Package ‘sssc’

June 15, 2018

Title  Same Species Sample Contamination Detection
Version  1.0.0
Description  Imports Variant Calling Format file into R. It can detect whether a sample contains contaminant from the same species. In the first stage of the approach, a change-point detection method is used to identify copy number variations for filtering. Next, features are extracted from the data for a support vector machine model. For log-likelihood calculation, the deviation parameter is estimated by maximum likelihood method. Using a radial basis function kernel support vector machine, the contamination of a sample can be detected.

Depends  R (>= 3.4.0)
Imports  changepoint, e1071, ggplot2, stats, VGAM
License  GPL-2
Encoding  UTF-8
LazyData  true
RoxygenNote  6.0.1
NeedsCompilation  no
Author  Tao Jiang [aut, cre]
Maintainer  Tao Jiang <tjiang8@ncsu.edu>
Repository  CRAN
Date/Publication  2018-06-15 11:22:54 UTC

R topics documented:

  config_df  ................................................................. 2
  generate_feature  ...................................................... 3
  getAlt2  ................................................................. 4
  getAnnoRate  ............................................................ 4
  getAvgLL  ................................................................. 5
  getLowDepth  ............................................................ 5
config_df

Default parameters of config.

Description

A dataframe containing default parameters.

Usage

config_df

Format

A data frame with 12 variables:

- threshold  Threshold for allele frequency
- skew  Skewness for allele frequency
- lower  Lower bound for allele frequency region
- upper  Upper bound for allele frequency region
- ldpthr  Threshold to determine low depth
- hom_mle  Hom MLE of p in Beta-Binomial model
- het_mle  Het MLE of p in Beta-Binomial model
generate_feature

Hom_thred  Threshold between hom and high
High_thred  Threshold between high and het
Het_thred  Threshold between het and low
hom_rho  Hom MLE of rho in Beta-Binomial model
het_rho  Het MLE of rho in Beta-Binomial model

Source

Created by Tao Jiang

---

**generate_feature**  
*Feature Generation for Contamination Detection Model*

### Description

Generates features from each pair of input VCF objects for training contamination detection model.

### Usage

```r
generate_feature(file, hom_p = 0.999, het_p = 0.5, hom_rho = 0.005, het_rho = 0.1, mixture, homcut = 0.99, highcut = 0.7, hetcut = 0.3)
```

### Arguments

- **file**: VCF input object
- **hom_p**: The initial value for p in Homozygous Beta-Binomial model, default is 0.999
- **het_p**: The initial value for p in Heterozygous Beta-Binomial model, default is 0.5
- **hom_rho**: The initial value for rho in Homozygous Beta-Binomial model, default is 0.005
- **het_rho**: The initial value for rho in Heterozygous Beta-Binomial model, default is 0.1
- **mixture**: A vector of whether the sample is contaminated: 0 for pure; 1 for contaminated
- **homcut**: Cutoff allele frequency value between hom and high, default is 0.99
- **highcut**: Cutoff allele frequency value between high and het, default is 0.7
- **hetcut**: Cutoff allele frequency value between het and low, default is 0.3

### Value

A data frame with all features for training model of contamination detection
getAlt2  

**Second alternative allele percentage**

**Description**
Second alternative allele percentage

**Usage**
getAlt2(f)

**Arguments**
- **f**: Input raw file

**Value**
Percent of the second alternative allele

getAnnoRate  

**Annotation rate**

**Description**
Annotation rate

**Usage**
getAnnoRate(f)

**Arguments**
- **f**: Input raw file

**Value**
Percentage of annotation locus
**getAvgLL**

*Calculate average log-likelihood*

**Description**

Calculate average log-likelihood

**Usage**

```r
getAvgLL(df, hom_mle, het_mle, hom_rho, het_rho)
```

**Arguments**

- `df` Input modified file
- `hom_mle` Hom MLE of p in Beta-Binomial model, default is 0.9981416 from NA12878_1_L5
- `het_mle` Het MLE of p in Beta-Binomial model, default is 0.4737897 from NA12878_1_L5
- `hom_rho` Hom MLE of rho in Beta-Binomial model, default is 0.04570275 from NA12878_1_L5
- `het_rho` Het MLE of rho in Beta-Binomial model, default is 0.02224098 from NA12878_1_L5

**Value**

`meanLL`

---

**getLowDepth**

*Low depth percentage*

**Description**

Low depth percentage

**Usage**

```r
getLowDepth(f, ldepthred)
```

**Arguments**

- `f` Input raw file
- `ldepthred` Threshold to determine low depth, default is 20

**Value**

Percentage of low depth
getRatio

*Get the ratio of allele frequencies with a region*

**Description**

Get the ratio of allele frequencies with a region

**Usage**

```
getRatio(subdf, lower, upper)
```

**Arguments**

- **subdf**: Dataframe with calculated statistics
- **lower**: Lower bound for allele frequency region
- **upper**: Upper bound for allele frequency region

**Value**

Ratio of allele frequencies with a region

getSkewness

*Get absolute value of skewness*

**Description**

Get absolute value of skewness

**Usage**

```
getSkewness(subdf)
```

**Arguments**

- **subdf**: Input dataframe

**Value**

Absolute value of skewness
**getSNVRate**

<table>
<thead>
<tr>
<th>getSNVRate</th>
<th>SNV percentage</th>
</tr>
</thead>
</table>

**Description**

SNV percentage

**Usage**

getSNVRate(df)

**Arguments**

df: Input raw file

**Value**

Percentage of SNV

---

**getVar**

<table>
<thead>
<tr>
<th>getVar</th>
<th>Calculate zygosity variable</th>
</tr>
</thead>
</table>

**Description**

Calculate zygosity variable

**Usage**

getVar(df, state, hom_mle, het_mle)

**Arguments**

df: Input modified file
state: Zygosity state
hom_mle: MLE in hom model
het_mle: MLE in het model

**Value**

Zygosity variable
locateFile  
**Check input filename**

**Description**
Check input filename

**Usage**
locateFile(fn, extension)

**Arguments**
- `fn`: Exact full file name of input file, including directory
- `extension`: Expected input file extension: vcf & txt

**Value**
Valid directory

---

negll  
**Negative Log Likelihood**

**Description**
Calculates negative log likelihood for beta binomial distribution.

**Usage**
negll(x, size, prob, rho)

**Arguments**
- `x`: Depth of alternative allele
- `size`: Total depth
- `prob`: Theoretical probability for heterozygous is 0.5, for homozygous is 0.999
- `rho`: Rho parameter of Beta-Binomial distribution of alternative allele
**readGATK**  
*Read in input vcf data in GATK format for Contamination detection*

**Description**  
Read in input vcf data in GATK format for Contamination detection

**Usage**  
readGATK(dr, dbOnly, depCut, thred, content, extnum, keepall)

**Arguments**  
- **dr**: A valid input object
- **dbOnly**: Use dbSNP as filter, default is FALSE, passed from read_vcf
- **depCut**: Use a threshold for min depth, default is False
- **thred**: Threshold for min depth, default is 20
- **content**: Column names in VCF files
- **extnum**: The column number or numbers to be extracted from vcf, default is 10; 0 for not extracting any columns
- **keepall**: Keep unextracted column in output, default is TRUE, passed from read_vcf

**Value**  
Dataframe from VCF file

---

**readStrelka**  
*Read in input vcf data in strelka2 format for Contamination detection*

**Description**  
Read in input vcf data in strelka2 format for Contamination detection

**Usage**  
readStrelka(dr, dbOnly, depCut, thred, content, extnum, keepall)
### Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>dr</code></td>
<td>A valid input object</td>
</tr>
<tr>
<td><code>dbOnly</code></td>
<td>Use dbSNP as filter, default is FALSE, passed from read_vcf</td>
</tr>
<tr>
<td><code>depcut</code></td>
<td>Use a threshold for min depth, default is False</td>
</tr>
<tr>
<td><code>thred</code></td>
<td>Threshold for min depth, default is 20</td>
</tr>
<tr>
<td><code>content</code></td>
<td>Column names in VCF files</td>
</tr>
<tr>
<td><code>extnum</code></td>
<td>The column number or numbers to be extracted from vcf, default is 10; 0 for not extracting any columns</td>
</tr>
<tr>
<td><code>keepall</code></td>
<td>Keep unextracted column in output, default is TRUE, passed from read_vcf</td>
</tr>
</tbody>
</table>

### Value

Dataframe from VCF file

---

### Description

Read in input vcf data in VarDict format for Contamination detection

### Usage

```r
readVarDict(dr, dbOnly, depCut, thred, content, extnum, keepall)
```

### Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>dr</code></td>
<td>A valid input object</td>
</tr>
<tr>
<td><code>dbOnly</code></td>
<td>Use dbSNP as filter, default is FALSE, passed from read_vcf</td>
</tr>
<tr>
<td><code>depcut</code></td>
<td>Use a threshold for min depth, default is False</td>
</tr>
<tr>
<td><code>thred</code></td>
<td>Threshold for min depth, default is 20</td>
</tr>
<tr>
<td><code>content</code></td>
<td>Column names in VCF files</td>
</tr>
<tr>
<td><code>extnum</code></td>
<td>The column number to be extracted from vcf, default is 10; 0 for not extracting any column</td>
</tr>
<tr>
<td><code>keepall</code></td>
<td>Keep unextracted column in output, default is TRUE, passed from read_vcf</td>
</tr>
</tbody>
</table>

### Value

Dataframe from VCF file
**readVarPROWL**

**Read in input vcf data in VarPROWL format**

**Description**

Read in input vcf data in VarPROWL format

**Usage**

```
readVarPROWL(dr, dbOnly, depCut, thred, content, extnum, keepall)
```

**Arguments**

- **dr**: A valid input object
- **dbOnly**: Use dbSNP as filter, default is FALSE, passed from read_vcf
- **depCut**: Use a threshold for min depth, default is False
- **thred**: Threshold for min depth, default is 20
- **content**: Column names in VCF files
- **extnum**: The column number or numbers to be extracted from vcf, default is 10; 0 for not extracting any columns
- **keepall**: Keep unextracted column in output, default is TRUE, passed from read_vcf

**Value**

vcf Dataframe from VCF file

---

**read_vcf**

**VCF Data Input**

**Description**

Reads a file in vcf or vcf.gz file and creates a list containing Content, Meta, VCF and file_sample_name

**Usage**

```
read_vcf(fn, vcfFor, dbOnly = FALSE, depCut = FALSE, thred = 20, 
metaline = 200, extnum = 10, keepall = T)
```
rho_est

Estimate Rho for Alternative Allele Frequency

Description

Estimates Rho parameter in beta binomial distribution for alternative allele frequency

Usage

rho_est(vl)

Arguments

vl A list of vcf objects from read_vcf function.

Value

A list containing (1) het_rho: Rho parameter of heterozygous location; (2) hom_rho: Rho parameter homozygous location;
Examples

data("vcf_example")
vcf_list <- list()
vcf_list[[1]] <- vcf_example$VCF
res <- rho_est(v1 = vcf_list)
res$het_rho[[1]]$par
res$hom_rho[[1]]$par

---

rmChangePoint

Remove CNV regions within VCF files by changepoint method

Description

Remove CNV regions within VCF files by changepoint method

Usage

rmChangePoint(vcf, threshold, skew, lower, upper)

Arguments

- **vcf**: Input VCF files
- **threshold**: Threshold for allele frequency
- **skew**: Skewness for allele frequency
- **lower**: Lower bound for allele frequency region
- **upper**: Upper bound for allele frequency region

Value

VCF object without changepoint region

---

rmCNVinVCF

Remove CNV regions within VCF files given cnv file

Description

Remove CNV regions within VCF files given cnv file

Usage

rmCNVinVCF(vcf, cnvobj)
sssc

Same Species Sample Contamination

Arguments

- vcf: Input VCF files
- cnvobj: cnv object

Value

VCF object without changepoint region

Description

Detects whether a sample is contaminated another sample of its same species. The input file should be in vcf format.

Usage

sssc(file, rmCNV = FALSE, cnvobj = NULL, config = NULL, class_model = NULL, regression_model = NULL)

Arguments

- file: VCF input object
- rmCNV: Remove CNV regions, default is FALSE
- cnvobj: cnv object, default is NULL
- config: config information of parameters. A default set is generated as part of the model and is included in a model object, which contains
- class_model: An SVM classification model
- regression_model: An SVM regression model

Value

A list containing (1) stat: a data frame with all statistics for contamination estimation; (2) result: contamination estimation (Class = 0, pure; Class = 1, contaminated)

Examples

data(vcf_example)
result <- sssc(file = vcf_example)
summary_vcf

VCF Data Summary

Description
Summarizes allele frequency information in scatter and density plots.

Usage
summary_vcf(vcf, ZG = NULL, CHR = NULL)

Arguments
- vcf: VCF object from read_vcf function
- ZG: zygosity: (1) null, for both het and hom, default; (2) het; (3) hom
- CHR: chromosome number: (1) null, all chromosome, default; (2) any specific number

Value
A list containing (1) scatter: allele frequency scatter plot; (2) density: allele frequency density plot

Examples
```r
data("vcf_example")
tmp <- summary_vcf(vcf = vcf_example, ZG = 'het', CHR = c(1,2))
plot(tmp$scatter)
plot(tmp$density)
```

svm_class_model

Default svm classification model.

Description
An svm object containing default svm classification model.

Usage
svm_class_model

Format
An svm object:

Source
Created by Tao Jiang
**svm_regression_model**  Default svm regression model.

**Description**

An svm object containing default svm regression model.

**Usage**

```r
svm_regression_model
```

**Format**

An svm object:

**Source**

Created by Tao Jiang

---

**train_ct**  Train Contamination Detection Model

**Description**

Trains two SVM models (classification and regression) to detect whether a sample is contaminated another sample of its same species.

**Usage**

```r
train_ct(feature)
```

**Arguments**

- `feature`  Feature list objects from `generate_feature()`

**Value**

A list contains two trained svm models: regression & classification
**update_vcf**

Remove CNV regions within VCF files

**Usage**

update_vcf(rmcnv = FALSE, vcf, cnvobj = NULL, threshold = 0.1, skew = 0.5, lower = 0.45, upper = 0.55)

**Arguments**

- **rmcnv**: Remove CNV regions, default is FALSE
- **vcf**: Input VCF files
- **cnvobj**: cnv object, default is NULL
- **threshold**: Threshold for allele frequency, default is 0.1
- **skew**: Skewness for allele frequency, default is 0.5
- **lower**: Lower bound for allele frequency region, default is 0.45
- **upper**: Upper bound for allele frequency region, default is 0.55

**Value**

VCF file without CNV region

---

**vcf_example**

*VCF example file.*

**Description**

An example containing a list of 4 data frames.

**Usage**

vcf_example

**Format**

A list of 4 data frames:

**Source**

Created by Tao Jiang
Index

*Topic datasets
  config_df, 2
  svm_class_model, 15
  svm_regression_model, 16
  vcf_example, 17

config_df, 2

generate_feature, 3
getAlt2, 4
getAnnoRate, 4
getAvgLL, 5
getLowDepth, 5
getRatio, 6
getSkewness, 6
getSNVRate, 7
getVar, 7

locateFile, 8

negll, 8

read_vcf, 11
readGATK, 9
readStrelka, 9
readVarDict, 10
readVarPROWL, 11
rho_est, 12
rmChangePoint, 13
rmCNVinVCF, 13

sssc, 14
summary_vcf, 15
svm_class_model, 15
svm_regression_model, 16

train_ct, 16

update_vcf, 17

vcf_example, 17