

Package ‘convaq’

April 30, 2018

Type Package

Title Perform CNV-based association studies

Version 0.1.1

Author Simon J. Larsen

Maintainer Simon J. Larsen <simonhffh@gmail.com>

Description

CoNVaQ is a package for finding CNV-regions significantly associated to a disease or phenotype. It provides two models: a standard statistical association model using Fisher's exact test and a query-based model, finding CNV regions matching user-specified predicates.

License MIT + file LICENSE

Encoding UTF-8

LazyData true

Imports Rcpp (>= 0.12.15)

LinkingTo Rcpp, BH

RoxygenNote 6.0.1

Suggests knitr, rmarkdown

VignetteBuilder knitr

URL <https://convaq.compbio.sdu.dk>

BugReports <https://github.com/SimonLarsen/convaq/issues>

R topics documented:

convaq	2
example	4
frequencies	5
print.convaq	5
regions	6
states	6

Index	7
--------------	----------

convaq	<i>Perform CNV-based association study.</i>
--------	---------------------------------------------

Description

CoNVaQ is a method for performing CNV-based association studies. It provides two models: a query-based model and a standard statistical model using Fisher's exact test.

convaq

Usage

```
convaq(segments1, segments2, model, name1 = "Group 1", name2 = "Group 2",
       qvalues = FALSE, qvalues.rep = 4000, merge = FALSE,
       merge.threshold = 0, p.cutoff = 0.05, pred1 = NULL, pred2 = NULL,
       nthreads = NULL)
```

Arguments

segments1	Data frame of segments for group 1. See details.
segments2	Data frame of segments for group 2. See details.
model	Model type. Either "statistical" or "query".
name1	Name of first group.
name2	Name of second group.
qvalues	TRUE if q-values should be computed, FALSE otherwise.
qvalues.rep	Number of repetitions to use in q-value computation.
merge	TRUE if adjacent regions of same type should be merged.
merge.threshold	Maximum number of base pairs allowed between two regions in order to be adjacent.
p.cutoff	(statistical model) P-value cutoff in statistical model.
pred1	(query model) Predicate for group 1 in query model.
pred2	(query model) Predicate for group 2 in query model.
nthreads	Number of threads to use. Defaults to number of cores available.

Value

An object of class `convaq` with the following elements:

regions	Data frame of significant regions.
freq	Data frame of within-group variation frequencies for each reported region.
state	The states of individual patients/samples for each region.
model	Type of model used.

name1	Name of first group.
name2	Name of second group.
qvalues	True if q-values were computed.
qvalues.rep	Number of repetitions used in q-value computation.
merge	True if adjacent regions of same type should be merged.
merge.threshold	Maximum distance (in base pairs) allowed between merged regions.
p.cutoff	P-value cutoff (statistical model only).
pred1	Predicate for group 1 (query model only).
pred2	Predicate for group 2 (query model only).

Segment data format

The CNV segment sets `segments1` and `segments2` must be data frame objects with the following five columns:

patient Patient identifier for the patient/sample the segment was found in.

chr Chromosome the segment is located in.

start First position of the segment in base pairs.

end Last position of the segment in base pairs.

type Segment type. One of "Gain", "Loss" or "LOH".

The order of the columns is used in order to determine the contents, not the column names.

Predicate format

Predicates are given as a character vector with the following format:

```
"[COMP] [FREQ] [EQ] [TYPE]"
```

where:

COMP is a comparison operator. One of "<" (less than), ">" (greater than), "<=" (less than or equal to) or ">=" (greater than or equal to).

FREQ is a numerical value between 0 and 1.

EQ is either "==" (equal to) or "!=" (not equal to).

TYPE is a segment type. One of "Gain", "Loss", "LOH" or "Normal".

The four arguments must be separated by a single space.

A predicate is evaluated for a specific group in a each region, and will either evaluate to TRUE or FALSE. If we want to find regions where at least 50% of the patients in a group are reported as "Gain", we can use the predicate:

```
">= 0.5 == Gain"
```

Likewise if we are searching for regions where less than 25% of the patients in a group have any kind of variation, we can use the predicate

```
"< 0.25 != Normal"
```

The query model works by combining two predicates, one for each group of segments. For instance, we can combine two predicates to search for regions where at least 50% of patients in the first group have a "Gain", and less than 25% of patients in the second group:

```
convaq(s1, s2, model="query", pred1=">= 0.5 == Gain", pred2="< 0.25 == Gain")
```

Author(s)

Simon J. Larsen <simonhffh@gmail.com>

Examples

```
data("example", package="convaq")
s1 <- example$disease
s2 <- example$healthy

# statistical model
convaq(s1, s2, model="statistical", p.cutoff=0.05, qvalues=FALSE)
convaq(s1, s2, model="statistical", p.cutoff=0.05, qvalues=TRUE, qvalues.rep=2000)

# query model
convaq(s1, s2, model="query", pred1=">= 0.5 == Gain", pred2="<= 0.2 == Gain")
convaq(s1, s2, model="query", pred1=">= 0.6 != Normal", pred2=">= 0.6 == Normal")
```

example

A small synthetic sample data set.

Description

A small data set containing generated segments to use with CoNVaQ.

Usage

```
example
```

Format

An object of class `list` of length 2.

Details

A list containing two data frames.

disease Data frame containing segments for the disease group

healthy Data frame containing segments for the healthy group

frequencies	<i>Extract variation frequencies for each region as a data frame.</i>
-------------	-----------------------------------------------------------------------

Description

Extract variation frequencies for each region as a data frame.

Usage

```
frequencies(x, ...)
```

Arguments

x	A convaq object.
...	Further arguments passed to or from other methods.

print.convaq	<i>Print description of CoNVaQ results object.</i>
--------------	----------------------------------------------------

Description

Print description of CoNVaQ results object.

Usage

```
## S3 method for class 'convaq'  
print(x, ...)
```

Arguments

x	A convaq object.
...	Further arguments passed to or from other methods.

regions	<i>Extract matching CNV regions from CoNVaQ result as a data frame.</i>
---------	-------------------------------------------------------------------------

Description

Extract matching CNV regions from CoNVaQ result as a data frame.

Usage

```
regions(x, ...)
```

Arguments

x	A convaq object.
...	Further arguments passed to or from other methods.

states	<i>Extract states of individual samples from CoNVaQ result as a data frame.</i>
--------	---------------------------------------------------------------------------------

Description

Extract states of individual samples from CoNVaQ result as a data frame.

Usage

```
states(x, ...)
```

Arguments

x	A convaq object.
...	Further arguments passed to or from other methods.

Index

*Topic **datasets**

example, [4](#)

convaq, [2](#)

convaq-package (convaq), [2](#)

example, [4](#)

frequencies, [5](#)

print.convaq, [5](#)

regions, [6](#)

states, [6](#)