

Clonality

October 25, 2011

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:	0.99.2
Date:	2011-03-30
License:	GPL-3
LazyLoad:	yes

Author(s)

Irina Ostrovnaya <ostrovni@mskcc.org>

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)
```

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 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 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
- Ostrovskaya 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)
```

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 DNA-copy). 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()
#Analysis of paired breast samples from study
#Hwang ES, Nyante SF, Chen YY, Moore D, DeVries S, Korkola JE, Esserman LJ, and Waldman F
#Clonality of lobular carcinoma in situ and synchronous invasive lobular cancer. Cancer 1
#library(gdata)                   #needed to read .xls files
#library(DNAcopy)
#arrayinfo<-read.xls("http://waldman.ucsf.edu/Colon/nakao.data.xls") #needed to extract
#data<-read.xls("http://waldman.ucsf.edu/Breast/Hwang.data.xls")
#data<-data[!is.na(data[,2]),]
#data<-data[apply(is.na(data),1,sum)<=50,]
#data<-data[,apply(is.na(data),2,sum)<=1000]
#data$Position<-arrayinfo$Mb[match(toupper(as.character(data[,1])),toupper(as.character(a
#data<-data[!is.na(data$Position),]
#data<-data[c(TRUE,data$Position[-1]!=data$Position[-1864]),] #discard probes with repeat
```

```
#data<-data[data$Chromosome<=22,] #getting rid of X and Y chromosomes
#dataCNA<-CNA(as.matrix(data[,c(4:51)]),maploc=data$Position,chrom=data$Chromosome,sample
#The averaging is not needed. Used only as example.
#dataAve<- ave.adj.probes (dataCNA,2)
```

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

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

```
# Same example as in clonality.analysis()
#Analysis of paired breast samples from study
#Hwang ES, Nyante SF, Chen YY, Moore D, DeVries S, Korkola JE, Esserman LJ, and Waldman F
#Clonality of lobular carcinoma in situ and synchronous invasive lobular cancer. Cancer 1
#library(gdata) #needed to read .xls files
#library(DNAcopy)
#arrayinfo<-read.xls("http://waldman.ucsf.edu/Colon/nakao.data.xls") #needed to extract
#data<-read.xls("http://waldman.ucsf.edu/Breast/Hwang.data.xls")
#data<-data[!is.na(data[,2]),]
#data<-data[apply(is.na(data),1,sum)<=50,]
#data<-data[,apply(is.na(data),2,sum)<=1000]
#data$Position<-arrayinfo$Mb[match(toupper(as.character(data[,1])),toupper(as.character(a
#data<-data[!is.na(data$Position),]
#dim(data)
#length(unique(paste(data$Chromosome, data$Position))) #there are repeated genomic locati
#data<-data[c(TRUE,data$Position[-1]!=data$Position[-1864]),] #discard probes with repeat
#data<-data[data$Chromosome<=22,] #getting rid of X and Y chromosomes
```

```

#dataCNA<-CNA(as.matrix(data[,c(4:6,28:30)]),maploc=data$Position,chrom=data$Chromosome,s

#dataCNA$maploc<-dataCNA$maploc*1000 #transforming maploc to Kb scale
#dataCNA$chrom<- splitChromosomes( dataCNA$chrom,dataCNA$maploc) #splits the chromosomes

#ptlist<-substr(names(dataCNA)[-c(1,2)],1,4)
#samnms<-names(dataCNA)[-c(1,2)]

#results<-clonality.analysis(dataCNA, ptlist, pfreq = NULL, refdata = NULL, nmad = 1.25,
# reference = TRUE, allpairs = FALSE)

#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[
#}
#}
#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("pt",i,".pdf",sep=""),height=7,width=11)
#chromosomePlots(results$OneStepSeg, ptlist,ptname=i,nmad=1.25)
#dev.off()
#}

```

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, refe
```

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 level. 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.
segspar	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 along time.

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

Author(s)

Irina Ostrovnyaya <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 paired breast samples from study
#Hwang ES, Nyante SF, Chen YY, Moore D, DeVries S, Korkola JE, Esserman LJ, and Waldman F
#Clonality of lobular carcinoma in situ and synchronous invasive lobular cancer. Cancer 1
library(gdata)      #needed to read .xls files
library(DNAcopy)
arrayinfo<-read.xls("http://waldman.ucsf.edu/Colon/nakao.data.xls") #needed to extract
data<-read.xls("http://waldman.ucsf.edu/Breast/Hwang.data.xls")
data<-data[!is.na(data[,2]),]
data<-data[apply(is.na(data),1,sum)<=50,]
data<-data[,apply(is.na(data),2,sum)<=1000]
data$Position<-arrayinfo$Mb[match(toupper(as.character(data[,1])),toupper(as.character(ar
data<-data[!is.na(data$Position),]
dim(data)
length(unique(paste(data$Chromosome, data$Position))) #there are repeated genomic locatio
data<-data[c(TRUE,data$Position[-1]!=data$Position[-1864]),] #discard probes with repeate
data<-data[data$Chromosome<=22,] #getting rid of X and Y chromosomes
dataCNA<-CNA(as.matrix(data[,c(4:6,28:30)]),maploc=data$Position,chrom=data$Chromosome,sa

dataCNA$maploc<-dataCNA$maploc*1000 #transforming maploc to Kb scale
dataCNA$chrom<- splitChromosomes( dataCNA$chrom,dataCNA$maploc) #splits the chromosomes

ptlist<-substr(names(dataCNA)[-c(1,2)],1,4)
samnms<-names(dataCNA)[-c(1,2)]

results<-clonality.analysis(dataCNA, ptlist, pfreq = NULL, refdata = NULL, nmad = 1.25,
  reference = TRUE, allpairs = FALSE)

#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[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("pt",i,".pdf",sep=""),height=7,width=11)
chromosomePlots(results$OneStepSeg, ptlist,ptname=i,nmad=1.25)
dev.off()
}
```

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 =
```

Arguments

<code>data.seg1</code>	Output of one-step segmentation - output <code>OneStepSeg</code> of <code>clonality.analysis()</code> . The chromosomes should be in the format "chr01p", "chr01q" etc.
<code>classall</code>	Classifications of the chromosomes - output <code>ChromClass</code> of <code>clonality.analysis()</code>
<code>ptlist</code>	Vector of the patient IDs in the order of the samples appearing in the data.
<code>ptpair</code>	Two sample names for which the plot is desired
<code>ptLR</code>	Matrix with the likelihood ratios - output <code>LR</code> of <code>clonality.analysis()</code>
<code>plot.as.in.analysis</code>	

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

```

# Same example as in clonality.analysis()
#Analysis of paired breast samples from study
#Hwang ES, Nyante SF, Chen YY, Moore D, DeVries S, Korkola JE, Esserman LJ, and Waldman F
#Clonality of lobular carcinoma in situ and synchronous invasive lobular cancer. Cancer 1
#library(gdata) #needed to read .xls files
#library(DNAcopy)
#arrayinfo<-read.xls("http://waldman.ucsf.edu/Colon/nakao.data.xls") #needed to extract
#data<-read.xls("http://waldman.ucsf.edu/Breast/Hwang.data.xls")
#data<-data[!is.na(data[,2]),]
#data<-data[apply(is.na(data),1,sum)<=50,]
#data<-data[,apply(is.na(data),2,sum)<=1000]
#data$Position<-arrayinfo$Mb[match(toupper(as.character(data[,1])),toupper(as.character(a
#data<-data[!is.na(data$Position),]
#dim(data)
#length(unique(paste(data$Chromosome, data$Position))) #there are repeated genomic locati
#data<-data[c(TRUE,data$Position[-1]!=data$Position[-1864]),] #discard probes with repeat
#data<-data[data$Chromosome<=22,] #getting rid of X and Y chromosomes
#dataCNA<-CNA(as.matrix(data[,c(4:6,28:30)]),maploc=data$Position,chrom=data$Chromosome,s

#dataCNA$maploc<-dataCNA$maploc*1000 #transforming maploc to Kb scale
#dataCNA$chrom<- splitChromosomes( dataCNA$chrom,dataCNA$maploc) #splits the chromosomes

#ptlist<-substr(names(dataCNA)[-c(1,2)],1,4)
#samnms<-names(dataCNA)[-c(1,2)]

#results<-clonality.analysis(dataCNA, ptlist, pfreq = NULL, refdata = NULL, nmad = 1.25,
# reference = TRUE, allpairs = FALSE)

#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[
#}
#}
#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("pt",i,".pdf",sep=""),height=7,width=11)
#chromosomePlots(results$OneStepSeg, ptlist,ptname=i,nmad=1.25)

```

```
#dev.off()
#}
```

histogramPlot *Histograms 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

```
# Same example as in clonality.analysis()
#Analysis of paired breast samples from study
#Hwang ES, Nyante SF, Chen YY, Moore D, DeVries S, Korkola JE, Esserman LJ, and Waldman F
#Clonality of lobular carcinoma in situ and synchronous invasive lobular cancer. Cancer 1
#library(gdata) #needed to read .xls files
#library(DNAcopy)
#arrayinfo<-read.xls("http://waldman.ucsf.edu/Colon/nakao.data.xls") #needed to extract
#data<-read.xls("http://waldman.ucsf.edu/Breast/Hwang.data.xls")
#data<-data[!is.na(data[,2]),]
#data<-data[apply(is.na(data),1,sum)<=50,]
#data<-data[,apply(is.na(data),2,sum)<=1000]
#data$Position<-arrayinfo$Mb[match(toupper(as.character(data[,1])),toupper(as.character(a
#data<-data[!is.na(data$Position),]
#dim(data)
#length(unique(paste(data$Chromosome, data$Position))) #there are repeated genomic locati
#data<-data[c(TRUE,data$Position[-1]!=data$Position[-1864]),] #discard probes with repeat
#data<-data[data$Chromosome<=22,] #getting rid of X and Y chromosomes
#dataCNA<-CNA(as.matrix(data[,c(4:6,28:30)]),maploc=data$Position,chrom=data$Chromosome,s

#dataCNA$maploc<-dataCNA$maploc*1000 #transforming maploc to Kb scale
#dataCNA$chrom<- splitChromosomes( dataCNA$chrom,dataCNA$maploc) #splits the chromosomes

#ptlist<-substr(names(dataCNA)[-c(1,2)],1,4)
#samnms<-names(dataCNA)[-c(1,2)]
```

```

#results<-clonality.analysis(dataCNA, ptlist, pfreq = NULL, refdata = NULL, nmad = 1.25,
# reference = TRUE, allpairs = FALSE)

#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])])
#}
#}
#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("pt",i,".pdf",sep=""),height=7,width=11)
#chromosomePlots(results$OneStepSeg, ptlist,ptname=i,nmad=1.25)
#dev.off()
#}

```

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.

Examples

```

# Same example as in clonality.analysis()
#Analysis of paired breast samples from study
#Hwang ES, Nyante SF, Chen YY, Moore D, DeVries S, Korkola JE, Esserman LJ, and Waldman F
#Clonality of lobular carcinoma in situ and synchronous invasive lobular cancer. Cancer 1
#library(gdata) #needed to read .xls files
#library(DNAcopy)
#arrayinfo<-read.xls("http://waldman.ucsf.edu/Colon/nakao.data.xls") #needed to extract
#data<-read.xls("http://waldman.ucsf.edu/Breast/Hwang.data.xls")
#data<-data[!is.na(data[,2]),]
#data<-data[apply(is.na(data),1,sum)<=50,]
#data<-data[,apply(is.na(data),2,sum)<=1000]
#data$Position<-arrayinfo$Mb[match(toupper(as.character(data[,1])),toupper(as.character(a
#data<-data[!is.na(data$Position),]
#dim(data)
#length(unique(paste(data$Chromosome, data$Position))) #there are repeated genomic locati
#data<-data[c(TRUE,data$Position[-1]!=data$Position[-1864]),] #discard probes with repeat
#data<-data[data$Chromosome<=22,] #getting rid of X and Y chromosomes
#dataCNA<-CNA(as.matrix(data[,c(4:6,28:30)]),maploc=data$Position,chrom=data$Chromosome,s

#dataCNA$maploc<-dataCNA$maploc*1000 #transforming maploc to Kb scale
#dataCNA$chrom<- splitChromosomes( dataCNA$chrom,dataCNA$maploc) #splits the chromosomes

#ptlist<-substr(names(dataCNA)[-c(1,2)],1,4)
#samnms<-names(dataCNA)[-c(1,2)]

#results<-clonality.analysis(dataCNA, ptlist, pfreq = NULL, refdata = NULL, nmad = 1.25,
# reference = TRUE, allpairs = FALSE)

#genomewide plots of pairs of tumors from the same patient
#pdf(".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[
#}
#}
#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("pt",i,".pdf",sep=""),height=7,width=11)
#chromosomePlots(results$OneStepSeg, ptlist,ptname=i,nmad=1.25)

```

```
#dev.off()  
#}
```

Index

ave.adj.probes, [3](#)
chromosomePlots, [4](#)
Clonality (*Clonality-package*), [1](#)
Clonality-package, [1](#)
clonality.analysis, [5](#)
genomewidePlots, [9](#)
histogramPlot, [11](#)
LOHclonality, [1](#)
splitChromosomes, [12](#)