Package ‘OncoPhase’

March 31, 2016

Type Package

Title SOMATIC MUTATION CELLULAR PREVALENCE COMPUTATION

Version 0.1

Date 2016-02-25

Description This package offers a direct method to quantify the cellular prevalence of single nucleotide variants (SNVs) using phase information. The method utilizes three sources of information: the phasing information, the copy number variation, and the allele counts. The method is demonstrated to bring more capabilities in Cancer Genomic and allows computing the cell prevalence of a mutation in various cancer contexts.

LazyData TRUE

License GPL-2

Imports limSolve

Suggests

#VignetteBuilder knitr

RoxygenNote 5.0.1

NeedsCompilation no

Author Donatien Chedom-Fotso [aut, cre],
Ahmed Ahmed [aut],
Christopher Yau [aut]

Maintainer Donatien Chedom-Fotso <donatien.chedom@well.ox.ac.uk>

R topics documented:

aOncoPhase *OncoPhase package for somatic mutations cellular prevalence quantification using haplotype phasing*

Description

The main function for somatic mutation cellular prevalence computation is `getPrevalence`. This function computes the cellular prevalence of a list of mutations located at a given region of the genome. It can also work on a whole genome scale. See the manual and examples at `getPrevalence` for more details.

Details

To compute the prevalence at a single mutation use the function `getPhasedSNPPrevalence`. The function `getPrevalenceLinear` compute the prevalence of a given mutation by directly solving the linear system associated to the model.

Input data for simple case studies can be generated with the function `build_casestudy`.

The package include experimental data for chromosome 10, 15, 18 and 22 for two patients retrieved from a parallel clinical study. (see for example `chr10_OP1019` and `chr22_OP1019`)

For more detailed information on usage, see the package vignette, by typing `vignette("OncoPhase")`. All support questions should be emailed to the authors.

Author(s)

Donatien Chedom-Fotso, Ahmed Ahmed, Christopher Yau.

References

OncoPhase reference:

OncoPhase: A package for computing Somatic Mutation cellular Prevalence in cancer using haplotype phasing. Bioinformatics 2016. Submitted

OncoPhase reference:

" Ovarian cancer haplotype sequencing reveals ubiquitous SOX2 overexpression in the premalignant fallopian tube epithelium"

buildModelMatrices *Generate the matrices C, W and M from a set of parameters.*

Description

This is a generic function to generate the matrices of the linear system (see the paper) from the allele counts and the copy number information.

Usage

```
buildModelMatrices(lambda_G, mu_G, lambda_S, mu_S, major_cn, minor_cn, context)
```

Arguments

`lambda_G` A count of alleles supporting the variant sequence of the Germline SNP

`mu_G` : A count of alleles supporting the reference sequence of the Germline SNP

`lambda_S` : A count of alleles supporting the variant sequence of the somatic mutation

`mu_S` : A count of alleles supporting the reference sequence of the somatic mutation

`minor_cn` : Minor copy number at the locus of the mutation

`context` represents either the situation of a mutation which occurred after the CNV ("C1") or the context of a mutation which occurred before the CNV ("C2"). If not provided, the right context will be estimated from the input

`major_cn:` Major copy number at the locus of the mutation

Value

the matrices W, C and M for the linear system of prevalence computation.

Examples

```
Matrices = buildModelMatrices(8, 5,3,10,2,1,"C1")

 print(Matrices)
# $context
# [1] "C1"
#
# $W
# SNP       SNV
# SNP 0.6153846 0.0000000
# SNV 0.0000000 0.2307692
#
# $M
# Germ Alt Both
# SNP    1   2    2
# SNV    0   0    1
#
# $C
# Germ Alt Both
# SNP    2   3    3
# SNV    2   3    3
```

`build_casestudy` *Build the input data matrices for a case study*

Description

This is a generic function to automatically build the five input data frame (snp_allelecount_df, ref_allelecount_df, phasing_association_df, major_copynumber_df,minor_copynumber_df,CNVfraction_df if method is PhasedSNPGeneral) for a case study with one somatic mutation, one germline mutation and one or more tumor sample.

Usage

```
build_casestudy(lambda_G, mu_G, lambda_S, mu_S, major_cn, minor_cn,
  cnv_fraction = NULL, depthOfCoverage = NULL)
```

Arguments

`lambda_G`	: A count or a vector of counts (In the case of multiple tumor samples) of alleles supporting the variant sequence of the Germline SNP
`mu_G`	: A count or a vector of counts (In the case of multiple tumor samples) of alleles supporting the reference sequence of the Germline SNP
`lambda_S`	: A count or a vector of counts (In the case of multiple tumor samples) of alleles supporting the variant sequence of the somatic mutation
`mu_S`	: A count or a vector of counts (In the case of multiple tumor samples) of alleles supporting the reference sequence of the somatic mutation
`minor_cn`	: Minor copy number (or a vector of copy number if multiple tumor samples) at the locus of the mutation
`depthOfCoverage`	: Coverage depth (or a vector of depth coverage if multiple tumor samples) at the locus of the mutation. If not provided the exact value of the counts passed as parameters are considered. If provided then a binomial sampling with replacement is performed to generate the counts. For the germline, the sampling is done with the parameters p=lambda_G / (lambda_G + mu_G) and N= depthOfCoverage and will yield the count of allele supporting the variant sequence of the germline and the count of allele supporting the reference. The same sampling is apply to the somatic mutation with the parameters : p=lambda_S/(lambda_S + mu_S) and N = depthOfCoverage.
`cnv_fraction:`	Estimated fraction (or a vector of fractions if multiple tumor samples) of cells affected by the CNV (1- normal genotype cell fraction). Used only in the case of the PhasedSNPgeneral method
`major_cn:`	Major copy number (or a vector of copy number if multiple tumor samples) at the locus of the mutation

Value

A list containing the following data frames:

snp_allelecount_df A data frame containing the count of alleles at each tumour samples supporting the variant sequence at the somatic and germline mutations. Chrom is set to chr3, Position of the germline and somatic mutations are respectively set to 100 and 1000

ref_allelecount_df A data frame containing the count of alleles at each tumour sample supporting the reference sequence at the somatic and germline mutation. Chrom is set to chr3, Position of the germline and somatic mutations are respectively set to 100 and 1000

phasing_association_df A data frame containing the phasing association between the somatic and the germline mutation

major_copynumber_df A data frame containing the major copy number at each tumour sample at the mutation locus

minor_copynumber_df A data frame containing the minor copy number at each tumour sample at the mutation locus

normalfraction_df A data frame containing the proportion of cells with a normal genotype at each tumour sample. Present only if the method is "PhasedSNPgeneral"

Examples

```
#Example 1
# We reproduce here the case study No 6 of the paper
 #Build the input data
 cs = build_casestudy(lambda_G=16, mu_G=8,lambda_S=14,mu_S=10,cnv_fraction=4/8,major_cn=3,minor_cn=1  )
 #Run the case
prevalence=getPrevalence(cs$snp_allelecount_df, cs$ref_allelecount_df, cs$phasing_association_df,
                     cs$major_copynumber_df,cs$minor_copynumber_df,cs$normalfraction_df)
#print the result
print(prevalence)
#Chrom  End IsGermline Tumour1
#somaticM  chr3 1000          0    0.75


#Example 2
#Multiple tumours and stochastic generation of the counts.
CaseStudy_10 = build_casestudy(lambda_G=c(8,12,10), mu_G=c(5,4,8),lambda_S=c(3,6,10),mu_S=c(10,8,12),
major_cn=c(2,2,3),minor_cn=c(1,1,2) , depthOfCoverage = c(60,100,200))

cs = CaseStudy_10
prevalence=getPrevalence(cs$snp_allelecount_df, cs$ref_allelecount_df, cs$phasing_association_df,
                     cs$major_copynumber_df,cs$minor_copynumber_df,cs$normalfraction_df)
#print the result
print(prevalence)
```

chr10_OP1019 *chr10_OP1019 : chromosome 10 of Patient OP1019.*

Description

A dataset containing Allele counts, haplotype phasing and copy number information on chromosome 10 of Patient OP1019 of the SOX2 Study.

Usage

```
chr10_OP1019
```

Format

Contains the following data : :

tumoursamples The list of tumor samples of the study

SNP_allelecount_df Data frame containing the count of allele supporting the variant of each mutations

Chrom : Chromosomes
Start : Starting position
End : End position
Vartype : variant Type
IsGermline : is the mutation a Germline SNP or a Somatic mutation
Ref : Reference sequence

All : Variant sequence

One entry per tumor samples

ref_allelecount_df Data frame containing the count of allele supporting the reference at each mutation

phasing_association_df A data frame containing for each somatic mutations, a colon separated list of Germline mutations phased to it.

major_copynumber_df A data frame containing the major copy number of the mutations at each tumor samples

minor_copynumber_df A data frame containing the minor copy number of the mutations at each tumor samples

minor_copynumber_df A data frame containing the normal cell contamination rate for each mutations at each tumour samples

Details

Chromosome also available on the same patient: chr10, chr15, chr18 and chr22

`chr22_OP1019` *chr22_OP1019 : chromosome 22 of Patient OP1019.*

Description

A dataset containing allele counts, haplotype phasing and copy number information on chromosome 22 of Patient OP1019 of the SOX2 Study.

Usage

```
chr22_OP1019
```

Format

Contains the following data : :

tumoursamples The list of tumor samples of the study

SNP_allelecount_df Data frame containing the count of allele supporting the variant of each mutations

Chrom : Chromosomes

Start : Starting position

End : End position

Vartype : variant Type

IsGermline : is the mutation a Germline SNP or a Somatic mutation

Ref : Reference sequence

All : Variant sequence

One entry per tumor samples

ref_allelecount_df Data frame containing the count of allele supporting the reference at each mutation

phasing_association_df A data frame containing for each somatic mutations, a colon separated list of Germline mutations phased to it.

major_copynumber_df A data frame containing the major copy number of the mutations at each tumor samples

minor_copynumber_df A data frame containing the minor copy number of the mutations at each tumor samples

minor_copynumber_df A data frame containing the normal cell contamination rate for each mutations at each tumor samples

Details

Also available on the same patient : chr10, chr15, chr18 and chr22

getPhasedSNPPrevalence

Compute detailed prevalence at a single mutation point

Description

This is a generic function to compute the prevalence at a single somatic mutation point using a phased Germline SNP.

Usage

```
getPhasedSNPPrevalence(lambda_G, mu_G, lambda_S, mu_S, major_cn, minor_cn,
  cnv_fraction = NULL, context = NULL, form = "Matrix", detail = FALSE)
```

Arguments

`lambda_G` A count of alleles supporting the variant sequence of the Germline SNP

`mu_G` A count of alleles supporting the reference sequence of the Germline SNP

`lambda_S` : A count of alleles supporting the variant sequence of the somatic mutation

`mu_S` : A count of alleles supporting the reference sequence of the somatic mutation

`major_cn` Major copy number at the locus of the mutation

`minor_cn` : Minor copy number at the locus of the mutation

`cnv_fraction` If provided, represents the fraction of cells affected by the copy number alteration. This value, if not provided, is computed from the allelic count information and copy number information. Default NULL

`context` If provided, it represents either the situation of a mutation which occurred after the CNV ("C1") or the context of a mutation which occurred before the CNV ("C2"). If not provided, the right context will be estimated from the input

`form` Can be either "Matrix" either "General", specify if the prevalence should be computed using the linear form formula or the General form formula. Default "Matrix"

`detail` In case form="Matrix", when set to TRUE, a detailed output is generated containing, the context and the detailed prevalence for each group of cells (germline cells (Germ), cells affected by one of the two genomic alterations (Alt), cells affected by by both genomic alterations (Both)).

Value

if form="general", the function return a numerical value representing the prevalence at the somatic mutation.

if form="matrix", the function return a list containing the following data frames:

Context The associated context

Prevalence The computed prevalence

fullPrevalence Detailed prevalence for each of the three genotype groups separated by "|". The three groups are Germline mutations, mutations harboring one of the two alterations (CNV or SNP) mutations harboring both alterations

Examples

```
# We reproduce here the case study No 6 of the paper
 #General form
 prevalence = getPhasedSNPPrevalence(lambda_G=16, mu_G=8,lambda_S=14,mu_S=10,major_cn=3,minor_cn=1,form=”Ge
 print(prevalence)
 # Matrix form
 prevalence = getPhasedSNPPrevalence(lambda_G=16, mu_G=8,lambda_S=14,mu_S=10,major_cn=3,minor_cn=1, form=”Ma
 print(prevalence)
```

getPrevalence *Somatic mutations cellular prevalence using haplotype phasing.*

Description

This is a generic function to compute the cellular prevalence of somatic mutations in cancer using haplotype phasing. The function applies the model to a range of mutations located at a given genomic region or at the whole genome scale. The model computes the prevalence of a somatic mutation relatively to close and eventually phased germline mutations. It uses three sources of information as input : The allelic counts, the phasing information and the copy number alteration. Multiple tumor samples can be provided for the prevalence computation.

Usage

```
getPrevalence(snp_allelecount_df, ref_allelecount_df, phasing_association_df,
  major_copynumber_df, minor_copynumber_df, cnv_fraction_df = NULL,
  nbFirstColumns = 3, method = "PhasedSNP", tumoursamples = NULL,
  region = NULL, min_cells = 2, min_alleles = 4, detail = TRUE)
```

Arguments

`snp_allelecount_df`
: A data frame containing for each mutation the allelic counts of the variant at each tumor samples. The data frame should contains at least the following three columns among its firsts columns: Chrom (The mutation chromosome) , End (The mutation position) and IsGermline (is the mutation a germline or somatic mutation).

ref_allelecount_df
A data frame containing for each mutation the allelic count of the reference at each tumor sample. The data frame should contains at least the following three columns among its firsts columns: Chrom (The mutation or Somatic mutation)

phasing_association_df
A data frame containing for each somatic mutation, a colon separated list of germline SNP phased to it.

major_copynumber_df
A data frame containing for each mutation, its major

minor_copynumber_df
A data frame containing for each mutation the minor chromosomal copy number at each tumor samples.

nbFirstColumns
Number of first columns in snp_allelecount_df to reproduce in the output dataframe e.g: Chrom, Pos, Vartype. Columns from nbFirstColumns +1 to the last column should contains the information needed for the prevalence computation at each tumour sample

method
The method to be used for prevalence computation (default : PhasedSNP , alternatives methods are PhasedSNPGeneral, FlankingSNP,FlankingSNPGeneral)

tumoursamples
: The list of tumor samples to consider for the prevalence snp_allelecount_df, ref_allelecount_df, major_copynumber_df,minor_copynumber_df and CNVfraction_df. If not provided, the headers from nbFirstColumns + 1 to the last column of snp_allelecount_df is retrieved and its intersection with the other inputted data frames headers is considered.

region
The region of the genome to consider for the prevalence computation in the format chrom:start-end e.g "chr22:179800-98767

min_cells
Minimum number of cells (default 2). In case the estimated number of cells sequenced at the locus of the mutation is less than min_cells, NA is returned.

min_alleles
Minimum number of alleles. (default 4). In case the estimated number of alleles sequenced at the locus of the mutation is less than min_alleles, NA is returned.

detail
when set to TRUE, a detailed output is generated containing, the context and the detailed prevalence for each group of cells (germline cells, cells affected by one of the two genomic alterations (SNV or CNV) but not both, cells affected by by both copynumber alteration and SNV). Default : TRUE.

CNVfraction_df,
If provided, represents a data frame containing for each mutation, the fraction of cells affected by a copy number alteration. If not provided theses values will be implicitly deduced from the other inputs. Mostly useful if the method is "PhasedSNPGeneral".

Value

A data frame containing :

Column 1 to NbFirstcolumn of the input data frame snp_allelecount_df. This will generally include the chromosome and the position of the mutation plus any other columns to report in the prevalence dataframe (e.g REF, ALL, ...)

One column per tumour sample reporting the prevalence of the mutation at each samples

Examples

```
#Example 1: Loading a simple example data set with two somatic mutations, 5 germlines SNP, and 3 tumor samples
data(simpleExample2)
se=simpleExample2
prevalence_df=getPrevalence(se$snp_allelecount_df, se$ref_allelecount_df, se$phasing_association_df, se$maj
print(prevalence_df)

#Chrom     End IsGermline  Tumour1        Tumour2        Tumour3
#mutation2  chr2 3003000          0 C2:0|0|1 C2:0.15|0|0.85 C2:0.12|0|0.88
#mutation6  chr2 4008000          0 C1:1|0|0       C1:1|0|0 C2:0|0.24|0.76

# Example 2: Running a case study as illustrated in the accompanying paper. Available case studies: A, B, C, 1
data(CaseStudy_6)
cs=CaseStudy_6
prevalence_CaseStudy6=getPrevalence(cs$snp_allelecount_df, cs$ref_allelecount_df, cs$phasing_association_df
print(prevalence_CaseStudy6)
#Chrom  End IsGermline          Tumour1
#somaticM  chr3 1000          0 C2:0.25|0.25|0.5

data(CaseStudy_A)
cs=CaseStudy_A
prevalence_CaseStudy_A=getPrevalence(cs$snp_allelecount_df, cs$ref_allelecount_df, cs$phasing_association_d
print(prevalence_CaseStudy_A)
#  Chrom  End IsGermline   Tumour1
# somaticM  chr3 1000          0 0.66


#Example 3 : Computing somatic mutation cellular prevalence on chromosome 15 of  patient 11152 (data retrieved

data(”chr15_OP1019”)
ds=chr15_OP1019
masterprevalence_df=getPrevalence(ds$snp_allelecount_df, ds$ref_allelecount_df, ds$phasing_association_df,
print(head(masterprevalence_df))

data(”chr10_OP1019”)
df=chr10_OP1019
masterprevalence_df=getPrevalence(df$snp_allelecount_df, df$ref_allelecount_df, df$phasing_association_df,
print(head(masterprevalence_df))


# Example 4 : Creating a simple example with one somatic mutation and one germline mutation on a single tumor s

#Empty dataframe
snpcount_df=as.data.frame(matrix(ncol=4,nrow=2))
names(snpcount_df) = c(”Chrom”,”End”,”IsGermline”,”Tumour1”)
rownames(snpcount_df) = c(”mutation1”,”mutation2”)
refcount_df = snpcount_df
major_cn_df= as.data.frame(matrix(ncol=1,nrow=2))
names(major_cn_df) = ”Tumour1”
rownames(major_cn_df) = c(”mutation1”,”mutation2”)
minor_cn_df = major_cn_df
CNVFraction_df = major_cn_df

#Filling the dataframes
snpcount_df[”mutation1”,] = c(”chr1”, 200100,0,40)
```

```
snpcount_df["mutation2",] = c("chr1",  200900,1,60)
refcount_df["mutation1",] = c("chr1",  200100,0,20)
refcount_df["mutation2",] = c("chr1",  200900,1,40)
major_cn_df["Tumour1"] = c(1,1)
minor_cn_df["Tumour1"] = c(1,1)
CNVFraction_df["Tumour1"] = c(0.2,0.2)

#Phasing association
phasing_association_df = as.data.frame(matrix(ncol=1,nrow=1))
colnames(phasing_association_df) = c("PhasedGermline")
rownames(phasing_association_df) = c("mutation1")
phasing_association_df["mutation1","PhasedMutations"] = "mutation2"

#Computing the prevalence
prevalence_df=getPrevalence(snpcount_df, refcount_df, phasing_association_df, major_cn_df, minor_cn_df,cnv_

print(prevalence_df)

#Chrom    End IsGermline      Tumour1
#mutation1  chr1 200100         0 C2:0|0.5|0.5
```

getPrevalenceLinear *Compute the cellular prevalence of each group of cells*

Description

This is a generic function to compute the detailed prevalence of a single mutation using the linear system making the model.

Usage

```
getPrevalenceLinear(lambda_G, mu_G, lambda_S, mu_S, major_cn, minor_cn, context)
```

Arguments

`lambda_G` : A count of alleles supporting the variant sequence of the Germline SNP

`mu_G` : A count of alleles supporting the reference sequence of the Germline SNP

`lambda_S` : A count of alleles supporting the variant sequence of the somatic mutation

`mu_S` : A count of alleles supporting the reference sequence of the somatic mutation

`minor_cn` : Minor copy number (or a vector of copy number if multiple tumor samples)

`context` represents either the situation of a mutation which occurred after the CNV ("C1") or the context of a mutation which occurred before the CNV ("C2"). If not provided, the right context will be estimated from the input

`major_cn:` Major copy number at the locus of the mutation

Value

A list of the three cellular prevalence of each of the three groups of cells

Examples

```
Prevalences = getPrevalenceLinear(8, 5,3,10,2,1,"C1")

  print(Prevalences)
# Germ  Alt Both
# 0.4  0.0  0.6
```

`hg19_dfsize` *@export*

Description

@export

Usage

```
hg19_dfsize
```

Format

An object of class `list` of length 25.

Index