |
| wts         | A numeric vector of weights.  |
| run.options | List of options (see details).  |
| Robj.file   | Output file created with the <a href="#">save</a> function.             |
| myseed      | Random seed.  |

**Details****run.options list:**

Below are the names for the options list run.options. All must be specified.

- cart.option List of options for [rpart](#). See [rpart.control](#).
- converge.cut Numeric value
- num.treemodel Integer
- max.iter Integer
- num.perm Integer
- method.tree.select Character string "top"
- permute Character string "restrict.xz"
- test.method 1 or 2. 1=LLR test, 2=Wald test

## Value

The returned object is NULL. The output is saved in the R object file Robj.file.

## References

Yu K, Zhang H, Wheeler W, Home HN, Chen J, Figueroa JD. A robust association test for detecting genetic variants with heterogeneous effects (submitted)

## Examples

```
myseed      <- 1
inputfile   <- system.file("sampleData", "data_1", package="Het.Tree")
outputfile  <- paste(getwd(), "/out.rda", sep="")

data        <- read.table(inputfile, header=1)
cc.var      <- "y.all"
T.vars      <- c("t1", "t2", "t3", "t4", "t5")

# The intercept is first (for X.vars), also, only the same intercept variable
# should be used
X.vars      <- c("int.x")
Z.vars      <- c("int.x", "x.all")
G.vars      <- c("g.all")

wts         <- rep(1, nrow(data))
test.method <- 2
xval        <- 0
minsplit    <- 50
maxdepth    <- 5
maxcompete  <- 0
maxsurrogate <- 0
usesurrogate <- 2
minbucket   <- 20
cart.option <- rpart.control(xval=xval, minbucket=minbucket,
                             minsplit=minsplit, maxdepth=maxdepth, maxcompete=maxcompete,
                             maxsurrogate=maxsurrogate, usesurrogate=usesurrogate)
converge.cut <- 0.001
num.treemodel <- 5
max.iter     <- 0
num.perm     <- 50
permute.method <- "restrict.xz"
method.tree.select <- "top"

run.options <- list(converge.cut=converge.cut,
                    num.treemodel=num.treemodel,
                    max.iter=max.iter,
                    method.tree.select=method.tree.select,
                    permute = permute.method,
                    test.method = test.method,
                    num.perm=num.perm, cart.option=cart.option)

# Main function
het.tree(data, cc.var, T.vars, X.vars, Z.vars, G.vars,
          wts, run.options, outputfile, myseed)
```

```
load(outputfile)

## the test result
obj$test.result

## the full tree
obj[[1]][[2]]
```

# Index

\*Topic **model**

het.tree, [2](#)

\*Topic **package**

Het.Tree, [1](#)

Het.Tree, [1](#)

het.tree, [1](#), [2](#)

rpart, [2](#)

rpart.control, [2](#)

save, [2](#)