Package ‘Clonality’

October 17, 2017

Type Package

Title Clonality testing

Version 1.24.0

Date 2017-18-04

Author Irina Ostrovnaya

Maintainer Irina Ostrovnaya <ostrovni@mskcc.org>

Depends R (>= 2.12.2), DNAcopy

Imports grDevices, graphics, stats, utils

Suggests gdata

Description Statistical tests for clonality versus independence of tumors from the same patient based on their LOH or genomewide copy number profiles

License GPL-3

LazyLoad yes

biocViews Microarray, CopyNumberVariation, Classification, aCGH

NeedsCompilation no

R topics documented:

`Clonality-package` *Clonality testing*

Description

Statistical tests for clonality versus independence of tumors from the same patient based on their LOH or genomewide copy number profiles.

Details

Package:	Clonality
Type:	Package
Version:	0.99.3
Date:	2014-9-07
License:	GPL-3
LazyLoad:	yes

Author(s)

Irina Ostrovnaya `<ostrovni@mskcc.org>`

`ave.adj.probes` *Averaging of adjacent probes in copy number arrays*

Description

For each sample the log-ratios at each consecutive K number of probes are averaged.

Usage

```
ave.adj.probes(data, K)
```

Arguments

`data`	Copy Number Array object (output of function CNA() from the package DNAcopy). First column contains chromosomes, second column contains genomic locations. Each remaining column contains log-ratios from a particular tumor or sample.
`K`	Number of markers to be averaged. Should be selected so that the final resolution of the averaged data would be 5,000-10,000 markers.

Details

Averages log-ratios in every K consecutive markers. The purpose of this step is to reduce the noise in the data, eliminate possible very small germline copy number variations, and get rid of a possible wave effect.

Value

Returns CNA object of reduced resolution

Examples

```
# Same example as in clonality.analysis()

set.seed(100)
chrom<-rep(c(1:22),each=100)
maploc<- runif(2200)* 200000
chromarm<-splitChromosomes(chrom,maploc)


#Simulate the dataset with 10 pairs of tumors with 22 chromosomes, 100 markers each
#Simulated log-ratios are equal to signal + noise
#Signal: each chromosome has 50% chance to be normal, 30% to be whole-arm loss/gain, and 20% to be partial arm
#There are no chromosomes with recurrent losses/gains
#Noise: drawn from normal distribution with mean 0, standard deviation 0.25
#First 9 patients have independent tumors, last patient has two tumors with identical signal, independent nois


set.seed(100)
chrom<-paste("chr",rep(c(1:22),each=100),"p",sep="")
chrom[nchar(chrom)==5]<-paste("chr0",substr(chrom[nchar(chrom)==5] ,4,5),sep="")
maploc<- rep(c(1:100),22)
data<-NULL
for (pt in 1:9)  #first 9 patients have independent tumors
{
tumor1<-tumor2<- NULL
mean1<- rnorm(22)
mean2<- rnorm(22)
for (chr in 1:22)
{
  r<-runif(2)
if (r[1]<=0.5) tumor1<-c(tumor1,rep(0,100))
  else if   (r[1]>0.7)  tumor1<-c(tumor1,rep(mean1[chr],100))
  else  { i<-sort(sample(1:100,2))
       tumor1<-c(tumor1,mean1[chr]*c(rep(0,  i[1]),rep(1, i[2]-i[1]), rep(0,  100-i[2])))
        }
if (r[2]<=0.5) tumor2<-c(tumor2,rep(0,100))
  else if   (r[2]>0.7)  tumor2<-c(tumor2,rep(mean2[chr],100))
  else   {i<-sort(sample(1:100,2))
      tumor2<-c(tumor2,mean2[chr]*c(rep(0,  i[1]),rep(1, i[2]-i[1]), rep(0,  100-i[2])))
         }
}
data<-cbind(data,tumor1,tumor2)
}

#last patient has identical profiles
tumor1<- NULL
```

```
mean1<- rnorm(22)
for (chr in 1:22)
{
  r<-runif(1)
if (r<=0.4) tumor1<-c(tumor1,rep(0,100))
  else if   (r>0.6)  tumor1<-c(tumor1,rep(mean1[chr],100))
  else  { i<-sort(sample(1:100,2))
       tumor1<-c(tumor1,mean1[chr]*c(rep(0,  i[1]),rep(1, i[2]-i[1]), rep(0,  100-i[2])))
        }

}
data<-cbind(data,tumor1,tumor1)

data<-data+matrix(rnorm( 44000,mean=0,sd=0.4) ,nrow=2200,ncol=20)
dataCNA<-CNA(data,chrom=chrom,maploc=maploc,sampleid=paste("pt",rep(1:10,each=2),rep(1:2,10)))
dim(dataCNA)
dataCNA2<-ave.adj.probes(dataCNA, 2)
dim(dataCNA2)
```

chromosomePlots *Per-chromosome plots of the copy number arrays from a particular patient*

Description

The function produces a sequence of plots for each chromosome with one-step segmented data of all samples of a particular patient.

Usage

```
chromosomePlots(data.seg1, ptlist, ptname,nmad)
```

Arguments

data.seg1	Output of one-step segmentation - output OneStepSeg of clonality.analysis().
ptlist	Vector of the patient IDs in the order of the samples appearing in the data. For example, if the first three tumors belong to patient A, and the following two belong to patient B, then ptlist=c('ptA', 'ptA', 'ptA', 'ptB', 'ptB').
ptname	Name of the patient from ptlist for which the data should be plotted
nmad	Number of MADs (median absolute deviations) that is used for Gain/Loss calls. Used to mark the Gain/Loss threshold on the plots.

Details

The function produces a sequence of plots for each chromosome with one-step segmented data of all samples of a particular patient. The dotted horizontal lines denote the gain and loss thresholds.

Examples

```
# See example as in clonality.analysis()
```

`clonality.analysis` *Clonality testing using copy number data*

Description

Function to test clonality of two tumors from the same patient based on their genomewide copy number profiles. This function calculates likelihood ratios and the reference distribution under the hypothesis of independence.

Usage

```
clonality.analysis(data, ptlist, pfreq = NULL, refdata = NULL, nmad = 1.25, reference = TRUE, allpa
```

Arguments

`data` Copy Number Array object (output of function CNA() from package DNAcopy). First column contains chromosomes, second column contains genomic locations. Each remaining column contains log-ratios from a particular tumor or sample. Chromosomes X and Y should be removed prior to analysis, and chromosomes should be split into p and q arms to improve the power (use function splitChromosomes()).

`ptlist` Vector of the patient IDs in the order of the samples appearing in the data. For example, if the first three tumors (columns 3, 4, 5 of data) belong to patient A, and the following two (columns 6, 7 of data) belong to patient B, then ptlist=c('ptA', 'ptA', 'ptA', 'ptB', 'ptB'). Note that while sample names in data should be unique the ptlist should have repeated labels.

`pfreq` Marginal frequencies of Gains, Losses and Normals for all the chromosomes. If it is not known, pfreq should be set to NULL and frequencies will be estimated from all the samples in the dataset. If frequencies are known, pfreq should be a data frame with 4 columns: 1) chromosome arm in the format 'chr01p', probability of 2) gain, 3) loss and 4) normal.

`refdata` If available, additional cohort of patients with the same disease that should be used to estimate the marginal gain/loss frequencies. If NULL, the original set of tumors is used, otherwise, refdata should be a CNA object. It will be segmented with 1 step CBS and each chromosome will be classified as gain/loss as described in the manuscript, leading to frequency estimates. No averaging or chromosome splitting is done for this dataset, so users should make sure refdata has chromosomes in the format 'chr01p' and that its resolution is similar to the one of the original data.

`nmad` Number of MADs (median absolute deviations) that is used for Gain/Loss calls. For each array MAD of its residuals (that is, data minus segmentation means) is calculated. Residuals represent the array's noise revel. Any segment of this array that has a mean at least nmad MADs above or below array's median is called a gain or a loss. We use value of 1.25, while values in the range of 0.5 to 2 can also be admissible depending on the resolution and presence of artifacts.

`reference` If TRUE the reference distribution of likelihood ratios is created under hypothesis of independence by pairing (independent) tumors from different patients.

`allpairs`	If TRUE all possible pairs of tumors from different patients will be used for reference distribution. If two tumors in a pair are not exchangeable, for example primary tumor vs recurrence, or pre-cancerous lesion vs tumor, then allpairs should be set to FALSE and the 'first' tumor should always come earlier in the data before the 'second' tumor for all the patients. Then 'first' tumors of patients will only be paired with 'second' tumors of other patients for the reference distribution.
`segmethod`	The segmentation algorithm to be used. The default is "oneseg" which uses the built in function of the same name based on the CBS algorithm. An alternative segmentation algorithm can be used. A function should be created and the name passed as described in the vignette.
`segpar`	The parameters necessary for the segmentation algorithm as a list. For "oneseg" you can specify alpha (default = 0.01) and nperm (default = 2000) necessary for the CBS algorithm.

Details

The function implements the statistical procedure designed to distinguish whether the two tumors from the same patient are clonal (have the same progenitor cancer cell) or independent (developed from normal cells independently). At first data are segmented with one step CBS (Olshen, A. B., Venkatraman, E. S., Lucito, R., Wigler, M. (2004). Circular binary segmentation for the analysis of array-based DNA copy number data. Biostatistics 5: 557-572) that picks at most one copy number change per chromosome arm. Then each chromosome arm is classified as Gain/Loss/Normal based on a middle segment if there are 3 segments, or based on the most outstanding segment if there are 2 segments. The multinomial likelihood ratio comparing these classifications is computed (LR1). For each concordant partial arm gain or loss we also calculate likelihood ratio that this change is exactly the same in both tumors. These likelihood ratios are multiplied by LR1 to obtain our final statistic, LR2. If LR2 is much greater than 1, that indicates clonality. If LR2 is much smaller than 1, it indicates independence. The reference distribution of LR2 under the hypothesis of independence is obtained by pairing up tumors from different patients, which are independent by default.

Since only one gain/loss is admissible per chromosome arm it is highly recommended to apply this methodology to arrays with at most 10,000-15,000 markers. We suggest averaging blocks of consecutive probes for arrays with larger resolution, see function ave.adj.probes.

Value

If the reference is TRUE, function returns the list with 4 elements: LR, OneStepSeg, ChromClass, refLR.

LR - matrix with the within patient comparisons. Each row corresponds to a pair of samples being compared. Columns are: Sample1 - name of sample 1; Sample2 - name of sample 2; LR1 - likelihood ratio without comparisons of specific concordant gains/losses; LR2 - final likelihood ratio with individual comparisons; GGorLL - number of chromosome arms that are classified as Gains in both tumors or Losses in both tumors; NN - number of chromosome arms that are classified as Normal in both tumors; GL - number of chromosome arms that are classified as Gain in one tumors and Loss in another; GNorLN - number of chromosome arms that are classified as Gain(Loss) in one tumors and Normal in another; IndividualComparisons - list of chromosome arms that had comparisons of specific concordant gains/losses in both tumors and the corresponding likelihood ratio for them being exactly the same. p-value - quantile of the reference distribution under the null hypothesis (refLR$LR2) that the value of LR2 match.

OneStepSeg - is the output of one step segmentation of the data. It has the same structure as the output of 'segment' from DNAcopy, but only one most prominent change per arm is allowed.

ChromClass - is the matrix of chromosome classifications based on the one step segmentation. Rows correspond to chromosome arms, columns correspond to samples. Chromosome arms are classified by the middle segment if there are 3 segments, and by the most outstanding segment if there are 2 segments.

refLR - matrix with the between patient comparisons (reference distribution under the hypothesis of independence). Has the same structure as LR but the pairs of tumors are selected from different patients.

Note that calculating the reference distribution might take a long time.

If the reference is FALSE, there is no p-value column in LR and no refLR output.

Author(s)

Irina Ostrovnaya `<ostrovni@mskcc.org>`

References

Ostrovnaya, I., Olshen, A. B., Seshan, V.E., Orlow, I., Albertson, D. G. and Begg, C. B. (2010), A metastasis or a second independent cancer? Evaluating the clonal origin of tumors using array copy number data. Statistics in Medicine, 29: 1608-1621

Ostrovnaya, I. and Begg, C. Testing Clonal Relatedness of Tumors Using Array Comparative Genomic Hybridization: A Statistical Challenge Clin Cancer Res March 1, 2010 16:1358-1367

Venkatraman, E. S. and Olshen, A. B. (2007). A faster circular binary segmentation algorithm for the analysis of array CGH data. Bioinformatics, 23:657 63.

Olshen, A. B., Venkatraman, E. S., Lucito, R., Wigler, M. (2004). Circular binary segmentation for the analysis of array-based DNA copy number data. Biostatistics 5: 557-572.

Examples

```
#Analysis of simulated data


#Simulate the dataset with 10 pairs of tumors with 22 chromosomes, 100 markers each
#Simulated log-ratios are equal to signal + noise
#Signal: each chromosome has 50% chance to be normal, 30% to be whole-arm loss/gain, and 20% to be partial arm
#There are no chromosomes with recurrent losses/gains
#Noise: drawn from normal distribution with mean 0, standard deviation 0.4
#First 9 patients have independent tumors, last patient has two tumors with identical signal, independent noi


set.seed(100)
chrom<-paste("chr",rep(c(1:22),each=100),"p",sep="")
chrom[nchar(chrom)==5]<-paste("chr0",substr(chrom[nchar(chrom)==5] ,4,5),sep="")
maploc<- rep(c(1:100),22)
data<-NULL
for (pt in 1:9)  #first 9 patients have independent tumors
{
tumor1<-tumor2<- NULL
mean1<- rnorm(22)
mean2<- rnorm(22)
for (chr in 1:22)
{
  r<-runif(2)
if (r[1]<=0.5) tumor1<-c(tumor1,rep(0,100))
```

```
  else if   (r[1]>0.7)  tumor1<-c(tumor1,rep(mean1[chr],100))
  else  { i<-sort(sample(1:100,2))
       tumor1<-c(tumor1,mean1[chr]*c(rep(0,  i[1]),rep(1, i[2]-i[1]), rep(0,  100-i[2])))
        }
if (r[2]<=0.5) tumor2<-c(tumor2,rep(0,100))
  else if   (r[2]>0.7)  tumor2<-c(tumor2,rep(mean2[chr],100))
  else   {i<-sort(sample(1:100,2))
      tumor2<-c(tumor2,mean2[chr]*c(rep(0,  i[1]),rep(1, i[2]-i[1]), rep(0,  100-i[2])))
         }
}
data<-cbind(data,tumor1,tumor2)
}

#last patient has identical profiles
tumor1<- NULL
mean1<- rnorm(22)
for (chr in 1:22)
{
  r<-runif(1)
if (r<=0.4) tumor1<-c(tumor1,rep(0,100))
  else if   (r>0.6)  tumor1<-c(tumor1,rep(mean1[chr],100))
  else  { i<-sort(sample(1:100,2))
       tumor1<-c(tumor1,mean1[chr]*c(rep(0,  i[1]),rep(1, i[2]-i[1]), rep(0,  100-i[2])))
        }

}
data<-cbind(data,tumor1,tumor1)

data<-data+matrix(rnorm( 44000,mean=0,sd=0.4) ,nrow=2200,ncol=20)
dataCNA<-CNA(data,chrom=chrom,maploc=maploc,sampleid=paste("pt",rep(1:10,each=2),rep(1:2,10)))
ptlist<- paste("pt",rep(1:10,each=2),sep=".")
samnms<-paste("pt",rep(1:10,each=2),rep(1:2,10),sep=".")
results<-clonality.analysis(dataCNA, ptlist,  pfreq = NULL, refdata = NULL, nmad = 1,
 reference = TRUE, allpairs = TRUE)



#genomewide plots of pairs of tumors from the same patient
pdf("genomewideplots.pdf",height=7,width=11)
for (i in unique(ptlist))
{
w<-which(ptlist==i)
ns<- length(w)
if (ns>1)
{
for (p1 in c(1:(ns-1)))
for (p2 in c((p1+1):ns))
genomewidePlots(results$OneStepSeg, results$ChromClass,ptlist , ptpair=samnms[c(w[p1],w[p2])],results$LR,  p
}
}
dev.off()



pdf("hist.pdf",height=7,width=11)
histogramPlot(results$LR[,4], results$refLR[,4])
dev.off()
```

```
for (i in unique(ptlist))
{
pdf(paste("Patient", i,".pdf",sep=""),height=7,width=11)
chromosomePlots(results$OneStepSeg, ptlist,ptname=i,nmad=1.25)
dev.off()
}
```

ECMtesting — *Clonality testing of >=3 tumors using Extended Concordant Mutations (ECM) test based on LOH (Loss of Heterozygosity) profiles*

Description

Function to test clonality of three and more tumors from the same patient based on their LOH profiles. This function implements Extended Concordant Mutations for all possible subsets of tumors from the same patient and minP multiplicity adjustment using simulated tumors.

Usage

```
ECMtesting(LOHtable,ptlist,noloh,loh1,loh2,Nsim=100)
```

Arguments

`LOHtable`	Matrix of LOH calls. Each row corresponds to a marker. First column contains the names of the markers. Each other column represents a sample and contains LOH calls.
`ptlist`	Vector of the patient IDs in the order the samples appear in the data. For example, if the first three tumors (columns 2, 3, 4 of data) belong to patient A, and the following two (columns 5, 6 of data) belong to patient B, then ptlist=c('ptA', 'ptA', 'ptA', 'ptB', 'ptB').
`noloh`	The string or a number that denotes absence of LOH.
`loh1`	The string or a number that denotes presence of LOH.
`loh2`	The string or a number that denotes presence of LOH that is discordant from loh1.
`Nsim`	Number of simulations used to calculate minP adjusted p-values

Details

Extended Concordant Mutations test is done for every subset of tumors. It uses number of concordant mutations in all tumors of the subset as a test statistic, and its reference distribution is calculated assuming fixed counts of LOH per tumor and equal probability of maternal and paternal alleles being affected. Note that ECM test for 2 tumors and original CM test will give slightly different p-values since continuity correction is done in ECM test.

Value

The function returns a list with number of elements equal to the number of patients. Each element is a matrix with two rows: ECM p-values for all possible subsets of tumors from this patient, and minP adjusted p-values. The tumors are denoted 1,2,3,... in the order they appear in LOHtable. Any tumor subsets with minP adjusted p-value <=0.05 can be considered clonal.

References

Ostrovnaya, I. "Testing clonality of three and more tumors using their loss of heterozygosity profiles", Statistical Applications in Genetics and Molecular Biology, 2012

Examples

```
set.seed(25)
LOHtable<-cbind(1:15,matrix(sample(c(0,1,2),15*12,replace=TRUE),ncol=12))
ECMtesting(LOHtable,rep(1:3,each=4),noloh=0,loh1=1,loh2=2,Nsim=100)
```

`genomewidePlots` *Plot of the genomewide copy number profiles of a pair of tumors.*

Description

Plot contains genomewide profiles from a pair of tumors. It uses the output from the function clonality.analysis().

Usage

```
genomewidePlots(data.seg1, classall, ptlist, ptpair, ptLR, plot.as.in.analysis = TRUE)
```

Arguments

`data.seg1` Output of one-step segmentation - output OneStepSeg of clonality.analysis(). The chromosomes should be in the format "chr01p", "chr01q" etc.

`classall` Classifications of the chromosomes - output ChromClass of clonality.analysis()

`ptlist` Vector of the patient IDs in the order of the samples appearing in the data.

`ptpair` Two sample names for which the plot is desired

`ptLR` Matrix with the likelihood ratios - output LR of clonality.analysis()

`plot.as.in.analysis`
If TRUE then the gain/loss patterns will be highlighted in accordance with the chromosome classification. For example, if there are three segments in a chromosome, then the middle one determines the chromosome status. If it is normal, no color will be plotted in the chromosome even if the 1st and 3rd segments are gains or losses. Another example: if there are 2 or 3 different segments of gains, they will be combined and only one segment will be plotted. If plot.as.in.analysis is equal to FALSE, the original one-step CBS segmentation will be plotted.

Details

Function produces genomewide plots of a pair of tumors. The log-ratios are plotted in grey in the order of their genomic locations, gains are plotted in blue, and losses are plotted in red.

Examples

```
 # See example as in clonality.analysis()
```

grid.lik *Auxiliary function: Grid of conditional probabilities*

Description

This auxiliary function generates the grid of likelihood values for each tumor pair (rows) and each value of xi (columns): P(observed mutations | xi)

Usage

```
grid.lik(xigrid, mutns, probamut)
```

Arguments

`xigrid`	Grid of the values of xi, corresponding to its domain of definition.
`mutns`	Matrix of the mutations observed, with all mutations in rows and the cases (tumor pairs) in columns. The data are coded as 0=mutation not observed, 1=shared mutation (observed in both tumors), 2=private mutation (observed in one tumor only).
`probamut`	Vector of the probabilities of occurence for each mutation.

Value

Return the matrix of the likelihood values for each tumor pair (rows) and each value of xi (columns). This matrix is called by the auxiliary function grid.lik, returned as a parameter by the function clonal.est, and used as a parameter by the function clonal.proba.

histogramPlot *Histrograms of Log-Likelihood Ratios*

Description

Function produces the histograms of the within-patient and between-patient log-Likelihood Ratios.

Usage

```
histogramPlot(ptLRvec, refLRvec)
```

Arguments

`ptLRvec`	Vector with the within-patient likelihood ratios - output LR of clonality.analysis()
`refLRvec`	Vector with the between-patient likelihood ratios - output refLR of clonality.analysis()

Details

Functions plots two overlapping histograms: within-patient log-likelihood ratios are in red and between-patient log-likelihood ratios (reference distribution under the hypothesis of independence) are in black.

Examples

```
# See example as in clonality.analysis()
```

`lcis` *Breast cancer data*

Description

For each sample the log-ratios at each consecutive K number of probes are averaged.

Usage

```
data(lcis)
```

Details

This is exome sequencing data from study of Lobular Carcinoma in Situ (LCIS) and Invaisve lobular carcinomas (ILC) or Invasive Ductal Carcinomas (IDC) in the same patients. First column called probi contains marginal probabilities that are obtained from breast cancer TCGA data and are not directly applicable to other cancers. Each subsequent column contains a pair of tumors where value 0 denotes that mutation is not observed, 1 if shared mutation is observed in both tumors, and 2 if it is a private mutation observed in only one tumor.

References

Begg CB, Ostrovnaya I, Carniello JV, Sakr RA, Giri D, Towers R, Schizas M, De Brot M, Andrade VP, Mauguen A, Seshan VE, King TA. "Clonal relationships between lobular carcinoma in situ and other breast malignancies.", Breast Cancer Res. 2016 Jun 23;18(1):66. doi: 10.1186/s13058-016-0727-z.

`LOHclonality` *Clonality testing using LOH (Loss of Heterozygosity) profiles*

Description

Function to test clonality of two tumors from the same patient based on their LOH profiles. This function implements Concordant Mutations and Likelihood Ratio tests.

Usage

```
LOHclonality(LOHtable, ptlist, refLOHtable = NULL, pfreq = NULL, noloh, loh1, loh2,method="both")
```

Arguments

`LOHtable` Matrix of LOH calls. Each row corresponds to a marker. First column contains the names of the markers. Each other column represents a sample and contains LOH calls.

`ptlist` Vector of the patient IDs in the order the samples appear in the data. For example, if the first three tumors (columns 3, 4, 5 of data) belong to patient A, and the following two (columns 6, 7 of data) belong to patient B, then ptlist=c('ptA', 'ptA', 'ptA', 'ptB', 'ptB').

`refLOHtable` Matrix of LOH calls that should be used to calculate the LOH frequencies used in Likelihood Ration calculation. The structure is similar to LOHtable. If refLOHtable is not specified, frequencies are calculated from LOHtable.

`pfreq` Vector of LOH frequencies known from the literature. Should be in the same order as the markers in LOHtable. If pfreq is not specified, frequencies are calculated from LOHtable.

`noloh` The string or a number that denotes absence of LOH.

`loh1` The string or a number that denotes presence of LOH.

`loh2` The string or a number that denotes presence of LOH that is discordant from loh1.

`method` Takes values "CM", "LR" or "both" if only Concordant Mutations test, or only Likelihood Ratio test, or both should be performed. Default value is "both".

Details

Function tests clonality of LOH profiles of tumors from the same patient using two tests. Concordant Mutations test has number of markers with concordant LOH as its test statistic. Its theoretical reference distribution under independence is calculated assuming that the maternal and paternal alleles are equally likely to be lost and that the frequencies of LOH are about the same across different markers.

Likelihood Ratio test uses pre-specified frequencies of LOH to compute Likelihood Ratio statistic. Its reference distribution is obtained by simulating tumors with the given LOH probabilities, and probability of maternal/paternal mutation estimated from the data. If LOH frequencies are not specified then they are estimated from the data.

Value

The function returns a data frame where each row corresponds to the pair of samples that are compared. Columns are: Sample1 - name of sample 1; Sample2 - name of sample 2; a - number of markers with concordant LOH in both tumors (test statistic for Concordant Mutations test); e - number of markers with LOH in both tumors, concordant or discordant; f - number of markers with LOH in the first tumor and Normal in the 2nd tumor; g - number of markers with LOH in the second tumor and Normal in the first tumor; h - number of markers that are Normal in both tumors; Ntot - total number of informative markers for both tumors; CMpvalue - p-value for Concordant Mutations test; LRpvalue - p-value for Likelihood Ratio test.

References

Begg CB, Eng KH, Hummer AJ. Statistical tests for clonality. Biometrics 2007; 63:522-530

Ostrovnaya I, Seshan VE, Begg CB. Comparison of properties of tests for assessing tumor clonality. Biometrics 2008; 68:1018-1022.

Examples

```
set.seed(25)
LOHtable<-cbind(1:20,matrix(sample(c(0,1,2),20*20,replace=TRUE),20))
LOHclonality(LOHtable,rep(1:10,each=2),pfreq=NULL,noloh=0,loh1=1,loh2=2)
```

`LRtesting3or4tumors` *Clonality testing of 3 or 4 tumors using Likelihood model based on LOH (Loss of Heterozygosity) profiles*

Description

Function to test clonality of 3 or 4 tumors from the same patient based on their LOH profiles.

Usage

```
LRtesting3or4tumors(LOHtable,ptlist,refLOHtable=NULL, pfreq=NULL,noloh,loh1,loh2,Nsim=100,m=0.5)
```

Arguments

`LOHtable` Matrix of LOH calls. Each row corresponds to a marker. First column contains the names of the markers. Each other column represents a sample and contains LOH calls.

`ptlist` Vector of the patient IDs in the order the samples appear in the data. For example, if the first three tumors (columns 2, 3, 4 of data) belong to patient A, and the following two (columns 5, 6 of data) belong to patient B, then ptlist=c('ptA', 'ptA', 'ptA', 'ptB', 'ptB').

`refLOHtable` Matrix of LOH calls that should be used to calculate the LOH frequencies used in Likelihood Ratio calculation. The structure is similar to LOHtable. If refLOHtable is not specified, frequencies are calculated from LOHtable.

`pfreq` Vector of LOH frequencies known from the literature. Should be in the same order as the markers in LOHtable. If pfreq is not specified, frequencies are calcualted from LOHtable.

`noloh` The string or a number that denotes absence of LOH.

`loh1` The string or a number that denotes presence of LOH.

`loh2` The string or a number that denotes presence of LOH that is discordant from loh1.

`Nsim` Number of simulations used to calculate minP adjusted p-values

`m` Probability that a favored allele is affected given that LOH has occurred. Must be a number above 0.5 (equal probability of maternal and paternal allelic loss)

Details

Likelihood ratio test for 3 and 4 tumors. For 3 tumors there are 3 possible tumor orderings, and for 4 tumors there are 2 topologies with 3 and 12 orderings each. The test calculates maximum likelihood ratio across all possible orderings, and the p-value is calculated using simulated reference distribution.

Value

The function returns a list with number of elements equal to the number of patients. Each element is list with two elements. First contains log maximum likelihood ratio value, p-value, and estimates of parameters c, the topology and tumor ordering that have maximum likelihood ratio. If p-value is significant, then the null hypothesis that all tumors are independent can be rejected. The second element has a matrix with all possible topologies and tumor orderings and their corresponding log likelihood ratios.

References

Ostrovnaya, I. "Testing clonality of three and more tumors using their loss of heterozygosity profiles", Statistical Applications in Genetics and Molecular Biology, 2012

Examples

```
set.seed(25)
LOHtable<-cbind(1:15,matrix(sample(c(0,1,2),15*12,replace=TRUE),ncol=12))
q<-LRtesting3or4tumors(LOHtable,rep(1:4,each=3),refLOHtable=NULL, pfreq=NULL,noloh=0,loh1=1,loh2=2,Nsim=100
```

`model.lik` *Auxiliary likelihood Function*

Description

This function computes the likelihood of the model.

Usage

```
model.lik(para, likmat, out0, xigrid)
```

Arguments

`para` Value of the model parameters, in the form c(mu, sigma, pi).

`likmat` Grid of conditional probabilities for each tumor pair (rows) and each value of xi (columns). This matrix is generated by the function grid.lik.

`out0` a small value that is used when the likelihood goes to infinite values, posing problem for the maximization. The corresponding combination of the parameters will thus be excluded from the search.

`xigrid` Grid of the values of xi, corresponding to its domain of definition.

Value

Return the likelihood value of the model for the given parameters. This likelihood function is the one that is maximized in the clonal.est function.

mutation.proba *Probability of being clonal*

Description

This function uses the results from mutation.rem to estimate the diagnostic probability of clonal relatedness for new cases. It is obtained from Bayes theorem using the prior probability of clonal relatedness (pi) and the contributions to the likelihood based on the mutations observed for the case. We recommand to use this function to estimate probabilities of clonality for new subjects, ie who are not used for the model estimation. To obtain estimate for the subjects on which the model estimation is based, the option "proba=TRUE" can be used in the mutation.rem function.

Usage

```
mutation.proba(para, likmat, xigrid = c(0, seq(0.0005, 0.9995, by=0.001)))
```

Arguments

`para`	Value of the model parameters, in the form c(mu, sigma, pi).
`likmat`	Grid of conditional probabilities for each tumor pair (rows) and each value of xi (columns). This matrix is generated by the auxiliary function grid.lik, and returned as a parameter by the principal function mutation.rem.
`xigrid`	Grid of the values of xi, corresponding to its domain of definition. The default is c(0, seq(0.0005, 0.9995, by=0.001)).

Value

Returns the vectors of probability of clonality for each pairs of tumors contained in the matrix likmat (the number of pairs is the number of rows of the matrix).

Author(s)

Audrey Mauguen <mauguena@mskcc.org> and Venkatraman E. Seshan.

References

Mauguen A, Seshan VE, Ostrovnaya I, Begg CB. Estimating the Probability of Clonal Relatedness of Pairs of Tumors in Cancer Patients. Submitted.

Examples

```
#___ Analysis of LCIS data
data(lcis)

#__ Parameters estimation
mod <- mutation.rem(lcis)
mod

#__ Probability of being clonal for a new subject
# generate a case with 30 mutations
# probabilities of each observed mutation
pi <- runif(30,0.001,0.13)
```

```
# mutation 1=shared or 2=private
newpair <- cbind(pi,rbinom(30,1,1-pi^2)+1)
# generate the matrix of likelihood values
new.likmat <- grid.lik(xigrid=c(0, seq(0.0005, 0.9995, by=0.001)), as.matrix(newpair[,c(-1)]), newpair[,1])
# probability of being clonal using the model previoulsy estimated
proba <- mutation.proba(c(mod$mu, mod$sigma, mod$pi), t(as.matrix(new.likmat)) )
```

`mutation.rem` *Estimation of the random-effect model for clonality based on mutations.*

Description

The model estimates the proportion of clonal cases in a population, and the distribution of the clonality signal.

Usage

```
mutation.rem(mutmat, proba=FALSE, sd.err = FALSE, print.proba=proba, print.sd.err=sd.err, xigrid
```

Arguments

`mutmat` Matrix containing the data, with all mutations in rows and the tumor pairs in columns. The data are coded as 0=mutation not observed, 1=shared mutation (observed in both tumors), 2=private mutation (observed in one tumor only). The first column contains the probabilities of occurence for each mutation.

`proba` Indicates whether to compute the individual probabilities of clonality for each pair. The default is FALSE.

`sd.err` Indicates whether to compute the standard errors of the estimated parameters. The default is FALSE.

`print.proba` Indicates whether the individual probabilities of clonality should be printed in the output. The default is TRUE if proba=TRUE and FALSE if proba=FALSE.

`print.sd.err` Indicates whether the the standard errors of the estimated parameters should be printed in the output. The default is TRUE if sd.err=TRUE and FALSE if sd.err=FALSE.

`xigrid` Grid of the values of xi used to compute the integration; it corresponds to the domain of definition of xi. The default is c(0, seq(0.0005, 0.9995, by=0.001)).

`init.para` Initial values of the parameters for the optimization. The order of the parameters is c(mu, sigma, pi), where mu and sigma are the mean and variance of the lognormal distribution of the random-effect xi, and pi is the proportion of clonal cases. The default is c(0,1,0.5).

Details

The function estimates a random effects model in which the random effect (the clonality signal, denoted xi_i for the ith case) reflects the somatic similarity of the tumors on a scale from 0 to 1, where 0 represents independence and higher values represent clonal tumors that are increasingly similar. The proportion of cases that are clonal is represented by the parameter pi. Thus the likelihood is a compound of (1-pi) cases that have a clonality signal of exactly 0, and pi cases that have a clonality signal drawn from a normal random effects distribution with mean mu and variance sigma^2. The program estimates all of the parameters and their variances using maximum likelihood. The output provides parameter estimates (mu, sigma, pi). The example dataset presented contains data from a study in which each patient has both a pre-malignant lobular carcinoma in situ (LCIS) and an invasive breast cancer, and we wish to estimate the proportion of these cases for which the LCIS was a direct precursor to the invasive cancer. The standard errors are computed using the inverse of minus the Hessian matrix.

Value

`mu` Estimated mean of the random-effect distribution.

`sigma` Estimated standard-deviation of the random-effect distribution.

`pi` Estimated proportion of clonal pairs in the population.

`likmat` Grid of likelihood values for each tumor pair (rows) and each value of xi (columns) needed for the function clonal.proba that computes the individual probabilities of clonality.

`likelihood` Value of the maximized likelihood.

`convergence` Convergence status (from the function optim).

`conv.message` Convergence message (from the function optim).

`se.mu` Standard error of the parameter mu.

`se.sigma` Standard error of the parameter sigma.

`se.pi` Standard error of the parameter pi.

`pr.clonal` Individual probabilities of clonality.

Author(s)

Audrey Mauguen <`mauguena@mskcc.org`> and Venkatraman E. Seshan.

References

Mauguen A, Seshan VE, Ostrovnaya I, Begg CB. Estimating the Probability of Clonal Relatedness of Pairs of Tumors in Cancer Patients. Submitted.

Examples

```
#___ Analysis of LCIS data
data(lcis)

#__ Parameters estimation
mod <- mutation.rem(lcis)
mod

#__ Parameters estimation with standard errors
mod <- mutation.rem(lcis, sd.err=TRUE)
mod

#__ Probability of being clonal
mod <- mutation.rem(lcis, proba=TRUE)
mod
```

`print.mutation.proba` *Print for the mutation.proba function*

Description

Print a summary of results for the probabilities of clonality estimated by the mutation.proba function.

Usage

```
## S3 method for class 'mutation.proba'
## S3 method for class 'mutation.proba'
print(x, ...)
```

Arguments

`x`	a mutation.proba object
`...`	Other unused arguments.

Value

Print results for the individual probabilities of clonality.

See Also

mutation.proba

`print.mutation.rem` *Print for the mutation.rem function*

Description

Print a summary of results for the random-effect model estimation estimated by the clonal.est function.

Usage

```
## S3 method for class 'mutation.rem'

## S3 method for class 'mutation.rem'
print(x, ...)
```

Arguments

`x`	a mutation.rem object
`...`	Other unused arguments.

Value

Print results for the model estimates.

See Also

mutation.rem

SNVtest — *Testing relatedness (clonality) of two tumors from the same patient using profiles of somatic mutations*

Description

Function to test clonality of two tumors from the same patient based on their mutational profiles. This function calculates conditional likelihood ratio relying only on loci where at least one of the tumors have a mutation, and p-values is calculated under the reference distribution under the hypothesis of independence.

Usage

```
SNVtest(tumor1, tumor2, pfreq, nrep = 1000)
```

Arguments

`tumor1` Vector of the binary mutation calls from tumor 1, where 0 denotes no mutation, 1 denotes a mutation. Mutations should be in the same order as frequencies in pfreq.

`tumor2` Vector of the binary mutation calls from tumor 2, where 0 denotes no mutation, 1 denotes a mutation. Mutations should be in the same order as frequencies in pfreq.

`pfreq` Marginal frequencies of mutations known a priori. These can be obtained from TCGA or similar databases. We recommend setting these frequencies to (x+y)(nx+ny), where x is the number of patients with the mutations in the TCGA(or other databases), and nx is the total number of the patients in TCGA; y and ny is number of patients with mutations and total number of patients in this study.

`nrep` Number of simulations used for generating the reference distribution under the hypothesis of independence.

Details

Only loci where at least one tumor has a mutation contribute to the model. The null distribution is patient specific since it is generated assuming the same total number of mutations in two tumors.

Value

The output is a vector with 5 values: c("n1","n2","n_match", "LRstat","maxKsi","LRpvalue")

`n1` Number of mutations in the first tumor.

`n2` Number of mutations in the second tumor

`n_match` Number of matches. i.e. loci where both tumors have an identical mutation

`LRstat` Likelihood ratio statistic

`maxKsi` Maximum likelhood estimate of Ksi, parameter of the likelihood representing clonality strength. Value close to 0 indicates independence, value close to 1 indicates perfect concordance in mutational profiles.

`LRpvalue` p-value calculated using the null distribution generated using prespecified mutational frequencies pfreq.

Author(s)

Irina Ostrovnaya <ostrovni@mskcc.org>

References

Ostrovnaya I, Seshan VE, Begg CB. "USING SOMATIC MUTATION DATA TO TEST TUMORS FOR CLONAL RELATEDNESS.", Ann Appl Stat. 2015 Sep;9(3):1533-1548

See Also

clonality.analysis() for test using genomewide copy number profiles; mutation.proba() for bayseian inference of clonality probability.

Examples

```
#___ Analysis of LCIS data from the following paper:
#Begg CB, Ostrovnaya I, Carniello JV, Sakr RA, Giri D, Towers R, Schizas M, De Brot M, Andrade VP, Mauguen A, S

data(lcis)
n<-nrow(lcis)

#Example of artificially generated independent tumor pair with marginal mutation frequencies lcis$probi

x1<-as.numeric(runif(n)<=lcis$probi)
x2<-as.numeric(runif(n)<=lcis$probi)
SNVtest(x1,x2,lcis$probi)


#Analysis of data from patient 47
table(lcis$TK47IDC.TK47LCIS1 )
#variable TK47IDC.TK47LCIS1  takes values 0 if mutation not observed, 1 if shared mutation (observed in both t

x1<-x2<-rep(0,n)
x1[lcis$TK47IDC.TK47LCIS1 ==1]<-x2[lcis$TK47IDC.TK47LCIS1 ==1]<-1
#we will assign private mutations to tumor 1 here since the likelihood doesn't depend on which tumor has the p
x1[lcis$TK47IDC.TK47LCIS1 ==2]<-1
SNVtest(x1,x2,lcis$probi)
```

splitChromosomes *Chromosome splitting*

Description

Divides the chromosomes into p and q arms.

Usage

```
splitChromosomes(chrom,maploc)
```

Arguments

chrom	Vector of chromosomes. They should be numeric 1 to 22.
maploc	Vector of genomic locations. They should be in Kilobases.

Details

The function returns the vector of chromosome arms labeled "chr01p", "chr01q", etc. The split into arms is accomplished using the following centers (in Kb) for chromosomes 1 through 22: (122356.96, 93189.90, 92037.54 , 50854.87 ,47941.40, 60438.12 , 59558.27, 45458.05 , 48607.50, 40434.94 , 52950.78, 35445.46 , 16934.00, 16570.00, 16760.00 , 36043.30 , 22237.13, 16082.90 , 28423.62 , 27150.40, 11760.00, 12830.00).

Examples

```
#simulated data

set.seed(100)
chrom<-rep(c(1:22),each=100)
maploc<- runif(2200)* 200000
chromarm<-splitChromosomes(chrom,maploc)
```

xidens *Auxiliary function computing the density of xi*

Description

Density function for the random variable xi, using a lognormal density for phi=-log(1-xi)

Usage

```
xidens(pmu, psig, xigrid)
```

Arguments

pmu	Mean parameter of the distribution.
psig	Variance parameter of the distribution.
xigrid	Grid of the values of xi, corresponding to its domain of definition.

Value

Returns the density value for the given values of xi.

Index