Package 'PureCN'

October 12, 2016

Type Package

Title Estimating tumor purity, ploidy, LOH, and SNV status using hybrid capture NGS data

Version 1.0.4

Date 2016-06-05

Author Markus Riester

Maintainer Markus Riester <markus.riester@novartis.com>

Description This package estimates tumor purity, copy number, loss of heterozygosity (LOH), and status of single nucleotide variants (SNVs). PureCN is designed for hybrid capture sequencing data, integrates well with standard somatic variant detection pipelines, and has support for tumor samples without matching normal samples.

Depends R (>= 3.3), DNAcopy, VariantAnnotation (>= 1.14.1)

Imports GenomicRanges (>= 1.20.3), IRanges (>= 2.2.1), RColorBrewer, S4Vectors, data.table, grDevices, graphics, stats, utils, SummarizedExperiment, GenomeInfoDb

Suggests PSCBS, RUnit, BiocStyle, BiocGenerics, knitr

VignetteBuilder knitr

License Artistic-2.0

biocViews CopyNumberVariation, Software, Sequencing, VariantAnnotation, VariantDetection

NeedsCompilation no

R topics documented:

`chr.hash` *A data.frame of chromosome names.*

Description

A table of chromosome names and their corresponding numerical representation. This is needed because DNAcopy requires numerical chromosome names.

Usage

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

Value

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)
```

correctCoverageBias *Correct for GC bias*

Description

Takes as input coverage data in GATK format (or data read by readCoverageGatk) and a mapping file for GC content, and then uses a loess normalization for bias correction. Largely follows the GC correction of the TitanCNA package.

Usage

```
correctCoverageBias(gatk.coverage.file, gc.gene.file,
    output.file = NULL)
```

Arguments

gatk.coverage.file	Exon coverage file as produced by GATK. Either a file name or data parsed with the readCoverageGatk function.
gc.gene.file	File providing GC content for each exon in the coverage files. First column in format CHR:START-END. Second column GC content (0 to 1). Third column provides gene symbols, which are optional, but used in runAbsoluteCN to generate gene level calls.
output.file	Optionally, write minimal GATK coverage file with GC corrected coverage.

Value

GC normalized coverage.

Author(s)

Markus Riester

Examples

```
gatk.normal.file <- system.file("extdata", "example_normal.txt",
    package="PureCN")
gc.gene.file <- system.file("extdata", "example_gc.gene.file.txt",
    package="PureCN")
coverage <- correctCoverageBias(gatk.normal.file, gc.gene.file)
```

createCurationFile *Create file to curate PureCN results*

Description

Function to create a CSV file that can be used to mark the correct solution in the output of a runAbsoluteCN() run.

Usage

```
createCurationFile(file.rds, overwrite.uncurated = TRUE)
```

Arguments

file.rds	Output of the runAbsoluteCN() function, serialized with saveRDS()
overwrite.uncurated	Overwrite existing files unless flagged as "Curated".

Value

A data.frame with the tumor purity and ploidy of the maximum likelihood solution

Author(s)

Markus Riester

Examples

```
data(purecn.example.output)
file.rds <- 'Sample1_PureCN.rds'
saveRDS(purecn.example.output, file=file.rds)
createCurationFile(file.rds)
```

createExonWeightFile *Calculate exon weights*

Description

Creates an exon weight file useful for segmentation. Requires a set of GATK coverage files from normal samples. A small number of tumor (or other normal) samples is then tested against all normals. Exon weights will be set proportional to the inverse of coverage standard deviation across all normals. Exons with high variance in coverage in the pool of normals are thus down-weighted.

Usage

```
createExonWeightFile(gatk.tumor.files, gatk.normal.files,
    exon.weight.file)
```

Arguments

`gatk.tumor.files`
: A small number (1-3) of GATK tumor or normal coverage samples.

`gatk.normal.files`
: A large number of GATK normal coverage samples (>20) to estimate exon log-ratio standard deviations. Should not overlap with files in gatk.tumor.files.

`exon.weight.file`
: Output filename.

Value

A data.frame with exon weights.

Author(s)

Markus Riester

Examples

```
exon.weight.file <- "exon_weights.txt"
gatk.normal.file <- system.file("extdata", "example_normal.txt",
    package="PureCN")
gatk.normal2.file <- system.file("extdata", "example_normal2.txt",
    package="PureCN")
gatk.normal.files <- c(gatk.normal.file, gatk.normal2.file)
gatk.tumor.file <- system.file("extdata", "example_tumor.txt",
    package="PureCN")

createExonWeightFile(gatk.tumor.file, gatk.normal.files, exon.weight.file)
```

---

createNormalDatabase *Create database of normal samples*

Description

Function to create a database of normal samples, used to find a good match for tumor copy number normalization.

Usage

```
createNormalDatabase(gatk.normal.files, ...)
```

Arguments

`gatk.normal.files`
: Vector with file names pointing to GATK coverage files of normal samples.

`...`
: Arguments passed to the prcomp function.

Value

A normal database that can be used in the findBestNormal function to retrieve good matching normal samples for a given tumor sample.

Author(s)

Markus Riester

Examples

```
gatk.normal.file <- system.file("extdata", "example_normal.txt",
    package="PureCN")
gatk.normal2.file <- system.file("extdata", "example_normal2.txt",
    package="PureCN")
gatk.normal.files <- c(gatk.normal.file, gatk.normal2.file)
normalDB <- createNormalDatabase(gatk.normal.files)
```

---

createSNPBlacklist *Create SNP black list*

Description

Function to create a black list of germline SNPs with expected allelic fraction (AF) smaller than 0.5 in diploid genomes.

Usage

```
createSNPBlacklist(vcf.files, n = min(10, length(vcf.files)),
    low.af = 0.025, high.af = 0.1, genome = "hg19")
```

Arguments

vcf.files	List of VCF files. When a VCF file contains multiple samples, it will ignore all samples except the first.
n	Required number of VCF files showing low allelic fraction to blacklist a SNP id.
low.af	Defines a low AF p-value.
high.af	Defines a high AF p-value. For every sample with high AF p-value, there must be one more sample with low AF to reach the cutoff.
genome	Version of the reference genome, required for the readVcf() function.

Value

A list with elements snp.black.list and segmented. "snp.black.list" is just a list of SNP ids. "segmented" blacklists whole regions.

Author(s)

Markus Riester

Examples

```
# Assume VCF files of normals (for example obtained by a MuTect artifact
# detection run) are in directory poolofnormals:
mutect.normal.files <- dir("poolofnormals", pattern="vcf$", full.names=TRUE)

# These files do not exist in our example, so we do not run the function here.
#snp.blacklist <- createSNPBlacklist(mutect.normal.files)
```

---

filterVcfBasic *Basic VCF filter function*

Description

Function to remove artifacts and low confidence/quality variant calls.

Usage

```
filterVcfBasic(vcf, tumor.id.in.vcf = NULL, use.somatic.status = TRUE,
    snp.blacklist = NULL, af.range = c(0.03, 0.97),
    contamination.cutoff = c(0.05, 0.075), coverage.cutoff = 20,
    min.supporting.reads = 3, verbose = TRUE)
```

Arguments

vcf	CollapsedVCF object, read in with the readVcf function from the VariantAnnotation package.
tumor.id.in.vcf	The tumor id in the CollapsedVCF (optional).
use.somatic.status	If somatic status and germline data is available, then use this information to remove non-heterozygous germline SNPs or germline SNPS with biased allelic fractions.
snp.blacklist	CSV file with SNP ids with expected allelic fraction significantly different from 0.5 in diploid genomes. Can be an array of lists. The function createSNPBlacklist can provide appropriate black lists.
af.range	Exclude SNPs with allelic fraction smaller or greater than the two values, respectively.
contamination.cutoff	Count SNPs in dbSNP with allelic fraction smaller than the first value, if found on most chromosomes, remove all with AF smaller than the second value.
coverage.cutoff	Minimum coverage in tumor. Variants with lower coverage are ignored.

`min.supporting.reads`
: Minimum number of reads supporting the alt allele.

`verbose`

Value

A list with elements

`vcf`	The filtered CollapsedVCF object.
`flag`	A flag (TRUE/FALSE) if problems were identified.
`flag_comment`	A comment describing the flagging.

Author(s)

Markus Riester

Examples

```
# This function is typically only called by runAbsolute via the
# fun.filterVcf and args.filterVcf comments.
library(VariantAnnotation)
vcf.file <- system.file("extdata", "example_vcf.vcf", package="PureCN")
vcf <- readVcf(vcf.file, "hg19")
vcf.filtered <- filterVcfBasic(vcf)
```

---

filterVcfMuTect *Filter VCF MuTect*

Description

Function to remove artifacts and low confidence/quality calls from a MuTect generated VCF file.

Usage

```
filterVcfMuTect(vcf, tumor.id.in.vcf = NULL, stats.file = NULL,
    ignore = c("clustered_read_position", "fstar_tumor_lod",
        "nearby_gap_events", "poor_mapping_region_alternate_allele_mapq",
        "poor_mapping_region_mapq0", "possible_contamination",
        "strand_artifact"), verbose = TRUE, ...)
```

Arguments

`vcf`
: VCF object, read in with the readVcf function from the VariantAnnotation package.

`tumor.id.in.vcf`
: The tumor id in the VCF file, optional.

`stats.file`
: MuTect stats file

ignore

verbose Verbose output.

... Additional arguments passed to filterVcfBasic

Value

A list with elements vcf, flag and flag_comment. "vcf" contains the filtered CollapsedVCF, "flag" a flag if problems were identified, further described in "flag_comment".

Author(s)

Markus Riester

Examples

```
### This function is typically only called by runAbsolute via the
### fun.filterVcf and args.filterVcf comments.
library(VariantAnnotation)
vcf.file <- system.file("extdata", "example_vcf.vcf", package="PureCN")
vcf <- readVcf(vcf.file, "hg19")
vcf.filtered <- filterVcfMuTect(vcf)
```

---

findBestNormal *Find best normal sample in database*

Description

Function to find the best matching normal for a provided tumor sample.

Usage

```
findBestNormal(gatk.tumor.file, normalDB, pcs = 1:3,
    num.normals = 1, ignore.sex = FALSE, verbose = TRUE)
```

Arguments

gatk.tumor.file

GATK coverage file of a tumor sample.

normalDB Database of normal samples, created with createNormalDatabase().

pcs Principal components to use for distance calculation.

num.normals Return the num.normals best normals.

ignore.sex If FALSE, detects sex of sample and returns best normals with matching sex.

verbose Verbose output.

Value

Filename of the best matching normal.

Author(s)

Markus Riester

Examples

```
gatk.normal.file <- system.file("extdata", "example_normal.txt",
    package="PureCN")
gatk.normal2.file <- system.file("extdata", "example_normal2.txt",
    package="PureCN")
gatk.normal.files <- c(gatk.normal.file, gatk.normal2.file)
normalDB <- createNormalDatabase(gatk.normal.files)

gatk.tumor.file <- system.file("extdata", "example_tumor.txt",
    package="PureCN")
gatk.best.normal.file <- findBestNormal(gatk.tumor.file, normalDB)
```

---

findFocal *Find focal amplifications*

Description

Function to find focal amplifications in segmented data. This is automatically called in runAbsoluteCN.

Usage

```
findFocal(seg, size.cutoff = 2e+06, cn.diff = 2, amp.cutoff = 6)
```

Arguments

seg	Segmentation data.
size.cutoff	Cutoff for focal in base pairs.
cn.diff	Minimum copy number delta between neighboring segments.
amp.cutoff	Minimum amplification integer copy number.

Value

Boolean vector for each segment wether it is focally amplified or not.

Author(s)

Markus Riester

Examples

```
gatk.normal.file <- system.file("extdata", "example_normal.txt",
    package="PureCN")
gatk.tumor.file <- system.file("extdata", "example_tumor.txt",
    package="PureCN")
vcf.file <- system.file("extdata", "example_vcf.vcf",
    package="PureCN")
gc.gene.file <- system.file("extdata", "example_gc.gene.file.txt",
    package="PureCN")

# Speed-up the runAbsoluteCN call by using the stored grid-search
# (purecn.example.output$candidates).
data(purecn.example.output)

# The max.candidate.solutions parameter is set to a very low value only to
# speed-up this example.  This is not a good idea for real samples.
ret <-runAbsoluteCN(gatk.normal.file=gatk.normal.file,
    gatk.tumor.file=gatk.tumor.file,
    vcf.file=vcf.file, sampleid='Sample1', gc.gene.file=gc.gene.file,
    candidates=purecn.example.output$candidates, max.candidate.solutions=2,
    args.focal=list(size.cutoff = 2e+06), fun.focal=findFocal)
```

`getSexFromCoverage` *Get sample sex from coverage*

Description

This function determines the sex of a sample by the coverage ratio of chrX and chrY. Loss of chromosome Y (LOY) can result in a wrong female call.

Usage

```
getSexFromCoverage(gatk.coverage, min.ratio = 20, verbose = TRUE)
```

Arguments

`gatk.coverage`	GATK coverage file or data read with readCoverageGatk.
`min.ratio`	Min chrX/chrY coverage ratio to call sample as female.
`verbose`	Verbose output.

Value

Returns "M" for male, "F" for female, or NA if unknown.

Author(s)

Markus Riester

Examples

```
gatk.tumor.file <- system.file("extdata", "example_tumor.txt",
    package="PureCN")
sex <- getSexFromCoverage(gatk.tumor.file)
```

---

plotAbs *Plots for analyzing PureCN solutions*

Description

This function provides various plots for finding correct purity and ploidy combinations in the results of a runAbsoluteCN call.

Usage

```
plotAbs(res, ids = NULL, type = c("hist", "overview",
    "BAF", "AF", "LOH", "all"), chr = NULL, germline.only = TRUE,
    show.contour = FALSE, purity = NULL, ploidy = NULL,
    ...)
```

Arguments

res	Return object of the runAbsoluteCN() function.
ids	Candidate solutions to be plotted. ids=1 will draw the plot for the maximum likelihood solution.
type	Different types of plots. "hist" will plot a histogram, assigning log-ratio peaks to integer values. "overview" will plot all local optima, sorted by likelihood. "BAF" plots something like a B-allele frequency plot known from SNP arrays: it plots allele frequencies of germline variants (or most likely germline when status is not available) against copy number. AF plots observed allelic fractions against expected (purity), maximum likelihood (optimal multiplicity) allelic fractions. "all" plots all, and is useful for generate a PDF for a sample for manual inspection.
chr	If NULL, show all chromosomes, otherwise only the ones specified (type=BAF only).
germline.only	If TRUE, show only variants most likely being germline in BAF plot. Useful to set to FALSE (in combination with chr) to study potential artifacts.
show.contour	For type overview, display contour plot.
purity	Display expected integer copy numbers for purity, defaults to purity of the solution (type=hist only).
ploidy	Display expected integer copy numbers for ploidy, defaults to ploidy of the solution (type=hist only).
...	Additonal parameters passed to the plot() function.

Value

Returns NULL

Author(s)

Markus Riester

Examples

```
data(purecn.example.output)
plotAbs(purecn.example.output, type="overview")
# plot details for the maximum likelihood solution (rank 1)
plotAbs(purecn.example.output, 1, type="hist")
plotAbs(purecn.example.output, 1, type="BAF")
```

---

plotBestNormal *Plot the PCA of tumor and its best normal(s)*

Description

This function can be used to understand how a best normal is chosen by the findBestNormal function. It can be also used to tune the best normal selection by finding good parameter values for num.normals and pcs.

Usage

```
plotBestNormal(gatk.normal.files, gatk.tumor.file,
    normalDB, x = 1, y = 2, col.tumor = "red", col.best.normal = "blue",
    col.other.normals = "black", ...)
```

Arguments

gatk.normal.files	GATK coverage file of normal files, typically identified via findBestNormal.
gatk.tumor.file	GATK coverage file of a tumor sample.
normalDB	Database of normal samples, created with createNormalDatabase().
x	PC to be plotted on x-axis.
y	PC to be plotted on y-axis.
col.tumor	Color of tumor in plot.
col.best.normal	Color of best normals in plot.
col.other.normals	Color of best normals in plot.
...	Arguments passed to the plot function.

Value

Returns NULL

Author(s)

Markus Riester

Examples

```
gatk.normal.file <- system.file("extdata", "example_normal.txt",
    package="PureCN")
gatk.normal2.file <- system.file("extdata", "example_normal2.txt",
     package="PureCN")
gatk.normal.files <- c(gatk.normal.file, gatk.normal2.file)
normalDB <- createNormalDatabase(gatk.normal.files)

gatk.tumor.file <- system.file("extdata", "example_tumor.txt",
    package="PureCN")
gatk.best.normal.file <- findBestNormal(gatk.tumor.file, normalDB)
plotBestNormal(gatk.best.normal.file, gatk.tumor.file, normalDB)
```

---

poolCoverage *Pool coverage from multiple samples*

Description

Averages the coverage of a list of samples.

Usage

```
poolCoverage(all.data, remove.chrs = c(), w = NULL)
```

Arguments

all.data	List of normals, read with readCoverageGatk.
remove.chrs	Remove these chromosomes from the pool.
w	List of weights for each sample

Value

A data.frame with the averaged coverage over all normals.

Author(s)

Markus Riester

Examples

```
gatk.normal.file <- system.file("extdata", "example_normal.txt",
    package="PureCN")
gatk.normal2.file <- system.file("extdata", "example_normal2.txt",
    package="PureCN")
gatk.normal.files <- c(gatk.normal.file, gatk.normal2.file)
gatk.tumor.file <- system.file("extdata", "example_tumor.txt",
    package="PureCN")

normalDB <- createNormalDatabase(gatk.normal.files)

# get the best 2 normals and average them
gatk.best.normal.files <- findBestNormal(gatk.tumor.file, normalDB,
    num.normals=2)
pool <- poolCoverage(lapply(gatk.best.normal.files, readCoverageGatk),
     remove.chrs=c('chrX', 'chrY'))
```

---

predictSomatic *Predict germline vs. somatic status*

Description

This function takes as input the output of a runAbsoluteCN run and annotates variants more correctly as germline vs. somatic by inferring maternal and paternal chromosome numbers.

Usage

```
predictSomatic(res, id = 1, cutoff = 0.1)
```

Arguments

res	Return object of the runAbsoluteCN() function.
id	Candidate solutions to be analyzed. id=1 will analyze the maximum likelihood solution.
cutoff	Exclude maternal/paternal chromosome number in segment if posterior probability is lower than cutoff.

Value

A data.frame with adjusted SNV state posterior probabilities.

Author(s)

Markus Riester

Examples

```
data(purecn.example.output)
# the output data was created using a matched normal sample, but in case
# no matched normal is available, this will help predicting somatic vs.
# germline status
purecn.snvs <- predictSomatic(purecn.example.output)
```

`purecn.example.output` *Example output*

Description

This provides the output of the runAbsoluteCN call used in the vignette and examples.

Usage

```
data(purecn.example.output)
```

Value

Output of the runAbsoluteCN call used in the vignette.

`readCoverageGatk` *Read GATK coverage files*

Description

Read coverage file produced by The Genome Analysis Toolkit. The three only important columns in the GATK-generated file are: Target, total_coverage, and average_coverage.

Usage

```
readCoverageGatk(file)
```

Arguments

`file`	Exon coverage file as produced by GATK.

Value

A data.frame with the parsed coverage information.

Author(s)

Markus Riester

Examples

```
gatk.tumor.file <- system.file("extdata", "example_tumor.txt",
    package="PureCN")
coverage <- readCoverageGatk(gatk.tumor.file)
```

`readCurationFile` *Read curation file*

Description

Function that can be used to read the curated output of the runAbsoluteCN function.

Usage

```
readCurationFile(file.rds, file.curation = gsub(".rds$",
    ".csv", file.rds), remove.failed = FALSE, report.best.only = FALSE,
    min.ploidy = NULL, max.ploidy = NULL)
```

Arguments

`file.rds`	Output of the runAbsoluteCN() function, serialized with saveRDS()
`file.curation`	Filename of a curation file that points to the correct tumor purity and ploidy solution.
`remove.failed`	Do not return solutions that failed.
`report.best.only`	Only return correct/best solution (useful on low memory machines when lots of samples are loaded)
`min.ploidy`	Minimum ploidy to be considered. If NULL, all. Can be used to automatically ignore unlikely solutions.
`max.ploidy`	Maximum ploidy to be considered. If NULL, all. Can be used to automatically ignore unlikely solutions.

Value

The return value of the corresponding runAbsoluteCN call, but with the results array manipulated according the curation CSV file and arguments of this function.

Author(s)

Markus Riester

Examples

```
data(purecn.example.output)
file.rds <- 'Sample1_PureCN.rds'
createCurationFile(file.rds)
# User can change the maximum likelihood solution manually in the generated
# CSV file. The correct solution is then loaded with readCurationFile.
purecn.curated.example.output <-readCurationFile(file.rds)
```

`runAbsoluteCN` *Run PureCN implementation of ABSOLUTE*

Description

This function takes as input tumor and normal control coverage and allelic fractions of germline variants and somatic mutations. Coverage data is provided in GATK DepthOfCoverage format, allelic fraction in VCF format (e.g. obtained by MuTect). Normal control does not need to be matched (from the same patient). In case VCF does not contain somatic status, it should contain dbSNP and optionally COSMIC annotation. Returns purity and ploidy combinations, sorted by likelihood score. Provides copy number and LOH data, by both gene and genomic region.

Usage

```
runAbsoluteCN(gatk.normal.file = NULL, gatk.tumor.file,
    log.ratio = NULL, seg.file = NULL, seg.file.sdev = 0.4,
    vcf.file = NULL, genome = "hg19", sex = c("?",
        "F", "M"), fun.filterVcf = filterVcfMuTect,
    args.filterVcf = list(), fun.setPriorVcf = setPriorVcf,
    args.setPriorVcf = list(), fun.segmentation = segmentationCBS,
    args.segmentation = list(), fun.focal = findFocal,
    args.focal = list(), sampleid = NULL, min.ploidy = 1,
    max.ploidy = 6, test.num.copy = 0:7, test.purity = seq(0.05,
      0.95, by = 0.01), prior.purity = rep(1, length(test.purity))/length(test.purity),
    max.candidate.solutions = 15, candidates = NULL,
    coverage.cutoff = 15, max.non.clonal = 0.2, max.homozygous.loss = 0.1,
    iterations = 30, log.ratio.calibration = 0.25,
    gc.gene.file = NULL, filter.lowhigh.gc.exons = 0.001,
    filter.targeted.base = 4, max.logr.sdev = 0.75,
    max.segments = 200, plot.cnv = TRUE, verbose = TRUE,
    post.optimize = FALSE, ...)
```

Arguments

`gatk.normal.file`

GATK coverage file of normal control (optional if log.ratio is provided - then it will be only used to filter low coverage exons). Should be already GC-normalized. Needs to be either a file name or data read with the readCoverageGatk function.

`gatk.tumor.file`
: GATK coverage file of tumor. Should be already GC-normalized. Needs to be either a file name or data read with the readCoverageGatk function.

`log.ratio`
: Copy number log-ratios for all exons in the coverage files. If NULL, calculated based on coverage files.

`seg.file`
: Segmented data. Optional, to support matched SNP6 data. If null, use coverage files or log.ratio to segment the data.

`seg.file.sdev`
: If seg.file provided, the log-ratio standard deviation, used to model likelihood of sub-clonal copy number events.

`vcf.file`
: VCF file, tested with MuTect output files. Optional, but typically needed to select between local optima of similar likelihood. Can also be a CollapsedVCF, read with the readVcf function. Requires a DB info flag for dbSNP membership. The default fun.setPriorVcf function will also look for a Cosmic.CNT slot, containing the hits in the COSMIC database. Again, do not expect very useful results without a VCF file.

`genome`
: Genome version, required for the readVcf function.

`sex`
: Sex of sample. If ?, detect.

`fun.filterVcf`
: Function for filtering variants. Expected output is a list with elements vcf (CollapsedVCF), flag (TRUE/FALSE) and flag_comment (string). The flags will be added to the output data and can be used to warn users, for example when samples look too noisy. Default filter will remove variants flagged by MuTect, but will keep germline variants. If ran in matched normal mode, it will by default use somatic status of variants and filter non-somatic calls with allelic fraction significantly different from 0.5 in normal.

`args.filterVcf`
: Arguments for variant filtering function. Arguments vcf, tumor.id.in.vcf, coverage.cutoff and verbose are required in the filter function and are automatically set (do NOT set them here again).

`fun.setPriorVcf`
: Function to set prior for somatic status for each variant in the VCF.

`args.setPriorVcf`
: Arguments for somatic prior function.

`fun.segmentation`
: Function for segmenting the copy number log-ratios. Expected return value is a list with elements seg (the segmentation) and size (the size in bp for all segments).

`args.segmentation`
: Arguments for segmentation function. Arguments normal, tumor, log.ratio, plot.cnv, coverage.cutoff, sampleid, vcf, tumor.id.in.vcf, verbose are required in the segmentation function and automatically set (do NOT set them here again).

`fun.focal`
: Function for identifying focal amplifications.

`args.focal`
: Arguments for focal amplification function.

`sampleid`
: Sample id, provided in output files etc.

`min.ploidy`
: Minimum ploidy to be considered.

`max.ploidy`
: Maximum ploidy to be considered.

`test.num.copy` Copy numbers tested in the grid search. Note that focal amplifications can have much higher copy numbers, but they will be labeled as subclonal (because they do not fit the integer copy numbers).

`test.purity` Considered tumor purity values.

`prior.purity` Priors for purity if they are available. Only change when you know what you are doing.

`max.candidate.solutions`
Number of local optima considered in optimization and variant fitting steps. If there are too many local optima, it will use specified number of top candidate solutions, but will also include all optima close to diploid, because silent genomes have often lots of local optima.

`candidates` Candidates to optimize from a previous run (return.object$candidates). If NULL, do 2D grid search and find local optima.

`coverage.cutoff`
Minimum exon coverage in both normal and tumor. Exons with lower coverage are ingored. The cutoff choice depends on the expected purity and overall coverage. High purity samples might need a lower cutoff to call homozygous deletions. If an exon.weigh.file (below) is NOT specified, it is recommended to set a higher cutoff (e.g. 20) to remove noise from unreliable exon measurements.

`max.non.clonal` Maximum genomic fraction assigned to a subclonal copy number state.

`max.homozygous.loss`
Maximum genomic fraction assigned to homozygous loss. This is set to a fairly high default value to not exclude correct solutions, especially in noisy segmentations.

`iterations` Maximum number of iterations in the Simulated Annealing copy number fit optimization.

`log.ratio.calibration`
re-calibrate log-ratios in the window sd(log.ratio)*log.ratio.calibration.

`gc.gene.file` A mapping file that assigns GC content and gene symbols to each exon in the coverage files. Used for generating gene level calls. First column in format CHR:START-END. Second column GC content (0 to 1). Third column gene symbol.

`filter.lowhigh.gc.exons`
Quantile q (defines lower q and upper 1-q) for removing exons with outlier GC profile. Assuming that GC correction might not have been worked on those. Requires gc.gene.file.

`filter.targeted.base`
Exclude exons with targeted base (size) smaller than this cutoff. This is useful when the same interval file was used to calculate GC content. For such small exons, the GC content is likely very different from the true GC content of the probes.

`max.logr.sdev` Flag noisy samples with segment log-ratio standard deviation larger than this. Assay specific and needs to be calibrated.

`max.segments` Flag noisy samples with a large number of segments. Assay specific and needs to be calibrated.

plot.cnv	Generate segmentation plots.
verbose	Verbose output.
post.optimize	Optimize purity using final SCNA-fit and SNVs. This might take a long time when lots of SNVs need to be fitted, but will typically result in a slightly more accurate purity, especially for rather silent genomes or very low purities. Otherwise, it will just use the purity determined via the SCNA-fit.
...	Additional parameters passed to the segmentation function.

Value

A list with elements

candidates	Results of the grid search.
results	All local optima, sorted by final rank.
input	The input data.

Author(s)

Markus Riester

Examples

```
gatk.normal.file <- system.file("extdata", "example_normal.txt",
    package="PureCN")
gatk.tumor.file <- system.file("extdata", "example_tumor.txt",
    package="PureCN")
vcf.file <- system.file("extdata", "example_vcf.vcf",
    package="PureCN")
gc.gene.file <- system.file("extdata", "example_gc.gene.file.txt",
    package="PureCN")

# Speed-up the runAbsoluteCN call by using the stored grid-search
# (purecn.example.output$candidates).
data(purecn.example.output)

# The max.candidate.solutions parameter is set to a very low value only to
# speed-up this example.  This is not a good idea for real samples.

ret <-runAbsoluteCN(gatk.normal.file=gatk.normal.file,
    gatk.tumor.file=gatk.tumor.file,
    candidates=purecn.example.output$candidates, max.candidate.solutions=2,
    vcf.file=vcf.file, sampleid='Sample1', gc.gene.file=gc.gene.file)
```

segmentationCBS *CBS segmentation*

Description

The default segmentation function. This function is called via the fun.segmentation argument of runAbsoluteCN. The arguments are passed via args.segmentation.

Usage

```
segmentationCBS(normal, tumor, log.ratio, plot.cnv,
    coverage.cutoff, sampleid = sampleid, exon.weight.file = NULL,
    alpha = 0.005, vcf = NULL, tumor.id.in.vcf = 1,
    verbose = TRUE)
```

Arguments

normal	GATK coverage data for normal sample.
tumor	GATK coverage data for tumor sample.
log.ratio	Copy number log-ratios, one for each exon in coverage file.
plot.cnv	Segmentation plots.
coverage.cutoff	Minimum coverage in both normal and tumor.
sampleid	Sample id, used in output files.
exon.weight.file	Can be used to assign weights to exons.
alpha	Alpha value for CBS, see documentation for the segment function.
vcf	Optional VCF object with germline allelic ratios.
tumor.id.in.vcf	Id of tumor in case multiple samples are stored in VCF.
verbose	Verbose output.

Value

A list with elements

seg	The segmentation.
size	The size of all segments (in base pairs).

Author(s)

Markus Riester

Examples

```
gatk.normal.file <- system.file("extdata", "example_normal.txt",
    package="PureCN")
gatk.tumor.file <- system.file("extdata", "example_tumor.txt",
    package="PureCN")
vcf.file <- system.file("extdata", "example_vcf.vcf",
    package="PureCN")
gc.gene.file <- system.file("extdata", "example_gc.gene.file.txt",
    package="PureCN")

# speed-up the runAbsoluteCN by using the stored grid-search.
# (purecn.example.output$candidates).
data(purecn.example.output)

# The max.candidate.solutions argument is set to a very low value only to
# speed-up this example. This is not a good idea for real samples.
ret <-runAbsoluteCN(gatk.normal.file=gatk.normal.file,
    gatk.tumor.file=gatk.tumor.file,
    vcf.file=vcf.file, sampleid='Sample1', gc.gene.file=gc.gene.file,
    candidates=purecn.example.output$candidates, max.candidate.solutions=2,
    fun.segmentation=segmentationCBS, args.segmentation=list(alpha=0.001))
```

`segmentationPSCBS` *PSCBS segmentation*

Description

Segmentation function. Uses the PSCBS package. This function is called via the fun.segmentation argument of runAbsoluteCN. The arguments are passed via args.segmentation.

Usage

```
segmentationPSCBS(normal, tumor, log.ratio, plot.cnv,
    coverage.cutoff, sampleid = sampleid, exon.weight.file = NULL,
    flavor = "tcn&dh", tauA = 0.03, vcf = NULL, tumor.id.in.vcf = 1,
    verbose = TRUE, ...)
```

Arguments

`normal`	GATK coverage data for normal sample.
`tumor`	GATK coverage data for tumor sample.
`log.ratio`	Copy number log-ratios, one for each exon in coverage file.
`plot.cnv`	Segmentation plots.
`coverage.cutoff`	Minimum coverage in both normal and tumor.
`sampleid`	Sample id, used in output files.

exon.weight.file
: Can be used to assign weights to exons. NOT SUPPORTED YET.

flavor
: Flavor value for PSBCS. See segmentByNonPairedPSCBS.

tauA
: tauA argument for PSCBS. See segmentByNonPairedPSCBS.

vcf
: Optional VCF object with germline allelic ratios.

tumor.id.in.vcf
: Id of tumor in case multiple samples are stored in VCF.

verbose
: Verbose output.

...
: Additional parameters passed to the segmentByNonPairedPSCBS function.

Value

A list with elements seg and size. "seg" contains the segmentation, "size" the size of all segments in base pairs.

Author(s)

Markus Riester

Examples

```
gatk.normal.file <- system.file("extdata", "example_normal.txt",
    package="PureCN")
gatk.tumor.file <- system.file("extdata", "example_tumor.txt",
    package="PureCN")
vcf.file <- system.file("extdata", "example_vcf.vcf",
    package="PureCN")
gc.gene.file <- system.file("extdata", "example_gc.gene.file.txt",
    package="PureCN")

ret <-runAbsoluteCN(gatk.normal.file=gatk.normal.file,
    gatk.tumor.file=gatk.tumor.file, vcf.file=vcf.file, sampleid='Sample1',
    gc.gene.file=gc.gene.file, fun.segmentation=segmentationPSCBS)
```

---

setPriorVcf *Set Somatic Prior VCF*

Description

Function to set prior for somatic mutation status for each variant in the provided CollapsedVCF object.

Usage

```
setPriorVcf(vcf, prior.somatic = c(0.5, 5e-04, 0.999,
    1e-04, 0.95, 0.01), verbose = TRUE)
```

Arguments

vcf	CollapsedVCF object, read in with the readVcf function from the VariantAnnotation package.
prior.somatic	Prior probabilities for somatic mutations. First value is for the case when no matched normals are available and the variant is not in dbSNP (second value). Third value is for variants with MuTect somatic call. Different from 1, because somatic mutations in segments of copy number 0 have 0 probability and artifacts can thus have dramatic influence on likelihood score. Forth value is for variants not labeled as somatic by MuTect. Last two values are optional, if vcf contains a flag Cosmic.CNT, it will set the prior probability for variants with CNT > 2 to the first of those values in case of no matched normal available (0.95 default). Final value is for the case that variant is in both dbSNP and COSMIC > 2.
verbose	Verbose output.

Value

A vector with the prior probability of somatic status for each variant in the CollapsedVCF.

Author(s)

Markus Riester

Examples

```
# This function is typically only called by runAbsoluteCN via the
# fun.setPriorVcf and args.setPriorVcf comments.
vcf.file <- system.file("extdata", "example_vcf.vcf", package="PureCN")
vcf <- readVcf(vcf.file, "hg19")
vcf.priorsomatic <- setPriorVcf(vcf)
```

Index