Package 'Canopy'

December 18, 2017

Type Package

Title Accessing Intra-Tumor Heterogeneity and Tracking Longitudinal and Spatial Clonal Evolutionary History by Next-Generation Sequencing

Version 1.3.0

Author Yuchao Jiang, Nancy R. Zhang

Maintainer Yuchao Jiang <yuchaoj@email.unc.edu>

Description A statistical framework and computational procedure for identifying the sub-populations within a tumor, determining the mutation profiles of each subpopulation, and inferring the tumor's phylogenetic history. The input are variant allele frequencies (VAFs) of somatic single nucleotide alterations (SNAs) along with allele-specific coverage ratios between the tumor and matched normal sample for somatic copy number alterations (CNAs). These quantities can be directly taken from the output of existing software. Canopy provides a general mathematical framework for pooling data across samples and sites to infer the underlying parameters. For SNAs that fall within CNA regions, Canopy infers their temporal ordering and resolves their phase. When there are multiple evolutionary configurations consistent with the data, Canopy outputs all configurations along with their confidence assessment.

License GPL-2

Depends R (>= 3.4), ape, fields, pheatmap, scatterplot3d

Imports grDevices, graphics, stats, utils

URL https://github.com/yuchaojiang/Canopy

NeedsCompilation no

Repository CRAN

Date/Publication 2017-12-18 19:12:06 UTC

R topics documented:

`addsamptree` *To determine whether the sampled tree will be accepted*

Description

To determine whether the sampled tree will be accepted by comparing the likelihood, used in `canopy.sample`.

Usage

```
addsamptree(tree,tree.new)
```

Arguments

tree	input tree (current)
tree.new	input tree (newly sampled)

Value

returned tree (either retain the old tree or accept the new tree).

Author(s)

Yuchao Jiang <yuchaoj@wharton.upenn.edu>

Examples

```
data(MDA231)
data(MDA231_tree)
sna.name = MDA231$sna.name
Y = MDA231$Y
C = MDA231$C
R = MDA231$R
X = MDA231$X
WM = MDA231$WM
Wm = MDA231$Wm
epsilonM = MDA231$epsilonM
epsilonm = MDA231$epsilonm
# sampling location of SNAs
tree.new = MDA231_tree
tree.new$sna = sampsna(MDA231_tree)
tree.new$Z = getZ(tree.new, sna.name)
tree.new$Q = getQ(tree.new, Y, C)
tree.new$H = tree.new$Q
tree.new$VAF = getVAF(tree.new, Y)
tree.new$likelihood = getlikelihood(tree.new, R, X, WM, Wm, epsilonM, epsilonm)
tree = addsamptree(MDA231_tree,tree.new)
```

AML43 *SNA input for primary tumor and relapse genome of leukemia patient from Ding et al. Nature 2012.*

Description

1242 SNAs from sequencing of leukemia patient at two timepoints. All SNAs are filtered to be from copy-number-neutral region.

Usage

```
data(AML43)
```

Value

List of simulated SNA input data for Canopy.

Author(s)

Yuchao Jiang <yuchaoj@wharton.upenn.edu>

Examples

```
data(AML43)
```

`canopy.BIC` *To get BIC as a model selection criterion*

Description

To get BIC as a model selection criterion from MCMC sampling results.

Usage

```
canopy.BIC(sampchain,projectname,K,numchain,burnin,thin,pdf)
```

Arguments

`sampchain`	list of sampled trees returned by `canopy.sample`
`projectname`	name of project
`K`	number of subclones (vector)
`numchain`	number of MCMC chains with random initiations
`burnin`	burnin of MCMC chains
`thin`	MCMC chains thinning
`pdf`	whether a pdf plot of BIC should be generated, default to be TRUE

Value

BIC values (vector) for model selection with plot generated (pdf format).

Author(s)

Yuchao Jiang <yuchaoj@wharton.upenn.edu>

Examples

```
data(MDA231_sampchain)
sampchain = MDA231_sampchain
projectname = 'MD231'
K = 3:6
numchain = 20
burnin = 150
thin = 5
bic = canopy.BIC(sampchain = sampchain, projectname = projectname, K = K,
                 numchain = numchain, burnin = burnin, thin = thin)
```

`canopy.cluster` *EM algorithm for multivariate clustering of SNAs*

Description

EM algorithm for multivariate clustering of SNAs.

Usage

```
canopy.cluster(R, X, num_cluster, num_run, Mu.init = NULL, Tau_Kplus1 = NULL)
```

Arguments

`R`	alternative allele read depth matrix
`X`	total read depth matrix
`num_cluster`	number of mutation clusters (BIC as model selection metric)
`num_run`	number of EM runs for estimation for each specific number of clusters (to avoid EM being stuck in local optima)
`Mu.init`	(optional) initial value of the VAF centroid for each mutation cluster in each sample
`Tau_Kplus1`	(optional) pre-specified proportion of noise component in clustering, uniformly distributed between 0 and 1

Value

Matrix of posterior probability of cluster assignment for each mutation.

Author(s)

Yuchao Jiang <yuchaoj@wharton.upenn.edu>

Examples

```
data(AML43)
R = AML43$R
X = AML43$X
Mu = AML43$Mu
Tau = AML43$Tau
pG = canopy.cluster.Estep(Tau, Mu, R, X)
```

---

canopy.cluster.Estep *E-step of EM algorithm for multivariate clustering of SNAs*

Description

E-step of EM algorithm for multivariate clustering of SNAs. Used in canopy.cluster.

Usage

```
canopy.cluster.Estep(Tau, Mu, R, X)
```

Arguments

Tau	prior for proportions of mutation clusters
Mu	MAF centroid for each mutation cluster in each sample
R	alternative allele read depth matrix
X	total read depth matrix

Value

Matrix of posterior probability of cluster assignment for each mutation.

Author(s)

Yuchao Jiang <yuchaoj@wharton.upenn.edu>

Examples

```
data(AML43)
R = AML43$R
X = AML43$X
Mu = AML43$Mu
Tau = AML43$Tau
pG = canopy.cluster.Estep(Tau, Mu, R, X)
```

canopy.cluster.Mstep *M-step of EM algorithm for multivariate clustering of SNAs*

Description

M-step of EM algorithm for multivariate clustering of SNAs. Used in canopy.cluster.

Usage

```
canopy.cluster.Mstep(pG, R, X, Tau_Kplus1)
```

Arguments

pG	matrix of posterior probability of cluster assignment for each mutation
R	alternative allele read depth matrix
X	total read depth matrix
Tau_Kplus1	proportion mutation cluster that is uniformly distributed to capture noise

Value

List of bic, converged Mu, Tau, and SNA cluster assignment.

Author(s)

Yuchao Jiang <yuchaoj@wharton.upenn.edu>

Examples

```
data(AML43)
R = AML43$R; X = AML43$X
num_cluster = 4 # Range of number of clusters to run
num_run = 6 # How many EM runs per clustering step
Tau_Kplus1=0.05 # Proportion of noise component
Mu.init=cbind(c(0.01,0.15,0.25,0.45),c(0.2,0.2,0.01,0.2)) # initial value
                                                          # of centroid
canopy.cluster=canopy.cluster(R = R, X = X, num_cluster = num_cluster,
                              num_run = num_run, Mu.init = Mu.init,
                              Tau_Kplus1=Tau_Kplus1)
```

canopy.output *To generate a posterior tree*

Description

To generate a posterior tree from the sub-tree space of trees with the same configurations.

Usage

```
canopy.output(post, config.i, C)
```

Arguments

post	list returned by canopy.post
config.i	configuration of sub-tree space to be output
C	CNA and CNA-region overlapping matrix, only needed if overlapping CNAs are used as input

Value

posterior tree from the sub-tree space of trees with the same configurations.

Author(s)

Yuchao Jiang <yuchaoj@wharton.upenn.edu>

Examples

```
data(MDA231_sampchain)
data(MDA231)
sampchain = MDA231_sampchain
projectname = 'MD231'
K = 3:6
numchain = 20
burnin = 150
thin = 5
optK = 4
C = MDA231$C
post = canopy.post(sampchain = sampchain, projectname = projectname, K = K,
                   numchain = numchain, burnin = burnin, thin = thin,
                   optK = optK, C = C)
config.i = 3
output.tree = canopy.output(post = post, config.i = config.i, C = C)
```

canopy.plottree *To plot tree inferred by Canopy*

Description

To plot Canopy's reconstructed phylogeny. Major plotting function of Canopy.

Usage

```
canopy.plottree(tree, pdf, pdf.name, txt, txt.name)
```

Arguments

tree	input tree to be plotted
pdf	whether a pdf plot should be generated, default to be FALSE
pdf.name	name of pdf to be generated, has to be provided if pdf is to be generated
txt	whether a txt file should be generated with information on mutations along the tree branches, default to be FALSE
txt.name	name of txt to be generated, has to be provided if txt is to be generated

Value

Plot of tree structure, clonal frequency and mutation legends (pdf format).

Author(s)

Yuchao Jiang <yuchaoj@wharton.upenn.edu>

Examples

```
data(MDA231_tree)
canopy.plottree(MDA231_tree, pdf = TRUE, pdf.name = 'MDA231_tree.pdf')
```

canopy.post *Posterior evaluation of MCMC sampled trees*

Description

Burnin, thinning, and posterior evaluation of MCMC sampled trees.

Usage

```
canopy.post(sampchain, projectname, K, numchain, burnin, thin, optK,
            C, post.config.cutoff)
```

Arguments

sampchain	list of sampled trees returned by canopy.sample
projectname	name of project
K	number of subclones (vector)
numchain	number of MCMC chains with random initiations
burnin	burnin of MCMC chains
thin	MCMC chain thinning.
optK	optimal number of subclones determined by canopy.BIC
C	CNA and CNA-region overlapping matrix, only needed if overlapping CNAs are used as input
post.config.cutoff	cutoff value for posterior probabilities of tree configurations, default is set to be 0.05 (only tree configurations with greater than 0.05 posterior probabilities will be reported by Canopy)

Value

samptreethin	list of sampled posterior trees
samptreethin.lik	vector of likelihood of sampled posterior trees
config	vector of configuration of sampled posterior trees (integer values)
config.summary	summary of configurations of sampled posterior trees

Author(s)

Yuchao Jiang <yuchaoj@wharton.upenn.edu>

Examples

```
data(MDA231_sampchain)
data(MDA231)
sampchain = MDA231_sampchain
projectname = 'MD231'
K = 3:6
numchain = 20
burnin = 150
thin = 5
optK = 4
C = MDA231$C
post = canopy.post(sampchain = sampchain, projectname = projectname, K = K,
                   numchain = numchain, burnin = burnin, thin = thin,
                   optK = optK, C = C)
```

canopy.sample *MCMC sampling in tree space*

Description

To sample the posterior trees. Major function of Canopy.

Usage

```
canopy.sample(R, X, WM, Wm, epsilonM, epsilonm, C=NULL,
              Y, K, numchain, max.simrun, min.simrun, writeskip, projectname,
              cell.line=NULL, plot.likelihood=NULL)
```

Arguments

R	alternative allele read depth matrix
X	total read depth matrix
WM	observed major copy number matrix
Wm	observed minor copy number matrix
epsilonM	observed standard deviation of major copy number (scalar input is transformed into matrix)
epsilonm	observed standard deviation of minor copy number (scalar input is transformed into matrix)
C	CNA and CNA-region overlapping matrix, only needed if overlapping CNAs are used as input
Y	SNA and CNA-region overlapping matrix
K	number of subclones (vector)
numchain	number of MCMC chains with random initiations
max.simrun	maximum number of simutation iterations for each chain
min.simrun	minimum number of simutation iterations for each chain
writeskip	interval to store sampled trees
projectname	name of project
cell.line	default to be FALSE, TRUE if input sample is cell line (no normal cell contamination)
plot.likelihood	default to be TRUE, posterior likelihood plot generated for check of convergence and selection of burnin and thinning in canopy.post

Value

List of sampleed trees in subtree space with different number of subclones; plot of posterior likelihoods in each subtree space generated (pdf format).

Author(s)

Yuchao Jiang <yuchaoj@wharton.upenn.edu>

Examples

```
data(MDA231)
R = MDA231$R; X = MDA231$X
WM = MDA231$WM; Wm = MDA231$Wm
epsilonM = MDA231$epsilonM; epsilonm = MDA231$epsilonm
C = MDA231$C
Y = MDA231$Y
K = 3:6
numchain = 20
projectname = 'MDA231'
# sampchain = canopy.sample(R = R, X = X, WM = WM, Wm = Wm, epsilonM = epsilonM,
#             epsilonm = epsilonm, C = C, Y = Y, K = K, numchain = numchain,
#             max.simrun = 50000, min.simrun = 10000, writeskip = 200,
#             projectname = projectname, cell.line = TRUE, plot.likelihood = TRUE)
```

---

canopy.sample.cluster *MCMC sampling in tree space with pre-clustering of SNAs*

Description

To sample the posterior trees with pre-clustering step of SNAs. Major function of Canopy.

Usage

```
canopy.sample.cluster(R, X, sna_cluster, WM, Wm, epsilonM, epsilonm, C=NULL,
            Y, K, numchain, max.simrun, min.simrun, writeskip, projectname,
              cell.line=NULL, plot.likelihood=NULL)
```

Arguments

R	alternative allele read depth matrix
X	total read depth matrix
sna_cluster	cluster assignment for each mutation from the EM Binomial clustering algorithm
WM	observed major copy number matrix
Wm	observed minor copy number matrix
epsilonM	observed standard deviation of major copy number (scalar input is transformed into matrix)
epsilonm	observed standard deviation of minor copy number (scalar input is transformed into matrix)
C	CNA and CNA-region overlapping matrix, only needed if overlapping CNAs are used as input

Y	SNA and CNA-region overlapping matrix
K	number of subclones (vector)
numchain	number of MCMC chains with random initiations
max.simrun	maximum number of simutation iterations for each chain
min.simrun	minimum number of simutation iterations for each chain
writeskip	interval to store sampled trees
projectname	name of project
cell.line	default to be FALSE, TRUE if input sample is cell line (no normal cell contamination)
plot.likelihood	default to be TRUE, posterior likelihood plot generated for check of convergence and selection of burnin and thinning in canopy.post

Value

List of sampleed trees in subtree space with different number of subclones; plot of posterior likelihoods in each subtree space generated (pdf format).

Author(s)

Yuchao Jiang <yuchaoj@wharton.upenn.edu>

Examples

```
data(MDA231)
R = MDA231$R; X = MDA231$X
WM = MDA231$WM; Wm = MDA231$Wm
epsilonM = MDA231$epsilonM; epsilonm = MDA231$epsilonm
C = MDA231$C
Y = MDA231$Y
K = 3:6
numchain = 20
projectname = 'MDA231'
# sampchain = canopy.sample.cluster(R = R, X = X, sna_cluster=c(1,2,3,4),
#             WM = WM, Wm = Wm, epsilonM = epsilonM,
#             epsilonm = epsilonm, C = C, Y = Y, K = K, numchain = numchain,
#             max.simrun = 50000, min.simrun = 10000, writeskip = 200,
#             projectname = projectname, cell.line = TRUE, plot.likelihood = TRUE)
```

---

canopy.sample.cluster.nocna

MCMC sampling in tree space with pre-clustering of SNAs

Description

To sample the posterior trees with pre-clustering step of SNAs. Major function of Canopy.

Usage

```
canopy.sample.cluster.nocna(R, X, sna_cluster, K, numchain,
                            max.simrun, min.simrun, writeskip, projectname,
                            cell.line=NULL, plot.likelihood=NULL)
```

Arguments

R	alternative allele read depth matrix
X	total read depth matrix
sna_cluster	cluster assignment for each mutation from the EM Binomial clustering algorithm
K	number of subclones (vector)
numchain	number of MCMC chains with random initiations
max.simrun	maximum number of simutation iterations for each chain
min.simrun	minimum number of simutation iterations for each chain
writeskip	interval to store sampled trees
projectname	name of project
cell.line	default to be FALSE, TRUE if input sample is cell line (no normal cell contamination)
plot.likelihood	default to be TRUE, posterior likelihood plot generated for check of convergence and selection of burnin and thinning in canopy.post

Value

List of sampleed trees in subtree space with different number of subclones; plot of posterior likelihoods in each subtree space generated (pdf format).

Author(s)

Yuchao Jiang <yuchaoj@wharton.upenn.edu>

Examples

```
data(toy3)
R = toy3$R; X = toy3$X
sna_cluster = toy3$sna_cluster
K = 3:5
numchain = 10
projectname = 'toy3'
# sampchain = canopy.sample.cluster.nocna(R = R, X = X,
#             sna_cluster=sna_cluster, K = K, numchain = numchain,
#             max.simrun = 40000, min.simrun = 10000, writeskip = 200,
#             projectname = projectname,
#             cell.line = TRUE, plot.likelihood = TRUE)
```

canopy.sample.nocna *MCMC sampling in tree space*

Description

To sample the posterior trees without CNA input. Major function of Canopy.

Usage

```
canopy.sample.nocna(R, X, K, numchain, max.simrun, min.simrun, writeskip,
              projectname, cell.line=NULL, plot.likelihood=NULL)
```

Arguments

R	alternative allele read depth matrix
X	total read depth matrix
K	number of subclones (vector)
numchain	number of MCMC chains with random initiations
max.simrun	maximum number of simutation iterations for each chain
min.simrun	minimum number of simutation iterations for each chain
writeskip	interval to store sampled trees
projectname	name of project
cell.line	default to be FALSE, TRUE if input sample is cell line (no normal cell contamination)
plot.likelihood	default to be TRUE, posterior likelihood plot generated for check of convergence and selection of burnin and thinning in canopy.post

Value

List of sampleed trees in subtree space with different number of subclones; plot of posterior likelihoods in each subtree space generated (pdf format).

Author(s)

Yuchao Jiang <yuchaoj@wharton.upenn.edu>

Examples

```
data(toy3)
R = toy3$R; X = toy3$X
K = 3:5
numchain = 10
projectname = 'toy3'
# sampchain = canopy.sample.nocna(R = R, X = X, K = K, numchain = numchain,
```

```
#             max.simrun = 50000, min.simrun = 10000, writeskip = 200,
#             projectname = projectname,
#             cell.line = TRUE, plot.likelihood = TRUE)
```

`getclonalcomposition` *To get clonal composition*

Description

To get clonal composition (mutational profile of each clone) of tree. Used in `canopy.post`.

Usage

```
getclonalcomposition(tree)
```

Arguments

`tree`	input tree

Value

List of each clone's mutational profile.

Author(s)

Yuchao Jiang <yuchaoj@wharton.upenn.edu>

Examples

```
data(MDA231_tree)
getclonalcomposition(MDA231_tree)
```

`getCMCm` *To get major and minor copy per clone*

Description

To get major and minor copy per clone. Used in `canopy.sample`.

Usage

```
getCMCm(tree, C)
```

Arguments

`tree`	input tree
`C`	CNA regions and CNA overlapping matrix

Value

CM	Matrix of major copy per clone.
Cm	Matrix of minor copy per clone.

Author(s)

Yuchao Jiang <yuchaoj@wharton.upenn.edu>

Examples

```
data(MDA231_tree)
data(MDA231)
C = MDA231$C
getCMCm(MDA231_tree, C)
```

---

getCZ *To get CNA genotyping matrix CZ*

Description

To get CNA genotyping matrix CZ from location of CNAs on the tree. Used in canopy.sample.

Usage

```
getCZ(tree)
```

Arguments

tree	input tree

Value

CNA genotyping matrix CZ.

Author(s)

Yuchao Jiang <yuchaoj@wharton.upenn.edu>

Examples

```
data(MDA231_tree)
getCZ(MDA231_tree)
```

getlikelihood *To get likelihood of the tree*

Description

To get likelihood of the tree given tree struture and data input. Used in canopy.sample.

Usage

```
getlikelihood(tree,R,X,WM,Wm,epsilonM,epsilonm)
```

Arguments

tree	input tree
R	alternative allele read depth matrix
X	total read depth matrix
WM	observed major copy number matrix
Wm	observed minor copy number matrix
epsilonM	observed standard deviation of major copy number (scalar input is transformed into matrix)
epsilonm	observed standard deviation of minor copy number (scalar input is transformed into matrix)

Value

Likelihood of sampled tree.

Author(s)

Yuchao Jiang <yuchaoj@wharton.upenn.edu>

Examples

```
data(MDA231)
data(MDA231_tree)
R = MDA231$R
X = MDA231$X
WM = MDA231$WM
Wm = MDA231$Wm
epsilonM = MDA231$epsilonM
epsilonm = MDA231$epsilonm
getlikelihood(MDA231_tree, R, X, WM, Wm, epsilonM, epsilonm)
```

getlikelihood.sna *To get SNA likelihood of the tree*

Description

To get SNA likelihood of the tree given tree struture and data input. Used in `canopy.sample.nocna` and `canopy.sample.cluster.nocna`.

Usage

```
getlikelihood.sna(tree, R, X)
```

Arguments

`tree`	input tree
`R`	alternative allele read depth matrix
`X`	total read depth matrix

Value

Likelihood of sampled tree.

Author(s)

Yuchao Jiang <yuchaoj@wharton.upenn.edu>

Examples

```
data(MDA231)
data(MDA231_tree)
R = MDA231$R
X = MDA231$X
getlikelihood.sna(MDA231_tree, R, X)
```

getQ *To get SNA-CNA genotyping matrix*

Description

To get SNA-CNA genotyping matrix Q, which specifies whether an SNA precedes a CNA. Used in `canopy.sample`.

Usage

```
getQ(tree, Y, C)
```

Arguments

tree	input tree
Y	SNA CNA overlapping matrix
C	CNA and CNA region overlapping matrix

Value

Genotyping matrix Q.

Author(s)

Yuchao Jiang <yuchaoj@wharton.upenn.edu>

Examples

```
data(MDA231_tree)
data(MDA231)
Y = MDA231$Y
C = MDA231$C
getQ(MDA231_tree, Y, C)
```

---

getVAF *To get variant allele frequency (VAF)*

Description

To get variant allele frequency (VAF) matrix, which contains percentage of mutant SNA alleles across samples. Used in canopy.sample.

Usage

```
getVAF(tree,Y)
```

Arguments

tree	input tree
Y	SNA CNA overlapping matrix

Value

Variant allele frequency matrix VAF.

Author(s)

Yuchao Jiang <yuchaoj@wharton.upenn.edu>

Examples

```
data(MDA231_tree)
data(MDA231)
Y = MDA231$Y
getVAF(MDA231_tree, Y)
```

`getZ` *To get SNA genotyping matrix Z*

Description

To get SNA genotyping matrix Z from location of SNAs on the tree. Used in `canopy.sample`.

Usage

```
getZ(tree, sna.name)
```

Arguments

`tree`	input tree
`sna.name`	vector of SNA names

Value

Genotyping matrix Z.

Author(s)

Yuchao Jiang <yuchaoj@wharton.upenn.edu>

Examples

```
data(MDA231_tree)
data(MDA231)
sna.name = rownames(MDA231$R)
getZ(MDA231_tree, sna.name)
```

initialcna *To initialize positions of CNAs*

Description

To initialize positions of CNAs on the tree. Used in initialization step of canopy.sample.

Usage

```
initialcna(tree,cna.name)
```

Arguments

tree	input tree
cna.name	vector of input CNA names

Value

Matrix specifying positions of CNAs (start and end node).

Author(s)

Yuchao Jiang <yuchaoj@wharton.upenn.edu>

Examples

```
data(MDA231_tree)
data(MDA231)
cna.name = rownames(MDA231$WM)
initialcna(MDA231_tree, cna.name)
```

initialcnacopy *To initialize major and minor copies of CNAs*

Description

To initialize major and minor copies of CNAs. Used in initialization step of canopy.sample.

Usage

```
initialcnacopy(tree)
```

Arguments

tree	input tree

Value

Matrix specifying major and minor copies of CNAs.

Author(s)

Yuchao Jiang <yuchaoj@wharton.upenn.edu>

Examples

```
data(MDA231_tree)
initialcnacopy(MDA231_tree)
```

`initialP` *To initialize clonal frequency matrix*

Description

To initialize clonal frequency matris P. Used in initialization step of `canopy.sample`.

Usage

```
initialP(tree,sampname,cell.line)
```

Arguments

`tree`	input tree
`sampname`	vector of input sample names
`cell.line`	default to be FALSE, TRUE if input sample is cell line (no normal cell contamination)

Value

Clonal frequency matrix P.

Author(s)

Yuchao Jiang <yuchaoj@wharton.upenn.edu>

Examples

```
data(MDA231_tree)
data(MDA231)
sampname = colnames(MDA231$R)
initialP(MDA231_tree, sampname, cell.line = TRUE)
```

initialsna *To initialize positions of SNAs*

Description

To initialize positions of SNAs on the tree. Used in initialization step of `canopy.sample`.

Usage

```
initialsna(tree,sna.name)
```

Arguments

`tree`	input tree
`sna.name`	vector of input SNA names

Value

Matrix specifying positions of SNAs (start and end node).

Author(s)

Yuchao Jiang <yuchaoj@wharton.upenn.edu>

Examples

```
data(MDA231_tree)
data(MDA231)
sna.name = rownames(MDA231$R)
initialsna(MDA231_tree, sna.name)
```

MDA231 *Dataset for project MDA231*

Description

Pre-stored dataset for project MDA231. A transplantable metastasis model system was derived from a heterogeneous human breast cancer cell line MDA-MB-231. Cancer cells from the parental line MDA-MB-231 were engrafted into mouse hosts leading to organ-specific metastasis. Mixed cell populations (MCPs) were in vivo selected from either bone or lung metastasis and grew into phenotypically stable and metastatically competent cancer cell lines. The parental line as well as the MCP sublines were whole-exome sequenced with somatic SNAs and CNAs profiled.

Usage

```
data(MDA231)
```

Value

List of input data for Canopy from project MDA231.

Author(s)

Yuchao Jiang <yuchaoj@wharton.upenn.edu>

Examples

```
data(MDA231)
```

MDA231_sampchain *List of pre-sampled trees*

Description

List of sampleed trees in subtree space with different number of subclones for project MDA231.

Usage

```
data(MDA231_sampchain)
```

Value

List of sampled trees from different subtree space

Author(s)

Yuchao Jiang <yuchaoj@wharton.upenn.edu>

Examples

```
data(MDA231_sampchain)
```

MDA231_tree *Most likely tree from project MDA231*

Description

Most likely tree from project MDA231 as a tree example.

Usage

```
data(MDA231_tree)
```

Value

Most likely tree from project MDA231

Author(s)

Yuchao Jiang <yuchaoj@wharton.upenn.edu>

Examples

```
data(MDA231_tree)
```

sampcna *To sample CNA positions*

Description

To sample CNA positions along the tree. Used in `canopy.sample`.

Usage

```
sampcna(tree)
```

Arguments

tree	input tree

Value

Newly sampled matrix specifying positions of CNAs (start and end node).

Author(s)

Yuchao Jiang <yuchaoj@wharton.upenn.edu>

Examples

```
data(MDA231_tree)
sampcna(MDA231_tree)
```

---

sampcnacopy *To sample major and minor copies of CNAs*

Description

To sample major and minor copies of CNAs. Used in `canopy.sample`.

Usage

```
sampcnacopy(tree)
```

Arguments

`tree`	input tree

Value

Newly sampled matrix specifying major and minor copies of CNAs.

Author(s)

Yuchao Jiang <yuchaoj@wharton.upenn.edu>

Examples

```
data(MDA231_tree)
sampcnacopy(MDA231_tree)
```

---

sampP *To sample clonal frequency*

Description

To sample clonal frequency matrix P. Used in `canopy.sample`.

Usage

```
sampP(tree, cell.line)
```

Arguments

tree	input tree
cell.line	default to be FALSE, TRUE if input sample is cell line (no normal cell contamination)

Value

Newly sampled clonal frequency matrix P.

Author(s)

Yuchao Jiang <yuchaoj@wharton.upenn.edu>

Examples

```
data(MDA231_tree)
sampP(MDA231_tree, cell.line = TRUE)
```

---

sampsna *To sample SNA positions*

Description

To sample SNA positions along the tree. Used in canopy.sample.

Usage

```
sampsna(tree)
```

Arguments

tree	input tree

Value

Newly sampled matrix specifying positions of SNAs (start and end node).

Author(s)

Yuchao Jiang <yuchaoj@wharton.upenn.edu>

Examples

```
data(MDA231_tree)
sampsna(MDA231_tree)
```

sampsna.cluster *To sample positions of SNA clusters*

Description

To sample SNA cluster positions along the tree. Used in canopy.sample.cluster and canopy.sample.cluster.nocna.

Usage

```
sampsna.cluster(tree)
```

Arguments

tree	input tree

Value

Newly sampled matrix specifying positions of SNA clusters (start and end node).

Author(s)

Yuchao Jiang <yuchaoj@wharton.upenn.edu>

Examples

```
data(MDA231_tree)
MDA231_tree$sna.cluster=initialsna(MDA231_tree,paste('cluster',1:4,sep=''))
sampsna.cluster(MDA231_tree)
```

sortcna *To sort identified overlapping CNAs.*

Description

To sort identified overlapping CNAs by their major and minor copy numbers. Used in canopy.post.

Usage

```
sortcna(tree,C)
```

Arguments

tree	input tree
C	CNA and CNA-region overlapping matrix

Value

Tree whose overlapping CNAs are sorted by major and minor copy numbers.

Author(s)

Yuchao Jiang <yuchaoj@wharton.upenn.edu>

Examples

```
data(MDA231_tree)
data(MDA231)
C = MDA231$C
sortcna(MDA231_tree, C)
```

---

toy *Toy dataset for Canopy*

Description

Pre-stored simulated toy dataset.

Usage

```
data(toy)
```

Value

List of simulated input data for Canopy.

Author(s)

Yuchao Jiang <yuchaoj@wharton.upenn.edu>

Examples

```
data(toy)
```

toy2 *Toy dataset 2 for Canopy*

Description

Pre-stored simulated toy dataset.

Usage

```
data(toy2)
```

Value

List of simulated input data for Canopy.

Author(s)

Yuchao Jiang <yuchaoj@wharton.upenn.edu>

Examples

```
data(toy2)
```

toy3 *Toy dataset 3 for Canopy*

Description

Pre-stored simulated toy dataset. 200 simulated SNAs from a tree with 4 branches. No CNA events at play.

Usage

```
data(toy3)
```

Value

List of simulated SNA input data for Canopy.

Author(s)

Yuchao Jiang <yuchaoj@wharton.upenn.edu>

Examples

```
data(toy3)
```

Index