gaia

October 25, 2011

`crc` *Real aCGH Dataset of Colorectal Cancer (CRC)*

Description

Dataset of CRC published by Venkatachalam et al. (Identification of candidate predisposing copy number variants in familial and early-onset colorectal cancer patients. Int. J. Cancer, 2010). The dataset contains 30 samples that were hybridized on SNP 250k Affymetrix GeneChip arrays. Raw data are available in GEO with identifier GSE13429.

Usage

```
data(crc)
```

Format

Data were preprocessed by PennCNV tool and discretized by Vega algorithm (Morganella et al. VEGA: variational segmentation for copy number detection. Bioinformatics, 2010) which is available as an R/Bioconductor package at the url (http://www.bioconductor.org/packages/devel/bioc/html/Vega.html).
Data are organized as a matrix having a row for each observed aberrant region. Each aberrant region is described by the following columns:
Sample Name - Chromosome - Start - End - Num of Markers - CN

Author(s)

Sandro Morganella et al.

Maintainer: S. Morganella <morganellaalx@gmail.com>

Examples

```
data(crc)
```

crc_markers — *Markers Descriptors for Real aCGH Dataset of Colorectal Cancer (CRC)*

Description

Markers Descriptors of CRC dataset for SNP 250k Affymetrix GeneChip arrays.

Usage

```
data(crc_markers)
```

Format

The marker descriptor matrix is organized as a matrix having a row for each measured probe. Each probe is described by the following columns:
Probe Name - Chromosome - Start

Author(s)

Sandro Morganella et al.

Maintainer: S. Morganella <morganellaalx@gmail.com>

Examples

```
data(crc_markers)
```

gaia — *GAIA: An R package for genomic analysis of significant chromosomal*

Description

GAIA (Genomic Analysis of Important Aberrations) allows to assess the statistical significance of chromosomal aberrations. A permutation test is used to compute the probability distribution of the normal case (no significant aberrations are present in the data) so that we can estimate the statistical significance of the observed data. In order to correct for multiple hypothesis testing the False Discovery Rate approach proposed by Storey et al. (2004) is used. Finally an iterative "peel-off" procedure is used to identify the most significant independent regions.

GAIA is described in Morganella et al. (2011).

Details

Package:	gaia
Type:	Package
Version:	1.0.1
Date:	2010-09-13
License:	GNU GPL
LazyLoad:	yes

Author(s)

Sandro Morganella et al.

Maintainer: S. Morganella <morganellaalx@gmail.com>

References

Morganella S. et al. (2011). Finding recurrent copy number alterations preserving within-sample homogeneity. Bioinformatics. DOI: 10.1093/bioinformatics/btr488.

Storey JD. et al. (2004). Strong control, conservative point estimation, and simultaneous conservative consistency of false discovery rates: A unified approach. Journal of the Royal Statistical Society. 66:187-205.

Examples

```
# Load the matrix containing the informations about the markers
data(synthMarkers_Matrix)

# Use the function load_markers to obtain the marker descriptor data object
markers_obj <- load_markers(synthMarkers_Matrix)

# Load the matrix containing the informations about the aberrant regions
data(synthCNV_Matrix)

# Use the function load_cnv to obtain the aberrant region descriptor data object
cnv_obj <- load_cnv(synthCNV_Matrix, markers_obj, 10)

# run GAIA algorithm and save the results within the file "results.txt"
runGAIA(cnv_obj, markers_obj, "results.txt")
```

---

generate_approx_null_hypothesis

This function computes the probability distribution of the null

---

Description

The probability distribution is computed by performing an approximation of independent random permutations of the rows. The null hypothesis is modeled as an histogram with a bin size of 1.

Author(s)

Sandro Morganella et al.

Maintainer: S. Morganella <morganellaalx@gmail.com>

`generate_null_hypothesis` *This function computes the probability distribution of the null*

Description

The probability distribution is computed by performing a number independent random permutation of the rows. The null hypothesis is modeled as an histogram with a bin size of 1.

Author(s)

Sandro Morganella et al.

Maintainer: S. Morganella <morganellaalx@gmail.com>

`load_cnv` *This function create the object containing all needed informations*

Description

This function loads the informations about the aberrant regions contained within the matrix passed as argument. It creates for all chromosomes and all kind of aberration (e.g. loss and gain) a matrix of dimension NxM (N observed samples and M observed probes).

Usage

```
load_cnv(segmentation_matrix, markers_list, num_of_samples)
```

Arguments

`segmentation_matrix`
: A matrix containing the aberrant regions where each row in the file reports the information of an aberrant region. In particular the matrix has the following column:
Sample Name - Chromosome - Start - End - Num of Markers - CN
"Sample Name" indicates the name of the sample. "Chromosome", "Start", "End", "Num of Markers" and "CN" indicate for each aberrant region the respective chromosome, the start and the end position the number of markers contained within the region and the found aberrations. Note that "CN" represents the estimated copy number for the segmented region and it must be an integer in the range 0..(K-1) where K is the number of the considered aberrations Therefore if we are considering only loss, LOH, gain in the file passed as argument the only possible kind of aberrations is 0, 1 and 2.

`markers_list` The marker descriptor object obtained by the function `load_markers`.

`num_of_samples`
: The number of analyzed samples.

Value

This function returns a list having the following structure:

`CNV_matrix_list[[i]][[j]]`
: contains the informations for the j-th chromosome on the i-th aberration. This element is a matrix of dimension NxM (N observed samples and M observed probes).

An example of the data produced by this function can be found in `synthCNV`

Author(s)

Sandro Morganella et al.

Maintainer: S. Morganella <morganellaalx@gmail.com>

Examples

```
# Load the matrix containing the informations about the markers
data(synthMarkers_Matrix)

# Use the function load_markers to obtain the marker descriptor data object
marks <- load_markers(synthMarkers_Matrix)

# Load the matrix containing the informations about the aberrant regions
data(synthCNV_Matrix)

# Use the function load_cnv to obtain the aberrant region descriptor data object
cnv <- load_cnv(synthCNV_Matrix, marks, 10)
```

---

load_markers *This function create the marker descriptor object containing all needed*

Description

This function loads the markers contained within the matrix passed as argument and creates for all chromosomes an ordered vector containing the position of each marker.These vectors are loaded within a list.

Usage

```
load_markers(marker_matrix)
```

Arguments

`marker_matrix`
: contains the marker descriptions as a matrix with the following structure:
Probe Name - Chromosome - Start - End
Note that the End position column is optional (in this case start and the end positions coincide). The sex chromosomes X and Y must be indicated with 23 and 24 respectively.

Value

This function returns a list having the following structure:

`chromosome_marker_list[[i]]`
: is a matrix of dimension 2xN (N is the number of observed probes for the i-th chromosome) the first and the second row contains the start and the end position of each marker of the i-th chromosome respectively.

An example of the data produced by this function can be found in `synthMarkers`.

Author(s)

Sandro Morganella et al.

Maintainer: S. Morganella <morganellaalx@gmail.com>

Examples

```
# Load the matrix containing the informations about the markers
data(synthMarkers_Matrix)

# Use the function load_markers to obtain the marker descriptor data object
marks <- load_markers(synthMarkers_Matrix)
```

---

peel_off *The iterative peel-off procedure to extract the independent peak*

Description

This function implements the peel-off algorithm to extract the independent regions having the minimum q-value (lower than the given threshold) within each chromosome. The function returns for each aberration and for each chromosome the list of aberrant regions. This function uses as support the function `search_peaks_in_region` that extract the primary peaks.

Note

This function uses the R package `qvalue` available at the bioconductor repository.
To install the `qvalue` package, start R and enter:
source("http://bioconductor.org/biocLite.R")
biocLite("qvalue")

Author(s)

Sandro Morganella et al.

Maintainer: S. Morganella <morganellaalx@gmail.com>

runGAIA *Run GAIA algorithm.*

Description

This function assess the significance of the chromosomal aberrations. Note that it uses the package qvalue.

Usage

```
runGAIA(cnv_obj, markers_obj, output_file_name, aberrations = -1, chromosomes =
```

Arguments

cnv_obj	an object returned by the function load_cnv describing the observed data.
markers_obj	an object returned by the function load_markers describing the observed markers.
output_file_name	the name of the file in which the significant aberrant regions are saved.
aberrations	[default=-1] the aberrations that will be analyzed. If it setted as -1 (default value) all aberrations will be analyzed.
chromosomes	[default=-1] the chromosomes that will be analyzed. If it setted as -1 (default value) all chromosomes will be analyzed.
num_iterations	[default=10] if the number of permutation steps (if approximation is equal to -1) - the number of column of the approximation matrix (if approximation is different to -1).
threshold	[default=0.25] markers having q-value lower than this threshold are labeled as significantly aberrant.
hom_threshold	[default=0.12] Threshold used for homogeneous peel-off. For values lower then 0 homogeneous peel-off is disabled.
approximation	[default=FALSE] if approximation is FALSE then GAIA explicitly performs the permutations, if it is TRUE then GAIA uses an approximated approach to compute the null distribution.

Value

This function return a matrix containing all significant aberrant regions.

Note

In order to execute this script you need of the R package qvalue available at the bioconductor repository.
To install the qvalue package, start R and enter:
source("http://bioconductor.org/biocLite.R")
biocLite("qvalue")

Author(s)

Sandro Morganella et al.

Maintainer: S. Morganella <morganellaalx@gmail.com>

References

GAIA home page: http://www.dsba.unisannio.it/Members/ceccarelli/GAIA

Examples

```
# Load the matrix containing the informations about the markers
data(synthMarkers_Matrix)

# Use the function load_markers to obtain the marker descriptor data object
markers_obj <- load_markers(synthMarkers_Matrix)

# Load the matrix containing the informations about the aberrant regions
data(synthCNV_Matrix)

# Use the function load_cnv to obtain the aberrant region descriptor data object
cnv_obj <- load_cnv(synthCNV_Matrix, markers_obj, 10)

# run GAIA algorithm and save the results within the file "results.txt"
runGAIA(cnv_obj, markers_obj, "results.txt")

# run GAIA algorithm in its approximated version generating 5000 approximations
runGAIA(cnv_obj, markers_obj, "results.txt", num_iterations=5000, approximation=TRUE)
```

---

search_peaks_in_regions

This function extracts the primary peak within a well-specified region.

Description

This function is used as support by the function `peel_off`. This function searches and returns the minimum q-value peak (the primary peak) within the region defined by the points `start_region` and `end_region`.

Author(s)

Sandro Morganella et al.

Maintainer: S. Morganella <morganellaalx@gmail.com>

synthCNV_Matrix *Example of aberration descriptor data matrix*

Description

This data matrix simulates 10 observations (samples) for 24 chromosomes (of 1000 probes) and for 3 kinds of aberrations 0, 1 and 2 with the following aberrant regions:

Chromosome 1: aberrant region of kind 0 from 301 to 700 in 100% of samples
Chromosome 2: aberrant region of kind 0 from 301 to 700 in 80% of samples
Chromosome 3: aberrant region of kind 0 from 301 to 700 in 60% of samples
Chromosome 4: aberrant region of kind 0 from 301 to 700 in 40% of samples
Chromosome 5: aberrant region of kind 0 from 301 to 700 in 20% of samples

Chromosome 10: aberrant region of kind 1 from 1 to 700 in 100% of samples
Chromosome 11: aberrant region of kind 1 from 1 to 700 in 80% of samples
Chromosome 12: aberrant region of kind 1 from 1 to 700 in 60% of samples
Chromosome 13: aberrant region of kind 1 from 1 to 700 in 40% of samples
Chromosome 14: aberrant region of kind 1 from 1 to 700 in 20% of samples

Chromosome 20: aberrant region of kind 2 from 801 to 1000 in 100% of samples
Chromosome 21: aberrant region of kind 2 from 801 to 1000 in 80% of samples
Chromosome 22: aberrant region of kind 2 from 801 to 1000 in 60% of samples
Chromosome 23: aberrant region of kind 2 from 801 to 1000 in 40% of samples
Chromosome 24: aberrant region of kind 2 from 801 to 1000 in 20% of samples

Usage

```
data(synthCNV_Matrix)
```

Format

This data matrix is organized as a matrix having a row for each observed aberrant region. Each aberrant region is described by the following columns:
Sample Name - Chromosome - Start - End - Num of Markers - CN
"Sample Name" indicates the name of the sample. "Chromosome", "Start", "End", "Num of Markers" and "CN" indicate for each aberrant region the respective chromosome, the start and the end position (in bp) the number of markers contained within the region and the found aberrations. Note that "CN" represents the estimated copy number for the segmented region and it must be an integer in the range 0..(K-1) where K is the number of the considered aberrations. In this data matrix three different aberration kinds are considered: 0, 1 and 2.

Author(s)

Sandro Morganella et al.

Maintainer: S. Morganella <morganellaalx@gmail.com>

Examples

```
data(synthCNV_Matrix)
```

`synthMarkers_Matrix`
Example of marker descriptor data matrix

Description

This matrix simulates the marker descriptors of 24 chromosomes (of 1000 probes) where the start and the end position of each probe coincide.

Usage

```
data(synthMarkers_Matrix)
```

Format

This data matrix is organized as a matrix having a row for each measured probe. Each probe is described by the following columns:
Probe Name - Chromosome - Start

Where: "Probe Name" is the name of the observed probe; "Chromosome" is the chromosome where the probe is located and "Start" is the start point (in bp) of the probe. Note that the matrix can have also a column ("End") specifying the position in which the probe ends, this column is optional and if it is missed than start and the end positions will coincide.
`synthMarkers_Matrix` reports the marker descriptor for 24 chromosomes of 1000 probes (where the sex chromosomes X and Y are indicated with 23 and 24 respectively.

Author(s)

Sandro Morganella et al.

Maintainer: S. Morganella <morganellaalx@gmail.com>

Examples

```
data(synthMarkers_Matrix)
```

`write_significant_regions` *This function writes into a tab delimited file the significant aberrant*

Description

This function writes into a tab delimited file the significant aberrant regions having the following format:
Chromosome - Aberration Kind - Region Start [bp] - Region End [bp] - Region Size [bp] - q-value
Chromosome: The chromosome where the aberration is found
Aberration Kind: The kind of the aberration found for the region
Region Start [bp]: The bp starting point of the aberrant region
Region End [bp]: The bp ending point of the aberrant region
Region Size [bp]: The size of the region in terms of bp
q-value: The q-value assessed for the aberrant region

Value

This function return a matrix containing all significant aberrant regions.

Author(s)

Sandro Morganella et al.

Maintainer: S. Morganella <morganellaalx@gmail.com>

Index