Package: Oncotree (via r-universe)

August 23, 2024

Type Package

Title Estimating Oncogenetic Trees

Version 0.3.5

Date 2023-09-26

Author Aniko Szabo, Lisa Pappas

Maintainer Aniko Szabo <aszabo@mcw.edu>

Depends R (>= 2.3.1), boot

Suggests lattice

Description Construct and evaluate directed tree structures that model the process of occurrence of genetic alterations during carcinogenesis as described in Szabo, A. and Boucher, K (2002) <doi:10.1016/S0025-5564(02)00086-X>.

License GPL (>=2)

URL https://github.com/anikoszabo/Oncotree

BugReports https://github.com/anikoszabo/Oncotree/issues

Repository https://anikoszabo.r-universe.dev

RemoteUrl https://github.com/anikoszabo/oncotree

RemoteRef HEAD

RemoteSha 156fd9e2eb5d163b3c5062477cd8b1971d931df8

Contents

`Oncotree-package` *Constructing and evaluating oncogenetic trees*

Description

Oncogenetic trees are directed tree structures that model the process of occurrence of genetic alterations during carcinogenesis.

Details

A **pure oncogenetic tree** is a directed rooted tree *T* with a probability $\pi(e)$ attached to each edge *e* such that for every vertex there is a unique directed path from the root to it along the edges of the tree. This tree generates observations on the presence/absence of genetic events the following way: each edge *e* is independently retained with probability $\pi(e)$; the set of vertices that are still reachable from the root gives the set of the observed genetic events.

To describe random deviations from the pure tree model an error model is added.

Error model

1. The tumor develops according to the pure oncogenetic tree model
2. The presence/absence of each alteration is independently measured
3. If the alteration is present it is not observed with probability ϵ_-.
   If the alteration is absent it is observed with probability ϵ_+.

Author(s)

Lisa Pappas, Aniko Szabo

Maintainer: Aniko Szabo <aszabo@mcw.edu>

References

[1] Desper R., Jiang F., Kallioniemi O.P., Moch H., Papadimitriou C.H., and Sch\"affer A.A. (1999) Inferring tree models for oncogenesis from comparative genome hybridization data. *Journal of Computational Biology*. **6**m 37–51. [2] Szabo, A. and Boucher, K. (2002) Estimating an oncogenetic tree when false negative and positives are present. *Mathematical Biosciences*, **176/**2, 219–236.

Examples

```
data(ov.cgh)
ov.tree <- oncotree.fit(ov.cgh)
plot(ov.tree, edge.weights="estimated")
```

ancestors *Find ancestors within an oncogenetic tree.*

Description

ancestors finds all the ancestors of the given vertex within the tree starting from itself up to the root. least.common.ancestor finds the common ancestor of two vertices that is closest to them (and farthest from the root).

Usage

```
ancestors(otree, vertex)
least.common.ancestor(otree, v1, v2)
```

Arguments

otree	An object of class oncotree.
vertex, v1, v2	Character values giving the names of the nodes.

Value

For ancestors: a character vector giving the names of the ancestors of vertex. The first element is vertex, and the last one is "Root".

For least.common.ancestor: a character value with the name of the least common ancestor of v1 and v2.

See Also

oncotree.fit

Examples

```
data(ov.cgh)
ov.tree <- oncotree.fit(ov.cgh)
ancestors(ov.tree, "4q-")
ancestors(ov.tree, "Xp-")
least.common.ancestor(ov.tree, "4q-","Xp-")  #"5q-"
```

bootstrap *Bootstrap an oncogenetic tree to assess stability*

Description

bootstrap.oncotree provides a set of resampling based estimates of the oncogenetic tree. Both a parametric and non-parametric approach is available. The print and plot methods provide interfaces for printing a summary and plotting the resulting set of trees.

Usage

```
   bootstrap.oncotree(otree, R, type = c("nonparametric", "parametric"))
   ## S3 method for class 'boottree'
print(x, ...)
   ## S3 method for class 'boottree'
plot(x, minfreq=NULL, minprop=NULL, nboots=NULL, draw.orig=TRUE,
                          draw.consensus=TRUE, fix.nodes=FALSE,
                      ask=(prod(par("mfrow"))<ntrees)&&dev.interactive(), ...)
```

Arguments

otree	An object of class oncotree.
R	The number of bootstrap replicates.
type	The type of bootstrap - see Details for explanations.
x	An object of class boottree - the output of bootstrap.oncotree
minfreq	A lower limit on the occurrence frequency of the tree in "boottree" for plotting. By default, all unique trees are plotted, which can lead to a large number of plots.
minprop	A lower limit on the occurrence proportion of the tree in "boottree" for plotting.
nboots	A lower limit on the number of bootstrapped trees plotted.
draw.orig	logical; if TRUE the original tree is plotted.
draw.consensus	logical; if TRUE the consensus tree is plotted (see Details).
fix.nodes	logical; if TRUE, the nodes for all trees are kept in the same position. If node.coords is passed as an argument to plot.oncotree, then those coordinates are used for all trees, otherwise the coordinates computed for the original tree are used.
ask	logical; if TRUE, the user is *ask*ed before each plot, see par(ask=.).
...	Ignored for print. Passed to plot.oncotree for the plot method.

Details

Parametric bootstrap: This approach assumes that the model is correct. Based on `otree`, a random data set is generated R times using `generate.data`. An oncogenetic tree is fitted to each of these random data sets.

Non-parametric bootstrap: The samples (rows) from the data associated with the tree are resampled with replacement R times, each time obtaining a data set with the same sample size. An oncogenetic tree is fitted to each of these resampled data sets.

For both approaches, a *consensus tree* that assigns to each vertex the parent that occurs most frequently in the bootstrapped trees, is also computed.

Value

For `bootstrap.oncotree`: an object of class `boottree` with the following components:

`original`	The `parent` component of the original tree (`otree`).
`consensus`	A numeric vector with the `parent$parent.num` component of the consensus tree - this defines the tree structure uniquely.
`parent.freq`	A matrix giving the number of trees with each possible child-parent edge. The rows correspond to children while the column to parents.
`tree.list`	A data frame with each row representing a unique tree obtained during the bootstrap. The 'Tree' variable contains the `parent$parent.num` component of the tree (each pasted into one dot-separated string), while the 'Freq' variable gives the frequency of the tree among the R bootstrap replicates.
`type`	A character value with the type of the bootstrap performed.

For `print.boottree`: the original object is returned invisibly. It prints a summary showing the number of replicates, the number of unique trees found, and the number of times that the original tree was obtained.

For `plot.oncotree`: nothing is returned. It is used for its side effect of producing a sequence of plots of the bootstrapped trees. Specifically, it plots the original tree (if `draw.orig=TRUE`), the consensus tree (if `draw.consensus=TRUE`), and then the other trees by frequency of occurrence. To limit the number of bootstrapped trees plotted, specify exactly one of `minfreq`, `minprop` or `nboots`. By default, if the session is interactive, the user is asked for confirmation before each new tree is drawn. To avoid this, either use `ask=FALSE` in the function call, or set up a layout that fits all the trees.

Author(s)

Lisa Pappas, Aniko Szabo

See Also

`oncotree.fit`

Examples

```
data(ov.cgh)
ov.tree <- oncotree.fit(ov.cgh[1:5])
set.seed(43636)
ov.b1 <- bootstrap.oncotree(ov.tree, R=100, type="parametric")
ov.b1
opar <- par(mfrow=c(3,2), mar=c(2,0,0,0))
plot(ov.b1, nboots=4)
plot(ov.b1, nboots=4, fix.nodes=TRUE)
par(opar)
```

---

distribution.oncotree *Find the event distribution defined by an oncogenetic tree*

Description

`distribution.oncotree` calculates the joint distribution of the events defined by the tree, while `marginal.distr` calculates the marginal probability of occurrence of each event.

Usage

```
distribution.oncotree(otree, with.probs = TRUE, with.errors=FALSE,
       edge.weights=if (with.errors) "estimated" else "observed")
marginal.distr(otree, with.errors = TRUE,
       edge.weights=if (with.errors) "estimated" else "observed")
```

Arguments

`otree`	An object of class `oncotree`.
`with.probs`	A logical value specifying if only the set of possible outcomes should be returned (if TRUE), or the associated probabilities of occurrence as well.
`with.errors`	A logical value specifying whether false positive and negative error rates should be incorporated into the distribution.
`edge.weights`	A choice of whether the observed or estimated edge transition probabilities should be used in the calculation of probabilities. See `oncotree.fit` for explanation of the difference. By default, estimated edge transition probabilities if `with.errors=TRUE` and the observed ones if `with.errors=FALSE`.

Value

For `distribution.oncotree`: a data frame each row of which gives a possible outcome.

For `marginal.distr`: a named numeric vector - the names are the event names (+ 'Root') and the values are the corresponding marginal probability of occurrence.

Author(s)

Aniko Szabo

See Also

oncotree.fit

Examples

```
data(ov.cgh)
ov.tree <- oncotree.fit(ov.cgh[1:5])

#joint distribution
jj <- distribution.oncotree(ov.tree, edge.weights="obs")
head(jj)
# including errors - time/size exponential in number of events
jj.eps <- distribution.oncotree(ov.tree, with.errors=TRUE)
head(jj.eps)

#marginal distribution
marginal.distr(ov.tree, with.error=FALSE)
#marginal distribution calculated from the joint
apply(jj[1:ov.tree$nmut], 2, function(x){sum(x*jj$Prob)})

##Same with errors incorporated
#marginal distribution
marginal.distr(ov.tree, with.error=TRUE)
#marginal distribution calculated from the joint
apply(jj.eps[1:ov.tree$nmut], 2, function(x){sum(x*jj.eps$Prob)})
```

---

error.rates<- *Set the error rates of an oncotree manually*

Description

Allows to set the false positive and false negative error rate associated with an object of class oncotree to values other than those found by the optimization in oncotree.fit. The estimated edge transition probabilities are updated appropriately.

Usage

```
error.rates(x) <- value
```

Arguments

x	An object of class oncotree.
value	A numeric vector of length 2. The false positive error rate will be set to value[1], while the false negative error rate to value[2].

See Also

oncotree.fit

Examples

```
data(ov.cgh)
ov.tree <- oncotree.fit(ov.cgh)
ov.tree
error.rates(ov.tree) <- c(0,0)
ov.tree
```

`generate.data` *Generate random data from an oncogenetic tree*

Description

Generates random event occurrence data based on an oncogenetic tree model.

Usage

```
generate.data(N, otree, with.errors=TRUE,
          edge.weights=if (with.errors) "estimated" else "observed",
          method=c("S","D1","D2"))
```

Arguments

`N`	The required sample size.
`otree`	An object of the class `oncotree`.
`with.errors`	A logical value specifying whether false positive and negative errors should be applied.
`edge.weights`	A choice of whether the observed or estimated edge transition probabilities should be used in the calculation of probabilities. See `oncotree.fit` for explanation of the difference. By default, estimated edge transition probabilities if `with.errors=TRUE` and the observed ones if `with.errors=FALSE`.
`method`	Simulation method, see Details for explanation of the options.

Details

There are three choices for the method of simulation; the best choice depends on the size of the tree, required sample size, and whether errors are needed.

Method "S" generates the data based on the conditional probability definition of the oncogenetic tree, and then 'corrupts' the resulting sample by introducing random errors. This method is applicable in all circumstances, but can be slower than other methods if `N` is large and `with.errors=FALSE` is used.

Method "D1" calculates the joint distribution generated by the tree exactly (using `distribution.oncotree`), and the observations are generated by sampling this distribution. Thus if `with.errors=TRUE` and the tree is large, this method might fail due to the exponential growth in the number of potential outcomes. On the other hand, for a moderately sized tree and a large desired sample size `N` this is the most efficient method.

Method "D2" calculates the joint distribution generated by the tree without false positives/negatives, samples from it, and then 'corrupts' the resulting sample. If `with.errors=FALSE` is used then this method is equivalent to method "D1".

Value

A data set where each row is an independent observation.

Author(s)

Aniko Szabo

See Also

`oncotree.fit`

Examples

```
data(ov.cgh)
ov.tree <- oncotree.fit(ov.cgh[1:5])

set.seed(7365)
rd <- generate.data(200, ov.tree, with.errors=TRUE)

#compare timing of methods
system.time(generate.data(20, ov.tree, with.errors=TRUE, method="S"))
system.time(generate.data(20, ov.tree, with.errors=TRUE, method="D1"))
system.time(generate.data(20, ov.tree, with.errors=TRUE, method="D2"))
```

`oncotree` *Build and display an oncogenetic tree*

Description

Build a directed tree structure to model the process of occurrence of genetic alterations (events) in carcinogenesis. The model is described in more detail in `Oncotree-package`. Methods for printing a short summary, displaying the tree on an R plot, and producing latex code for drawing the tree (using the 'pstricks' and 'pst-tree' LaTeX packages) are provided.

Usage

```
  oncotree.fit(dataset, error.fun = function(x, y){sum((x - y)^2)})
  ## S3 method for class 'oncotree'
print(x, ...)
  ## S3 method for class 'oncotree'
plot(x, edge.weights = c("none", "observed", "estimated"),
    edge.digits=2, node.coords=NULL, plot=TRUE, cex = par("cex"),
  col.edge=par("col"), col.text=par("col"), col.weight=par("col"),...)
```

```
pstree.oncotree(x, edge.weights=c("none","observed","estimated"), edge.digits=2,
                  shape=c("none","oval", "circle", "triangle", "diamond"),
                  pstree.options=list(arrows="->", treefit="loose",
                                      arrowscale="1.5 0.8", nodesep="3pt"))
```

Arguments

dataset	A data frame or a matrix with variable names as a listing of genetic events taking on binary values indicating missing (0) or present (1). Each row is an independent sample.
error.fun	A function of two variables that measures the deviation of the observed marginal frequencies of the events (which will be the first argument in the call) from the estimated ones. The false positive and negative error rates are obtained by minimizing error.fun. If error.fun=NULL is used, the error rates are not estimated.
x	An object of class oncotree.
edge.weights	Choice of edge weights to show on the plot.
edge.digits	The number of significant digits to use when displaying edge weights.
node.coords	A matrix with node-coordinates or NULL if the coordinates should be computed automatically (default).
plot	Logical; indicates whether the tree should be plotted.
cex	Scaling factor for the text in the nodes.
col.edge	color of the tree edges.
col.text	color of the node label.
col.weight	color of the edge weights.
...	Ignored for print. For plot these can be graphical parameters passed to lines when the edges are drawn
shape	The shape of the node in the pst-tree representation.
pstree.options	Additional options for pst-tree. See the pstricks documentation for possible values.

Details

'pst-tree' is a very flexible package, and very detailed formatting of the tree is possible. pstree.oncotree provides some default settings for drawing trees, but they can be easily overridden: most options can be set in pstree.options, while the appearance of the tree nodes can be controlled by defining a one-parameter \lab command that gives the desired appearance. For example, if red, non-mathematical test is desired in an oval, you could use \newcommand{\lab}[1]{\Toval[name=#1]{{\red #1}}}.

Value

For oncotree.fit: an object of class oncotree which has components

data	data frame used, after dropping events with zero observed frequency, and adding a column for the artificial 'Root' node

`nmut` number of tree nodes: the number genetic events present in data +1 for the 'Root' node

`parent` a list containing information about the tree structure with the following components

- childa character vector of the event names starting with 'Root'
- parenta character vector of the names of the parents of `child`
- parent.numa numeric vector with column indices corresponding to `parent`
- obs.weightsraw edge transition probabilities P(child|parent)
- est.weightsedge transition probabilities adjusted for the error rates `eps`

`level` a numeric vector of the depth of each node in the tree (1 for the root, 2 for its children, etc.)

`numchild` a numeric vector giving the number of children for each node

`levelnodes` a numeric vector of the number of nodes found at each level of the tree

`levelgrp` a character matrix with its rows giving the ordered nodes at each level

`eps` a numeric vector of length two showing the estimated false positive and negative error rates (if `error.fun` is not NULL). Do not modify directly, but rather through `error.rates<-`.

For `print.oncotree`:

the original object is returned invisibly. It prints a summary showing the number of nodes, the parent-child relationships, and the false positive and negative error rates.

For `plot.oncotree`:

a matrix with node-coordinates is returned invisibly. The column names of the matrix are the names of the nodes/events (including 'Root'), the rows gives the x- and y-coordinates, respectively. This matrix provides a valid input for `node.coords`. If `plot=TRUE`, a plot of the tree is produced.

For `pstree.oncotree`:

a character string with the LaTeX code needed to draw a tree. `\usepackage{pstricks,pst-tree}` is required in the preamble of the LaTeX file, and it should be processed through a PostScript intermediary (DVIPS or similar) and not through PDFLaTeX.

Author(s)

Lisa Pappas

References

Szabo, A. and Boucher, K. (2002) Estimating an oncogenetic tree when false negative and positives are present. Mathematical Biosciences, 176/2, 219-236.

See Also

`bootstrap.oncotree`,`error.rates<-`, `generate.data`,`ancestors`,`distribution.oncotree`

Examples

```
data(ov.cgh)
ov.tree <- oncotree.fit(ov.cgh, error.fun=function(x,y){max(abs(x-y))})
ov.tree
nodes <- plot(ov.tree, edge.weights="est")
#move the Root node to the left
nodes["x","Root"] <- nodes["x","8q+"]
plot(ov.tree, node.coords=nodes)
#output for pstricks+pst-tree
pstree.oncotree(ov.tree, edge.weights="obs", shape="oval")
```

---

ov.cgh *Ovarian cancer CGH data*

Description

This is a data set obtained using the comparative genomic hybridization technique (CGH) on samples from papillary serous cystadenocarcinoma of the ovary. Only the seven most commonly occurring events are given.

Usage

```
data(ov.cgh)
```

Format

A data frame with 87 observations on the following 7 variables.

`8q+` a 0/1 indicator of the presence of the '8q+' event

`3q+` a 0/1 indicator of the presence of the '3q+' event

`5q-` a 0/1 indicator of the presence of the '5q-' event

`4q-` a 0/1 indicator of the presence of the '4q-' event

`8p-` a 0/1 indicator of the presence of the '8p-' event

`1q+` a 0/1 indicator of the presence of the '1q+' event

`Xp-` a 0/1 indicator of the presence of the 'Xp-' event

Details

The CGH technique uses fluorescent staining to detect abnormal (increased or decreased) number of DNA copies. Often the results are reported as a gain or loss on a certain arm, without further distinction for specific regions. It is common to denote a change in DNA copy number on a specific chromosome arm by prefixing a "-" sign for decrease and a "+" for increase. Thus, say, -3q denotes abnormally low DNA copy number on the q arm of the 3rd chromosome.

Source

NCBI's SKY-CGH database

Examples

```
data(ov.cgh)
heatmap(data.matrix(ov.cgh), Colv=NA, scale="none", col=c("gray90","red"))
```

Index