# Package ‘ZygosityPredictor’

January 6, 2024

<table>
<thead>
<tr>
<th>Type</th>
<th>Package</th>
</tr>
</thead>
<tbody>
<tr>
<td>Title</td>
<td>Package for prediction of zygosity for variants/genes in NGS data</td>
</tr>
<tr>
<td>Version</td>
<td>1.2.0</td>
</tr>
<tr>
<td>Date</td>
<td>2022-10-26</td>
</tr>
<tr>
<td>Imports</td>
<td>GenomicAlignments, GenomicRanges, Rsamtools, IRanges, VariantAnnotation, DelayedArray, dplyr, stringr, purrr, tibble, methods, igraph</td>
</tr>
<tr>
<td>License</td>
<td>GPL-2</td>
</tr>
<tr>
<td>Description</td>
<td>The ZygosityPredictor allows to predict how many copies of a gene are affected by small variants. In addition to the basic calculations of the affected copy number of a variant, the ZygosityPredictor can integrate the influence of several variants on a gene and ultimately make a statement if and how many wild-type copies of the gene are left. This information proves to be of particular use in the context of translational medicine. For example, in cancer genomes, the ZygosityPredictor can address whether unmutated copies of tumor-suppressor genes are present. Beyond this, it is possible to make this statement for all genes of an organism. The ZygosityPredictor was primarily developed to handle SNVs and INDELs (later addressed as small-variants) of somatic and germline origin. In order not to overlook severe effects outside of the small-variant context, it has been extended with the assessment of large scale deletions, which cause losses of whole genes or parts of them.</td>
</tr>
<tr>
<td>RoxygenNote</td>
<td>7.2.3</td>
</tr>
<tr>
<td>Encoding</td>
<td>UTF-8</td>
</tr>
<tr>
<td>biocViews</td>
<td>BiomedicalInformatics, FunctionalPrediction, SomaticMutation, GenePrediction</td>
</tr>
<tr>
<td>Depends</td>
<td>R (&gt;= 4.3.0)</td>
</tr>
<tr>
<td>LazyData</td>
<td>false</td>
</tr>
<tr>
<td>Suggests</td>
<td>knitr, rmarkdown, testthat, BiocStyle</td>
</tr>
<tr>
<td>VignetteBuilder</td>
<td>knitr</td>
</tr>
<tr>
<td>git_url</td>
<td><a href="https://git.bioconductor.org/packages/ZygosityPredictor">https://git.bioconductor.org/packages/ZygosityPredictor</a></td>
</tr>
<tr>
<td>git_branch</td>
<td>RELEASE_3_18</td>
</tr>
</tbody>
</table>
aff_germ_copies

Calculates how many copies are affected by a germline small variant

Usage

aff_germ_copies(chr, af, tcn, purity, sex, c_normal = NULL, af_normal = 0.5)

Arguments

chr  chromosome of the variant (either format 1,2,...,X,Y or chr1,...,chrX)
af   Allele-frequency of the variant (numeric value between 0 and 1)
tcn  total-copynumber at position of the variant (numeric value >0)
purity purity of the sample (numeric value between 0 and 1 indicating the fraction of relevant sample with control/unrelevant tissue)
aff_som_copies

sex                sex of the sample (character: "male", "female", "m", "f")
c_normal           expected copy number at position of the variant in normal tissue, 1 for gono-
somes in male samples, and 2 for male autosomes and all chromosomes in fe-
male samples. (The function can also assess the c_normal parameter by itself,
but then the following two inputs must be provided: chr and sex)
af_normal          Allele-frequency in normal tissue (numeric value between 0 and 1) 0.5 repre-
sents heterozygous variants in diploid genome, 1 would be homozygous. Could
be relevant if germline CNVs are present at the position. Then also the c_normal
parameter would have to be adjusted.

Value

A numeric value indicating the affecting copies for the variant
A numeric value indicating the affecting copies for the variant

Examples

library(dplyr)
library(purrr)
library(stringr)
aff_germ_copies(af=0.67, tcn=2, purity=0.9, chr="chrX", sex="female")
library(dplyr)
library(purrr)
library(stringr)
aff_germ_copies(af=0.67, tcn=2, purity=0.9, chr="chrX", sex="female")

aff_som_copies calculates how many copies are affected by a somatic small variant

Description

calculates how many copies are affected by a somatic small variant
calculates how many copies are affected by a somatic small variant

Usage

aff_som_copies(chr, af, tcn, purity, sex, c_normal = NULL)
aff_som_copies(chr, af, tcn, purity, sex, c_normal = NULL)

Arguments

chr  chromosome of the variant (either format 1,2...X,Y or chr1,...,chrX)
af    Allele-frequency of the variant (numeric value between 0 and 1)
tcn   total-copynumber at position of the variant (numeric value >0)
purity  purity of the sample (numeric value between 0 and 1 indicating the fraction of relevant sample with control/unrelevant tissue)

sex    sex of the sample (character: "male", "female", "m", "f")

c_normal  expected copy number at the position of the variant in normal tissue, 1 for gonomes in male samples, and 2 for male autosomes and all chromosomes in female samples. (The function can also assess the c_normal parameter by itself, but then the following two inputs must be provided: chr and sex)

Value

A numeric value indicating the affecting copies for the variant

A numeric value indicating the affecting copies for the variant

Examples

```r
library(dplyr)
library(purrr)
library(stringr)
aff_som_copies(chr="chrX", af=0.67, tcn=2, purity=0.9, sex="female")
library(dplyr)
library(purrr)
library(stringr)
aff_som_copies(chr="chrX", af=0.67, tcn=2, purity=0.9, sex="female")
```

---

GR_GENE_MODEL  germline small variant object

Description

germline small variant object

Usage

data(GR_GENE_MODEL)

Format

```r
## 'GR_GENE_MODEL' GRanges object
```
<table>
<thead>
<tr>
<th>Package</th>
<th>Description</th>
<th>Usage</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>GR_GERM_SMALL_VARS</strong></td>
<td>germline small variant object</td>
<td>data(GR_GERM_SMALL_VARS)</td>
</tr>
<tr>
<td><strong>GR_SCNA</strong></td>
<td>copynumber object</td>
<td>data(GR_SCNA)</td>
</tr>
<tr>
<td><strong>GR_SOM_SMALL_VARS</strong></td>
<td>somatic small variant object</td>
<td>data(GR_SOM_SMALL_VARS)</td>
</tr>
</tbody>
</table>
predict_zygosity predicts zygosity of a set of genes of a sample

**Description**

predicts zygosity of a set of genes of a sample

**Usage**

```
predict_zygosity(
  purity,
  sex,
  somCna,
  geneModel,
  bamDna,
  somSmallVars = NULL,
  germSmallVars = NULL,
  bamRna = NULL,
  ploidy = NULL,
  colnameTcn = NULL,
  colnameCnaType = NULL,
  includeHomoDel = TRUE,
  includeIncompleteDel = TRUE,
  showReadDetail = FALSE,
  printLog = FALSE,
  assumeSomCnaGaps = FALSE,
  byTcn = TRUE,
  vcf = NULL,
  distCutOff = 5000
)
```

```
predict_zygosity(
  purity,
  sex,
  somCna,
  geneModel,
  bamDna,
  somSmallVars = NULL,
  germSmallVars = NULL,
  bamRna = NULL,
  ploidy = NULL,
  colnameTcn = NULL,
  colnameCnaType = NULL,
  includeHomoDel = TRUE,
  includeIncompleteDel = TRUE,
```
predict_zygosity

showReadDetail = FALSE,
printLog = FALSE,
assumeSomCnaGaps = FALSE,
byTcn = TRUE,
vcf = NULL,
distCutOff = 5000
)

Arguments

purity
purity of the sample (numeric value between 0 and 1 indicating the fraction of relevant sample with control/unrelevant tissue)

sex
sex of the sample (character: "male", "female", "m", "f")

somCna
GRanges object containing all genomic regions with annotated total copynumber and cna_type as metadata columns. The total-copynumber column should be named "tcn" but also some other commonly used names. It should contain numeric values or characters that can be converted to numeric values. The cna_type column must contain the information about loss of heterozygosity (LOH). Therefore the term "LOH" must be explicitly mentioned in the column. If a genomic region is not present in the object, it will be taken as heterozygous with neutral TCN of 2.

geneModel
GRanges object containing the gene-annotation of the used reference genome with metadata column of the gene name (gene)

bamDna
path to bam-file

somSmallVars
GRanges object containing all somatic small variants (SNV and INDEL). Required metadata columns are reference base (ref/REF), alternative base (alt/ALT), annotation of the gene name (gene/GENE) and the allele-frequency (af/AF). If the object is not provided the tool assumes there are no somatic small variants.

germsmallVars
GRanges object containing all germline small variants (SNV and INDEL). Required metadata columns are reference base (ref/REF), alternative base (alt/ALT), annotation of the gene name (gene/GENE) and the allele-frequency (af/AF). If the object is not provided the tool assumes there are no germline small variants.

bamRna
optional; path to rna file (bam format)

ploidy
ploidy of the sample (numeric value)

colnameTcn
character indicating the name of the metadata containing the tcn information in the somCna object. If not provided the tool tries to detect the column according to default names

colnameCnaType
character indicating the name of the metadata containing cna type information in the somCna object. If not provided the tool tries to detect the column according to default names

includeHomoDel
default = TRUE; if FALSE homozygous deletions are excluded

includeIncompleteDel
default = TRUE; if FALSE heterzygous deletions are excluded

showReadDetail
default = FALSE; if TRUE a table is added to the output,
predict_zygosity

printLog
default = FALSE; if TRUE the gene which is evaluated is printed in console, containing the query-name of each read which was used to perform haplotype-phasing and the info into which class it was assigned.

assumeSomCnaGaps
(logical, default=FALSE) Only required if the somCna object lacks copy number information for genomic segments on which small variants are detected. By default, variants in such regions will be excluded from the analysis as required information about the copy number is missing. These variants will be attached to the final output list in a separate tibble. To include them, this flag must be set TRUE and the ground ploidy must be given as an input. This ground ploidy will then be taken as tcn in the missing regions. If no ploidy is given the tool will assume the ground ploidy of 2 when this flag is TRUE.

byTcn
logical, default=TRUE; optional if includeHomoDel or includeIncompleteDelS is TRUE. If FALSE the tool will not use tcn as a criterion to assign large deletions. It will use the cna_type column and check for indicating strings like HOMDEL/HomoDel/DEL. Some commonly used strings are covered. It is recommended to leave this flag TRUE.

vcf
character; path to variant call file (.vcf.gz format). Will be used (if provided) for extended SNP phasing if variants on the same gene are too far away from each other for direct haplotype phasing.

distCutOff
numeric, default=5000; if input vcf is provided and SNP phasing is performed, this will limit the distance at which the SNP phasing should not be tried anymore. As the probability of finding overlapping reads at such a long distance is very low and the runtime will increase exponentially.

Value
A list of dataframes. Those are the evaluation per variant, the evaluation per gene and, if performed, the info about the haplotype-phasing.

Examples

```R
# Generating example data for CNVs and somatic variants

# CNVs
library(GenomicRanges)
library(dplyr)
cns = GRanges(dplyr::tibble(
  chr = "chr17",
  start = c(170060, 34520990),
  end = c(34520990, 83198614),
  tcn = c(2, 1),
  cna_type = c("neutral", "LOH")
))

# Somatic variants
somatic_vars = GRanges(dplyr::tibble(
  chr="chr17",
  start = 7675088,
  end = 7675088,
  ref = "C",
  alt = "G"
))
```


predict_zygosity

alt = "T",
af = 0.65,
gene = "TP53"
)

germline_vars = GenomicRanges::GRanges(
dplyr::tibble(
  chr="chr17",
  start = 41771694,
  end = 41771694,
  ref = "GTGT",
  alt = "G",
  af = 0.95,
  gene = "JUP"
)

reference = GenomicRanges::GRanges(
dplyr::tibble(
  chr = "chr17",
  start = c(7661778, 41754603),
  end = c(7687538, 41786931),
  gene = c("TP53", "JUP")
)

sex = "female"
purity = 0.9

bamfile <- system.file("extdata", "ZP_example.bam",
  package = "ZygosityPredictor")
predict_zygosity(purity = purity, sex = sex,
somCna = cnvs,
somSmallVars = somatic_vars,
germsSmallVars = germline_vars,
geneModel = reference,
bamDna = bamfile)

cnvs = GenomicRanges::GRanges(
dplyr::tibble(
  chr = "chr17",
  start = c(170060, 34520990),
  end = c(34520990, 83198614),
  tcn = c(2, 1),
  cna_type = c("neutral", "LOH")
)

somatic_vars = GenomicRanges::GRanges(
dplyr::tibble(
  chr="chr17",
  start = 7675088,
  end = 7675088,
  ref = "C",
  alt = "T",
  af = 0.65,
  gene = "TP53"
germline_vars = GenomicRanges::GRanges(dplyr::tibble(
    chr="chr17",
    start = 41771694,
    end = 41771694,
    ref = "GTGT",
    alt = "G",
    af = 0.95,
    gene = "JUP"
))
reference = GenomicRanges::GRanges(dplyr::tibble(
    chr = "chr17",
    start = c(7661778, 41754603),
    end = c(7687538, 41786931),
    gene = c("TP53", "JUP")
))
sex = "female"
purity = 0.9
bamfile <- system.file("extdata", "ZP_example.bam", 
    package = "ZygosityPredictor")
predict_zygosity(purity = purity, sex = sex, 
    somCna = cnvs, 
    somSmallVars = somatic_vars, 
    germSmallVars = germline_vars, 
    geneModel = reference, 
    bamDna = bamfile)
Index

* datasets
  GR_GENE_MODEL, 4
  GR_GERM_SMALL_VARS, 5
  GR_SCNA, 5
  GR_SOM_SMALL_VARS, 5

aff_germ_copies, 2
aff_som_copies, 3

GR_GENE_MODEL, 4
GR_GERM_SMALL_VARS, 5
GR_SCNA, 5
GR_SOM_SMALL_VARS, 5

predict_zygosity, 6