Package 'aroma.cn'

October 28, 2015

Version 1.6.1

Depends R (>= 3.1.1), R.utils (>= 2.1.0), aroma.core (>= 2.14.0)

Imports R.methodsS3 (>= 1.7.0), R.oo (>= 1.19.0), R.filesets (>= 2.9.0), R.cache (>= 0.10.0), matrixStats (>= 0.15.0), PSCBS (>= 0.50.0)

Suggests aroma.light (>= 2.2.1), DNAcopy (>= 1.40.0), GLAD (>= 1.12.0)

SuggestsNote BioC (>= 3.0), Recommended: aroma.light, DNAcopy

Date 2015-10-27

Title Copy-Number Analysis of Large Microarray Data Sets

Description Methods for analyzing DNA copy-number data. Specifically, this package implements the multi-source copy-number normalization (MSCN) method for normalizing copy-number data obtained on various platforms and technologies. It also implements the TumorBoost method for normalizing paired tumor-normal SNP data.

License LGPL (>= 2.1)

LazyLoad TRUE

biocViews ProprietaryPlatforms, aCGH, CopyNumberVariants, SNP, Microarray, OneChannel, TwoChannel, DataImport, DataRepresentation, Preprocessing, QualityControl

URL http://www.aroma-project.org/, https://github.com/HenrikBengtsson/aroma.cn

BugReports https://github.com/HenrikBengtsson/aroma.cn/issues

NeedsCompilation no

Author Henrik Bengtsson [aut, cre, cph], Pierre Neuvial [aut]

Maintainer Henrik Bengtsson <henrikb@braju.com>

Repository CRAN

Date/Publication 2015-10-28 00:08:16

R topics documented:

`aroma.cn-package` *Package aroma.cn*

Description

Methods for analyzing DNA copy-number data. Specifically, this package implements the multi-source copy-number normalization (MSCN) method for normalizing copy-number data obtained on various platforms and technologies. It also implements the TumorBoost method for normalizing paired tumor-normal SNP data.

This package should be considered to be in an alpha or beta phase. You should expect the API to be changing over time.

Installation and updates

To install this package, call `install.packages("aroma.cn")`.

To get started

To get started, see:

1. ...

License

The releases of this package is licensed under LGPL version 2.1 or newer.

The development code of the packages is under a private licence (where applicable) and patches sent to the author fall under the latter license, but will be, if incorporated, released under the "release" license above.

Author(s)

Henrik Bengtsson, Pierre Neuvial

References

Please cite aroma.cn one or more of approprite reference below

H. Bengtsson, P. Neuvial and T.P. Speed. TumorBoost: Normalization of allele-specific tumor copy numbers from a single pair of tumor-normal genotyping microarrays, BMC Bioinformatics, 2010

H. Bengtsson, A. Ray, P. Spellman and T.P. Speed. A single-sample method for normalizing and combining full-resolutioncopy numbers from multiple platforms, labs and analysis methods, Bioinformatics, 2009

H. Bengtsson; K. Simpson; J. Bullard; K. Hansen. aroma.affymetrix: A generic framework in R for analyzing small to very large Affymetrix data sets in bounded memory, Tech Report 745, Department of Statistics, University of California, Berkeley, February 2008

H. Bengtsson, R. Irizarry, B. Carvalho, & T.P. Speed. Estimation and assessment of raw copy numbers at the single locus level, Bioinformatics, 2008

`AbstractCurveNormalization`

The AbstractCurveNormalization class

Description

Package: aroma.cn
Class AbstractCurveNormalization

```
Object
~~|
~~+--AbstractCurveNormalization
```

Directly known subclasses:
PrincipalCurveNormalization, XYCurveNormalization

public abstract static class **AbstractCurveNormalization**
extends Object

Usage

```
AbstractCurveNormalization(dataSet=NULL, targetSet=NULL, subsetToFit=NULL, tags="*",
  copyTarget=TRUE, ...)
```

Arguments

`dataSet`	An `AromaUnitTotalCnBinarySet` of "test" samples to be normalized.
`targetSet`	An `AromaUnitTotalCnBinarySet` of paired target samples.
`subsetToFit`	The subset of loci to be used to fit the normalization functions. If `NULL`, loci on chromosomes 1-22 are used, but not on ChrX and ChrY.

tags	(Optional) Sets the tags for the output data sets.
copyTarget	If TRUE, target arrays are copied to the output data set, otherwise not.
...	Not used.

Fields and Methods

Methods:

getFullName	-
getInputDataSet	-
getName	-
getOutputDataSet	-
getTags	-
getTargetDataSet	-
process	-
setTags	-

Methods inherited from Object:
$, $<-, [[, [[<-, as.character, attach, attachLocally, clearCache, clearLookupCache, clone, detach, equals, extend, finalize, getEnvironment, getFieldModifier, getFieldModifiers, getFields, getInstantiationTime, getStaticInstance, hasField, hashCode, ll, load, objectSize, print, save, asThis

Author(s)

Henrik Bengtsson

---

callXXorXY.numeric *Calls XX or XY from ChrX allele B fractions of a normal sample*

Description

Calls XX or XY from ChrX allele B fractions of a normal sample.

Usage

```
## S3 method for class 'numeric'
callXXorXY(betaX, betaY=NULL, flavor=c("density"), adjust=1.5, ...,
  censorAt=c(-0.5, +1.5), verbose=FALSE)
```

Arguments

betaX	A numeric vector containing ChrX allele B fractions.
betaY	A optional numeric vector containing ChrY allele B fractions.
flavor	A character string specifying the type of algorithm used.

adjust	A postive double specifying the amount smoothing for the empirical density estimator.
...	Additional arguments passed to findPeaksAndValleys.
censorAt	A double vector of length two specifying the range for which values are considered finite. Values below (above) this range are treated as -Inf (+Inf).
verbose	A logical or a Verbose object.

Value

Returns a ...

Missing and non-finite values

Missing and non-finite values are dropped before trying to call XX or XY.

Author(s)

Henrik Bengtsson, Pierre Neuvial

See Also

Internally findPeaksAndValleys is used to identify the thresholds.

MultiSourceCopyNumberNormalization
The MultiSourceCopyNumberNormalization class

Description

Package: aroma.cn
Class MultiSourceCopyNumberNormalization

```
Object
~~|
~~+--ParametersInterface
~~~~~~~|
~~~~~~~+--MultiSourceCopyNumberNormalization
```

Directly known subclasses:

public static class **MultiSourceCopyNumberNormalization**
extends ParametersInterface

The multi-source copy-number normalization (MSCN) method [1] is a normalization method that normalizes copy-number estimates measured by multiple sites and/or platforms for common samples. It normalizes the estimates toward a common scale such that for any copy-number level the mean level of the normalized data are the same.

Usage

```
MultiSourceCopyNumberNormalization(dsList=NULL, fitUgp=NULL, subsetToFit=NULL,
  targetDimension=1, align=c("byChromosome", "none"), tags="*", ...)
```

Arguments

dsList	A list of K AromaUnitTotalCnBinarySet:s.
fitUgp	An AromaUgpFile that specifies the common set of loci used to normalize the data sets at.
subsetToFit	The subset of loci (as mapped by the fitUgp object) to be used to fit the normalization functions. If NULL, loci on chromosomes 1-22 are used, but not on ChrX and ChrY.
targetDimension	A numeric index specifying the data set in dsList to which each platform in standardize towards. If NULL, the arbitrary scale along the fitted principal curve is used. This always starts at zero and increases.
align	A character specifying type of alignment applied, if any. If "none", no alignment is done. If "byChromosome", the signals are shifted chromosome by chromosome such the corresponding smoothed signals have the same median signal across sources. For more details, see below.
tags	(Optional) Sets the tags for the output data sets.
...	Not used.

Details

The multi-source normalization method is by nature a single-sample method, that is, it normalizes arrays for one sample at the time and independently of all other samples/arrays.

However, the current implementation is such that it first generates smoothed data for *all* samples/arrays. Then, it normalizes the sample one by one.

Fields and Methods

Methods:

```
getAllNames        -
getAsteriskTags    -
getInputDataSets   -
getOutputDataSets  -
getTags            -
nbrOfDataSets      -
process            -
```

Methods inherited from ParametersInterface:
getParameterSets, getParameters, getParametersAsString

Methods inherited from Object:
$, $<-, [[, [[<-, as.character, attach, attachLocally, clearCache, clearLookupCache, clone, detach, equals, extend, finalize, getEnvironment, getFieldModifier, getFieldModifiers, getFields, getInstantiationTime, getStaticInstance, hasField, hashCode, ll, load, objectSize, print, save, asThis

Different preprocessing methods normalize ChrX \& ChrY differently

Some preprocessing methods estimate copy numbers on sex chromosomes differently from the autosomal chromosomes. The way this is done may vary from method to method and we cannot assume anything about what approach is. This is the main reason why the estimation of the normalization function is by default based on signals from autosomal chromosomes only; this protects the estimate of the function from being biased by specially estimated sex-chromosome signals. Note that the normalization function is still applied to all chromosomes.

This means that if the transformation applied by a particular preprocessing method is not the same for the sex chromosomes as the autosomal chromosomes, the normalization applied on the sex chromosomes is not optimal one. This is why multi-source normalization sometimes fails to bring sex-chromosome signals to the same scale across sources. Unfortunately, there is no automatic way to handle this. The only way would be to fit a specific normalization function to each of the sex chromosomes, but that would require that there exist copy-number abberations on those chromosomes, which could be a too strong assumption.

A more conservative approach is to normalize the signals such that afterward the median of the smoothed copy-number levels are the same across sources for any particular chromosome. This is done by setting argument `align="byChromosome"`.

Author(s)

Henrik Bengtsson

References

[1] H. Bengtsson, A. Ray, P. Spellman & T.P. Speed, *A single-sample method for normalizing and combining full-resolution copy numbers from multiple platforms, labs and analysis methods*, Bioinformatics 2009.

`PairedPscbsModel` *The PairedPscbsModel class*

Description

Package: aroma.cn
Class PairedPscbsModel

Object
~~|

```
~~+--ParametersInterface
~~~~~~~|
~~~~~~~+--PairedPscbsModel
```

Directly known subclasses:

public static class **PairedPscbsModel**
extends ParametersInterface

This class represents the Paired PSCBS method [1], which segments matched tumor-normal parental copy-number data into piecewise constant segments.

Usage

```
PairedPscbsModel(dsT=NULL, dsN=NULL, tags="*", ..., dropTcnOutliers=TRUE,
  gapMinLength=1e+06, seed=NULL)
```

Arguments

dsT, dsN	The tumor and the normal AromaUnitPscnBinarySet.
tags	Tags added to the output data sets.
...	(Optional) Additional arguments passed to segmentByPairedPSCBS.
dropTcnOutliers	If TRUE, then TCN outliers are dropped using dropSegmentationOutliers.
gapMinLength	Genomic regions with no data points that are of this length and greater are considered to be "gaps" and are ignored in the segmentation. If +Inf, no gaps are identified.
seed	An optional integer specifying the random seed to be used in the segmentation. Seed needs to be set for exact numerical reproducibility.

Fields and Methods

Methods:

```
fit               -
getChipType       -
getChromosomes    -
getDataSets       -
getFullName       -
getName           -
getNormalDataSet  -
getOutputDataSet  -
getTags           -
getTumorDataSet   -
indexOf           -
```

```
nbrOfFiles        -
setTags           -
```

Methods inherited from ParametersInterface:
getParameterSets, getParameters, getParametersAsString

Methods inherited from Object:
$, $<-, [[, [[<-, as.character, attach, attachLocally, clearCache, clearLookupCache, clone, detach, equals, extend, finalize, getEnvironment, getFieldModifier, getFieldModifiers, getFields, getInstantiationTime, getStaticInstance, hasField, hashCode, ll, load, objectSize, print, save, asThis

References

[1] ...

See Also

...

Examples

```
## Not run:
  dataSet <- "GSE12702"
tags <- "ASCRMAv2"
chipType <- "Mapping250K_Nsp"
ds <- AromaUnitPscnBinarySet$byName(dataSet, tags=tags, chipType=chipType)
print(ds)

# Extract tumors and normals
idxs <- seq(from=1, to=nbrOfFiles(ds), by=2)
dsT <- extract(ds, idxs);
idxs <- seq(from=2, to=nbrOfFiles(ds), by=2)
dsN <- extract(ds, idxs);

# Setup Paired PSCBS model
seg <- PairedPscbsModel(dsT=dsT, dsN=dsN)
print(seg)

# Segment all tumor-normal pairs
fit(seg, verbose=-10)


## End(Not run)
```

PrincipalCurveNormalization
The PrincipalCurveNormalization class

Description

Package: aroma.cn
Class PrincipalCurveNormalization

```
Object
~~|
~~+--AbstractCurveNormalization
~~~~~~~|
~~~~~~~+--PrincipalCurveNormalization
```

Directly known subclasses:

public static class **PrincipalCurveNormalization**
extends *AbstractCurveNormalization*

Usage

```
PrincipalCurveNormalization(..., subset=1/20)
```

Arguments

`...`	Arguments passed to `AbstractCurveNormalization`.
`subset`	A `double` in (0,1] specifying the fraction of the `subsetToFit` to be used for fitting. Since the fit function for this class is rather slow, the default is to use a 1/20:th of the default data points.

Fields and Methods

Methods:
No methods defined.

Methods inherited from AbstractCurveNormalization:
as.character, backtransformOne, fitOne, getAsteriskTags, getDataSets, getFullName, getInputDataSet, getName, getOutputDataSet, getPairedDataSet, getPath, getRootPath, getSubsetToFit, getTags, getTargetDataSet, nbrOfFiles, process, setTags

Methods inherited from Object:
$, $<-, [[, [[<-, as.character, attach, attachLocally, clearCache, clearLookupCache, clone, detach, equals, extend, finalize, getEnvironment, getFieldModifier, getFieldModifiers, getFields, getInstantiationTime, getStaticInstance, hasField, hashCode, ll, load, objectSize, print, save, asThis

Author(s)

Henrik Bengtsson

`TotalCnBinnedSmoothing`
The TotalCnBinnedSmoothing class

Description

Package: aroma.cn
Class TotalCnBinnedSmoothing

```
Object
~~|
~~+--ParametersInterface
~~~~~~~|
~~~~~~~+--AromaTransform
~~~~~~~~~~~|
~~~~~~~~~~~+--TotalCnSmoothing
~~~~~~~~~~~~~~~~|
~~~~~~~~~~~~~~~~+--TotalCnBinnedSmoothing
```

Directly known subclasses:

public static class **TotalCnBinnedSmoothing**
extends *TotalCnSmoothing*

Usage

```
TotalCnBinnedSmoothing(..., robust=FALSE)
```

Arguments

`...`	Arguments passed to `TotalCnSmoothing`.
`robust`	If `TRUE`, a robust smoother is used, otherwise not.

Details

Note that `dsS <- TotalCnBinnedSmoothing(ds, targetUgp=ugp)` where `ugp <- getAromaUgpFile(ds)` returns a data set with an identical set of loci as the input data set and identical signals as the input ones, *except* for loci with duplicated positions. If all loci have unique positions, the the output is identical to the input.

Fields and Methods

Methods:
No methods defined.

Methods inherited from TotalCnSmoothing:
getAsteriskTags, getOutputDataSet0, getOutputFileClass, getOutputFileExtension, getOutputFileSetClass, getOutputFiles, getParameters, getPath, getRootPath, getTargetPositions, getTargetUgpFile, process, smoothRawCopyNumbers

Methods inherited from AromaTransform:
as.character, findFilesTodo, getAsteriskTags, getExpectedOutputFiles, getExpectedOutputFullnames, getFullName, getInputDataSet, getName, getOutputDataSet, getOutputDataSet0, getOutputFiles, getPath, getRootPath, getTags, isDone, process, setTags

Methods inherited from ParametersInterface:
getParameterSets, getParameters, getParametersAsString

Methods inherited from Object:
$, $<-, [[, [[<-, as.character, attach, attachLocally, clearCache, clearLookupCache, clone, detach, equals, extend, finalize, getEnvironment, getFieldModifier, getFieldModifiers, getFields, getInstantiationTime, getStaticInstance, hasField, hashCode, ll, load, objectSize, print, save, asThis

Author(s)

Henrik Bengtsson

`TotalCnKernelSmoothing`
The TotalCnKernelSmoothing class

Description

Package: aroma.cn
Class TotalCnKernelSmoothing

```
Object
~~|
~~+--ParametersInterface
~~~~~~~|
~~~~~~~+--AromaTransform
~~~~~~~~~~~~|
~~~~~~~~~~~~+--TotalCnSmoothing
~~~~~~~~~~~~~~~~~|
~~~~~~~~~~~~~~~~~+--TotalCnKernelSmoothing
```

Directly known subclasses:

public static class **TotalCnKernelSmoothing**
extends *TotalCnSmoothing*

Usage

```
TotalCnKernelSmoothing(..., kernel=c("gaussian", "uniform"), bandwidth=50000, censorH=3,
  robust=FALSE)
```

Arguments

`...`	Arguments passed to `TotalCnSmoothing`.
`kernel`	A `character` string specifying the type of kernel to be used.
`bandwidth`	A `double` specifying the bandwidth of the smoothing.
`censorH`	A positive `double` specifying the bandwidth threshold where values outside are ignored (zero weight).
`robust`	If `TRUE`, a robust smoother is used, otherwise not.

Fields and Methods

Methods:
No methods defined.

Methods inherited from TotalCnSmoothing:
getAsteriskTags, getOutputDataSet0, getOutputFileClass, getOutputFileExtension, getOutputFileSetClass, getOutputFiles, getParameters, getPath, getRootPath, getTargetPositions, getTargetUgpFile, process, smoothRawCopyNumbers

Methods inherited from AromaTransform:
as.character, findFilesTodo, getAsteriskTags, getExpectedOutputFiles, getExpectedOutputFullnames, getFullName, getInputDataSet, getName, getOutputDataSet, getOutputDataSet0, getOutputFiles, getPath, getRootPath, getTags, isDone, process, setTags

Methods inherited from ParametersInterface:
getParameterSets, getParameters, getParametersAsString

Methods inherited from Object:
$, $<-, [[, [[<-, as.character, attach, attachLocally, clearCache, clearLookupCache, clone, detach, equals, extend, finalize, getEnvironment, getFieldModifier, getFieldModifiers, getFields, getInstantiationTime, getStaticInstance, hasField, hashCode, ll, load, objectSize, print, save, asThis

Author(s)

Henrik Bengtsson

TotalCnSmoothing *The abstract TotalCnSmoothing class*

Description

Package: aroma.cn
Class TotalCnSmoothing

```
Object
~~|
~~+--ParametersInterface
~~~~~~~|
~~~~~~~+--AromaTransform
~~~~~~~~~~~~|
~~~~~~~~~~~~+--TotalCnSmoothing
```

Directly known subclasses:
TotalCnBinnedSmoothing, TotalCnKernelSmoothing

public abstract static class **TotalCnSmoothing**
extends *AromaTransform*

Usage

```
TotalCnSmoothing(dataSet=NULL, ..., targetUgp=NULL,
  .reqSetClass="AromaUnitTotalCnBinarySet")
```

Arguments

dataSet	An AromaUnitTotalCnBinarySet.
...	Arguments passed to AromaTransform.
targetUgp	An AromaUgpFile specifying the target loci for which smoothed copy-number are generated.
.reqSetClass	(internal only)

Fields and Methods

Methods:

getTargetUgpFile	-
process	-

Methods inherited from AromaTransform:
as.character, findFilesTodo, getAsteriskTags, getExpectedOutputFiles, getExpectedOutputFullnames, getFullName, getInputDataSet, getName, getOutputDataSet, getOutputDataSet0, getOutputFiles, getPath, getRootPath, getTags, isDone, process, setTags

Methods inherited from ParametersInterface:
getParameterSets, getParameters, getParametersAsString

Methods inherited from Object:
$, $<-, [[, [[<-, as.character, attach, attachLocally, clearCache, clearLookupCache, clone, detach, equals, extend, finalize, getEnvironment, getFieldModifier, getFieldModifiers, getFields, getInstantiationTime, getStaticInstance, hasField, hashCode, ll, load, objectSize, print, save, asThis

Author(s)

Henrik Bengtsson

`TumorBoostNormalization`
The TumorBoostNormalization class

Description

Package: aroma.cn
Class TumorBoostNormalization

```
Object
~~|
~~+--TumorBoostNormalization
```

Directly known subclasses:

public static class **TumorBoostNormalization**
extends Object

TumorBoost is normalization method that normalizes the allele B fractions of a tumor sample given the allele B fractions and genotype calls for a matched normal. The method is a single-sample (single-pair) method. It does not require total copy number estimates. The normalization is done such that the total copy number is unchanged afterwards.

Usage

```
TumorBoostNormalization(dsT=NULL, dsN=NULL, gcN=NULL, flavor=c("v4", "v3", "v2", "v1"),
  preserveScale=TRUE, collapseHomozygous=FALSE, tags="*", ...)
```

Arguments

dsT	An AromaUnitFracBCnBinarySet of tumor samples.
dsN	An AromaUnitFracBCnBinarySet of match normal samples.
gcN	An AromaUnitGenotypeCallSet of genotypes for the normals.
flavor	A character string specifying the type of correction applied.
preserveScale	If TRUE, SNPs that are heterozygous in the matched normal are corrected for signal compression using an estimate of signal compression based on the amount of correction performed by TumorBoost on SNPs that are homozygous in the matched normal.
collapseHomozygous	If TRUE, SNPs that are homozygous in the matched normal are also called homozygous in the tumor, that is, it's allele B fraction is collapsed to either 0 or 1. If FALSE, the homozygous values are normalized according the model. [NOT USED YET]
tags	(Optional) Sets the tags for the output data sets.
...	Not used.

Fields and Methods

Methods:

getFullName	-
getInputDataSet	-
getName	-
getNormalDataSet	-
getNormalGenotypeCallSet	-
getOutputDataSet	-
getTags	-
nbrOfFiles	-
process	-
setTags	-

Methods inherited from Object:
$, $<-, [[, [[<-, as.character, attach, attachLocally, clearCache, clearLookupCache, clone, detach, equals, extend, finalize, getEnvironment, getFieldModifier, getFieldModifiers, getFields, getInstantiationTime, getStaticInstance, hasField, hashCode, ll, load, objectSize, print, save, asThis

Author(s)

Henrik Bengtsson, Pierre Neuvial

`XYCurveNormalization` *The XYCurveNormalization class*

Description

Package: aroma.cn
Class XYCurveNormalization

```
Object
~~|
~~+--AbstractCurveNormalization
~~~~~~~|
~~~~~~~+--XYCurveNormalization
```

Directly known subclasses:

public static class **XYCurveNormalization**
extends *AbstractCurveNormalization*

Usage

```
XYCurveNormalization(...)
```

Arguments

`...`	Arguments passed to `AbstractCurveNormalization`.

Fields and Methods

Methods:
No methods defined.

Methods inherited from AbstractCurveNormalization:
as.character, backtransformOne, fitOne, getAsteriskTags, getDataSets, getFullName, getInputDataSet, getName, getOutputDataSet, getPairedDataSet, getPath, getRootPath, getSubsetToFit, getTags, getTargetDataSet, nbrOfFiles, process, setTags

Methods inherited from Object:
$, $<-, [[, [[<-, as.character, attach, attachLocally, clearCache, clearLookupCache, clone, detach, equals, extend, finalize, getEnvironment, getFieldModifier, getFieldModifiers, getFields, getInstantiationTime, getStaticInstance, hasField, hashCode, ll, load, objectSize, print, save, asThis

Author(s)

Henrik Bengtsson

Index