Package 'ExomeCNV'

February 15, 2013

Type Package

Title Detect CNV and LOH from Exome Sequencing Data

Version 1.4

Date 2012-08-24

Author J. Fah Sathirapongsasuti, Hane Lee, and Stanley F. Nelson

Maintainer J. Fah Sathirapongsasuti <fsathira@fas.harvard.edu>

Description ExomeCNV is a statistical method to detect CNV and LOH using depth-of-coverage and B-allele frequencies from mapped short sequence reads in exome sequencing data.

License LGPL-2.1

LazyLoad yes

Depends DNAcopy

Repository CRAN

Date/Publication 2012-08-25 15:07:22

NeedsCompilation no

R topics documented:

`ExomeCNV-package` *Exome Sequencing-Based CNV and LOH Detection.*

Description

ExomeCNV is an R package tailored to detection of CNV (Copy-Number Variants) and LOH (Loss of Heterozygosity) from exome sequencing data. It exploits the unique discrete feature of exon definitions and incredible cross-sample consistency of depth-of-coverage. ExomeCNV is most suitable when paired samples (e.g. tumor-normal pair) are available. Both of the paired samples should be processed and sequenced in a similar manner (e.g. same library prep, sequencer, average depth-of-coverage, etc.).

Details

Package:	ExomeCNV
Type:	Package
Version:	1.0
Date:	2011-01-27
License:	LGPL-2.1
LazyLoad:	yes

See user guide at https://secure.genome.ucla.edu/index.php/ExomeCNV_User_Guide

Author(s)

Fah Sathirapongsasuti <fsathira@fas.harvard.edu>

References

https://secure.genome.ucla.edu/index.php/ExomeCNV_User_Guide

Examples

```
chr.list=paste("chr",c("19","20","21"), sep="")

suffix = ".small.coverage"

prefix = "http://genome.ucla.edu/~fah/ExomeCNV/data/normal."
normal = read.all.coverage(prefix, suffix, chr.list, header=TRUE)

prefix = "http://genome.ucla.edu/~fah/ExomeCNV/data/tumor."
tumor = read.all.coverage(prefix, suffix, chr.list, header=TRUE)

demo.logR = calculate.logR(normal, tumor)

demo.eCNV = c()
for (i in 1:length(chr.list)) {
idx = (normal$chr == chr.list[i])
ecnv = classify.eCNV(normal=normal[idx,], tumor=tumor[idx,],
logR=demo.logR[idx], min.spec=0.9999, min.sens=0.9999,
option="spec", c=0.5, l=70)
demo.eCNV = rbind(demo.eCNV, ecnv)
}
```

`calculate.logR` *Calculate log ratio of depth of coverage between paired samples.*

Description

Calculate log ratio of depth of coverage between paired samples (e.g. tumor/normal). This is to be used as input for classify.logR(). The ratios are normalize by the total number of reads and adjusted so that median log ratio of exons on "normal" chromosomes is zero. Normal chromosome is defined by input `normal.chrs`.

Usage

```
calculate.logR(normal, tumor, normal.chrs = c("chr1", "chr2", "chr3",
                  "chr4", "chr5", "chr6", "chr7", "chr8", "chr9",
                  "chr10", "chr11", "chr12", "chr13", "chr14", "chr15",
                  "chr16", "chr17", "chr18", "chr19", "chr20", "chr21",
                  "chr22", "chrX", "chrY"))
```

Arguments

`normal`	a data.frame of depth of coverage of normal (control) sample. See `read.all.coverage()` for more information.
`tumor`	a data.frame of depth of coverage of tumor (case) sample. See `read.all.coverage()` for more information.
`normal.chrs`	a vector of strings indicating chromosomes that are believed to have normal copy numbers. Default to whole genome.

Value

a vector of log ratios.

Author(s)

Fah Sathirapongsasuti <fsathira@fas.harvard.edu>

References

https://secure.genome.ucla.edu/index.php/ExomeCNV_User_Guide

See Also

read.all.coverage

Examples

```
chr.list=c("chr19","chr20","chr21")
suffix = ".small.coverage"
prefix = "http://genome.ucla.edu/~fah/ExomeCNV/data/normal."
normal = read.all.coverage(prefix, suffix, chr.list, header=TRUE)
prefix = "http://genome.ucla.edu/~fah/ExomeCNV/data/tumor."
tumor = read.all.coverage(prefix, suffix, chr.list, header=TRUE)

demo.logR = calculate.logR(normal, tumor)
```

---

chr.hash *a data.frame of chromosome names to hash to numeric representation.*

Description

a table of chromosome name and their corresponding numerical representation. This is needed because the way DNAcopy encodes the chromosomes.

Usage

```
data(chr.hash)
```

Format

A data frame with 24 observations on the following 2 variables.

`chr` a factor with levels `chr1` `chr10` `chr11` `chr12` `chr13` `chr14` `chr15` `chr16` `chr17` `chr18` `chr19` `chr2` `chr20` `chr21` `chr22` `chr3` `chr4` `chr5` `chr6` `chr7` `chr8` `chr9` `chrX` `chrY`

`number` a numeric vector

References

https://secure.genome.ucla.edu/index.php/ExomeCNV_User_Guide

Examples

```
data(chr.hash)
```

`classify.eCNV` *Call CNV on each exon based on log ratio of read depth.*

Description

Calculate specificity and sensitivity (power) of detecting CNV based on depth of coverage and log ratio of all exons. Make a call when sufficient specificity and sensitivity are achieved.

Usage

```
classify.eCNV(normal, tumor, logR = NULL, min.spec = 0.9, min.sens = 0.9,
      option = "auc", admix = 0.3, c = admix, read.len = 70,
      l = read.len, normal.chrs = c("chr1",
     "chr2","chr3","chr4","chr5","chr6","chr7","chr8",
     "chr9","chr10","chr11","chr12","chr13","chr14",
     "chr15","chr16","chr17","chr18","chr19","chr20",
     "chr21","chr22","chrX","chrY"),
     test.num.copy = c(1,3))
```

Arguments

`normal`	a data.frame of depth of coverage of normal (control) sample. See `read.all.coverage()` for more information.
`tumor`	a data.frame of depth of coverage of tumor (case) sample. See `read.all.coverage()` for more information.
`logR`	a vector of log ratio as calculated by `calculate.logR`.
`min.spec`	desired minimum specificity.
`min.sens`	desired minimum sensitivity (power).
`option`	objective quantity to optimize over when minimum sensitivity and specificity are achieved. Possible opetions are `sens` for sensitivity, `spec` for specificity, `auc` for area under curve = (specificity + sensitivity)/2.
`admix`	contamination rate (admixture rate), the proportion of the normal cells in the tumor samples.
`c`	(depreciated) same as `admix`
`read.len`	sequence read length.
`l`	(depreciated) same as `read.len`

normal.chrs	a vector of strings indicating chromosomes that are believed to have normal copy numbers. Default to whole genome.
test.num.copy	copy numbers to be tested. 1 for deletion, 3 for duplication, 4 and beyond for amplification. Default to (1,3,4,5).

Details

This is the main function to call CNV at exon level. It first computes power based on depth of coverage of the exon. With sufficient power and specificity, a CNV call is made based on the log ratio of depth of coverage while optimizing for specificity, sensitivity or AUC (depending on the option set by user). Power calculation and CNV calling are based on the ratio of normal random variables.

Value

a data.frame with collowing fields:

tumor.average.coverage	average coverage of tumor exon
logR	log ratio of depth of coverage
ratio	ratio of of depth of coverage
copy.number	CNV call. 1 = deletion, 2 = normal, 3 and more = amplification
lower.cutoff	cutoff value used to call the CNV
upper.cutoff	same as lower.cutoff
spec	predicted specificity associated with the CNV call.
sens	predicted sensitivity (power) associated with the CNV call.

Author(s)

Fah Sathirapongsasuti <fsathira@fas.harvard.edu>

References

https://secure.genome.ucla.edu/index.php/ExomeCNV_User_Guide

See Also

calculate.logR

Examples

```
normal = read.table("http://genome.ucla.edu/~fah/ExomeCNV/data/normal.chr21.small.coverage",
header=TRUE, sep='\t')
tumor = read.table("http://genome.ucla.edu/~fah/ExomeCNV/data/tumor.chr21.small.coverage",
header=TRUE, sep='\t')
con = url("http://genome.ucla.edu/~fah/ExomeCNV/demo/logR/demo.small.logR.chr21.RData")
load(con)
close(con)
```

```
ecnv = classify.eCNV(normal, tumor, logR, min.spec=0.9999, min.sens=0.9999,
option="spec", c=0.5, l=70)
```

CNV.analyze *A wrapper function of DNAcopy that prepare input and run CBS.*

Description

Preprocess the data and run DNAcopy (to do Circular Binary Segmentation). It also offers an option to plot log ratios without performing DNAcopy.

Usage

```
CNV.analyze(normal, tumor, logR = NULL, coverage.cutoff = 15,
                 normal.chrs = c("chr1", "chr2", "chr3", "chr4", "chr5",
                 "chr6", "chr7", "chr8", "chr9", "chr10", "chr11",
                 "chr12", "chr13", "chr14", "chr15", "chr16", "chr17",
                 "chr18", "chr19", "chr20", "chr21", "chr22", "chrX",
                 "chrY"), normal.chr = normal.chrs, c = 0.5, write.file = FALSE,
                 file = NULL, doDNAcopy = TRUE, sdundo = 1, smooth = TRUE,
                 alpha = 0.01, plot.cnv = TRUE)
```

Arguments

normal	a data.frame of depth of coverage of normal (control) sample. See `read.all.coverage()` for more information.
tumor	a data.frame of depth of coverage of tumor (case) sample. See `read.all.coverage()` for more information.
logR	a vector of log ratio as calculated by `calculate.logR`.
coverage.cutoff	a hard cutoff to exclude exons with low depth of coverage from consideration.
normal.chrs	a vector of strings indicating chromosomes that are believed to have normal copy numbers. Default to whole genome.
normal.chr	degenerous. same as `normal.chrs` above.
c	contamination rate (admixture rate), the proportion of the normal cells in the tumor samples.
write.file	Boolean option to write the result of DNAcopy to file.
file	File name of the result of DNAcopy to be written.
doDNAcopy	Boolean option to do DNAcopy. If FALSE, it will plot log ratios without doing Circular Binary Segmentation.
sdundo	Option to pass on to DNAcopy. See `segment` in Package DNAcopy for more details.

smooth	Option to pass on to DNAcopy. See segment in Package DNAcopy for more details.
alpha	Option to pass on to DNAcopy. See segment in Package DNAcopy for more details.
plot.cnv	Boolean option to plot the results.

Details

The function first calculates log ratios using calculate.logR then call segment (from Package DNAcopy) using exon midpoints as the probe positions.

Value

cnv	A data.frame with the following fields: chr chromosome on which the exon is located; probe probe name; probe_start starting position of the exon; probe_end ending position of the exon; size size of the exon; targeted.base the number of bases targeted by exome sequencing; sequenced.base the number of bases sequenced (mapped); coverage total number of bases sequenced; average.coverage average depth of coverage of the exon; base.with..10.coverage the number of bases in the exon with >= 10 fold coverage
cna	output from segment, can be used to plot. See segment in Package DNAcopy for more details.
logR	a vector of log ratios returned from calculate.logR

Author(s)

Fah Sathirapongsasuti <fsathira@fas.harvard.edu>

References

https://secure.genome.ucla.edu/index.php/ExomeCNV_User_Guide

See Also

calculate.logR

Examples

```
data(chr.hash)
normal = read.table("http://genome.ucla.edu/~fah/ExomeCNV/data/normal.chr19.small.coverage",
header=TRUE, sep='\t')
tumor = read.table("http://genome.ucla.edu/~fah/ExomeCNV/data/tumor.chr19.small.coverage",
header=TRUE, sep='\t')
con = url("http://genome.ucla.edu/~fah/ExomeCNV/demo/logR/demo.small.logR.chr19.RData")
load(con)
close(con)
cnv = CNV.analyze(normal, tumor, logR = logR, coverage.cutoff = 15,
normal.chrs = row.names(chr.hash), c = 0.5, write.file = FALSE,
file = NULL, doDNAcopy = TRUE, sdundo = 1, smooth = TRUE,
alpha = 0.01, plot.cnv = TRUE)
```

colors *Constants used in the package.*

Description

A collection of all data and constants used/available in ExomeCNV.

Usage

```
data(colors)
```

Format

The format is: int [1:8] 2 3 4 5 6 7 8 9

References

https://secure.genome.ucla.edu/index.php/ExomeCNV_User_Guide

Examples

```
data(colors)
```

combine.CNV *Combine results of DNAcopy and classify.eCNV into one table.*

Description

combine results of DNAcopy and classify.eCNV into one table

Usage

```
combine.CNV(cnv.ls)
```

Arguments

cnv.ls	list of results from classify.eCNV, ranked by coarseness (finest to coarsest)

Details

algorithm: start from the finest classification (usually exon-level eCNV) for each of the coarser classification for each cnv interval match with exon in its range if the copy number is not classified (NA) or match assign/change copy# and log ratio value merge adjacent exons with same log ratio and copy number classification

Author(s)

Fah Sathirapongsasuti <fsathira@fas.harvard.edu>

References

https://secure.genome.ucla.edu/index.php/ExomeCNV_User_Guide

Examples

```
## this function is only for internal use
```

---

do.plot.eCNV *Plot CNV calls over log ratio.*

Description

Plot results of `classify.eCNV()` or `multi.CNV.analyze()`.

Usage

```
do.plot.eCNV(all.ecnv, pch = "*", lim.quantile = 0.99, style = "idx",
             bg.cnv = NULL, line.plot = FALSE)
```

Arguments

`all.ecnv`	Result of `classify.eCNV()` or `multi.CNV.analyze()`
`pch`	Character to be used for plotting one data point.
`lim.quantile`	Percentage of all data to display. If set to 1, display every data point (which can be highly skewed by the outliers).
`style`	What to use on the x-axis. The two options are: `idx` for exon idices and `bp` for actual base-pair coordinate of the exons.
`bg.cnv`	Raw data to plot as background for interval plot (`line.plot=TRUE`). This should be in data.frame that has "chr", "probe_end", and "logR".
`line.plot`	Boolean option to plot CNV as line.

Author(s)

Fah Sathirapongsasuti <fsathira@fas.harvard.edu>

References

https://secure.genome.ucla.edu/index.php/ExomeCNV_User_Guide

Examples

```
chr.list = paste("chr",c("19","20","21"), sep="")
suffix = ".RData"
prefix = paste("http://genome.ucla.edu/~fah/ExomeCNV/demo/",
"demo.eCNV.9999.9999.spec.c.5/demo.eCNV.9999.9999.spec.c.5.",
sep="")
demo.eCNV = read.eCNV(prefix, suffix, chr.list, url=TRUE)
do.plot.eCNV(demo.eCNV, lim.quantile=0.99, style="idx", line.plot=FALSE)

chr.list=paste("chr",c("19","20","21"), sep="")
suffix = ".small.coverage"
prefix = "http://genome.ucla.edu/~fah/ExomeCNV/data/normal."
normal = read.all.coverage(prefix, suffix, chr.list, header=TRUE)
prefix = "http://genome.ucla.edu/~fah/ExomeCNV/data/tumor."
tumor = read.all.coverage(prefix, suffix, chr.list, header=TRUE)
demo.logR = calculate.logR(normal, tumor)

## The following will take a while to run (~3-5 mins)
demo.cnv = multi.CNV.analyze(normal, tumor, logR=demo.logR, all.cnv.ls=NULL,
coverage.cutoff=5, min.spec=0.99, min.sens=0.99, option="auc",
c=0.5, sdundo=c(2), alpha=c(0.01))
do.plot.eCNV(demo.cnv, lim.quantile=0.99, style="bp", bg.cnv=demo.eCNV, line.plot=TRUE)
```

`do.plot.loh` *Plot output of LOH calls.*

Description

Plot results of `LOH.classify()` or `multi.LOH.analyze()` over the background of BAF or deviation of BAF.

Usage

```
do.plot.loh(the.loh, normal, tumor, method, lim.quantile = 0.99,
color = "red", plot.style = c("dev", "baf"))
```

Arguments

`the.loh`	LOH calls as a result of `multi.LOH.analyze()`
`normal`	BAF information for all heterozygous positions in the normal exome. Same as the input for `LOH.classify`.
`tumor`	BAF information for all heterozygous positions in the tumor exome. Same as the input for `LOH.classify`.
`method`	Method used to call the LOH in `LOH.classify` and `multi.LOH.analyze()`.
`lim.quantile`	Percentage of all data to display. If set to 1, display every data point (which can be highly skewed by the outliers).
`color`	Color string to signify LOH region.

plot.style	The style of background BAF information. If plot.style is "dev", uses |baf.tumor - baf.normal|; if "baf", just plot baf.normal (gray) and baf.tumor (blue).

Author(s)

Fah Sathirapongsasuti <fsathira@fas.harvard.edu>

References

https://secure.genome.ucla.edu/index.php/ExomeCNV_User_Guide

See Also

do.plot.eCNV

Examples

```
normal = read.delim("http://genome.ucla.edu/~fah/ExomeCNV/data/normal.small.baf.txt",
header=TRUE)
tumor = read.delim("http://genome.ucla.edu/~fah/ExomeCNV/data/tumor.small.baf.txt",
header=TRUE)
eLOH = LOH.analyze(normal, tumor, alpha=0.05, method="two.sample.fisher")
con = url("http://genome.ucla.edu/~fah/ExomeCNV/demo/demo.the.loh.RData")
load(con)
close(con)
do.plot.loh(the.loh, normal, tumor, "two.sample.fisher", plot.style="baf")
```

---

expand.loh *Propagate LOH calls to all heterozygous positions within each LOH segment.*

Description

LOH calls are done on (large) genomic intervals; this function help reassign those LOH calls to individual positions within in the interval, in case user wants LOH calls on each position (for example, when counting the number of heterozygous positions with LOH).

Usage

```
expand.loh(the.loh, data)
```

Arguments

the.loh	Result of multi.LOH.analyze
data	A data.frame of BAF data at all heterozygous positions. This can be the same as the input for LOH.analyze

Author(s)

Fah Sathirapongsasuti <fsathira@fas.harvard.edu>

References

https://secure.genome.ucla.edu/index.php/ExomeCNV_User_Guide

Examples

```
normal = read.delim("http://genome.ucla.edu/~fah/ExomeCNV/data/normal.baf.txt",
header=TRUE)
con = url("http://genome.ucla.edu/~fah/ExomeCNV/demo/demo.the.loh.RData")
load(con)
close(con)
expanded.loh = expand.loh(the.loh, normal)
```

---

get.AUC *Calculate theoretical specificity, sensitivity, area under curve (AUC).*

Description

Calculate theoretical specificity, sensitivity, area under curve (AUC) given copy number ratio, window size, and sequence read length. These are used internally.

Usage

```
get.AUC(chi, r, W, l, rho, overdisperse = "no", phi = 1, od.alpha = 0)
```

Arguments

chi	x-coverage, number of sequenced bases at one position
r	cutoff ratio used to call amplification/deletion
W	window size in bp
l	read length in bp
rho	true copy number ratio e.g. 1.5 for 1 copy gain, 0.5 for 1 copy loss
overdisperse	overdispersion model: "no" – no overdispersion, var = mu "quasi-likelihood" or "ql" – quasi-likelihood where var = phi*mu "negative binomial" or "nb" – negative binomial where var = mu + alpha*mu^2
phi	overdispersion constant as modeled by quasi-likelihood approach where var = phi*mu
od.alpha	overdispersion constant as modeled by negative binomial approach where var = mu + alpha*mu^2

Author(s)

Fah Sathirapongsasuti <fsathira@fas.harvard.edu>

References

https://secure.genome.ucla.edu/index.php/ExomeCNV_User_Guide

Examples

```
get.AUC(chi=35, r=1.4, W=500, l=70, rho=1.5, overdisperse="no")
```

`guesstimate.contamination` *Estimate admixture/contamination rate in a sample.*

Description

Admixture/contarmination here refers to the DNA content with normal copy number that is present in sample with copy number variation. An example is the non-cancer (normal) tissue found in a tumor biopsy sample. The approximation is made through deviation of log coverage ratio from zero in a region with evidence for LOH or deletion.

Usage

```
guesstimate.contarmination(logR, region.idx = NULL)
```

Arguments

`logR`	Log coverage ratio as calculated by `calculate.logR`
`region.idx`	A vector of indices or logical values indicating exons with evidence for LOH/deletion.

Author(s)

Fah Sathirapongsasuti <fsathira@fas.harvard.edu>

References

https://secure.genome.ucla.edu/index.php/ExomeCNV_User_Guide

Examples

```
normal = read.table("http://genome.ucla.edu/~fah/ExomeCNV/data/normal.chr21.coverage",
header=TRUE, sep='\t')
con = url("http://genome.ucla.edu/~fah/ExomeCNV/demo/logR/demo.logR.chr21.RData")
load(con)
close(con)
admix.rate = guesstimate.contamination( logR=logR, region.idx=(normal$chr == "chr21") )
```

LOH.analyze *Call LOH on each heterozygous position using specified test statistic.*

Description

Call LoH from BAF using specified test. It returns a vector of T/F indicating LOH status. If it runs CMH test, strata is required as well as normal or tumor with each row corresponding to each stratum.

Usage

```
LOH.analyze(normal = NULL, tumor = NULL, strata = NULL, alpha = 0.05,
method = c("deviation.half.norm", "variance.f", "deviation.wilcox",
"deviation.t", "CMH", "mantelhaen", "two.sample.fisher",
"two.sample.prop", "only.tumor", "only.normal") )
```

Arguments

normal	BAF information for all heterozygous positions in the normal exome. It should be a data.frame with four fields: chr, position, coverage, and baf "baf" here doesn't refer to frequency but the raw count. Example can be found at https://secure.genome.ucla.edu/inde
tumor	Same as normal. If method is "only.tumor" or "only.normal", only one of tumor or normal is required, respectively.
strata	A list of 2x2xN matrices, each corresponds to a segment. N is the number of heterozygous positions in a segment. Use make.loh.strata.
alpha	Type I error rate used in the exonic LOH test.
method	Type of test to use to call LOH. See details.

Details

Statistical tests that can be used in calling LOH are based on three test statistics: # BAF as count statistic # Variance of BAF, reflecting the amount of deviation of BAF away from its central value (~0.5) # Absolute deviation of BAF from the null value of 0.5 # Difference between BAF's in case and control samples

Each test statistic allows for different tests and is based on different assumptions.

Options "only.tumor" and "only.normal" use only one sample (case or control) to perform binomial test against null p=0.5. We can model LOH as a binomial event, asking among N reads mapped to the position, how likely is it to observe a certain number of B-allele (BAF).

Options "two.sample.fisher" and "two.sample.prop" are similar to the binomial test for one sample above but instead of testing the observed proportion against the null value of 0.5, they compare the observed proportion between case and control. This can be modeled by binomial distribution (two.sample.prop) or hypergeometric distribution (Fisher's exact test; two.sample.fisher), hence the two possible tests.

Option "variance.f" performs F-test to compare variances of case and control BAF's

Options "deviation.wilcox" and "deviation.t" perform t-test and Wilcoxon Rank Sum (Mann-Whitney) Test, respectively. This is to compare the mean value of the absolute deviation of BAF from 0.5 (i.e. |BAF - 0.5|).

Option "deviation.half.norm" is based on the observation that the distribution of BAF difference between case and control are normally distributed around 0. Thus the absolute value follows folded-normal distribution. Under LOH, the absolute difference will have a higher mean value, and we can measure and test the increase in the difference using half-normal distribution.

Option "CMH" or "mantelhaen" uses Cochran-Mantel-Haenszel Chi-sq test for common odds ratio equal to 1. It requires that the number of stata N >= 2. In case N = 1, it is equivalent to Pearson's Chi-sq (prop.test). This is useful when trying to call LOH for segments, which contain multiple heterozygous positions, each with its own contigency table. The only problem with this test is that it requires phasing information, which does not always exist. Thus it is not recommended for use.

Author(s)

Fah Sathirapongsasuti <fsathira@fas.harvard.edu>

References

https://secure.genome.ucla.edu/index.php/ExomeCNV_User_Guide

Examples

```
normal = read.delim("http://genome.ucla.edu/~fah/ExomeCNV/data/normal.small.baf.txt",
header=TRUE)
tumor = read.delim("http://genome.ucla.edu/~fah/ExomeCNV/data/tumor.small.baf.txt",
header=TRUE)
eLOH = LOH.analyze(normal, tumor, alpha=0.05, method="two.sample.fisher")
```

`multi.CNV.analyze` *Use Circular Binary Segmentation to create segments and call CNV on each segment.*

Description

Run `CNV.analyze` under configurations specified in `sdundo` and `alpha`. `sdundo` and `alpha` have to be of the same length. A list of other finer CNV interval may be supplied in `all.cnv.ls`. Finally it merges all results to form a final CNV interval list.

Usage

```
multi.CNV.analyze(normal, tumor, logR = NULL, all.cnv.ls = NULL,
  coverage.cutoff = 10, admix = 0.3, c = admix, read.len = 70,
  l = read.len, sdundo = c(1, 2), alpha = c(0.05, 0.01),
  min.spec = 0.99, min.sens = 0.9, option = "auc")
```

Arguments

normal	a data.frame of depth of coverage of normal (control) sample. See read.all.coverage() for more information.
tumor	a data.frame of depth of coverage of tumor (case) sample. See read.all.coverage() for more information.
logR	a vector of log ratio as calculated by calculate.logR.
all.cnv.ls	a list of other finer CNV interval (usually the outputs of classify.eCNV)
coverage.cutoff	a number to use as a cutoff for minimum average coverage that should be considered. If this minimum coverage is not met, CNV will not be called for that segment.
admix	contamination rate (admixture rate), the proportion of the normal cells in the tumor samples.
c	(depreciated) same as admix
read.len	sequence read length.
l	(depreciated) same as read.len
sdundo	the number of SDs between means to keep a split. This is a parameter for CBS as used in DNAcopy package.
alpha	significance levels for the test to accept change-points. This is a parameter for CBS as used in DNAcopy package.
min.spec	desired minimum specificity.
min.sens	desired minimum sensitivity (power).
option	objective quantity to optimize over when minimum sensitivity and specificity are achieved. Possible opetions are sens for sensitivity, spec for specificity, auc for area under curve = (specificity + sensitivity)/2.

Details

This is a wrapper function for classify.eCNV. It first segments the genome (exome) using CBS with variable fineness levels (as specified by sdundo and alpha). Then go through each segment and call CNV. Finally, merge all intervals from finest level (exon) to coarest level, prioritizing finer level to improve sensitivity.

Author(s)

Fah Sathirapongsasuti <fsathira@fas.harvard.edu>

References

https://secure.genome.ucla.edu/index.php/ExomeCNV_User_Guide

Examples

```
chr.list = c("chr19","chr20","chr21")

suffix = ".small.coverage"
prefix = "http://genome.ucla.edu/~fah/ExomeCNV/data/normal."
normal = read.all.coverage(prefix, suffix, chr.list, header=TRUE)
prefix = "http://genome.ucla.edu/~fah/ExomeCNV/data/tumor."
tumor = read.all.coverage(prefix, suffix, chr.list, header=TRUE)

demo.logR = calculate.logR(normal, tumor)

suffix = ".RData"
prefix = paste("http://genome.ucla.edu/~fah/ExomeCNV/demo/",
"demo.small.eCNV.9999.9999.spec.c.5/demo.small.eCNV.9999.9999.spec.c.5.",
sep="")
demo.eCNV = read.eCNV(prefix, suffix, chr.list, url=TRUE)

## The following will take a while to run (~3-5 mins)
# demo.cnv = multi.CNV.analyze(normal, tumor, logR=demo.logR, all.cnv.ls=list(demo.eCNV),
#  coverage.cutoff=5, min.spec=0.99, min.sens=0.99, option="auc",
# c=0.5, sdundo=c(2), alpha=c(0.05))
```

`multi.LOH.analyze` *Use Circular Binary Segmentation to create segments and call LOH on each segment based on BAF using specified method.*

Description

This is analogous to `multi.CNV.analysis` for LOH. User can control fineness of segmentation by adjusting sdundo and `alpha`.

Usage

```
multi.LOH.analyze(normal = NULL, tumor = NULL, all.loh.ls = NULL,
  min.spec = 0.95, test.alpha = NULL,
  method = c("deviation.half.norm", "variance.f",
  "deviation.wilcox", "deviation.t", "CHM", "mantelhaen",
  "two.sample.fisher", "two.sample.prop", "only.tumor",
  "only.normal"), sdundo = c(1, 2), alpha = c(0.05, 0.01))
```

Arguments

`normal`	BAF information for all heterozygous positions in the normal exome. It should be a data.frame with four fields: chr, position, coverage, and baf. "baf" here doesn't refer to frequency but the raw count. Example can be found at https://secure.genome.ucla.edu/inde
`tumor`	Same as `normal`. If method is "only.tumor" or "only.normal", only one of tumor or normal is required, respectively.

all.loh.ls	A list of LOH calls to be merged. Usually a result of LOH.analyze. The order the LOH calls determines priority of the calls.
min.spec	Minimum specificity acceptable. Always equal to 1-alpha, where alpha is Type I error rate.
test.alpha	Type I error rate for the statistical test (not to be confused with alpha below). Only one of alpha and min.spec needs to be specified.
method	Type of test to use to call LOH. See details.
sdundo	the number of SDs between means to keep a split. This is a parameter for CBS as used in DNAcopy package.
alpha	significance levels for the test to accept change-points. This is a parameter for CBS as used in DNAcopy package.

Details

Statistical tests that can be used in calling LOH are based on three test statistics: # BAF as count statistic # Variance of BAF, reflecting the amount of deviation of BAF away from its central value (~0.5) # Absolute deviation of BAF from the null value of 0.5 # Difference between BAF's in case and control samples

Each test statistic allows for different tests and is based on different assumptions.

Options "only.tumor" and "only.normal" use only one sample (case or control) to perform binomial test against null p=0.5. We can model LOH as a binomial event, asking among N reads mapped to the position, how likely is it to observe a certain number of B-allele (BAF).

Options "two.sample.fisher" and "two.sample.prop" are similar to the binomial test for one sample above but instead of testing the observed proportion against the null value of 0.5, they compare the observed proportion between case and control. This can be modeled by binomial distribution (two.sample.prop) or hypergeometric distribution (Fisher's exact test; two.sample.fisher), hence the two possible tests.

Option "variance.f" performs F-test to compare variances of case and control BAF's

Options "deviation.wilcox" and "deviation.t" perform t-test and Wilcoxon Rank Sum (Mann-Whitney) Test, respectively. This is to compare the mean value of the absolute deviation of BAF from 0.5 (i.e. |BAF - 0.5|).

Option "deviation.half.norm" is based on the observation that the distribution of BAF difference between case and control are normally distributed around 0. Thus the absolute value follows folded-normal distribution. Under LOH, the absolute difference will have a higher mean value, and we can measure and test the increase in the difference using half-normal distribution.

Option "CMH" or "mantelhaen" uses Cochran-Mantel-Haenszel Chi-sq test for common odds ratio equal to 1. It requires that the number of stata N >= 2. In case N = 1, it is equivalent to Pearson's Chi-sq (prop.test). This is useful when trying to call LOH for segments, which contain multiple heterozygous positions, each with its own contigency table. The only problem with this test is that it requires phasing information, which does not always exist. Thus it is not recommended for use.

Value

An object of class eCNV. See link{classify.eCNV} for more information.

Author(s)

Fah Sathirapongsasuti <fsathira@fas.harvard.edu>

References

https://secure.genome.ucla.edu/index.php/ExomeCNV_User_Guide

See Also

multi.CNV.analyze

Examples

```
normal = read.delim("http://genome.ucla.edu/~fah/ExomeCNV/data/normal.small.baf.txt",
header=TRUE)
tumor = read.delim("http://genome.ucla.edu/~fah/ExomeCNV/data/tumor.small.baf.txt",
header=TRUE)
eLOH = LOH.analyze(normal, tumor, alpha=0.05, method="two.sample.fisher")

### WARNING! These examples take a long time to run (~15-20 mins).
# the.loh = multi.LOH.analyze(normal, tumor, all.loh.ls=list(eLOH), test.alpha=0.001,
#   method="variance.f", sdundo=c(0,0), alpha=c(0.5,0.1))
# the.loh = multi.LOH.analyze(normal, tumor, all.loh.ls=list(eLOH), min.spec=0.99,
# method="deviation.wilcox", sdundo=c(0), alpha=c(0.1))
# the.loh = multi.LOH.analyze(normal, tumor, all.loh.ls=list(eLOH), min.spec=0.999999,
# method="CMH", sdundo=c(2), alpha=c(0.1))
# the.loh = multi.LOH.analyze(normal, tumor, all.loh.ls=list(eLOH), min.spec=0.9999,
# method="two.sample.fisher", sdundo=c(0,0), alpha=c(0.1,0.05))
```

---

pool.coverage *Pooling a compendium of "normal" exome samples.*

Description

Pooling a compendium of "normal" exome samples to form a reference sample to use as a normal control in case no matched normal is available. In germline CNV discovery one may not have a matched normal sample to compare against. Pooling is proposed an an alternative whereby many "normal" samples can be aveaged and used to serve as a reference. These functions facilitate the averaging of those samples. All samples are assumed to be in the same prescribed format.

Usage

```
pool.coverage(all.data)
pool.coverage.from.files(infile.prefix.list, infile.suffix =
                 "exon_parsed.coverage", exome, chr.list = c("chr1",
                 "chr2", "chr3", "chr4", "chr5", "chr6", "chr7",
                 "chr8", "chr9", "chr10", "chr11", "chr12", "chr13",
                 "chr14", "chr15", "chr16", "chr17", "chr18", "chr19",
                 "chr20", "chr21", "chr22", "chrX", "chrY"))
```

Arguments

all.data	A list of all data.frame's for coverages.
infile.prefix.list	A vector of file prefixes. Similar to that used in read.all.coverage.
infile.suffix	A suffix string for coverage files. Similar to that used in read.all.coverage.
exome	A data.frame defining exome (with chr, probe_start, probe_end, and name as columns).
chr.list	A vector of chromosomes in the exome. Similar to that used in read.all.coverage.

Author(s)

Fah Sathirapongsasuti <fsathira@fas.harvard.edu>

References

https://secure.genome.ucla.edu/index.php/ExomeCNV_User_Guide

Examples

```
# note that this makes no biological sense, just an example
normal = read.table("http://genome.ucla.edu/~fah/ExomeCNV/data/normal.chr21.coverage",
header=TRUE, sep='\t')
tumor = read.table("http://genome.ucla.edu/~fah/ExomeCNV/data/tumor.chr21.coverage",
header=TRUE, sep='\t')
pool = pool.coverage(list(normal, tumor))

exome = read.delim("http://genome.ucla.edu/~fah/ExomeCNV/data/exome.sample.bed",
header=FALSE)
names(exome) = c("chr", "probe_start", "probe_end", "name")
suffix = ".coverage"
prefix.list = c("http://genome.ucla.edu/~fah/ExomeCNV/data/tumor.",
"http://genome.ucla.edu/~fah/ExomeCNV/data/normal.")
pool = pool.coverage.from.files(prefix.list, suffix, exome,
chr.list=c("chr19","chr20","chr21"))
```

---

read.all.coverage *Read all coverage files by chromosome.*

Description

Go through each chromosome and read coverage files (as prepared by the script available at http://genome.ucla.edu/~fah/Exon

Usage

```
read.all.coverage(prefix, suffix, chr.list = c("chr1", "chr2", "chr3",
                  "chr4", "chr5", "chr6", "chr7", "chr8", "chr9",
                  "chr10", "chr11", "chr12", "chr13", "chr14", "chr15",
                  "chr16", "chr17", "chr18", "chr19", "chr20", "chr21",
                  "chr22", "chrX", "chrY"), header = TRUE)
```

Arguments

prefix	Prefix of a coverage file name, particularly the part right before "chr#". For exampe, if the file names are: normal.chr1.coverage, normal.chr2.coverage, ..., the prefix is "normal." and the suffix is ".coverage". This assumes that the coverage files are saved separately by chromosome, and the file names differ at just the chromosome number. See example at: http://genome.ucla.edu/~fah/ExomeCNV/data.
suffix	Suffix of the file name, particularly the part following "chr#". See `prefix` for example.
chr.list	A vector of chromosome names in the exome. Should be of format "chr#" where # is the chromosome number. This is assuming that the coverage files are broken up by chromosomes, one file per chromosome. If all chromosomes are contained in one file, set `chr.list=c("")` and modify prefix/suffix so that they concatenate into the right filename.
header	Logical, inidcating if the coverage file has header or not.

Author(s)

Fah Sathirapongsasuti <fsathira@fas.harvard.edu>

References

https://secure.genome.ucla.edu/index.php/ExomeCNV_User_Guide

Examples

```
chr.list=paste("chr",c("19","20","21"),sep="")

suffix = ".small.coverage"

prefix = "http://genome.ucla.edu/~fah/ExomeCNV/data/normal."
normal = read.all.coverage(prefix, suffix, chr.list, header=TRUE)

prefix = "http://genome.ucla.edu/~fah/ExomeCNV/data/tumor."
tumor = read.all.coverage(prefix, suffix, chr.list, header=TRUE)
```

---

read.coverage.gatk *Read coverage file produced by The Genome Analysis Toolkit (GATK).*

Description

Read coverage file produced by The Genome Analysis Toolkit (GATK; http://www.broadinstitute.org/gsa/wiki/index.php/The
and reformat it to be usable by ExomeCNV. For exact command to produce the coverage file, see
https://secure.genome.ucla.edu/index.php?title=ExomeCNV_User_Guide#GATK_DepthOfCoverage.
The only three important columns are in the GATK-generated file are: Target, total_coverage, and
average_coverage. Note that it is okay that the last column of the resulting data.frame is "NA".

Usage

```
read.coverage.gatk(file)
```

Arguments

file	Exon coverage file as produced by GATK. See example at http://genome.ucla.edu/~fah/ExomeCNV/data/s

Author(s)

Fah Sathirapongsasuti <fsathira@fas.harvard.edu>

References

https://secure.genome.ucla.edu/index.php/ExomeCNV_User_Guide

Examples

```
coverage = read.coverage.gatk("http://genome.ucla.edu/~fah/ExomeCNV/data/sampleCoverage.gatk.txt")
```

---

read.coverage.gtf *Read coverage file in GTF format as created by Howie Goodell.*

Description

Read coverage file in GTF format (as prepared by Howie Goodell) and reformat it to be usable by ExomeCNV.

Usage

```
read.coverage.gtf(file)
```

Arguments

file	Exon coverage file in GTF format. See example at http://genome.ucla.edu/~fah/ExomeCNV/data/sampleC

Author(s)

Fah Sathirapongsasuti <fsathira@fas.harvard.edu>

References

https://secure.genome.ucla.edu/index.php/ExomeCNV_User_Guide

Examples

```
coverage = read.coverage.gtf("http://genome.ucla.edu/~fah/ExomeCNV/data/sampleCoverage.gtf")
```

read.eCNV *Read all .RData files produced by classify.eCNV().*

Description

This is a very simple function that facilitate reading .RData files produced by `classify.eCNV`. It makes an assumption that the .RData file contains a variable named `ecnv`, which is a very strong assumption. So if you wish to use this function, you need to name the output from `classify.eCNV` as `ecnv`. The best way to go about this is to follow the example in http://genome.ucla.edu/~fah/ExomeCNV/demo/demo.R and http://genome.ucla.edu/~fah/ExomeCNV/demo/do.eCNV.R.

Usage

```
read.eCNV(eCNV.prefix, eCNV.suffix, chr.list=c("chr1", "chr2", "chr3",
                  "chr4", "chr5", "chr6", "chr7", "chr8", "chr9",
                  "chr10", "chr11", "chr12", "chr13", "chr14", "chr15",
                  "chr16", "chr17", "chr18", "chr19", "chr20", "chr21",
                  "chr22", "chrX", "chrY"), url=FALSE)
```

Arguments

`eCNV.prefix`	The part of the .RData file name that precedes chromosome name. For example, if your file is named "demo.chr19.RData", the prefix is "demo." and suffix is ".RData".
`eCNV.suffix`	The part of the .RData file name that follows chromosome name. For example, if your file is named "demo.chr19.RData", the prefix is "demo." and suffix is ".RData".
`chr.list`	A vector of chromosome names as appeared in the .RData file name. For example, if your files are "demo.chr19.RData", "demo.chr20.RData", and "demo.chr21.RData", `chr.list` should be `c("chr19","chr20","chr21")`.
`url`	A boolean indicating if the file source is a URL.

Author(s)

Fah Sathirapongsasuti <fsathira@fas.harvard.edu>

References

https://secure.genome.ucla.edu/index.php/ExomeCNV_User_Guide

Examples

```
chr.list = paste("chr",c("19","20","21"), sep="")
suffix = ".RData"
prefix = paste("http://genome.ucla.edu/~fah/ExomeCNV/demo/",
"demo.eCNV.9999.9999.spec.c.5/demo.eCNV.9999.9999.spec.c.5.",
```

```
sep="")
demo.eCNV = read.eCNV(prefix, suffix, chr.list, url=TRUE)
```

---

save.logR *Save logR in files by chromosome.*

Description

Save logR in files by chromosome. This is to prepare input for `classify.eCNV` It requires exome to have chr and each line correspond to logR.

Usage

```
save.logR(all.logR, exome, name)
```

Arguments

`all.logR`	A vector of all log coverage ration as a result of `calculate.logR`
`exome`	A data.frame representing exome. It needs to have `chr` field. Can be the same as `normal` used in `calculate.logR`.
`name`	File name to be saved.

Author(s)

Fah Sathirapongsasuti <fsathira@fas.harvard.edu>

References

https://secure.genome.ucla.edu/index.php/ExomeCNV_User_Guide

Examples

```
chr.list=paste("chr",c("19","20","21"),sep="")

suffix = ".small.coverage"

prefix = "http://genome.ucla.edu/~fah/ExomeCNV/data/normal."
normal = read.all.coverage(prefix, suffix, chr.list, header=TRUE)

prefix = "http://genome.ucla.edu/~fah/ExomeCNV/data/tumor."
tumor = read.all.coverage(prefix, suffix, chr.list, header=TRUE)

demo.logR = calculate.logR(normal, tumor)
save.logR(demo.logR, normal, "demo")
```

write.loh.output *Generate an output file for LOH calls.*

Description

Generate .loh.txt file for the LOH call. This is a very simple function; it's essentially a wrapper for `write.table`.

Usage

```
write.loh.output(loh, name)
```

Arguments

`loh`	A data.frame
`name`	Prefix for the file name (to follow by .loh.txt)

Author(s)

Fah Sathirapongsasuti <fsathira@fas.harvard.edu>

References

https://secure.genome.ucla.edu/index.php/ExomeCNV_User_Guide

Examples

```
con = url("http://genome.ucla.edu/~fah/ExomeCNV/demo/demo.the.loh.RData")
load(con)
close(con)
write.loh.output(the.loh, "demo.eloh")
```

write.output *Generate output files from ExomeCNV outputs.*

Description

Generate output files from ExomeCNV outputs. The files geneated are: 1. .cnv.txt file with all CNV calls 2. .exon.lrr.txt file containing log coverage ratio for each exon 3. .segment.lrr.txt file containing log coverage ratio for each segment (as defined by CBS) 4. .segment.copynumber.txt file containing copy number calls for each segment 5. .cnv.png file, a plot of the results

Usage

```
write.output(eCNV, cnv, name)
```

Arguments

eCNV	Result of multi.CNV.analyze
cnv	Result of classify.eCNV
name	File name prefix for all the output files

Author(s)

Fah Sathirapongsasuti

References

https://secure.genome.ucla.edu/index.php/ExomeCNV_User_Guide

Examples

```
chr.list=c("chr19","chr20","chr21")

suffix = ".RData"
prefix = paste("http://genome.ucla.edu/~fah/ExomeCNV/demo/",
"demo.small.eCNV.9999.9999.spec.c.5/demo.small.eCNV.9999.9999.spec.c.5.",
sep="")
demo.eCNV = read.eCNV(prefix, suffix, chr.list, url=TRUE)

con = url("http://genome.ucla.edu/~fah/ExomeCNV/demo/demo.cnv.RData")
load(con)
close(con)

write.output(demo.eCNV, demo.cnv, "demo")
```

Index