

# Package ‘aroma.core’

August 19, 2025

**Version** 3.3.2

**Depends** R (>= 3.2.1), R.utils (>= 2.10.1), R.filesets (>= 2.14.0),  
R.devices (>= 2.16.1)

**DependsNote** BioC (>= 3.2)

**Imports** stats, tools, utils, R.methodsS3 (>= 1.8.1), R.oo (>= 1.24.0),  
R.cache (>= 0.14.0), R.rsp (>= 0.44.0), matrixStats (>=  
0.57.0), RColorBrewer (>= 1.1-2), PSCBS (>= 0.65.0), listenv,  
future, BiocManager

**Suggests** KernSmooth, png (>= 0.1-7), Cairo (>= 1.5-9), EBImage (>=  
4.8.3), preprocessCore (>= 1.28.0), aroma.light (>= 2.2.1),  
DNAcopy (>= 1.40.0), GLAD (>= 2.30.0), sfit (>= 0.1.8),  
expectile (>= 0.2.5), HaarSeg (>= 0.0.2), mpcbs (>= 1.1.1)

**SuggestsNote** BioC (>= 3.0), Recommended: aroma.light, DNAcopy, png,  
preprocessCore, sfit

**Additional\_repositories** <https://henrikbengtsson.r-universe.dev>,  
<https://r-forge.r-project.org>

**Title** Core Methods and Classes Used by 'aroma.\*' Packages Part of the  
Aroma Framework

**Description** Core methods and classes used by higher-level 'aroma.\*' packages  
part of the Aroma Project, e.g. 'aroma.affymetrix' and 'aroma.cn'.

**License** LGPL (>= 2.1)

**URL** <https://github.com/HenrikBengtsson/aroma.core>,  
<https://www.aroma-project.org/>

**BugReports** <https://github.com/HenrikBengtsson/aroma.core/issues>

**LazyLoad** TRUE

**biocViews** Microarray, OneChannel, TwoChannel, MultiChannel,  
DataImport, DataRepresentation, GUI, Visualization,  
Preprocessing, QualityControl, aCGH, CopyNumberVariants

**NeedsCompilation** no

**Author** Henrik Bengtsson [aut, cre, cph],  
 Mark Robinson [ctb],  
 Ken Simpson [ctb]  
**Maintainer** Henrik Bengtsson <henrikb@braju.com>  
**Repository** CRAN  
**Date/Publication** 2025-08-18 23:10:02 UTC

## Contents

aroma.core-package . . . . .	3
AbstractCNData . . . . .	4
AbstractPSCNData . . . . .	6
AromaCellCpgFile . . . . .	7
AromaCellPositionFile . . . . .	8
AromaCellTabularBinaryFile . . . . .	9
AromaGenomeTextFile . . . . .	11
AromaMicroarrayDataFile . . . . .	14
AromaMicroarrayDataSet . . . . .	15
AromaMicroarrayDataSetTuple . . . . .	17
AromaMicroarrayTabularBinaryFile . . . . .	18
AromaPlatform . . . . .	21
AromaPlatformInterface . . . . .	22
AromaRepository . . . . .	23
AromaTabularBinaryFile . . . . .	25
AromaTabularBinarySet . . . . .	27
AromaTransform . . . . .	29
AromaUnitCallFile . . . . .	30
AromaUnitCallSet . . . . .	32
AromaUnitFracBCnBinaryFile . . . . .	34
AromaUnitFracBCnBinarySet . . . . .	37
AromaUnitGenotypeCallFile . . . . .	39
AromaUnitGenotypeCallSet . . . . .	41
AromaUnitPscnBinaryFile . . . . .	43
AromaUnitPscnBinarySet . . . . .	46
AromaUnitSignalBinaryFile . . . . .	48
AromaUnitSignalBinarySet . . . . .	50
AromaUnitTabularBinaryFile . . . . .	52
AromaUnitTotalCnBinaryFile . . . . .	54
AromaUnitTotalCnBinarySet . . . . .	57
AromaUnitTypesFile . . . . .	59
BinnedScatter . . . . .	61
CbsModel . . . . .	63
ChromosomalModel . . . . .	65
ChromosomeExplorer . . . . .	66
colBinnedSmoothing.matrix . . . . .	68
colKernelSmoothing.matrix . . . . .	71
CopyNumberChromosomalModel . . . . .	73

CopyNumberSegmentationModel . . . . .	74
doCBS . . . . .	76
Explorer . . . . .	77
GladModel . . . . .	78
HaarSegModel . . . . .	80
NonPairedPSCNData . . . . .	82
PairedPSCNData . . . . .	84
ParametersInterface . . . . .	87
RawAlleleBFractions . . . . .	88
RawCopyNumberModel . . . . .	89
RawCopyNumbers . . . . .	90
RawGenomicSignals . . . . .	94
RawMirroredAlleleBFractions . . . . .	97
RawSequenceReads . . . . .	98
SegmentedAlleleBFractions . . . . .	100
SegmentedCopyNumbers . . . . .	102
SegmentedGenomicSignalsInterface . . . . .	105
UnitAnnotationDataFile . . . . .	106
UnitNamesFile . . . . .	107
UnitTypesFile . . . . .	108

<b>Index</b>	<b>110</b>
--------------	------------

---

aroma.core-package	<i>Package aroma.core</i>
--------------------	---------------------------

---

## Description

Core methods and classes used by higher-level 'aroma.\*' packages part of the Aroma Project, e.g. 'aroma.affymetrix' and 'aroma.cn'.

This package is a support package for **aroma.affymetrix**. This package should be considered to be in an alpha or beta phase. You should expect the API to be changing over time. For practical purposes, consider this package a private package.

## Installation and updates

This package is available on CRAN, i.e. to install do `install.packages("aroma.core")`.

## 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

## References

[1] H. Bengtsson, *The R.oo package - Object-Oriented Programming with References Using Standard R Code*, In Kurt Hornik, Friedrich Leisch and Achim Zeileis, editors, Proceedings of the 3rd International Workshop on Distributed Statistical Computing (DSC 2003), March 20-22, Vienna, Austria. <https://www.r-project.org/conferences/DSC-2003/Proceedings/>

---

 AbstractCNData

*The AbstractCNData class*


---

## Description

Package: aroma.core

### Class AbstractCNData

```

data.frame
~~|
~~+--RichDataFrame
~~~~~|
~~~~~+--RawGenomicSignals
~~~~~|
~~~~~+--AbstractCNData
  
```

### Directly known subclasses:

[AbstractPSCNData](#), [NonPairedPSCNData](#), [PairedPSCNData](#)

```

public class AbstractCNData
  extends RawGenomicSignals
  
```

An AbstractCNData object holds copy number data.

## Usage

```
AbstractCNData(chromosome=NULL, x=NULL, y=NULL, ..., name=NULL, .virtuals=NULL)
```

## Arguments

chromosome	(Optional) An <a href="#">integer</a> scalar (or a <a href="#">vector</a> of length J), which can be used to specify which chromosome each locus belongs to in case multiple chromosomes are segments. This argument is also used for annotation purposes.
x	Optional <a href="#">numeric vector</a> of J genomic locations. If <a href="#">NULL</a> , index locations 1:J are used.
y	Optional <a href="#">numeric vector</a> of J genomic locations.
...	Optional named locus-specific signal <a href="#">vectors</a> of length J.
name	Optional <a href="#">character</a> string.
.virtuals	(Internal) a <a href="#">list</a> with virtual column name <a href="#">functions</a> .

## Fields and Methods

### Methods:

```

findLargeGaps -
getChipType -
getLocusData -
getPlatform -
setChipType -
setPlatform -

```

### Methods inherited from RawGenomicSignals:

\*, +, -, addBy, append, applyBinaryOperator, as.character, as.data.frame, assertOneChromosome, binnedSmoothing, binnedSmoothingByField, clearCache, clone, divideBy, drawDensity, estimateStandardDeviation, extractChromosome, extractChromosomes, extractDataForSegmentation, extractRegion, extractRegions, extractSubset, gaussianSmoothing, getBasicField, getCXY, getChromosome, getChromosomes, getDefaultLocusFields, getLocusFields, getPositions, getSigma, getSignalColumnName, getSignalColumnNames, getSignals, getWeights, getXScale, getXY, getYScale, hasWeights, kernelSmoothing, lines, multiplyBy, nbrOfChromosomes, nbrOfLoci, plot, points, print, segmentByCBS, segmentByGLAD, segmentByHaarSeg, segmentByMPCBS, setBasicField, setSigma, setSignals, setWeights, setXScale, setYScale, signalRange, sort, subtractBy, xMax, xMin, xRange, xSeq, yMax, yMin, yRange

### Methods inherited from RichDataFrame:

\$, \$<-, [, [[, [[<-, as.data.frame, as.list, dim, dropVirtualColumn, getColumnNames, getColumnNamesTranslator, getFullName, getName, getTags, getVirtualColumn, getVirtualColumnFunction, getVirtualColumnNames, hasColumn, hasColumns, hasVirtualColumn, hasVirtualColumns, length, names, newInstance, print, rbind, setAttributes, setColumnNamesMap, setColumnNamesTranslator, setName, setTags, setVirtualColumn, subset, translateColumnNames

### Methods inherited from data.frame:

\$<-, data.frame-method, \$<-, Math, Ops,nonStructure,vector-method, Ops,structure,vector-method, Ops,vector,nonStructure-method, Ops,vector,structure-method, Ops, Summary, [, [<-, data.frame-method, [<-, [[, [[<-, data.frame-method, [[<-, aggregate, anyDuplicated, anyNA, as.NonPairedPSCNData, as.PairedPSCNData, as.data.frame, as.list, as.matrix, as.vector, attachLocally, by, callSegmentationOutliers, cbind, coerce,ANY,list-method, coerce,oldClass,S3-method, dim, dimnames, dimnames<-, dropSegmentationOutliers, droplevels, duplicated, edit, findLargeGaps, format, formula, head, initialize,oldClass-method, is.na, merge, na.exclude, na.omit, plot, print, prompt, rbind, row.names, row.names<-, rowsum, segmentByCBS, segmentByPairedPSCBS, show,oldClass-method, slotsFromS3,data.frame-method, sort\_by, split, split<-, stack, str, subset, summary, t, tail, transform, type.convert, unique, unstack, unwrap, within, wrap, writeDataFrame, xtfm

### Author(s)

Henrik Bengtsson

---

AbstractPSCNData      *The AbstractPSCNData class*

---

## Description

Package: aroma.core

### Class AbstractPSCNData

```

data.frame
~~|
~~+--RichDataFrame
~~~~~|
~~~~~+--RawGenomicSignals
~~~~~|
~~~~~+--AbstractCNData
~~~~~|
~~~~~+--AbstractPSCNData

```

#### Directly known subclasses:

[NonPairedPSCNData](#), [PairedPSCNData](#)

```

public class AbstractPSCNData
extends AbstractCNData

```

A AbstractPSCNData object holds parent-specific copy number data.

## Usage

```
AbstractPSCNData(chromosome=NULL, x=NULL, isSNP=NULL, mu=NULL, ...)
```

## Arguments

chromosome	(Optional) An <a href="#">integer</a> scalar (or a <a href="#">vector</a> of length J), which can be used to specify which chromosome each locus belongs to in case multiple chromosomes are segments. This argument is also used for annotation purposes.
x	Optional <a href="#">numeric vector</a> of J genomic locations. If <a href="#">NULL</a> , index locations 1 : J are used.
isSNP	An optional <a href="#">logical vector</a> of length J specifying whether each locus is a SNP or not (non-polymorphic loci).
mu	An optional <a href="#">numeric vector</a> of J genotype calls in {0,1/2,1} for AA, AB, and BB, respectively, and <a href="#">NA</a> for non-polymorphic loci.
...	Optional named locus-specific signal <a href="#">vectors</a> of length J.

## Fields and Methods

### Methods:

*No methods defined.*

### Methods inherited from AbstractCNData:

findLargeGaps, getChipType, getLocusData, getPlatform, hasKnownPositions, orderAlongGenome, setChipType, setPlatform

### Methods inherited from RawGenomicSignals:

\*, +, -, addBy, append, applyBinaryOperator, as.character, as.data.frame, assertOneChromosome, binnedSmoothing, binnedSmoothingByField, clearCache, clone, divideBy, drawDensity, estimateStandardDeviation, extractChromosome, extractChromosomes, extractDataForSegmentation, extractRegion, extractRegions, extractSubset, gaussianSmoothing, getBasicField, getCXY, getChromosome, getChromosomes, getDefaultLocusFields, getLocusFields, getPositions, getSigma, getSignalColumnName, getSignalColumnNames, getSignals, getWeights, getXScale, getXY, getYScale, hasWeights, kernelSmoothing, lines, multiplyBy, nbrOfChromosomes, nbrOfLoci, plot, points, print, segmentByCBS, segmentByGLAD, segmentByHaarSeg, segmentByMPCBS, setBasicField, setSigma, setSignals, setWeights, setXScale, setYScale, signalRange, sort, subtractBy, xMax, xMin, xRange, xSeq, yMax, yMin, yRange

### Methods inherited from RichDataFrame:

\$, \$<-, [, [[, [[<-, as.data.frame, as.list, dim, dropVirtualColumn, getColumnNames, getColumnNamesTranslator, getFullName, getName, getTags, getVirtualColumn, getVirtualColumnFunction, getVirtualColumnNames, hasColumn, hasColumns, hasVirtualColumn, hasVirtualColumns, length, names, newInstance, print, rbind, setAttributes, setColumnNamesMap, setColumnNamesTranslator, setName, setTags, setVirtualColumn, subset, translateColumnNames

### Methods inherited from data.frame:

\$<-, data.frame-method, \$<-, Math, Ops,nonStructure,vector-method, Ops,structure,vector-method, Ops,vector,nonStructure-method, Ops,vector,structure-method, Ops, Summary, [, [<-, data.frame-method, [<-, [[, [[<-, data.frame-method, [[<-, aggregate, anyDuplicated, anyNA, as.NonPairedPSCNData, as.PairedPSCNData, as.data.frame, as.list, as.matrix, as.vector, attachLocally, by, callSegmentationOutliers, cbind, coerce, ANY, list-method, coerce, oldClass, S3-method, dim, dimnames, dimnames<-, dropSegmentationOutliers, droplevels, duplicated, edit, findLargeGaps, format, formula, head, initialize, oldClass-method, is.na, merge, na.exclude, na.omit, plot, print, prompt, rbind, row.names, row.names<-, rowsum, segmentByCBS, segmentByPairedPSCBS, show, oldClass-method, slotsFromS3, data.frame-method, sort\_by, split, split<-, stack, str, subset, summary, t, tail, transform, type.convert, unique, unstack, unwrap, within, wrap, writeDataFrame, xfrm

## Author(s)

Henrik Bengtsson

---

AromaCellCpgFile

*A binary file holding local CpG density for each cell (probe/feature)*

---

## Description

A binary file holding local CpG density for each cell (probe/feature).

**Usage**

```
AromaCellCpgFile(...)
```

**Arguments**

... Arguments passed to constructor of [AromaCellTabularBinaryFile](#).

**Details**

Note that this class does *not* assume a rectangular chip layout. In other words, there is no concept of mapping a *spatial* location on the array to a cell index and vice versa. The reason for this to be able to use this class also for non-rectangular chip types.

**Author(s)**

Mark Robinson

---

AromaCellPositionFile *A binary file holding chromosome/position for each cell*

---

**Description**

A binary file holding chromosome/position for each cell.

**Usage**

```
AromaCellPositionFile(...)
```

**Arguments**

... Arguments passed to constructor of [AromaCellTabularBinaryFile](#).

**Details**

Note that this class does *not* assume a rectangular chip layout. In other words, there is no concept of mapping a *spatial* location on the array to a cell index and vice versa. The reason for this to be able to use this class also for non-rectangular chip types.

**Author(s)**

Henrik Bengtsson



---

AromaCellTabularBinaryFile

*The AromaCellTabularBinaryFile class*


---

## Description

Package: aroma.core

### Class AromaCellTabularBinaryFile

#### Object

```

~~|
~~+--FullNameInterface
~~~~~|
~~~~~+--GenericDataFile
~~~~~|
~~~~~+--ColumnNamesInterface
~~~~~|
~~~~~+--GenericTabularFile
~~~~~|
~~~~~+--CacheKeyInterface
~~~~~|
~~~~~+--FileCacheKeyInterface
~~~~~|
~~~~~+--AromaTabularBinaryFile
~~~~~|
~~~~~+--AromaPlatformInterface
~~~~~|
~~~~~+--AromaMicroarrayTabularBinaryFile
~~~~~|
~~~~~+--AromaCellTabularBinaryFile

```

#### Directly known subclasses:

[AromaCellCpgFile](#), [AromaCellPositionFile](#), [AromaCellSequenceFile](#)

public abstract static class **AromaCellTabularBinaryFile**

extends [AromaMicroarrayTabularBinaryFile](#)

An AromaCellTabularBinaryFile is an [AromaTabularBinaryFile](#) with the constraint that the rows map one-to-one to the cells (features) of a microarray.

## Usage

```
AromaCellTabularBinaryFile(...)
```

**Arguments**

... Arguments passed to [AromaTabularBinaryFile](#).

**Fields and Methods****Methods:**

byChipType -  
 nbrOfCells -

**Methods inherited from AromaMicroarrayTabularBinaryFile:**

allocate, as.character, byChipType, findByChipType, getChipType, getFilenameExtension, getPlatform

**Methods inherited from AromaPlatformInterface:**

getAromaPlatform, getAromaUflFile, getAromaUgpFile, getChipType, getPlatform, getUnitAnnotationDataFile, getUnitNamesFile, getUnitTypesFile, isCompatibleWith

**Methods inherited from AromaTabularBinaryFile:**

[, [<-, [[, allocate, as.character, colApply, colMeans, colStats, colSums, dimnames<-, getBytesPerColumn, getColClasses, getDefaultColumnNames, getRootName, importFrom, nbrOfColumns, nbrOfRows, readColumns, readDataFrame, readFooter, readHeader, readRawFooter, setAttributesByTags, subset, summary, updateData, updateDataColumn, writeFooter, writeRawFooter

**Methods inherited from FileCacheKeyInterface:**

getCacheKey

**Methods inherited from CacheKeyInterface:**

getCacheKey

**Methods inherited from GenericTabularFile:**

[, as.character, dim, extractMatrix, head, nbrOfColumns, nbrOfRows, readColumns, readDataFrame, tail, writeColumnsToFiles

**Methods inherited from ColumnNamesInterface:**

appendColumnNamesTranslator, appendColumnNamesTranslatorByNULL, appendColumnNamesTranslatorBycharacter, appendColumnNamesTranslatorByfunction, appendColumnNamesTranslatorBylist, clearColumnNamesTranslator, clearListOfColumnNamesTranslators, getColumnNames, getColumnNamesTranslator, getDefaultColumnNames, getListOfColumnNamesTranslators, nbrOfColumns, setColumnNames, setColumnNamesTranslator, setListOfColumnNamesTranslators, updateColumnNames

**Methods inherited from GenericDataFile:**

as.character, clone, compareChecksum, copyTo, equals, fromFile, getAttribute, getAttributes, getChecksum, getChecksumFile, getCreatedOn, getDefaultFullName, getExtension, getExtensionPattern, getFileSize, getFileType, getFilename, getFilenameExtension, getLastAccessedOn, getLastModifiedOn, getOutputExtension, getPath, getPathname, gunzip, gzip, hasBeenModified, is.na, isFile, isGzipped, linkTo, readChecksum, renameTo, renameToUpperCaseExt, setAttribute, setAttributes, setAttributesBy, setAttributesByTags, setExtensionPattern, testAttributes, validate, validateChecksum, writeChecksum, getParentName

**Methods inherited from FullNameInterface:**

appendFullNameTranslator, appendFullNameTranslatorByNULL, appendFullNameTranslatorByTabularTextFile, appendFullNameTranslatorByTabularTextFileSet, appendFullNameTranslatorBycharacter, appendFullNameTranslatorBydata.frame, appendFullNameTranslatorByfunction, appendFullNameTranslatorBylist, clearFullNameTranslator, clearListOfFullNameTranslators, getDefaultFullName, getFullName, getFullNameTranslator, getListOfFullNameTranslators, getName, getTags, hasTag, hasTags, resetFullName, setFullName, setFullNameTranslator, setListOfFullNameTranslators, setName, setTags, updateFullName

**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, names, objectSize, print, save, asThis

**Author(s)**

Henrik Bengtsson

**See Also**

[AromaUnitTabularBinaryFile](#).

---

AromaGenomeTextFile    *The AromaGenomeTextFile class*

---

**Description**

Package: aroma.core

**Class AromaGenomeTextFile****Object**

```

~~|
~~+--FullNameInterface
~~~~~|
~~~~~+--GenericDataFile
~~~~~|
~~~~~+--ColumnNamesInterface
~~~~~|
~~~~~+--GenericTabularFile
~~~~~|
~~~~~+--TabularTextFile
~~~~~|
~~~~~+--CacheKeyInterface
~~~~~|
~~~~~+--FileCacheKeyInterface
~~~~~|
~~~~~+--AromaGenomeTextFile

```

**Directly known subclasses:**[AromaUcscGenomeTextFile](#)

public abstract static class **AromaGenomeTextFile**  
 extends [FileCacheKeyInterface](#)

An AromaGenomeTextFile represents a annotation tabular text file that specifies the number of bases (nucleotides) per chromosome for a particular genome/organism.

**Usage**

```
AromaGenomeTextFile(...)
```

**Arguments**

... Arguments passed to [TabularTextFile](#).

**Details**

An AromaGenomeTextFile is a tab-delimited text file with a header containing (at least) column names 'chromosome' and 'nbrOfBases'. The 'chromosome' column specifies the chromosomes (character strings) and the 'nbrOfBases' column specifies the lengths (integer) of the chromosomes in number of bases (nucleotides).

The filename of an AromaGenomeTextFile should have format "<genome>,chromosomes(<tag>)\*.txt", and be located in annotationData/genomes/<genome>/, e.g. annotationData/genomes/Human/Human,chromosomes,max,200

**Fields and Methods****Methods:**

```
byGenome -
readDataFrame -
```

**Methods inherited from FileCacheKeyInterface:**

```
getCacheKey
```

**Methods inherited from CacheKeyInterface:**

```
getCacheKey
```

**Methods inherited from TabularTextFile:**

```
as.character, getCommentChar, getDefaultColumnClassPatterns, getDefaultColumnClasses, getDefaultColumnNames, getHeader, getReadArguments, hasColumnHeader, nbrOfLines, nbrOfRows, readColumnNames, readColumns, readDataFrame, readLines, readRawHeader, setCommentChar
```

**Methods inherited from GenericTabularFile:**

```
[, as.character, dim, extractMatrix, head, nbrOfColumns, nbrOfRows, readColumns, readDataFrame, tail, writeColumnsToFiles
```

**Methods inherited from ColumnNamesInterface:**

appendColumnNamesTranslator, appendColumnNamesTranslatorByNULL, appendColumnNamesTranslatorBycharacter, appendColumnNamesTranslatorByfunction, appendColumnNamesTranslatorBylist, clearColumnNamesTranslator, clearListOfColumnNamesTranslators, getColumnNames, getColumnNamesTranslator, getDefaultColumnNames, getListOfColumnNamesTranslators, nbrOfColumns, setColumnNames, setColumnNamesTranslator, setListOfColumnNamesTranslators, updateColumnNames

**Methods inherited from GenericDataFile:**

as.character, clone, compareChecksum, copyTo, equals, fromFile, getAttribute, getAttributes, getChecksum, getChecksumFile, getCreatedOn, getDefaultFullName, getExtension, getExtensionPattern, getFileSize, getFileType, getFilename, getFilenameExtension, getLastAccessedOn, getLastModifiedOn, getOutputExtension, getPath, getPathname, gunzip, gzip, hasBeenModified, is.na, isFile, isGzipped, linkTo, readChecksum, renameTo, renameToUpperCaseExt, setAttribute, setAttributes, setAttributesBy, setAttributesByTags, setExtensionPattern, testAttributes, validate, validateChecksum, writeChecksum, getParentName

**Methods inherited from FullNameInterface:**

appendFullNameTranslator, appendFullNameTranslatorByNULL, appendFullNameTranslatorByTabularTextFile, appendFullNameTranslatorByTabularTextFileSet, appendFullNameTranslatorBycharacter, appendFullNameTranslatorBydata.frame, appendFullNameTranslatorByfunction, appendFullNameTranslatorBylist, clearFullNameTranslator, clearListOfFullNameTranslators, getDefaultFullName, getFullName, getFullNameTranslator, getListOfFullNameTranslators, getName, getTags, hasTag, hasTags, resetFullName, setFullName, setFullNameTranslator, setListOfFullNameTranslators, setName, setTags, updateFullName

**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, names, objectSize, print, save, asThis

**Author(s)**

Henrik Bengtsson

**Examples**

```
# Locate a Human,chromosomes(,*)*.txt file
db <- AromaGenomeTextFile$byGenome("Human")
print(db)

# Read the data
df <- readDataframe(db)
print(df)
str(df)

# Details on the file format
oopts <- options(width=40)
print(readLines(db))
options(oopts)
```

---

AromaMicroarrayDataFile

*The abstract AromaMicroarrayDataFile class*


---

## Description

Package: aroma.core

### Class AromaMicroarrayDataFile

#### Object

```

~|
~+---FullNameInterface
~~~~~|
~~~~~+---GenericDataFile
~~~~~|
~~~~~+---CacheKeyInterface
~~~~~|
~~~~~+---FileCacheKeyInterface
~~~~~|
~~~~~+---AromaMicroarrayDataFile

```

#### Directly known subclasses:

```

public abstract static class AromaMicroarrayDataFile
extends FileCacheKeyInterface

```

An AromaMicroarrayDataFile object represents a single microarray data file. Each such file originates from a specific chip type on a given platform.

## Usage

```
AromaMicroarrayDataFile(...)
```

## Arguments

```
... Arguments passed to GenericDataFile.
```

## Fields and Methods

### Methods:

```

getChipType -
getPlatform -
isAverageFile -

```

**Methods inherited from FileCacheKeyInterface:**

getCacheKey

**Methods inherited from CacheKeyInterface:**

getCacheKey

**Methods inherited from GenericDataFile:**

as.character, clone, compareChecksum, copyTo, equals, fromFile, getAttribute, getAttributes, getChecksum, getChecksumFile, getCreatedOn, getDefaultFullName, getExtension, getExtensionPattern, getFileSize, getFileType, getFilename, getFilenameExtension, getLastAccessedOn, getLastModifiedOn, getOutputExtension, getPath, getPathname, gunzip, gzip, hasBeenModified, is.na, isFile, isGzipped, linkTo, readChecksum, renameTo, renameToUpperCaseExt, setAttribute, setAttributes, setAttributesBy, setAttributesByTags, setExtensionPattern, testAttributes, validate, validateChecksum, writeChecksum, getParentName

**Methods inherited from FullNameInterface:**

appendFullNameTranslator, appendFullNameTranslatorByNULL, appendFullNameTranslatorByTabularTextFile, appendFullNameTranslatorByTabularTextFileSet, appendFullNameTranslatorBycharacter, appendFullNameTranslatorBydata.frame, appendFullNameTranslatorByfunction, appendFullNameTranslatorBylist, clearFullNameTranslator, clearListOfFullNameTranslators, getDefaultFullName, getFullName, getFullNameTranslator, getListOfFullNameTranslators, getName, getTags, hasTag, hasTags, resetFullName, setFullName, setFullNameTranslator, setListOfFullNameTranslators, setName, setTags, updateFullName

**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, names, objectSize, print, save, asThis

**Author(s)**

Henrik Bengtsson

**See Also**

An object of this class is typically part of an [AromaMicroarrayDataSet](#).

---

AromaMicroarrayDataSet

*The AromaMicroarrayDataSet class*

---

**Description**

Package: aroma.core

**Class AromaMicroarrayDataSet**

```

Object
~~|
~~+---FullNameInterface
~~~~~|
~~~~~+---GenericDataFileSet
~~~~~|
~~~~~+---AromaMicroarrayDataSet

```

### Directly known subclasses:

```

public abstract static class AromaMicroarrayDataSet
extends GenericDataFileSet

```

An AromaMicroarrayDataSet object represents a set of [AromaMicroarrayDataFiles](#) with *identical* chip types.

### Usage

```
AromaMicroarrayDataSet(files=NULL, ...)
```

### Arguments

```

files      A list of AromaMicroarrayDataFile:s.
...        Arguments passed to GenericDataFileSet.

```

### Fields and Methods

#### Methods:

```

as.AromaMicroarrayDataSetList -
as.AromaMicroarrayDataSetTuple -
getAverageFile -
getChipType -
getPlatform -

```

#### Methods inherited from GenericDataFileSet:

```

[, [[, anyDuplicated, anyNA, append, appendFiles, appendFullNamesTranslator, appendFullNamesTranslatorByNULL, appendFullNamesTranslatorByTabularTextFile, appendFullNamesTranslatorByTabularTextFileSet, appendFullNamesTranslatorBydata.frame, appendFullNamesTranslatorByfunction, appendFullNamesTranslatorBylist, as.character, as.list, byName, byPath, c, clearCache, clearFullNamesTranslator, clone, copyTo, dsApplyInPairs, duplicated, equals, extract, findByName, findDuplicated, getChecksum, getChecksumFileSet, getChecksumObjects, getDefaultFullName, getFile, getFileClass, getFileSize, getFiles, getFullNames, getNames, getOneFile, getPath, getPathnames, getSubdirs, gunzip, gzip, hasFile, indexOf, is.na, names, nbrOfFiles, rep, resetFullNames, setFullNamesTranslator, sortBy, unique, update2, updateFullName, updateFullNames, validate, getFullNameTranslatorSet, getParentName

```



**Methods inherited from FullNameInterface:**

appendFullNameTranslator, appendFullNameTranslatorByNULL, appendFullNameTranslatorByTabularTextFile, appendFullNameTranslatorByTabularTextFileSet, appendFullNameTranslatorBycharacter, appendFullNameTranslatorBydata.frame, appendFullNameTranslatorByfunction, appendFullNameTranslatorBylist, clearFullNameTranslator, clearListOfFullNameTranslators, getDefaultFullName, getFullName, getFullNameTranslator, getListOfFullNameTranslators, getName, getTags, hasTag, hasTags, resetFullName, setFullName, setFullNameTranslator, setListOfFullNameTranslators, setName, setTags, updateFullName

**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, names, objectSize, print, save, asThis

**Author(s)**

Henrik Bengtsson

---

AromaMicroarrayDataSetTuple

*The AromaMicroarrayDataSetTuple class*

---

**Description**

Package: aroma.core

**Class AromaMicroarrayDataSetTuple**

Object

~~|

~~+---FullNameInterface

~~~~~|

~~~~~+---GenericDataFileSetList

~~~~~|

~~~~~+---AromaMicroarrayDataSetTuple

**Directly known subclasses:**

*AromaUnitTotalCnBinarySetTuple*

public abstract static class **AromaMicroarrayDataSetTuple**

extends [GenericDataFileSetList](#)

**Usage**

AromaMicroarrayDataSetTuple(..., .setClass="AromaMicroarrayDataSet")

**Arguments**

... Arguments passed to [GenericDataFileSetList](#).  
 .setClass The name of the class of the input set.

**Fields and Methods****Methods:**

|                                |                                |
|--------------------------------|--------------------------------|
| as                             | -                              |
| as.AromaMicroarrayDataSetTuple | -                              |
| getChipTypes                   | -                              |
| getFullNames                   | -                              |
| getSets                        | -                              |
| getTags                        | -                              |
| <a href="#">nbrOfChipTypes</a> | Gets the number of chip types. |

**Methods inherited from GenericDataFileSetList:**

as, as.GenericDataFileSetList, as.character, as.data.frame, as.list, assertDuplicates, clone, extract, getAsteriskTags, getDefaultFullName, getFileList, getFileListClass, getFullNames, getNames, getSet, getSets, getTags, indexOf, length, nbrOfFiles, nbrOfSets, setTags

**Methods inherited from FullNameInterface:**

appendFullNameTranslator, appendFullNameTranslatorByNULL, appendFullNameTranslatorByTabularTextFile, appendFullNameTranslatorByTabularTextFileSet, appendFullNameTranslatorBycharacter, appendFullNameTranslatorBydata.frame, appendFullNameTranslatorByfunction, appendFullNameTranslatorBylist, clearFullNameTranslator, clearListOfFullNameTranslators, getDefaultFullName, getFullName, getFullNameTranslator, getListOfFullNameTranslators, getName, getTags, hasTag, hasTags, resetFullName, setFullName, setFullNameTranslator, setListOfFullNameTranslators, setName, setTags, updateFullName

**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, names, objectSize, print, save, asThis

**Author(s)**

Henrik Bengtsson

**Description**

Package: aroma.core

**Class AromaMicroarrayTabularBinaryFile****Object**

```

~~|
~~+--FullNameInterface
~~~~~|
~~~~~+--GenericDataFile
~~~~~|
~~~~~+--ColumnNamesInterface
~~~~~|
~~~~~+--GenericTabularFile
~~~~~|
~~~~~+--CacheKeyInterface
~~~~~|
~~~~~+--FileCacheKeyInterface
~~~~~|
~~~~~+--AromaTabularBinaryFile
~~~~~|
~~~~~+--AromaPlatformInterface
~~~~~|
~~~~~+--AromaMicroarrayTabularBinaryFile

```

**Directly known subclasses:**

[AromaCellCpgFile](#), [AromaCellPositionFile](#), [AromaCellSequenceFile](#), [AromaCellTabularBinaryFile](#), [AromaUhlFile](#), [AromaUgcFile](#), [AromaUgpFile](#), [AromaUnitChromosomeTabularBinaryFile](#), [AromaUnitGcContentFile](#), [AromaUnitTabularBinaryFile](#)

public abstract static class **AromaMicroarrayTabularBinaryFile**  
 extends [AromaPlatformInterface](#)

An [AromaMicroarrayTabularBinaryFile](#) is an abstract [AromaTabularBinaryFile](#).

**Usage**

```
AromaMicroarrayTabularBinaryFile(...)
```

**Arguments**

... Arguments passed to [AromaTabularBinaryFile](#).

**Fields and Methods****Methods:**

```

byChipType           -
getChipType          -
getFilenameExtension -
getPlatform          -

```

**Methods inherited from AromaPlatformInterface:**

getAromaPlatform, getAromaUflFile, getAromaUgpFile, getChipType, getPlatform, getUnitAnnotationDataFile, getUnitNamesFile, getUnitTypesFile, isCompatibleWith

**Methods inherited from AromaTabularBinaryFile:**

[, [<-, [[, allocate, as.character, colApply, colMeans, colStats, colSums, dimnames<-, getBytesPerColumn, getColClasses, getDefaultColumnNames, getRootName, importFrom, nbrOfColumns, nbrOfRows, readColumns, readDataFrame, readFooter, readHeader, readRawFooter, setAttributesByTags, subset, summary, updateData, updateDataColumn, writeFooter, writeRawFooter

**Methods inherited from FileCacheKeyInterface:**

getCacheKey

**Methods inherited from CacheKeyInterface:**

getCacheKey

**Methods inherited from GenericTabularFile:**

[, as.character, dim, extractMatrix, head, nbrOfColumns, nbrOfRows, readColumns, readDataFrame, tail, writeColumnsToFiles

**Methods inherited from ColumnNamesInterface:**

appendColumnNamesTranslator, appendColumnNamesTranslatorByNULL, appendColumnNamesTranslatorBycharacter, appendColumnNamesTranslatorByfunction, appendColumnNamesTranslatorBylist, clearColumnNamesTranslator, clearListOfColumnNamesTranslators, getColumnNames, getColumnNamesTranslator, getDefaultColumnNames, getListOfColumnNamesTranslators, nbrOfColumns, setColumnNames, setColumnNamesTranslator, setListOfColumnNamesTranslators, updateColumnNames

**Methods inherited from GenericDataFile:**

as.character, clone, compareChecksum, copyTo, equals, fromFile, getAttribute, getAttributes, getChecksum, getChecksumFile, getCreatedOn, getDefaultFullName, getExtension, getExtensionPattern, getFileSize, getFileType, getFilename, getFilenameExtension, getLastAccessedOn, getLastModifiedOn, getOutputExtension, getPath, getPathname, gunzip, gzip, hasBeenModified, is.na, isFile, isGzipped, linkTo, readChecksum, renameTo, renameToUpperCaseExt, setAttribute, setAttributes, setAttributesBy, setAttributesByTags, setExtensionPattern, testAttributes, validate, validateChecksum, writeChecksum, getParentName

**Methods inherited from FullNameInterface:**

appendFullNameTranslator, appendFullNameTranslatorByNULL, appendFullNameTranslatorByTabularTextFile, appendFullNameTranslatorByTabularTextFileSet, appendFullNameTranslatorBycharacter, appendFullNameTranslatorBydata.frame, appendFullNameTranslatorByfunction, appendFullNameTranslatorBylist, clearFullNameTranslator, clearListOfFullNameTranslators, getDefaultFullName, getFullName, getFullNameTranslator, getListOfFullNameTranslators, getName, getTags, hasTag, hasTags, resetFullName, setFullName, setFullNameTranslator, setListOfFullNameTranslators, setName, setTags, updateFullName

**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, names, objectSize, print, save, asThis

### Author(s)

Henrik Bengtsson

### See Also

[AromaTabularBinaryFile](#).

---

|               |                                |
|---------------|--------------------------------|
| AromaPlatform | <i>The AromaPlatform class</i> |
|---------------|--------------------------------|

---

### Description

Package: aroma.core

**Class AromaPlatform**

[Object](#)

~~|

~~+--AromaPlatform

#### Directly known subclasses:

public abstract static class **AromaPlatform**

extends [Object](#)

An AromaPlatform provides methods for a given platform, e.g. Affymetrix, Agilent, Illumina.

### Usage

AromaPlatform(...)

### Arguments

...                      Not used.

### Methods

#### Methods:

|        |   |
|--------|---|
| byName | - |
| equals | - |

```

getAromaUgpFile -
getName         -
getUnitNamesFile -
getUnitTypesFile -

```

**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, names, objectSize, print, save, asThis

**Author(s)**

Henrik Bengtsson

**Examples**

```

## Not run:
library("aroma.affymetrix")
platform <- AromaPlatform$byName("Affymetrix")
print(platform)
stopifnot(getName(platform) == "Affymetrix")

## End(Not run)

```

---

AromaPlatformInterface

*The AromaPlatformInterface class*

---

**Description**

Package: aroma.core

**Class AromaPlatformInterface****Interface**

~~|

~~+--AromaPlatformInterface

**Directly known subclasses:**

[AromaCellCpgFile](#), [AromaCellPositionFile](#), [AromaCellSequenceFile](#), [AromaCellTabularBinaryFile](#), [AromaMicroarrayTabularBinaryFile](#), [AromaUflFile](#), [AromaUgcFile](#), [AromaUgpFile](#), [AromaUnitCallFile](#), [AromaUnitChromosomeTabularBinaryFile](#), [AromaUnitFracBCnBinaryFile](#), [AromaUnitGcContentFile](#), [AromaUnitGenotypeCallFile](#), [AromaUnitPscnBinaryFile](#), [AromaUnitSignalBinaryFile](#), [AromaUnitTabularBinaryFile](#), [AromaUnitTotalCnBinaryFile](#), [AromaUnitTypesFile](#)

public abstract class **AromaPlatformInterface**  
 extends [Interface](#)

An AromaPlatformInterface provides methods for a given platform, e.g. Affymetrix, Agilent, Illumina.

### Usage

```
AromaPlatformInterface(...)
```

### Arguments

... Not used.

### Methods

#### Methods:

|                                  |                     |
|----------------------------------|---------------------|
| <a href="#">getAromaPlatform</a> | Gets the platform.  |
| <a href="#">getAromaUflFile</a>  | -                   |
| <a href="#">getAromaUgpFile</a>  | -                   |
| <a href="#">getChipType</a>      | Gets the chip type. |
| <a href="#">getPlatform</a>      | Gets the platform.  |
| <a href="#">getUnitNamesFile</a> | -                   |
| <a href="#">getUnitTypesFile</a> | -                   |

#### Methods inherited from Interface:

extend, print, uses

### Author(s)

Henrik Bengtsson

---

AromaRepository

*The AromaRepository class*

---

### Description

Package: aroma.core

**Class AromaRepository**

[Object](#)

~~|

~~+--AromaRepository

**Directly known subclasses:**

public static class **AromaRepository**  
 extends [Object](#)

An AromaRepository object provides methods for downloading annotation data from the Aroma repository.

**Usage**

```
AromaRepository(urlPath="https://www.aroma-project.org/data", verbose=FALSE, ...)
```

**Arguments**

|         |   |
|---------|---|
| urlPath | The URL to the Aroma repository.                          |
| verbose | The <a href="#">Verbose</a> to be used during processing. |
| ...     | Not used.   |

**Methods****Methods:**

|                                      |  |
|--------------------------------------|--|
| downloadACC                          | -  |
| downloadACM                          | -  |
| downloadACP                          | -  |
| downloadACS                          | -  |
| downloadAll                          | -  |
| downloadCDF                          | -  |
| <a href="#">downloadChipTypeFile</a> | Download a particular chip type annotation file. |
| downloadProbeSeqsTXT                 | -  |
| downloadTXT                          | -  |
| downloadUFL                          | -  |
| downloadUGC                          | -  |
| downloadUGP                          | -  |

**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, names, objectSize, print, save, asThis

**Author(s)**

Henrik Bengtsson



---

AromaTabularBinaryFile

*The AromaTabularBinaryFile class*


---

## Description

Package: aroma.core

### Class AromaTabularBinaryFile

#### Object

```

~|
~+--FullNameInterface
~~~~~|
~~~~~+--GenericDataFile
~~~~~|
~~~~~+--ColumnNamesInterface
~~~~~|
~~~~~+--GenericTabularFile
~~~~~|
~~~~~+--CacheKeyInterface
~~~~~|
~~~~~+--FileCacheKeyInterface
~~~~~|
~~~~~+--AromaTabularBinaryFile

```

#### Directly known subclasses:

[AromaCellCpgFile](#), [AromaCellPositionFile](#), [AromaCellSequenceFile](#), [AromaCellTabularBinaryFile](#), [AromaMicroarrayTabularBinaryFile](#), [AromaUffFile](#), [AromaUgcFile](#), [AromaUgpFile](#), [AromaUnitCallFile](#), [AromaUnitChromosomeTabularBinaryFile](#), [AromaUnitFracBCnBinaryFile](#), [AromaUnitGcContentFile](#), [AromaUnitGenotypeCallFile](#), [AromaUnitPscnBinaryFile](#), [AromaUnitSignalBinaryFile](#), [AromaUnitTabularBinaryFile](#), [AromaUnitTotalCnBinaryFile](#), [AromaUnitTypesFile](#)

public abstract static class **AromaTabularBinaryFile**

extends [FileCacheKeyInterface](#)

A AromaTabularBinaryFile represents a file with a binary format. It has a well defined header, a data section, and a footer.

## Usage

```
AromaTabularBinaryFile(...)
```

## Arguments

... Arguments passed to [GenericTabularFile](#).

**Fields and Methods****Methods:**

|                   |   |
|-------------------|---|
| [                 | -   |
| [<-               | -   |
| colMeans          | -   |
| colSums           | -   |
| getBytesPerColumn | -   |
| getColClasses     | -   |
| importFrom        | -   |
| nrOfColumns       | -   |
| nrOfRows          | -   |
| readColumns       | -   |
| readFooter        | Reads the file footer in XML format into a named nested list. |
| subset            | -   |
| summary           | -   |
| writeFooter       | Writes a named nested list to the file footer in XML format.  |

**Methods inherited from FileCacheKeyInterface:**

getCacheKey

**Methods inherited from CacheKeyInterface:**

getCacheKey

**Methods inherited from GenericTabularFile:**

[, as.character, dim, extractMatrix, head, nrOfColumns, nrOfRows, readColumns, readDataFrame, tail, writeColumnsToFiles

**Methods inherited from ColumnNamesInterface:**

appendColumnNamesTranslator, appendColumnNamesTranslatorByNULL, appendColumnNamesTranslatorBycharacter, appendColumnNamesTranslatorByfunction, appendColumnNamesTranslatorBylist, clearColumnNamesTranslator, clearListOfColumnNamesTranslators, getColumnNames, getColumnNamesTranslator, getDefaultColumnNames, getListOfColumnNamesTranslators, nrOfColumns, setColumnNames, setColumnNamesTranslator, setListOfColumnNamesTranslators, updateColumnNames

**Methods inherited from GenericDataFile:**

as.character, clone, compareChecksum, copyTo, equals, fromFile, getAttribute, getAttributes, getChecksum, getChecksumFile, getCreatedOn, getDefaultFullName, getExtension, getExtensionPattern, getFileSize, getFileType, getFilename, getFilenameExtension, getLastAccessedOn, getLastModifiedOn, getOutputExtension, getPath, getPathname, gunzip, gzip, hasBeenModified, is.na, isFile, isGzipped, linkTo, readChecksum, renameTo, renameToUpperCaseExt, setAttribute, setAttributes, setAttributesBy, setAttributesByTags, setExtensionPattern, testAttributes, validate, validateChecksum, writeChecksum, getParentName

**Methods inherited from FullNameInterface:**

appendFullNameTranslator, appendFullNameTranslatorByNULL, appendFullNameTranslatorByTabularTextFile, appendFullNameTranslatorByTabularTextFileSet, appendFullNameTranslatorBycharacter, appendFullNameTranslatorBydata.frame, appendFullNameTranslatorByfunction, appendFull-

NameTranslatorBylist, clearFullNameTranslator, clearListOfFullNameTranslators, getDefaultFullName, getFullName, getFullNameTranslator, getListOfFullNameTranslators, getName, getTags, hasTag, hasTags, resetFullName, setFullName, setFullNameTranslator, setListOfFullNameTranslators, setName, setTags, updateFullName

**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, names, objectSize, print, save, asThis

**Author(s)**

Henrik Bengtsson

**See Also**

[GenericDataFile](#).

AromaTabularBinarySet *The AromaTabularBinarySet class*

**Description**

Package: aroma.core

**Class AromaTabularBinarySet**

[Object](#)

~~|

~~+---FullNameInterface

~~~~~|

~~~~~+---GenericDataFileSet

~~~~~|

~~~~~+---GenericTabularFileSet

~~~~~|

~~~~~+---AromaTabularBinarySet

**Directly known subclasses:**

[AromaUnitCallSet](#), [AromaUnitFracBCnBinarySet](#), [AromaUnitGenotypeCallSet](#), [AromaUnitPscnBinarySet](#), [AromaUnitSignalBinarySet](#), [AromaUnitTotalCnBinarySet](#)

public static class **AromaTabularBinarySet**

extends [GenericTabularFileSet](#)

An AromaTabularBinarySet object represents a set of [AromaTabularBinaryFiles](#) with *identical* chip types.

**Usage**

```
AromaTabularBinarySet(files=NULL, ...)
```

**Arguments**

files            A list of [AromaTabularBinaryFile](#)s.  
 ...             Arguments passed to [GenericDataFileSet](#).

**Fields and Methods****Methods:**

*No methods defined.*

**Methods inherited from GenericTabularFileSet:**

extractMatrix, calculateAverageColumnAcrossFiles

**Methods inherited from GenericDataFileSet:**

[, [[, anyDuplicated, anyNA, append, appendFiles, appendFullNamesTranslator, appendFullNamesTranslatorByNULL, appendFullNamesTranslatorByTabularTextFile, appendFullNamesTranslatorByTabularTextFileSet, appendFullNamesTranslatorBydata.frame, appendFullNamesTranslatorByfunction, appendFullNamesTranslatorBylist, as.character, as.list, byName, byPath, c, clearCache, clearFullNamesTranslator, clone, copyTo, dsApplyInPairs, duplicated, equals, extract, findByName, findDuplicated, getChecksum, getChecksumFileSet, getChecksumObjects, getDefaultFullName, getFile, getFileClass, getFileSize, getFiles, getFullNames, getNames, getOneFile, getPath, getPathnames, getSubdirs, gunzip, gzip, hasFile, indexOf, is.na, names, nbrOfFiles, rep, resetFullNames, setFullNamesTranslator, sortBy, unique, update2, updateFullName, updateFullNames, validate, getFullNameTranslatorSet, getParentName

**Methods inherited from FullNameInterface:**

appendFullNameTranslator, appendFullNameTranslatorByNULL, appendFullNameTranslatorByTabularTextFile, appendFullNameTranslatorByTabularTextFileSet, appendFullNameTranslatorBycharacter, appendFullNameTranslatorBydata.frame, appendFullNameTranslatorByfunction, appendFullNameTranslatorBylist, clearFullNameTranslator, clearListOfFullNameTranslators, getDefaultFullName, getFullName, getFullNameTranslator, getListOfFullNameTranslators, getName, getTags, hasTag, hasTags, resetFullName, setFullName, setFullNameTranslator, setListOfFullNameTranslators, setName, setTags, updateFullName

**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, names, objectSize, print, save, asThis

**Author(s)**

Henrik Bengtsson

---

AromaTransform      *The AromaTransform class*


---

## Description

Package: aroma.core

### Class AromaTransform

[Object](#)

~~|

~~+--[ParametersInterface](#)

~~~~~|

~~~~~+--AromaTransform

### Directly known subclasses:

public abstract static class **AromaTransform**

extends [ParametersInterface](#)

This abstract class represents a transform (algorithm/operator) that transforms data. A transform has an input data set, which is transformed into an output data set.

## Usage

```
AromaTransform(dataSet=NULL, tags="*", ..., .reqSetClass="AromaMicroarrayDataSet")
```

## Arguments

|              |  |
|--------------|--|
| dataSet      | The input data set as an <a href="#">AromaMicroarrayDataSet</a> .                            |
| tags         | A <a href="#">character vector</a> of tags to be appended to the tags of the input data set. |
| ...          | Not used.  |
| .reqSetClass | Internal argument.   |

## Details

Subclasses must implement the process() method.

## Fields and Methods

### Methods:

|                                 |  |
|---------------------------------|--|
| <a href="#">getFullName</a>     | Gets the full name of the output data set. |
| <a href="#">getInputDataSet</a> | Gets the input data set.                   |

|                                  |   |
|----------------------------------|---|
| <a href="#">getName</a>          | Gets the name of the output data set.       |
| <a href="#">getOutputDataSet</a> | Gets the transformed data set.              |
| <a href="#">getPath</a>          | Gets the path of the output directory.      |
| <a href="#">getRootPath</a>      | Gets the root path of the output directory. |
| <a href="#">getTags</a>          | Gets the tags of the output data set.       |
| <a href="#">isDone</a>           | Checks if the data set is processed or not. |
| <a href="#">process</a>          | Processes the data set.                     |
| <a href="#">setTags</a>          | -   |

**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](#), [names](#), [objectSize](#), [print](#), [save](#), [asThis](#)

**Author(s)**

Henrik Bengtsson

---

AromaUnitCallFile

*The AromaUnitCallFile class*

---

**Description**

Package: aroma.core

**Class AromaUnitCallFile****Object**

```
~~|
```

```
~~+--FullNameInterface
```

```
~~~~~|
```

```
~~~~~+--GenericDataFile
```

```
~~~~~|
```

```
~~~~~+--ColumnNamesInterface
```

```
~~~~~|
```

```
~~~~~+--GenericTabularFile
```

```
~~~~~|
```

```
~~~~~+--CacheKeyInterface
```

```
~~~~~|
```

```
~~~~~+--FileCacheKeyInterface
```

```
~~~~~|
```

```
~~~~~+--AromaTabularBinaryFile
```

```
~~~~~|
```

```
~~~~~+--AromaPlatformInterface
```

```

~~~~~|
~~~~~+---AromaUnitSignalBinaryFile
~~~~~|
~~~~~+---AromaUnitCallFile

```

**Directly known subclasses:**

[AromaUnitGenotypeCallFile](#)

```

public static class AromaUnitCallFile
extends AromaUnitSignalBinaryFile

```

An [AromaUnitCallFile](#) is a [AromaUnitSignalBinaryFile](#).

**Usage**

```
AromaUnitCallFile(...)
```

**Arguments**

... Arguments passed to [AromaUnitSignalBinaryFile](#).

**Fields and Methods****Methods:**

```

extractCallArray -
extractCalls     -
extractMatrix    -
findUnitsTodo    -

```

**Methods inherited from [AromaUnitSignalBinaryFile](#):**

allocate, allocateFromUnitAnnotationDataFile, allocateFromUnitNamesFile, as.character, extractMatrix, extractRawGenomicSignals, fromFile, getChipType, getExtensionPattern, getFilenameExtension, getNumberOfFilesAveraged, getPlatform, isAverageFile, nbrOfUnits, readDataFrame, writeDataFrame

**Methods inherited from [AromaPlatformInterface](#):**

getAromaPlatform, getAromaUflFile, getAromaUgpFile, getChipType, getPlatform, getUnitAnnotationDataFile, getUnitNamesFile, getUnitTypesFile, isCompatibleWith

**Methods inherited from [AromaTabularBinaryFile](#):**

[, [<-, [[, allocate, as.character, colApply, colMeans, colStats, colSums, dimnames<-, getBytesPerColumn, getColClasses, getDefaultColumnNames, getRootName, importFrom, nbrOfColumns, nbrOfRows, readColumns, readDataFrame, readFooter, readHeader, readRawFooter, setAttributesByTags, subset, summary, updateData, updateDataColumn, writeFooter, writeRawFooter

**Methods inherited from [FileCacheKeyInterface](#):**

getCacheKey

**Methods inherited from CacheKeyInterface:**

getCacheKey

**Methods inherited from GenericTabularFile:**

[, as.character, dim, extractMatrix, head, nbrOfColumns, nbrOfRows, readColumns, readDataFrame, tail, writeColumnsToFiles

**Methods inherited from ColumnNamesInterface:**

appendColumnNamesTranslator, appendColumnNamesTranslatorByNULL, appendColumnNamesTranslatorBycharacter, appendColumnNamesTranslatorByfunction, appendColumnNamesTranslatorBylist, clearColumnNamesTranslator, clearListOfColumnNamesTranslators, getColumnNames, getColumnNamesTranslator, getDefaultColumnNames, getListOfColumnNamesTranslators, nbrOfColumns, setColumnNames, setColumnNamesTranslator, setListOfColumnNamesTranslators, updateColumnNames

**Methods inherited from GenericDataFile:**

as.character, clone, compareChecksum, copyTo, equals, fromFile, getAttribute, getAttributes, getChecksum, getChecksumFile, getCreatedOn, getDefaultFullName, getExtension, getExtensionPattern, getFileSize, getFileType, getFilename, getFilenameExtension, getLastAccessedOn, getLastModifiedOn, getOutputExtension, getPath, getPathname, gunzip, gzip, hasBeenModified, is.na, isFile, isGzipped, linkTo, readChecksum, renameTo, renameToUpperCaseExt, setAttribute, setAttributes, setAttributesBy, setAttributesByTags, setExtensionPattern, testAttributes, validate, validateChecksum, writeChecksum, getParentName

**Methods inherited from FullNameInterface:**

appendFullNameTranslator, appendFullNameTranslatorByNULL, appendFullNameTranslatorByTabularTextFile, appendFullNameTranslatorByTabularTextFileSet, appendFullNameTranslatorBycharacter, appendFullNameTranslatorBydata.frame, appendFullNameTranslatorByfunction, appendFullNameTranslatorBylist, clearFullNameTranslator, clearListOfFullNameTranslators, getDefaultFullName, getFullName, getFullNameTranslator, getListOfFullNameTranslators, getName, getTags, hasTag, hasTags, resetFullName, setFullName, setFullNameTranslator, setListOfFullNameTranslators, setName, setTags, updateFullName

**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, names, objectSize, print, save, asThis

**Author(s)**

Henrik Bengtsson

---

AromaUnitCallSet

*The AromaUnitCallSet class*

---

**Description**

Package: aroma.core

**Class AromaUnitCallSet**



```

Object
~~|
~~+---FullNameInterface
~~~~~|
~~~~~+---GenericDataFileSet
~~~~~|
~~~~~+---GenericTabularFileSet
~~~~~|
~~~~~+---AromaTabularBinarySet
~~~~~|
~~~~~+---AromaUnitSignalBinarySet
~~~~~|
~~~~~+---AromaUnitCallSet

```

**Directly known subclasses:**

[AromaUnitGenotypeCallSet](#)

```

public static class AromaUnitCallSet
extends AromaUnitSignalBinarySet

```

An AromaUnitCallSet object represents a set of [AromaUnitCallFiles](#) with *identical* chip types.

**Usage**

```
AromaUnitCallSet(...)
```

**Arguments**

... Arguments passed to [AromaUnitSignalBinarySet](#).

**Fields and Methods****Methods:**

```

extractCallArray      -
extractCalls          -
extractGenotypeMatrix -
findUnitsTodo         -

```

**Methods inherited from AromaUnitSignalBinarySet:**

byName, findByName, getAromaFullNameTranslatorSet, getAromaUgpFile, getChipType, getPlatform, validate, writeDataFrame

**Methods inherited from AromaTabularBinarySet:**

getDefaultFullName, getRootName, setAttributesBy, setAttributesBySampleAnnotationFile, setAttributesBySampleAnnotationSet, setAttributesByTags

**Methods inherited from GenericTabularFileSet:**

extractMatrix, calculateAverageColumnAcrossFiles

**Methods inherited from GenericDataFileSet:**

[, [[, anyDuplicated, anyNA, append, appendFiles, appendFullNamesTranslator, appendFullNamesTranslatorByNULL, appendFullNamesTranslatorByTabularTextFile, appendFullNamesTranslatorByTabularTextFileSet, appendFullNamesTranslatorBydata.frame, appendFullNamesTranslatorByfunction, appendFullNamesTranslatorBylist, as.character, as.list, byName, byPath, c, clearCache, clearFullNamesTranslator, clone, copyTo, dsApplyInPairs, duplicated, equals, extract, findByName, findDuplicated, getChecksum, getChecksumFileSet, getChecksumObjects, getDefaultFullName, getFile, getFileClass, getFileSize, getFiles, getFullNames, getNames, getOneFile, getPath, getPathnames, getSubdirs, gunzip, gzip, hasFile, indexOf, is.na, names, nbrOfFiles, rep, resetFullNames, setFullNamesTranslator, sortBy, unique, update2, updateFullName, updateFullNames, validate, getFullNameTranslatorSet, getParentName

**Methods inherited from FullNameInterface:**

appendFullNameTranslator, appendFullNameTranslatorByNULL, appendFullNameTranslatorByTabularTextFile, appendFullNameTranslatorByTabularTextFileSet, appendFullNameTranslatorBycharacter, appendFullNameTranslatorBydata.frame, appendFullNameTranslatorByfunction, appendFullNameTranslatorBylist, clearFullNameTranslator, clearListOfFullNameTranslators, getDefaultFullName, getFullName, getFullNameTranslator, getListOfFullNameTranslators, getName, getTags, hasTag, hasTags, resetFullName, setFullName, setFullNameTranslator, setListOfFullNameTranslators, setName, setTags, updateFullName

**Methods inherited from Object:**

\$, \$&lt;-, [[, [[&lt;-, as.character, attach, attachLocally, clearCache, clearLookupCache, clone, detach, equals, extend, finalize, getEnvironment, getFieldModifier, getFieldModifiers, getFields, getInstantiationTime, getStaticInstance, hasField, hashCode, ll, load, names, objectSize, print, save, asThis

**Author(s)**

Henrik Bengtsson

---

AromaUnitFracBCnBinaryFile*The AromaUnitFracBCnBinaryFile class*

---

**Description**

Package: aroma.core

**Class AromaUnitFracBCnBinaryFile****Object**

```

~~|
~~+--FullNameInterface
~~~~~|
~~~~~+--GenericDataFile
~~~~~|
~~~~~+--ColumnNamesInterface

```

```

~~~~~|
~~~~~+--GenericTabularFile
~~~~~|
~~~~~+--CacheKeyInterface
~~~~~|
~~~~~+--FileCacheKeyInterface
~~~~~|
~~~~~+--AromaTabularBinaryFile
~~~~~|
~~~~~+--AromaPlatformInterface
~~~~~|
~~~~~+--AromaUnitSignalBinaryFile
~~~~~|
~~~~~+--AromaUnitFracBCnBinaryFile

```

**Directly known subclasses:**

```

public static class AromaUnitFracBCnBinaryFile
extends AromaUnitSignalBinaryFile

```

An `AromaUnitFracBCnBinaryFile` is a `AromaUnitTabularBinaryFile`.

**Usage**

```
AromaUnitFracBCnBinaryFile(...)
```

**Arguments**

... Arguments passed to `AromaUnitTabularBinaryFile`.

**Fields and Methods****Methods:**

```
extractRawAlleleBFractions -
```

**Methods inherited from `AromaUnitSignalBinaryFile`:**

allocate, allocateFromUnitAnnotationDataFile, allocateFromUnitNamesFile, as.character, extractMatrix, extractRawGenomicSignals, fromFile, getChipType, getExtensionPattern, getFilenameExtension, getNumberOfFilesAveraged, getPlatform, isAverageFile, nbrOfUnits, readDataFrame, writeDataFrame

**Methods inherited from `AromaPlatformInterface`:**

getAromaPlatform, getAromaUflFile, getAromaUgpFile, getChipType, getPlatform, getUnitAnnotationDataFile, getUnitNamesFile, getUnitTypesFile, isCompatibleWith

**Methods inherited from AromaTabularBinaryFile:**

[, [<-, [[, allocate, as.character, colApply, colMeans, colStats, colSums, dimnames<-, getBytesPerColumn, getColClasses, getDefaultColumnNames, getRootName, importFrom, nbrOfColumns, nbrOfRows, readColumns, readDataFrame, readFooter, readHeader, readRawFooter, setAttributesByTags, subset, summary, updateData, updateDataColumn, writeFooter, writeRawFooter

**Methods inherited from FileCacheKeyInterface:**

getCacheKey

**Methods inherited from CacheKeyInterface:**

getCacheKey

**Methods inherited from GenericTabularFile:**

[, as.character, dim, extractMatrix, head, nbrOfColumns, nbrOfRows, readColumns, readDataFrame, tail, writeColumnsToFiles

**Methods inherited from ColumnNamesInterface:**

appendColumnNamesTranslator, appendColumnNamesTranslatorByNULL, appendColumnNamesTranslatorBycharacter, appendColumnNamesTranslatorByfunction, appendColumnNamesTranslatorBylist, clearColumnNamesTranslator, clearListOfColumnNamesTranslators, getColumnNames, getColumnNamesTranslator, getDefaultColumnNames, getListOfColumnNamesTranslators, nbrOfColumns, setColumnNames, setColumnNamesTranslator, setListOfColumnNamesTranslators, updateColumnNames

**Methods inherited from GenericDataFile:**

as.character, clone, compareChecksum, copyTo, equals, fromFile, getAttribute, getAttributes, getChecksum, getChecksumFile, getCreatedOn, getDefaultFullName, getExtension, getExtensionPattern, getFileSize, getFileType, getFilename, getFilenameExtension, getLastAccessedOn, getLastModifiedOn, getOutputExtension, getPath, getPathname, gunzip, gzip, hasBeenModified, is.na, isFile, isGzipped, linkTo, readChecksum, renameTo, renameToUpperCaseExt, setAttribute, setAttributes, setAttributesBy, setAttributesByTags, setExtensionPattern, testAttributes, validate, validateChecksum, writeChecksum, getParentName

**Methods inherited from FullNameInterface:**

appendFullNameTranslator, appendFullNameTranslatorByNULL, appendFullNameTranslatorByTabularTextFile, appendFullNameTranslatorByTabularTextFileSet, appendFullNameTranslatorBycharacter, appendFullNameTranslatorBydata.frame, appendFullNameTranslatorByfunction, appendFullNameTranslatorBylist, clearFullNameTranslator, clearListOfFullNameTranslators, getDefaultFullName, getFullName, getFullNameTranslator, getListOfFullNameTranslators, getName, getTags, hasTag, hasTags, resetFullName, setFullName, setFullNameTranslator, setListOfFullNameTranslators, setName, setTags, updateFullName

**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, names, objectSize, print, save, asThis

**Author(s)**

Henrik Bengtsson

---

AromaUnitFracBCnBinarySet

*The AromaUnitFracBCnBinarySet class*


---

## Description

Package: aroma.core

### Class AromaUnitFracBCnBinarySet

#### Object

```

~|
~+--FullNameInterface
~~~~~|
~~~~~+--GenericDataFileSet
~~~~~|
~~~~~+--GenericTabularFileSet
~~~~~|
~~~~~+--AromaTabularBinarySet
~~~~~|
~~~~~+--AromaUnitSignalBinarySet
~~~~~|
~~~~~+--AromaUnitFracBCnBinarySet

```

#### Directly known subclasses:

```

public static class AromaUnitFracBCnBinarySet
extends AromaUnitSignalBinarySet

```

An AromaUnitFracBCnBinarySet object represents a set of [AromaUnitFracBCnBinaryFiles](#) with *identical* chip types.

## Usage

```
AromaUnitFracBCnBinarySet(...)
```

## Arguments

... Arguments passed to [AromaUnitSignalBinarySet](#).

## Details

The term "allele B fraction" is also know as "allele B frequency", which was coined by Peiffer et al. (2006). Note that the term "frequency" is a bit misleading since it is not a frequency in neither the statistical nor the population sense, but rather only proportion relative to the total amount of allele A and allele B signals, which is calculated for each sample independently.

**Fields and Methods****Methods:**

byName -  
writeDataFrame -

**Methods inherited from AromaUnitSignalBinarySet:**

byName, findByName, getAromaFullNameTranslatorSet, getAromaUgpFile, getChipType, getPlatform, validate, writeDataFrame

**Methods inherited from AromaTabularBinarySet:**

getDefaultFullName, getRootName, setAttributesBy, setAttributesBySampleAnnotationFile, setAttributesBySampleAnnotationSet, setAttributesByTags

**Methods inherited from GenericTabularFileSet:**

extractMatrix, calculateAverageColumnAcrossFiles

**Methods inherited from GenericDataFileSet:**

[, [[, anyDuplicated, anyNA, append, appendFiles, appendFullNamesTranslator, appendFullNamesTranslatorByNULL, appendFullNamesTranslatorByTabularTextFile, appendFullNamesTranslatorByTabularTextFileSet, appendFullNamesTranslatorBydata.frame, appendFullNamesTranslatorByfunction, appendFullNamesTranslatorBylist, as.character, as.list, byName, byPath, c, clearCache, clearFullNamesTranslator, clone, copyTo, dsApplyInPairs, duplicated, equals, extract, findByName, findDuplicated, getChecksum, getChecksumFileSet, getChecksumObjects, getDefaultFullName, getFile, getFileClass, getFileSize, getFiles, getFullNames, getNames, getOneFile, getPath, getPathnames, getSubdirs, gunzip, gzip, hasFile, indexOf, is.na, names, nbrOfFiles, rep, resetFullNames, setFullNamesTranslator, sortBy, unique, update2, updateFullName, updateFullNames, validate, getFullNameTranslatorSet, getParentName

**Methods inherited from FullNameInterface:**

appendFullNameTranslator, appendFullNameTranslatorByNULL, appendFullNameTranslatorByTabularTextFile, appendFullNameTranslatorByTabularTextFileSet, appendFullNameTranslatorBycharacter, appendFullNameTranslatorBydata.frame, appendFullNameTranslatorByfunction, appendFullNameTranslatorBylist, clearFullNameTranslator, clearListOfFullNameTranslators, getDefaultFullName, getFullName, getFullNameTranslator, getListOfFullNameTranslators, getName, getTags, hasTag, hasTags, resetFullName, setFullName, setFullNameTranslator, setListOfFullNameTranslators, setName, setTags, updateFullName

**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, names, objectSize, print, save, asThis

**Author(s)**

Henrik Bengtsson

**References**

[1] Peiffer et al., *High-resolution genomic profiling of chromosomal aberrations using Infinium whole-genome genotyping*, Genome Res, 2006.

---

AromaUnitGenotypeCallFile

*The AromaUnitGenotypeCallFile class*


---

**Description**

Package: aroma.core

**Class AromaUnitGenotypeCallFile**

Object

```

~~|
~~+--FullNameInterface
~~~~~|
~~~~~+--GenericDataFile
~~~~~|
~~~~~+--ColumnNamesInterface
~~~~~|
~~~~~+--GenericTabularFile
~~~~~|
~~~~~+--CacheKeyInterface
~~~~~|
~~~~~+--FileCacheKeyInterface
~~~~~|
~~~~~+--AromaTabularBinaryFile
~~~~~|
~~~~~+--AromaPlatformInterface
~~~~~|
~~~~~+--AromaUnitSignalBinaryFile
~~~~~|
~~~~~+--AromaUnitCallFile
~~~~~|
~~~~~+--AromaUnitGenotypeCallFile

```

**Directly known subclasses:**

```

public static class AromaUnitGenotypeCallFile
extends AromaUnitCallFile

```

An AromaUnitGenotypeCallFile is a [AromaUnitTabularBinaryFile](#).

**Usage**

```
AromaUnitGenotypeCallFile(...)
```

**Arguments**

... Arguments passed to [AromaUnitTabularBinaryFile](#).

**Fields and Methods****Methods:**

```
extractGenotypeMatrix -
extractGenotypes      -
isHeterozygous        -
isHomozygous          -
updateGenotypes       -
```

**Methods inherited from AromaUnitCallFile:**

allocate, extractCallArray, extractCalls, extractMatrix, findUnitsTodo

**Methods inherited from AromaUnitSignalBinaryFile:**

allocate, allocateFromUnitAnnotationDataFile, allocateFromUnitNamesFile, as.character, extractMatrix, extractRawGenomicSignals, fromFile, getChipType, getExtensionPattern, getFilenameExtension, getNumberOfFilesAveraged, getPlatform, isAverageFile, nbrOfUnits, readDataFrame, writeDataFrame

**Methods inherited from AromaPlatformInterface:**

getAromaPlatform, getAromaUflFile, getAromaUgpFile, getChipType, getPlatform, getUnitAnnotationDataFile, getUnitNamesFile, getUnitTypesFile, isCompatibleWith

**Methods inherited from AromaTabularBinaryFile:**

[, [<-, [[, allocate, as.character, colApply, colMeans, colStats, colSums, dimnames<-, getBytesPerColumn, getColClasses, getDefaultColumnNames, getRootName, importFrom, nbrOfColumns, nbrOfRows, readColumns, readDataFrame, readFooter, readHeader, readRawFooter, setAttributesByTags, subset, summary, updateData, updateDataColumn, writeFooter, writeRawFooter

**Methods inherited from FileCacheKeyInterface:**

getCacheKey

**Methods inherited from CacheKeyInterface:**

getCacheKey

**Methods inherited from GenericTabularFile:**

[, as.character, dim, extractMatrix, head, nbrOfColumns, nbrOfRows, readColumns, readDataFrame, tail, writeColumnsToFiles

**Methods inherited from ColumnNamesInterface:**

appendColumnNamesTranslator, appendColumnNamesTranslatorByNULL, appendColumnNamesTranslatorBycharacter, appendColumnNamesTranslatorByfunction, appendColumnNamesTranslatorBylist, clearColumnNamesTranslator, clearListOfColumnNamesTranslators, getColumnNames,



getColumnNamesTranslator, getDefaultColumnNames, getListOfColumnNamesTranslators, nbrOfColumns, setColumnNames, setColumnNamesTranslator, setListOfColumnNamesTranslators, updateColumnNames

**Methods inherited from GenericDataFile:**

as.character, clone, compareChecksum, copyTo, equals, fromFile, getAttribute, getAttributes, getChecksum, getChecksumFile, getCreatedOn, getDefaultFullName, getExtension, getExtensionPattern, getFileSize, getFileType, getFilename, getFilenameExtension, getLastAccessedOn, getLastModifiedOn, getOutputExtension, getPath, getPathname, gunzip, gzip, hasBeenModified, is.na, isFile, isGzipped, linkTo, readChecksum, renameTo, renameToUpperCaseExt, setAttribute, setAttributes, setAttributesBy, setAttributesByTags, setExtensionPattern, testAttributes, validate, validateChecksum, writeChecksum, getParentName

**Methods inherited from FullNameInterface:**

appendFullNameTranslator, appendFullNameTranslatorByNULL, appendFullNameTranslatorByTabularTextFile, appendFullNameTranslatorByTabularTextFileSet, appendFullNameTranslatorBycharacter, appendFullNameTranslatorBydata.frame, appendFullNameTranslatorByfunction, appendFullNameTranslatorBylist, clearFullNameTranslator, clearListOfFullNameTranslators, getDefaultFullName, getFullName, getFullNameTranslator, getListOfFullNameTranslators, getName, getTags, hasTag, hasTags, resetFullName, setFullName, setFullNameTranslator, setListOfFullNameTranslators, setName, setTags, updateFullName

**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, names, objectSize, print, save, asThis

**Author(s)**

Henrik Bengtsson

---

AromaUnitGenotypeCallSet

*The AromaUnitGenotypeCallSet class*

---

**Description**

Package: aroma.core

**Class AromaUnitGenotypeCallSet**

Object

~~|

~~+--FullNameInterface

~~~~~|

~~~~~+--GenericDataFileSet

~~~~~|

~~~~~+--GenericTabularFileSet

~~~~~|

~~~~~+--AromaTabularBinarySet

```

~~~~~|
~~~~~+--AromaUnitSignalBinarySet
~~~~~|
~~~~~+--AromaUnitCallSet
~~~~~|
~~~~~+--AromaUnitGenotypeCallSet

```

### Directly known subclasses:

```

public static class AromaUnitGenotypeCallSet
extends AromaUnitCallSet

```

An AromaUnitGenotypeCallSet object represents a set of AromaUnitGenotypeCallFiles with *identical* chip types.

### Usage

```
AromaUnitGenotypeCallSet(...)
```

### Arguments

... Arguments passed to AromaUnitCallSet.

### Fields and Methods

#### Methods:

```

byName -
extractGenotypes -

```

#### Methods inherited from AromaUnitCallSet:

byPath, extractCallArray, extractCalls, extractGenotypeMatrix, findByName, findUnitsTodo

#### Methods inherited from AromaUnitSignalBinarySet:

byName, findByName, getAromaFullNameTranslatorSet, getAromaUgpFile, getChipType, getPlatform, validate, writeDataFrame

#### Methods inherited from AromaTabularBinarySet:

getDefaultFullName, getRootName, setAttributesBy, setAttributesBySampleAnnotationFile, setAttributesBySampleAnnotationSet, setAttributesByTags

#### Methods inherited from GenericTabularFileSet:

extractMatrix, calculateAverageColumnAcrossFiles

#### Methods inherited from GenericDataFileSet:

[, [[, anyDuplicated, anyNA, append, appendFiles, appendFullNamesTranslator, appendFullNamesTranslatorByNULL, appendFullNamesTranslatorByTabularTextFile, appendFullNamesTranslatorByTabularTextFileSet, appendFullNamesTranslatorBydata.frame, appendFullNamesTranslatorByfunction, appendFullNamesTranslatorBylist, as.character, as.list, byName, byPath, c, clearCache,

clearFullNamesTranslator, clone, copyTo, dsApplyInPairs, duplicated, equals, extract, findByName, findDuplicated, getChecksum, getChecksumFileSet, getChecksumObjects, getDefaultFullName, getFile, getFileClass, getFileSize, getFiles, getFullNames, getNames, getOneFile, getPath, getPathnames, getSubdirs, gunzip, gzip, hasFile, indexOf, is.na, names, nbrOfFiles, rep, resetFullNames, setFullNamesTranslator, sortBy, unique, update2, updateFullName, updateFullNames, validate, getFullNameTranslatorSet, getParentName

**Methods inherited from FullNameInterface:**

appendFullNameTranslator, appendFullNameTranslatorByNULL, appendFullNameTranslatorByTabularTextFile, appendFullNameTranslatorByTabularTextFileSet, appendFullNameTranslatorBycharacter, appendFullNameTranslatorBydata.frame, appendFullNameTranslatorByfunction, appendFullNameTranslatorBylist, clearFullNameTranslator, clearListOfFullNameTranslators, getDefaultFullName, getFullName, getFullNameTranslator, getListOfFullNameTranslators, getName, getTags, hasTag, hasTags, resetFullName, setFullName, setFullNameTranslator, setListOfFullNameTranslators, setName, setTags, updateFullName

**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, names, objectSize, print, save, asThis

**Author(s)**

Henrik Bengtsson

---

AromaUnitPscnBinaryFile

*The AromaUnitPscnBinaryFile class*

---

**Description**

Package: aroma.core

**Class AromaUnitPscnBinaryFile**

**Object**

```

~~|
~~+--FullNameInterface
~~~~~|
~~~~~+--GenericDataFile
~~~~~|
~~~~~+--ColumnNamesInterface
~~~~~|
~~~~~+--GenericTabularFile
~~~~~|
~~~~~+--CacheKeyInterface
~~~~~|
~~~~~+--FileCacheKeyInterface
~~~~~|

```

```

~~~~~+--AromaTabularBinaryFile
~~~~~|
~~~~~+--AromaPlatformInterface
~~~~~|
~~~~~+--AromaUnitSignalBinaryFile
~~~~~|
~~~~~+--CopyNumberDataFile
~~~~~|
~~~~~+--AromaUnitPscnBinaryFile

```

### Directly known subclasses:

public abstract static class **AromaUnitPscnBinaryFile**  
 extends *CopyNumberDataFile*

An *AromaUnitPscnBinaryFile* is a *AromaUnitSignalBinaryFile* that holds total copy number signals (TCNs) and allele B fractions (BAFs). The TCNs can either be on an unknown scale or ratios relative to a reference. The signals are always stored on the original scale, i.e. they are never stored on the logarithmic scale. The BAFs are always on a  $[0-\text{eps}, 1+\text{eps}]$  scale, where  $\text{eps} \geq 0$ .

### Usage

```
AromaUnitPscnBinaryFile(...)
```

### Arguments

... Arguments passed to *AromaUnitSignalBinaryFile*.

### Fields and Methods

#### Methods:

```

extractRawCopyNumbers -
hasAlleleBFractions -
hasStrandiness -
hasTotalCopyNumberRatios -

```

#### Methods inherited from *CopyNumberDataFile*:

as, as.CopyNumberDataFile, getNumberOfFilesAveraged, hasAlleleBFractions, hasStrandiness

#### Methods inherited from *AromaUnitSignalBinaryFile*:

allocate, allocateFromUnitAnnotationDataFile, allocateFromUnitNamesFile, as.character, extractMatrix, extractRawGenomicSignals, fromFile, getChipType, getExtensionPattern, getFilenameExtension, getNumberOfFilesAveraged, getPlatform, isAverageFile, nbrOfUnits, readDataFrame, writeDataFrame

**Methods inherited from AromaPlatformInterface:**

getAromaPlatform, getAromaUflFile, getAromaUgpFile, getChipType, getPlatform, getUnitAnnotationDataFile, getUnitNamesFile, getUnitTypesFile, isCompatibleWith

**Methods inherited from AromaTabularBinaryFile:**

[, [<-, [[, allocate, as.character, colApply, colMeans, colStats, colSums, dimnames<-, getBytesPerColumn, getColClasses, getDefaultColumnNames, getRootName, importFrom, nbrOfColumns, nbrOfRows, readColumns, readDataFrame, readFooter, readHeader, readRawFooter, setAttributesByTags, subset, summary, updateData, updateDataColumn, writeFooter, writeRawFooter

**Methods inherited from FileCacheKeyInterface:**

getCacheKey

**Methods inherited from CacheKeyInterface:**

getCacheKey

**Methods inherited from GenericTabularFile:**

[, as.character, dim, extractMatrix, head, nbrOfColumns, nbrOfRows, readColumns, readDataFrame, tail, writeColumnsToFiles

**Methods inherited from ColumnNamesInterface:**

appendColumnNamesTranslator, appendColumnNamesTranslatorByNULL, appendColumnNamesTranslatorBycharacter, appendColumnNamesTranslatorByfunction, appendColumnNamesTranslatorBylist, clearColumnNamesTranslator, clearListOfColumnNamesTranslators, getColumnNames, getColumnNamesTranslator, getDefaultColumnNames, getListOfColumnNamesTranslators, nbrOfColumns, setColumnNames, setColumnNamesTranslator, setListOfColumnNamesTranslators, updateColumnNames

**Methods inherited from GenericDataFile:**

as.character, clone, compareChecksum, copyTo, equals, fromFile, getAttribute, getAttributes, getChecksum, getChecksumFile, getCreatedOn, getDefaultFullName, getExtension, getExtensionPattern, getFileSize, getFileType, getFilename, getFilenameExtension, getLastAccessedOn, getLastModifiedOn, getOutputExtension, getPath, getPathname, gunzip, gzip, hasBeenModified, is.na, isFile, isGzipped, linkTo, readChecksum, renameTo, renameToUpperCaseExt, setAttribute, setAttributes, setAttributesBy, setAttributesByTags, setExtensionPattern, testAttributes, validate, validateChecksum, writeChecksum, getParentName

**Methods inherited from FullNameInterface:**

appendFullNameTranslator, appendFullNameTranslatorByNULL, appendFullNameTranslatorByTabularTextFile, appendFullNameTranslatorByTabularTextFileSet, appendFullNameTranslatorBycharacter, appendFullNameTranslatorBydata.frame, appendFullNameTranslatorByfunction, appendFullNameTranslatorBylist, clearFullNameTranslator, clearListOfFullNameTranslators, getDefaultFullName, getFullName, getFullNameTranslator, getListOfFullNameTranslators, getName, getTags, hasTag, hasTags, resetFullName, setFullName, setFullNameTranslator, setListOfFullNameTranslators, setName, setTags, updateFullName

**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, names, objectSize, print, save, asThis

**Author(s)**

Henrik Bengtsson

---

AromaUnitPscnBinarySet

*The AromaUnitPscnBinarySet class*


---

## Description

Package: aroma.core

### Class AromaUnitPscnBinarySet

#### Object

```

~|
~+--FullNameInterface
~~~~~|
~~~~~+--GenericDataFileSet
~~~~~|
~~~~~+--GenericTabularFileSet
~~~~~|
~~~~~+--AromaTabularBinarySet
~~~~~|
~~~~~+--AromaUnitSignalBinarySet
~~~~~|
~~~~~+--CopyNumberDataSet
~~~~~|
~~~~~+--AromaUnitPscnBinarySet

```

#### Directly known subclasses:

```

public static class AromaUnitPscnBinarySet
extends CopyNumberDataSet

```

An AromaUnitPscnBinarySet object represents a set of [AromaUnitPscnBinaryFiles](#) with *identical* chip types.

#### Usage

```
AromaUnitPscnBinarySet(...)
```

#### Arguments

... Arguments passed to [AromaUnitSignalBinarySet](#).

#### Fields and Methods

##### Methods:

byName -  
 getAverageFile -  
 writeDataFrame -

**Methods inherited from CopyNumberDataSet:**

as, as.CopyNumberDataSet, doCBS, hasAlleleBFractions, hasStrandiness

**Methods inherited from AromaUnitSignalBinarySet:**

byName, findByName, getAromaFullNameTranslatorSet, getAromaUgpFile, getChipType, getPlatform, validate, writeDataFrame

**Methods inherited from AromaTabularBinarySet:**

getDefaultFullName, getRootName, setAttributesBy, setAttributesBySampleAnnotationFile, setAttributesBySampleAnnotationSet, setAttributesByTags

**Methods inherited from GenericTabularFileSet:**

extractMatrix, calculateAverageColumnAcrossFiles

**Methods inherited from GenericDataFileSet:**

[, [[, anyDuplicated, anyNA, append, appendFiles, appendFullNamesTranslator, appendFullNamesTranslatorByNULL, appendFullNamesTranslatorByTabularTextFile, appendFullNamesTranslatorByTabularTextFileSet, appendFullNamesTranslatorBydata.frame, appendFullNamesTranslatorByfunction, appendFullNamesTranslatorBylist, as.character, as.list, byName, byPath, c, clearCache, clearFullNamesTranslator, clone, copyTo, dsApplyInPairs, duplicated, equals, extract, findByName, findDuplicated, getChecksum, getChecksumFileSet, getChecksumObjects, getDefaultFullName, getFile, getFileClass, getFileSize, getFiles, getFullNames, getNames, getOneFile, getPath, getPathnames, getSubdirs, gunzip, gzip, hasFile, indexOf, is.na, names, nbrOfFiles, rep, resetFullNames, setFullNamesTranslator, sortBy, unique, update2, updateFullName, updateFullNames, validate, getFullNameTranslatorSet, getParentName

**Methods inherited from FullNameInterface:**

appendFullNameTranslator, appendFullNameTranslatorByNULL, appendFullNameTranslatorByTabularTextFile, appendFullNameTranslatorByTabularTextFileSet, appendFullNameTranslatorBycharacter, appendFullNameTranslatorBydata.frame, appendFullNameTranslatorByfunction, appendFullNameTranslatorBylist, clearFullNameTranslator, clearListOfFullNameTranslators, getDefaultFullName, getFullName, getFullNameTranslator, getListOfFullNameTranslators, getName, getTags, hasTag, hasTags, resetFullName, setFullName, setFullNameTranslator, setListOfFullNameTranslators, setName, setTags, updateFullName

**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, names, objectSize, print, save, asThis

**Author(s)**

Henrik Bengtsson

---

AromaUnitSignalBinaryFile

*The AromaUnitSignalBinaryFile class*


---

## Description

Package: aroma.core

**Class AromaUnitSignalBinaryFile**

Object

```

~|
~+--FullNameInterface
~~~~~|
~~~~~+--GenericDataFile
~~~~~|
~~~~~+--ColumnNamesInterface
~~~~~|
~~~~~+--GenericTabularFile
~~~~~|
~~~~~+--CacheKeyInterface
~~~~~|
~~~~~+--FileCacheKeyInterface
~~~~~|
~~~~~+--AromaTabularBinaryFile
~~~~~|
~~~~~+--AromaPlatformInterface
~~~~~|
~~~~~+--AromaUnitSignalBinaryFile

```

**Directly known subclasses:**

[AromaUnitCallFile](#), [AromaUnitFracBCnBinaryFile](#), [AromaUnitGenotypeCallFile](#), [AromaUnitPsc-nBinaryFile](#), [AromaUnitTotalCnBinaryFile](#), [AromaUnitTypesFile](#)

```
public static class AromaUnitSignalBinaryFile
```

```
extends AromaPlatformInterface
```

An AromaUnitSignalBinaryFile is a [AromaTabularBinaryFile](#).

## Usage

```
AromaUnitSignalBinaryFile(...)
```

## Arguments

... Arguments passed to [AromaTabularBinaryFile](#).



**Fields and Methods****Methods:**

|                          |  |
|--------------------------|--|
| extractMatrix            | -  |
| getChipType              | -  |
| getNumberOfFilesAveraged | -  |
| getPlatform              | -  |
| isAverageFile            | -  |
| nbrOfUnits               | -  |
| readDataFrame            | -  |
| writeDataFrame           | Writes the data file as a tab-delimited text file. |

**Methods inherited from AromaPlatformInterface:**

getAromaPlatform, getAromaUflFile, getAromaUgpFile, getChipType, getPlatform, getUnitAnnotationDataFile, getUnitNamesFile, getUnitTypesFile, isCompatibleWith

**Methods inherited from AromaTabularBinaryFile:**

[, [<-, [[, allocate, as.character, colApply, colMeans, colStats, colSums, dimnames<-, getBytesPerColumn, getColClasses, getDefaultColumnNames, getRootName, importFrom, nbrOfColumns, nbrOfRows, readColumns, readDataFrame, readFooter, readHeader, readRawFooter, setAttributesByTags, subset, summary, updateData, updateDataColumn, writeFooter, writeRawFooter

**Methods inherited from FileCacheKeyInterface:**

getCacheKey

**Methods inherited from CacheKeyInterface:**

getCacheKey

**Methods inherited from GenericTabularFile:**

[, as.character, dim, extractMatrix, head, nbrOfColumns, nbrOfRows, readColumns, readDataFrame, tail, writeColumnsToFiles

**Methods inherited from ColumnNamesInterface:**

appendColumnNamesTranslator, appendColumnNamesTranslatorByNULL, appendColumnNamesTranslatorBycharacter, appendColumnNamesTranslatorByfunction, appendColumnNamesTranslatorBylist, clearColumnNamesTranslator, clearListOfColumnNamesTranslators, getColumnNames, getColumnNamesTranslator, getDefaultColumnNames, getListOfColumnNamesTranslators, nbrOfColumns, setColumnNames, setColumnNamesTranslator, setListOfColumnNamesTranslators, updateColumnNames

**Methods inherited from GenericDataFile:**

as.character, clone, compareChecksum, copyTo, equals, fromFile, getAttribute, getAttributes, getChecksum, getChecksumFile, getCreatedOn, getDefaultFullName, getExtension, getExtensionPattern, getFileSize, getFileType, getFilename, getFilenameExtension, getLastAccessedOn, getLastModifiedOn, getOutputExtension, getPath, getPathname, gunzip, gzip, hasBeenModified, is.na, isFile, isGzipped, linkTo, readChecksum, renameTo, renameToUpperCaseExt, setAttribute, setAttributes, setAttributesBy, setAttributesByTags, setExtensionPattern, testAttributes, validate, validateChecksum, writeChecksum, getParentName

**Methods inherited from FullNameInterface:**

appendFullNameTranslator, appendFullNameTranslatorByNULL, appendFullNameTranslatorByTabularTextFile, appendFullNameTranslatorByTabularTextFileSet, appendFullNameTranslatorBycharacter, appendFullNameTranslatorBydata.frame, appendFullNameTranslatorByfunction, appendFullNameTranslatorBylist, clearFullNameTranslator, clearListOfFullNameTranslators, getDefaultFullName, getFullName, getFullNameTranslator, getListOfFullNameTranslators, getName, getTags, hasTag, hasTags, resetFullName, setFullName, setFullNameTranslator, setListOfFullNameTranslators, setName, setTags, updateFullName

**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, names, objectSize, print, save, asThis

**Author(s)**

Henrik Bengtsson

**See Also**

[AromaTabularBinaryFile](#).

---

AromaUnitSignalBinarySet

*The AromaUnitSignalBinarySet class*

---

**Description**

Package: aroma.core

**Class AromaUnitSignalBinarySet****Object**

```

~~|
~~+--FullNameInterface
~~~~~|
~~~~~+--GenericDataFileSet
~~~~~|
~~~~~+--GenericTabularFileSet
~~~~~|
~~~~~+--AromaTabularBinarySet
~~~~~|
~~~~~+--AromaUnitSignalBinarySet

```

**Directly known subclasses:**

[AromaUnitCallSet](#), [AromaUnitFracBCnBinarySet](#), [AromaUnitGenotypeCallSet](#), [AromaUnitPscnBinarySet](#), [AromaUnitTotalCnBinarySet](#)

```
public static class AromaUnitSignalBinarySet
  extends AromaTabularBinarySet
```

An `AromaUnitSignalBinarySet` object represents a set of `AromaUnitSignalBinaryFiles` with *identical* chip types.

### Usage

```
AromaUnitSignalBinarySet(...)
```

### Arguments

... Arguments passed to `AromaTabularBinarySet`.

### Fields and Methods

#### Methods:

|                              |   |
|------------------------------|---|
| <code>byName</code>          | -   |
| <code>getAromaUgpFile</code> | -   |
| <code>getChipType</code>     | -   |
| <code>getPlatform</code>     | -   |
| <code>writeDataFrame</code>  | Writes the data set as a tab-delimited text file. |

#### Methods inherited from `AromaTabularBinarySet`:

`getDefaultFullName`, `getRootName`, `setAttributesBy`, `setAttributesBySampleAnnotationFile`, `setAttributesBySampleAnnotationSet`, `setAttributesByTags`

#### Methods inherited from `GenericTabularFileSet`:

`extractMatrix`, `calculateAverageColumnAcrossFiles`

#### Methods inherited from `GenericDataFileSet`:

`[`, `[[`, `anyDuplicated`, `anyNA`, `append`, `appendFiles`, `appendFullNamesTranslator`, `appendFullNamesTranslatorByNULL`, `appendFullNamesTranslatorByTabularTextFile`, `appendFullNamesTranslatorByTabularTextFileSet`, `appendFullNamesTranslatorBydata.frame`, `appendFullNamesTranslatorByfunction`, `appendFullNamesTranslatorBylist`, `as.character`, `as.list`, `byName`, `byPath`, `c`, `clearCache`, `clearFullNamesTranslator`, `clone`, `copyTo`, `dsApplyInPairs`, `duplicated`, `equals`, `extract`, `findByName`, `findDuplicated`, `getChecksum`, `getChecksumFileSet`, `getChecksumObjects`, `getDefaultFullName`, `getFile`, `getFileClass`, `getFileSize`, `getFiles`, `getFullNames`, `getNames`, `getOneFile`, `getPath`, `getPathnames`, `getSubdirs`, `gunzip`, `gzip`, `hasFile`, `indexOf`, `is.na`, `names`, `nbrOfFiles`, `rep`, `resetFullNames`, `setFullNamesTranslator`, `sortBy`, `unique`, `update2`, `updateFullName`, `updateFullNames`, `validate`, `getFullNameTranslatorSet`, `getParentName`

#### Methods inherited from `FullNameInterface`:

`appendFullNameTranslator`, `appendFullNameTranslatorByNULL`, `appendFullNameTranslatorByTabularTextFile`, `appendFullNameTranslatorByTabularTextFileSet`, `appendFullNameTranslatorBycharacter`, `appendFullNameTranslatorBydata.frame`, `appendFullNameTranslatorByfunction`, `appendFullNameTranslatorBylist`, `clearFullNameTranslator`, `clearListOfFullNameTranslators`, `getDefaultFullName`, `getFullName`, `getFullNameTranslator`, `getListOfFullNameTranslators`, `getName`, `getTags`,

hasTag, hasTags, resetFullName, setFullName, setFullNameTranslator, setListOfFullNameTranslators, setName, setTags, updateFullName

**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, names, objectSize, print, save, asThis

**Author(s)**

Henrik Bengtsson

---

AromaUnitTabularBinaryFile

*The AromaUnitTabularBinaryFile class*

---

**Description**

Package: aroma.core

**Class AromaUnitTabularBinaryFile**

Object

```

~~|
~~+--FullNameInterface
~~~~~|
~~~~~+--GenericDataFile
~~~~~|
~~~~~+--ColumnNamesInterface
~~~~~|
~~~~~+--GenericTabularFile
~~~~~|
~~~~~+--CacheKeyInterface
~~~~~|
~~~~~+--FileCacheKeyInterface
~~~~~|
~~~~~+--AromaTabularBinaryFile
~~~~~|
~~~~~+--AromaPlatformInterface
~~~~~|
~~~~~+--AromaMicroarrayTabularBinaryFile
~~~~~|
~~~~~+--UnitAnnotationDataFile
~~~~~|
~~~~~+--AromaUnitTabularBinaryFile

```

**Directly known subclasses:**

[AromaUflFile](#), [AromaUgcFile](#), [AromaUgpFile](#), [AromaUnitChromosomeTabularBinaryFile](#), [AromaUnitGcContentFile](#)

public abstract static class **AromaUnitTabularBinaryFile**  
extends [UnitAnnotationDataFile](#)

A [AromaUnitTabularBinaryFile](#) is an [AromaTabularBinaryFile](#) with the constraint that the rows map one-to-one to, and in the same order as, the units in a annotation chip type file (e.g. CDF file). The (full) chip type of the annotation chip type file is given by the mandatory file footer chipType.

**Usage**

```
AromaUnitTabularBinaryFile(...)
```

**Arguments**

... Arguments passed to [AromaTabularBinaryFile](#).

**Fields and Methods****Methods:**

|                                |  |
|--------------------------------|--|
| <a href="#">byChipType</a>     | -  |
| <a href="#">nbrOfUnits</a>     | -  |
| <a href="#">writeDataFrame</a> | Writes the data file as a tab-delimited text file. |

**Methods inherited from [UnitAnnotationDataFile](#):**

[byChipType](#), [getAromaUflFile](#), [getAromaUgpFile](#), [getChipType](#), [getDefaultExtension](#), [getPlatform](#), [nbrOfUnits](#)

**Methods inherited from [AromaMicroarrayTabularBinaryFile](#):**

[allocate](#), [as.character](#), [byChipType](#), [findByChipType](#), [getChipType](#), [getFilenameExtension](#), [getPlatform](#)

**Methods inherited from [AromaPlatformInterface](#):**

[getAromaPlatform](#), [getAromaUflFile](#), [getAromaUgpFile](#), [getChipType](#), [getPlatform](#), [getUnitAnnotationDataFile](#), [getUnitNamesFile](#), [getUnitTypesFile](#), [isCompatibleWith](#)

**Methods inherited from [AromaTabularBinaryFile](#):**

[\[](#), [\[<-](#), [\[\[](#), [allocate](#), [as.character](#), [colApply](#), [colMeans](#), [colStats](#), [colSums](#), [dimnames<-](#), [getBytesPerColumn](#), [getColClasses](#), [getDefaultColumnNames](#), [getRootName](#), [importFrom](#), [nbrOfColumns](#), [nbrOfRows](#), [readColumns](#), [readDataFrame](#), [readFooter](#), [readHeader](#), [readRawFooter](#), [setAttributesByTags](#), [subset](#), [summary](#), [updateData](#), [updateDataColumn](#), [writeFooter](#), [writeRawFooter](#)

**Methods inherited from [FileCacheKeyInterface](#):**

[getCacheKey](#)

**Methods inherited from [CacheKeyInterface](#):**

[getCacheKey](#)

**Methods inherited from GenericTabularFile:**

[, as.character, dim, extractMatrix, head, nbrOfColumns, nbrOfRows, readColumns, readDataFrame, tail, writeColumnsToFiles

**Methods inherited from ColumnNamesInterface:**

appendColumnNamesTranslator, appendColumnNamesTranslatorByNULL, appendColumnNamesTranslatorBycharacter, appendColumnNamesTranslatorByfunction, appendColumnNamesTranslatorBylist, clearColumnNamesTranslator, clearListOfColumnNamesTranslators, getColumnNames, getColumnNamesTranslator, getDefaultColumnNames, getListOfColumnNamesTranslators, nbrOfColumns, setColumnNames, setColumnNamesTranslator, setListOfColumnNamesTranslators, updateColumnNames

**Methods inherited from GenericDataFile:**

as.character, clone, compareChecksum, copyTo, equals, fromFile, getAttribute, getAttributes, getChecksum, getChecksumFile, getCreatedOn, getDefaultFullName, getExtension, getExtensionPattern, getFileSize, getFileType, getFilename, getFilenameExtension, getLastAccessedOn, getLastModifiedOn, getOutputExtension, getPath, getPathname, gunzip, gzip, hasBeenModified, is.na, isFile, isGzipped, linkTo, readChecksum, renameTo, renameToUpperCaseExt, setAttribute, setAttributes, setAttributesBy, setAttributesByTags, setExtensionPattern, testAttributes, validate, validateChecksum, writeChecksum, getParentName

**Methods inherited from FullNameInterface:**

appendFullNameTranslator, appendFullNameTranslatorByNULL, appendFullNameTranslatorByTabularTextFile, appendFullNameTranslatorByTabularTextFileSet, appendFullNameTranslatorBycharacter, appendFullNameTranslatorBydata.frame, appendFullNameTranslatorByfunction, appendFullNameTranslatorBylist, clearFullNameTranslator, clearListOfFullNameTranslators, getDefaultFullName, getFullName, getFullNameTranslator, getListOfFullNameTranslators, getName, getTags, hasTag, hasTags, resetFullName, setFullName, setFullNameTranslator, setListOfFullNameTranslators, setName, setTags, updateFullName

**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, names, objectSize, print, save, asThis

**Author(s)**

Henrik Bengtsson

---

AromaUnitTotalCnBinaryFile

*The AromaUnitTotalCnBinaryFile class*

---

**Description**

Package: aroma.core

**Class AromaUnitTotalCnBinaryFile**

[Object](#)

~~|

```

~+--FullNameInterface
~~~~~|
~~~~~+--GenericDataFile
~~~~~|
~~~~~+--ColumnNamesInterface
~~~~~|
~~~~~+--GenericTabularFile
~~~~~|
~~~~~+--CacheKeyInterface
~~~~~|
~~~~~+--FileCacheKeyInterface
~~~~~|
~~~~~+--AromaTabularBinaryFile
~~~~~|
~~~~~+--AromaPlatformInterface
~~~~~|
~~~~~+--AromaUnitSignalBinaryFile
~~~~~|
~~~~~+--CopyNumberDataFile
~~~~~|
~~~~~+--AromaUnitTotalCnBinaryFile

```

**Directly known subclasses:**

public abstract static class **AromaUnitTotalCnBinaryFile**  
 extends *CopyNumberDataFile*

An AromaUnitTotalCnBinaryFile is a *AromaUnitSignalBinaryFile*.

**Usage**

```
AromaUnitTotalCnBinaryFile(...)
```

**Arguments**

... Arguments passed to *AromaUnitSignalBinaryFile*.

**Fields and Methods****Methods:**

```

extractPSCNArray      -
extractPSCNMatrix    -
extractRawCopyNumbers -
getAM                 -
hasAlleleBFractions  -

```

hasStrandiness -

**Methods inherited from CopyNumberDataFile:**

as, as.CopyNumberDataFile, getNumberOfFilesAveraged, hasAlleleBFractions, hasStrandiness

**Methods inherited from AromaUnitSignalBinaryFile:**

allocate, allocateFromUnitAnnotationDataFile, allocateFromUnitNamesFile, as.character, extractMatrix, extractRawGenomicSignals, fromFile, getChipType, getExtensionPattern, getFilenameExtension, getNumberOfFilesAveraged, getPlatform, isAverageFile, nbrOfUnits, readDataFrame, writeDataFrame

**Methods inherited from AromaPlatformInterface:**

getAromaPlatform, getAromaUffFile, getAromaUgpFile, getChipType, getPlatform, getUnitAnnotationDataFile, getUnitNamesFile, getUnitTypesFile, isCompatibleWith

**Methods inherited from AromaTabularBinaryFile:**

[, [<-, [[, allocate, as.character, colApply, colMeans, colStats, colSums, dimnames<-, getBytesPerColumn, getColClasses, getDefaultColumnNames, getRootName, importFrom, nbrOfColumns, nbrOfRows, readColumns, readDataFrame, readFooter, readHeader, readRawFooter, setAttributesByTags, subset, summary, updateData, updateDataColumn, writeFooter, writeRawFooter

**Methods inherited from FileCacheKeyInterface:**

getCacheKey

**Methods inherited from CacheKeyInterface:**

getCacheKey

**Methods inherited from GenericTabularFile:**

[, as.character, dim, extractMatrix, head, nbrOfColumns, nbrOfRows, readColumns, readDataFrame, tail, writeColumnsToFiles

**Methods inherited from ColumnNamesInterface:**

appendColumnNamesTranslator, appendColumnNamesTranslatorByNULL, appendColumnNamesTranslatorBycharacter, appendColumnNamesTranslatorByfunction, appendColumnNamesTranslatorBylist, clearColumnNamesTranslator, clearListOfColumnNamesTranslators, getColumnNames, getColumnNamesTranslator, getDefaultColumnNames, getListOfColumnNamesTranslators, nbrOfColumns, setColumnNames, setColumnNamesTranslator, setListOfColumnNamesTranslators, updateColumnNames

**Methods inherited from GenericDataFile:**

as.character, clone, compareChecksum, copyTo, equals, fromFile, getAttribute, getAttributes, getChecksum, getChecksumFile, getCreatedOn, getDefaultFullName, getExtension, getExtensionPattern, getFileSize, getFileType, getFilename, getFilenameExtension, getLastAccessedOn, getLastModifiedOn, getOutputExtension, getPath, getPathname, gunzip, gzip, hasBeenModified, is.na, isFile, isGzipped, linkTo, readChecksum, renameTo, renameToUpperCaseExt, setAttribute, setAttributes, setAttributesBy, setAttributesByTags, setExtensionPattern, testAttributes, validate, validateChecksum, writeChecksum, getParentName

**Methods inherited from FullNameInterface:**

appendFullNameTranslator, appendFullNameTranslatorByNULL, appendFullNameTranslatorByTabularTextFile, appendFullNameTranslatorByTabularTextFileSet, appendFullNameTranslatorBycharacter, appendFullNameTranslatorBydata.frame, appendFullNameTranslatorByfunction, appendFullNameTranslatorBylist, clearFullNameTranslator, clearListOfFullNameTranslators, getDefaultFullName, getFullName, getFullNameTranslator, getListOfFullNameTranslators, getName, getTags,



hasTag, hasTags, resetFullName, setFullName, setFullNameTranslator, setListOfFullNameTranslators, setName, setTags, updateFullName

**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, names, objectSize, print, save, asThis

**Author(s)**

Henrik Bengtsson

---

AromaUnitTotalCnBinarySet

*The AromaUnitTotalCnBinarySet class*

---

**Description**

Package: aroma.core

**Class AromaUnitTotalCnBinarySet**

Object

```

~~|
~~+--FullNameInterface
~~~~~|
~~~~~+--GenericDataFileSet
~~~~~|
~~~~~+--GenericTabularFileSet
~~~~~|
~~~~~+--AromaTabularBinarySet
~~~~~|
~~~~~+--AromaUnitSignalBinarySet
~~~~~|
~~~~~+--CopyNumberDataSet
~~~~~|
~~~~~+--AromaUnitTotalCnBinarySet

```

**Directly known subclasses:**

```

public static class AromaUnitTotalCnBinarySet
extends CopyNumberDataSet

```

An AromaUnitTotalCnBinarySet object represents a set of [AromaUnitTotalCnBinaryFiles](#) with *identical* chip types.

**Usage**

```
AromaUnitTotalCnBinarySet(...)
```

**Arguments**

```
... Arguments passed to AromaUnitSignalBinarySet.
```

**Fields and Methods****Methods:**

```
as.AromaUnitTotalCnBinarySetTuple -
as.CopyNumberDataSetTuple -
byName -
exportAromaUnitPscnBinarySet -
exportTotalCnRatioSet -
extractPSCNArray -
getAverageFile -
getUnitNamesFile -
writeDataFrame -
```

**Methods inherited from CopyNumberDataSet:**

as, as.CopyNumberDataSet, doCBS, hasAlleleBFractions, hasStrandiness

**Methods inherited from AromaUnitSignalBinarySet:**

byName, findByName, getAromaFullNameTranslatorSet, getAromaUgpFile, getChipType, getPlatform, validate, writeDataFrame

**Methods inherited from AromaTabularBinarySet:**

getDefaultFullName, getRootName, setAttributesBy, setAttributesBySampleAnnotationFile, setAttributesBySampleAnnotationSet, setAttributesByTags

**Methods inherited from GenericTabularFileSet:**

extractMatrix, calculateAverageColumnAcrossFiles

**Methods inherited from GenericDataFileSet:**

[, [], anyDuplicated, anyNA, append, appendFiles, appendFullNamesTranslator, appendFullNamesTranslatorByNULL, appendFullNamesTranslatorByTabularTextFile, appendFullNamesTranslatorByTabularTextFileSet, appendFullNamesTranslatorBydata.frame, appendFullNamesTranslatorByfunction, appendFullNamesTranslatorBylist, as.character, as.list, byName, byPath, c, clearCache, clearFullNamesTranslator, clone, copyTo, dsApplyInPairs, duplicated, equals, extract, findByName, findDuplicated, getChecksum, getChecksumFileSet, getChecksumObjects, getDefaultFullName, getFile, getFileClass, getFileSize, getFiles, getFullNames, getNames, getOneFile, getPath, getPathnames, getSubdirs, gunzip, gzip, hasFile, indexOf, is.na, names, nbrOfFiles, rep, resetFullNames, setFullNamesTranslator, sortBy, unique, update2, updateFullName, updateFullNames, validate, getFullNameTranslatorSet, getParentName

**Methods inherited from FullNameInterface:**

appendFullNameTranslator, appendFullNameTranslatorByNULL, appendFullNameTranslatorByTabularTextFile, appendFullNameTranslatorByTabularTextFileSet, appendFullNameTranslatorBychar-

acter, appendFullNameTranslatorBydata.frame, appendFullNameTranslatorByfunction, appendFullNameTranslatorBylist, clearFullNameTranslator, clearListOfFullNameTranslators, getDefaultFullName, getFullName, getFullNameTranslator, getListOfFullNameTranslators, getName, getTags, hasTag, hasTags, resetFullName, setFullName, setFullNameTranslator, setListOfFullNameTranslators, setName, setTags, updateFullName

**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, names, objectSize, print, save, asThis

**Author(s)**

Henrik Bengtsson

---

AromaUnitTypesFile      *The AromaUnitTypesFile class*

---

**Description**

Package: aroma.core

**Class AromaUnitTypesFile**

Object

```

~~|
~~+--FullNameInterface
~~~~~|
~~~~~+--GenericDataFile
~~~~~|
~~~~~+--ColumnNamesInterface
~~~~~|
~~~~~+--GenericTabularFile
~~~~~|
~~~~~+--CacheKeyInterface
~~~~~|
~~~~~+--FileCacheKeyInterface
~~~~~|
~~~~~+--AromaTabularBinaryFile
~~~~~|
~~~~~+--AromaPlatformInterface
~~~~~|
~~~~~+--AromaUnitSignalBinaryFile
~~~~~|
~~~~~+--UnitAnnotationDataFile
~~~~~|
~~~~~+--UnitTypesFile
~~~~~|

```

~~~~~+---AromaUnitTypesFile

### Directly known subclasses:

public static class **AromaUnitTypesFile**  
 extends *UnitTypesFile*

### Usage

AromaUnitTypesFile(...)

### Arguments

... Arguments passed to [AromaUnitTabularBinaryFile](#).

### Fields and Methods

#### Methods:

|                         |   |
|-------------------------|---|
| getChipType             | - |
| getPlatform             | - |
| getUnitTypes            | - |
| importFromUnitTypesFile | - |

#### Methods inherited from UnitTypesFile:

getUnitTypes, nbrOfUnits

#### Methods inherited from UnitAnnotationDataFile:

byChipType, getAromaUflFile, getAromaUgpFile, getChipType, getDefaultExtension, getPlatform, nbrOfUnits

#### Methods inherited from AromaUnitSignalBinaryFile:

allocate, allocateFromUnitAnnotationDataFile, allocateFromUnitNamesFile, as.character, extractMatrix, extractRawGenomicSignals, fromFile, getChipType, getExtensionPattern, getFilenameExtension, getNumberOfFilesAveraged, getPlatform, isAverageFile, nbrOfUnits, readDataFrame, writeDataFrame

#### Methods inherited from AromaPlatformInterface:

getAromaPlatform, getAromaUflFile, getAromaUgpFile, getChipType, getPlatform, getUnitAnnotationDataFile, getUnitNamesFile, getUnitTypesFile, isCompatibleWith

#### Methods inherited from AromaTabularBinaryFile:

[, [<-, [[, allocate, as.character, colApply, colMeans, colStats, colSums, dimnames<-, getBytesPerColumn, getColClasses, getDefaultColumnNames, getRootName, importFrom, nbrOfColumns, nbrOfRows, readColumns, readDataFrame, readFooter, readHeader, readRawFooter, setAttributesByTags, subset, summary, updateData, updateDataColumn, writeFooter, writeRawFooter

**Methods inherited from FileCacheKeyInterface:**

getCacheKey

**Methods inherited from CacheKeyInterface:**

getCacheKey

**Methods inherited from GenericTabularFile:**

[, as.character, dim, extractMatrix, head, nbrOfColumns, nbrOfRows, readColumns, readDataFrame, tail, writeColumnsToFiles

**Methods inherited from ColumnNamesInterface:**

appendColumnNamesTranslator, appendColumnNamesTranslatorByNULL, appendColumnNamesTranslatorBycharacter, appendColumnNamesTranslatorByfunction, appendColumnNamesTranslatorBylist, clearColumnNamesTranslator, clearListOfColumnNamesTranslators, getColumnNames, getColumnNamesTranslator, getDefaultColumnNames, getListOfColumnNamesTranslators, nbrOfColumns, setColumnNames, setColumnNamesTranslator, setListOfColumnNamesTranslators, updateColumnNames

**Methods inherited from GenericDataFile:**

as.character, clone, compareChecksum, copyTo, equals, fromFile, getAttribute, getAttributes, getChecksum, getChecksumFile, getCreatedOn, getDefaultFullName, getExtension, getExtensionPattern, getFileSize, getFileType, getFilename, getFilenameExtension, getLastAccessedOn, getLastModifiedOn, getOutputExtension, getPath, getPathname, gunzip, gzip, hasBeenModified, is.na, isFile, isGzipped, linkTo, readChecksum, renameTo, renameToUpperCaseExt, setAttribute, setAttributes, setAttributesBy, setAttributesByTags, setExtensionPattern, testAttributes, validate, validateChecksum, writeChecksum, getParentName

**Methods inherited from FullNameInterface:**

appendFullNameTranslator, appendFullNameTranslatorByNULL, appendFullNameTranslatorByTabularTextFile, appendFullNameTranslatorByTabularTextFileSet, appendFullNameTranslatorBycharacter, appendFullNameTranslatorBydata.frame, appendFullNameTranslatorByfunction, appendFullNameTranslatorBylist, clearFullNameTranslator, clearListOfFullNameTranslators, getDefaultFullName, getFullName, getFullNameTranslator, getListOfFullNameTranslators, getName, getTags, hasTag, hasTags, resetFullName, setFullName, setFullNameTranslator, setListOfFullNameTranslators, setName, setTags, updateFullName

**Methods inherited from Object:**

\$, \$&lt;-, [[, [[&lt;-, as.character, attach, attachLocally, clearCache, clearLookupCache, clone, detach, equals, extend, finalize, getEnvironment, getFieldModifier, getFieldModifiers, getFields, getInstantiationTime, getStaticInstance, hasField, hashCode, ll, load, names, objectSize, print, save, asThis

**Author(s)**

Henrik Bengtsson

**Description**

Package: aroma.core

**Class BinnedScatter**

```
list
~~|
~~+--BinnedScatter
```

**Directly known subclasses:**

```
public class BinnedScatter
extends list
```

**Usage**

```
BinnedScatter(data=NULL, density=NULL, map=NULL, params=NULL)
```

**Arguments**

```
data          A Nx2 numeric matrix.
density       ...
map           ...
params        A list of parameters.
```

**Fields and Methods****Methods:**

```
plot          -
points        -
reorder       -
subsample     -
subset        -
```

**Methods inherited from list:**

Ops,nonStructure,vector-method, Ops,structure,vector-method, Ops,vector,nonStructure-method, Ops,vector,structure-method, all.equal, as.CopyNumberDataSetTuple, as.data.frame, attachLocally, callHooks, coerce,ANY,list-method, exportAromaUnitPscnBinarySet, listToXml, mergeBoxplotStats, relist, reset, type.convert, within

**Author(s)**

Henrik Bengtsson

**See Also**

The spatial density is estimated by internal functions of the **smoothScatter** package.

**Examples**

```
# Sample scatter data
n <- 10e3
x <- rnorm(n=n)
y <- rnorm(n=n)
xy <- cbind(x=x, y=sin(x)+y/5)

# Bin data and estimate densities
xyd <- binScatter(xy)

layout(matrix(1:4, nrow=2))
par(mar=c(5,4,2,1))

# Plot data
plot(xyd, pch=1)

# Thin scatter data by subsampling
rhos <- c(1/3, 1/4, 1/6)
for (kk in seq_along(rhos)) {
  xyd2 <- subsample(xyd, size=rhos[kk])
  points(xyd2, pch=1, col=kk+1)
}

for (kk in seq_along(rhos)) {
  xyd2 <- subsample(xyd, size=rhos[kk])
  plot(xyd2, pch=1, col=kk+1)
  mtext(side=3, line=0, sprintf("Density: %.1f%", 100*rhos[kk]))
}
```

CbsModel

*The CbsModel class***Description**

Package: aroma.core

**Class CbsModel**

**Object**

```
~|
~+--ChromosomalModel
~~~~~|
~~~~~+--CopyNumberChromosomalModel
~~~~~|
~~~~~+--CopyNumberSegmentationModel
~~~~~|
```

```
~~~~~+---CbsModel
```

### Directly known subclasses:

```
public static class CbsModel
extends CopyNumberSegmentationModel
```

This class represents the Circular Binary Segmentation (CBS) model [1].

### Usage

```
CbsModel(cesTuple=NULL, ..., seed=NULL)
```

### Arguments

|          |                                                                                                                                                                                                             |
|----------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| cesTuple | A <a href="#">CopyNumberDataSetTuple</a> .                                                                                                                                                                  |
| ...      | Arguments passed to the constructor of <a href="#">CopyNumberSegmentationModel</a> .                                                                                                                        |
| seed     | An (optional) <a href="#">integer</a> that if specified will (temporarily) set the random seed each time before calling the segmentation method. For more information, see <a href="#">segmentByCBS()</a> . |

### Fields and Methods

#### Methods:

*No methods defined.*

#### Methods inherited from [CopyNumberSegmentationModel](#):

fit, getAsteriskTags, getFitFunction, getFullNames, getRegions, getTags, plot, plotCopyNumberRegionLayers, writeRegions

#### Methods inherited from [CopyNumberChromosomalModel](#):

as.character, calculateChromosomeStatistics, calculateRatios, estimateSds, extractRawCopyNumbers, fit, getChromosomeLength, getDataFileMatrix, getMaxNAFraction, getNames, getOptionalArguments, getPairedNames, getRefSetTuple, getReference, getReferenceSetTuple, isPaired, newPlot, plotAxesLayers, plotChromosomesLayers, plotCytobandLayers, plotFitLayers, plotGridHorizontalLayers, plotRawCopyNumbers, plotSampleLayers, setReference

#### Methods inherited from [ChromosomalModel](#):

as.character, fit, getAlias, getAromaGenomeTextFile, getAsteriskTags, getChipType, getChipTypes, getChromosomes, getFullName, getFullNames, getGenome, getGenomeData, getGenomeFile, getListOfAromaUgpFiles, getName, getNames, getParentPath, getPath, getReportPath, getRootPath, getSetTuple, getSets, getTags, indexOf, nbrOfArrays, nbrOfChipTypes, setChromosomes, setGenome

#### 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, names, objectSize, print, save, asThis



**Author(s)**

Henrik Bengtsson

**References**

- [1] Olshen, A. B., Venkatraman, E. S., Lucito, R., Wigler, M. *Circular binary segmentation for the analysis of array-based DNA copy number data. Biostatistics 5: 557-572, 2004.*
- [2] Venkatraman, E. S. & Olshen, A. B. *A faster circular binary segmentation algorithm for the analysis of array CGH data. Bioinformatics, 2007.*

**See Also**[CopyNumberSegmentationModel](#).

---

ChromosomalModel      *The ChromosomalModel class*


---

**Description**

Package: aroma.core

**Class ChromosomalModel****Object**

~~|

~~+--ChromosomalModel

**Directly known subclasses:**[CbsModel](#), [CopyNumberChromosomalModel](#), [CopyNumberSegmentationModel](#), [GladModel](#), [HaarSeg-Model](#), [RawCopyNumberModel](#)

```
public abstract static class ChromosomalModel
  extends Object
```

This *abstract* class represents a chromosomal model.

**Usage**

```
ChromosomalModel(cesTuple=NULL, tags="*", genome="Human", chromosomes=NULL, ...)
```

**Arguments**

```
cesTuple      A AromaMicroarrayDataSetTuple.
tags          A character vector of tags.
genome        A character string specifying what genome is process.
chromosomes   (optional) A vector specifying which chromosomes to process.
...            Not used.
```

**Fields and Methods****Methods:**

|                                |                                         |
|--------------------------------|-----------------------------------------|
| fit                            | -                                       |
| <a href="#">getChipType</a>    | Gets a label for all chip types merged. |
| getChipTypes                   | -                                       |
| <a href="#">getChromosomes</a> | Gets the chromosomes to be processed.   |
| getFullName                    | -                                       |
| getFullNames                   | -                                       |
| getGenome                      | -                                       |
| getName                        | -                                       |
| <a href="#">getNames</a>       | Gets the names of the arrays.           |
| getPath                        | -                                       |
| getSets                        | -                                       |
| getTags                        | -                                       |
| indexOf                        | -                                       |
| <a href="#">nbrOfArrays</a>    | Gets the number of arrays.              |
| <a href="#">nbrOfChipTypes</a> | Gets the number of chip types.          |
| setChromosomes                 | -                                       |
| setGenome                      | -                                       |

**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, names, objectSize, print, save, asThis

**Requirements**

This class requires genome information annotation files for every chip type.

**Author(s)**

Henrik Bengtsson

---

ChromosomeExplorer      *The ChromosomeExplorer class*

---

**Description**

Package: aroma.core

**Class ChromosomeExplorer**

[Object](#)

~~|

```

~+---Explorer
~~~~~|
~~~~~+---ChromosomeExplorer

```

### Directly known subclasses:

```

public static class ChromosomeExplorer
extends Explorer

```

### Usage

```
ChromosomeExplorer(model=NULL, zooms=2^(0:6), ...)
```

### Arguments

|       |                                                                                                               |
|-------|---------------------------------------------------------------------------------------------------------------|
| model | A <a href="#">CopyNumberChromosomalModel</a> object.                                                          |
| zooms | An positive <a href="#">integer vector</a> specifying for which zoom levels the graphics should be generated. |
| ...   | Not used.                                                                                                     |

### Fields and Methods

#### Methods:

|                                |                                                                    |
|--------------------------------|--------------------------------------------------------------------|
| display                        | -                                                                  |
| <a href="#">getChromosomes</a> | Gets the chromosomes available.                                    |
| getFullNames                   | -                                                                  |
| <a href="#">getModel</a>       | Gets the model.                                                    |
| getNames                       | -                                                                  |
| getPath                        | -                                                                  |
| getZooms                       | -                                                                  |
| indexOf                        | -                                                                  |
| <a href="#">process</a>        | Generates image files, scripts and dynamic pages for the explorer. |
| <a href="#">setArrays</a>      | Sets the arrays.                                                   |
| setCytoband                    | -                                                                  |
| setZooms                       | -                                                                  |

#### Methods inherited from Explorer:

addIncludes, addIndexFile, as.character, display, getAlias, getArrayOfInput, getAsteriskTags, getFullName, getIncludePath, getMainPath, getName, getNameOfInput, getNames, getPath, getReportPathPattern, getRootPath, getSampleLayerPrefix, getSubname, getTags, getTagsOfInput, getTemplatePath, getVersion, nbrOfArrays, process, setAlias, setArrays, setReportPathPattern, setSubname, setup, splitByReportPathPattern, updateSetupExplorerFile

**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, names, objectSize, print, save, asThis

**Generating PNG images**

In order to get better looking graphs, but also to be able to generate bitmap images on systems without direct bitmap support, which is the case when running R in batch mode or on Unix without X11 support, images are created using the `png2` device (a wrapper for `bitmap()` imitating `png()`). The `png()` is only used if `png2()`, which requires Ghostscript, does not. Note, when images are created using `png2()`, the images does not appear immediately, although the function call is completed, so be patient.

**Author(s)**

Henrik Bengtsson

**See Also**

[CopyNumberChromosomalModel](#).

---

colBinnedSmoothing.matrix

*Binned smoothing of a matrix column by column*

---

**Description**

Binned smoothing of a matrix column by column.

**Usage**

```
## S3 method for class 'matrix'
colBinnedSmoothing(Y, x=seq_len(nrow(Y)), w=NULL, xOut=NULL, xOutRange=NULL,
  from=min(x, na.rm = TRUE), to=max(x, na.rm = TRUE), by=NULL, length.out=length(x),
  na.rm=TRUE, FUN="median", ..., verbose=FALSE)
```

**Arguments**

|      |                                                                                                                                         |
|------|-----------------------------------------------------------------------------------------------------------------------------------------|
| Y    | A <b>numeric</b> JxI <b>matrix</b> (or a <b>vector</b> of length J.)                                                                    |
| x    | A (optional) <b>numeric vector</b> specifying the positions of the J entries. The default is to assume uniformly distributed positions. |
| w    | A optional <b>numeric vector</b> of prior weights for each of the J entries.                                                            |
| xOut | Optional <b>numeric vector</b> of K bin center locations.                                                                               |

|                          |                                                                                                                                                                                                                                                                                                                                                                                         |
|--------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| xOutRange                | Optional Kx2 <a href="#">matrix</a> specifying the boundary locations for K bins, where each row represents a bin $[x_0, x_1)$ . If not specified, the boundaries are set to be the midpoints of the bin centers, such that the bins have maximum lengths without overlapping. Vice versa, if xOut is not specified, then xOut is set to be the mid points of the xOutRange boundaries. |
| from, to, by, length.out | If neither xOut nor xOutRange is specified, the xOut is generated uniformly from these arguments, which specify the center location of the first and the last bin, and the distance between the center locations, utilizing the <a href="#">seq()</a> function. Argument length.out can be used as an alternative to by, in case it specifies the total number of bins instead.         |
| FUN                      | A <a href="#">function</a> .                                                                                                                                                                                                                                                                                                                                                            |
| na.rm                    | If <a href="#">TRUE</a> , missing values are excluded, otherwise not.                                                                                                                                                                                                                                                                                                                   |
| ...                      | Not used.                                                                                                                                                                                                                                                                                                                                                                               |
| verbose                  | See <a href="#">Verbose</a> .                                                                                                                                                                                                                                                                                                                                                           |

### Details

Note that all zero-length bins  $[x_0, x_1)$  will get result in an [NA](#) value, because such bins contain no data points. This also means that `colBinnedSmoothing(Y, x=x, xOut=xOut)` where xOut contains duplicated values, will result in some zero-length bins and hence [NA](#) values.

### Value

Returns a [numeric](#) KxI [matrix](#) (or a [vector](#) of length K) where K is the total number of bins. The following attributes are also returned:

xOut The center locations of each bin.

xOutRange The bin boundaries.

count The number of data points within each bin (based solely on argument x).

binWidth The *average* bin width.

### Author(s)

Henrik Bengtsson

### See Also

[\\*colKernelSmoothing\(\)](#).

### Examples

```
# Number of tracks
I <- 4

# Number of data points per track
J <- 100
```

```

# Simulate data with a gain in track 2 and 3
x <- 1:J
Y <- matrix(rnorm(I*J, sd=1/2), ncol=I)
Y[30:50,2:3] <- Y[30:50,2:3] + 3

# Uniformly distributed equal-sized bins
Ys3 <- colBinnedSmoothing(Y, x=x, from=2, by=3)
Ys5 <- colBinnedSmoothing(Y, x=x, from=3, by=5)

# Custom bins
xOutOfRange <- t(matrix(c(
  1, 11,
  11, 31,
  31, 41,
  41, 51,
  51, 81,
  81, 91,
  91, 101
), nrow=2))
YsC <- colBinnedSmoothing(Y, x=x, xOutOfRange=xOutOfRange)

# Custom bins specified by center locations with
# maximized width relative to the neighboring bins.
xOut <- c(6, 21, 36, 46, 66, 86, 96)
YsD <- colBinnedSmoothing(Y, x=x, xOut=xOut)

xlim <- range(x)
ylim <- c(-3,5)
layout(matrix(1:I, ncol=1))
par(mar=c(3,3,1,1)+0.1, pch=19)
for (ii in 1:I) {
  plot(NA, xlim=xlim, ylim=ylim)
  points(x, Y[,ii], col="#999999")

  xOut <- attr(Ys3, "xOut")
  lines(xOut, Ys3[,ii], col=2)
  points(xOut, Ys3[,ii], col=2)

  xOut <- attr(Ys5, "xOut")
  lines(xOut, Ys5[,ii], col=3)
  points(xOut, Ys5[,ii], col=3)

  xOut <- attr(YsC, "xOut")
  lines(xOut, YsC[,ii], col=4)
  points(xOut, YsC[,ii], col=4, pch=15)

  xOut <- attr(YsD, "xOut")
  lines(xOut, YsD[,ii], col=5)
  points(xOut, YsD[,ii], col=5, pch=15)

  if (ii == 1) {
    legend("topright", pch=c(19,19,15,15), col=c(2,3,4,5),
          c("by=3", "by=5", "Custom #1", "Custom #2"), horiz=TRUE, bty="n")
  }
}

```

```

    }
  }

  # Sanity checks
  xOut <- x
  YsT <- colBinnedSmoothing(Y, x=x, xOut=xOut)
  stopifnot(all(YsT == Y))
  stopifnot(all(attr(YsT, "counts") == 1))

  xOut <- attr(YsD, "xOut")
  YsE <- colBinnedSmoothing(YsD, x=xOut, xOut=xOut)
  stopifnot(all(YsE == YsD))
  stopifnot(all(attr(YsE, "xOutRange") == attr(YsD, "xOutRange")))
  stopifnot(all(attr(YsE, "counts") == 1))

  # Scramble ordering of loci
  idxs <- sample(x)
  x2 <- x[idxs]
  Y2 <- Y[idxs, , drop=FALSE]
  Y2s <- colBinnedSmoothing(Y2, x=x2, xOut=x2)
  stopifnot(all(attr(Y2s, "xOut") == x2))
  stopifnot(all(attr(Y2s, "counts") == 1))
  stopifnot(all(Y2s == Y2))

  xOut <- x[seq(from=2, to=J, by=3)]
  YsT <- colBinnedSmoothing(Y, x=x, xOut=xOut)
  stopifnot(all(YsT == Ys3))
  stopifnot(all(attr(YsT, "counts") == 3))

  xOut <- x[seq(from=3, to=J, by=5)]
  YsT <- colBinnedSmoothing(Y, x=x, xOut=xOut)
  stopifnot(all(YsT == Ys5))
  stopifnot(all(attr(YsT, "counts") == 5))

```

---

```
colKernelSmoothing.matrix
```

*Kernel smoothing of a matrix column by column*

---

## Description

Kernel smoothing of a matrix column by column.

## Usage

```

## S3 method for class 'matrix'
colKernelSmoothing(Y, x=seq_len(nrow(Y)), w=NULL, xOut=x,
  kernel=c("gaussian", "uniform"), h, censorH=3, na.rm=TRUE, robust=FALSE, ...,
  verbose=FALSE)

```

**Arguments**

|         |                                                                                                                                         |
|---------|-----------------------------------------------------------------------------------------------------------------------------------------|
| Y       | A <b>numeric</b> JxI <b>matrix</b> (or a <b>vector</b> of length J.)                                                                    |
| x       | A (optional) <b>numeric vector</b> specifying the positions of the J entries. The default is to assume uniformly distributed positions. |
| w       | A optional <b>numeric vector</b> of prior weights for each of the J entries.                                                            |
| xOut    | A <b>numeric vector</b> specifying K target positions where the kernel is applied.                                                      |
| kernel  | A <b>character</b> string or a <b>function</b> specifying the kernel used.                                                              |
| h       | A single positive <b>numeric</b> specifying the bandwidth of the kernel.                                                                |
| tensorH | A single positive <b>numeric</b> specifying the where to truncate the kernel. If <b>Inf</b> , no truncation is done.                    |
| na.rm   | If <b>TRUE</b> , missing values are excluded, otherwise not.                                                                            |
| robust  | If <b>TRUE</b> , robust estimators are used, otherwise not.                                                                             |
| ...     | Not used.                                                                                                                               |
| verbose | See <b>Verbose</b> .                                                                                                                    |

**Value**

Returns a **numeric** KxI **matrix** (or a **vector** of length K).

**Author(s)**

Henrik Bengtsson

**See Also**

[\\*colBinnedSmoothing\(\)](#).

**Examples**

```
J <- 100
I <- 4
Y <- matrix(rnorm(I*J, sd=1/2), ncol=I)
# Introduce a gain in column 2 and 3
Y[30:50,2:3] <- Y[30:50,2:3] + 3
x <- 1:J

xOut <- x
Ys1 <- colKernelSmoothing(Y, x=x, xOut=xOut, kernel="gaussian", h=1)
Ys5 <- colKernelSmoothing(Y, x=x, xOut=xOut, kernel="gaussian", h=5)

xlim <- range(c(x,xOut))
ylim <- c(-3,5)
layout(matrix(1:I, ncol=1))
par(mar=c(3,3,1,1)+0.1, pch=19)
for (ii in 1:I) {
  plot(NA, xlim=xlim, ylim=ylim)
  points(x, Y[,ii], col="#999999")
}
```



```

    lines(xOut, Ys1[,ii], col=2)
    points(xOut, Ys1[,ii], col=2)
    lines(xOut, Ys5[,ii], col=3)
    points(xOut, Ys5[,ii], col=3)
}

```

---

CopyNumberChromosomalModel

*The CopyNumberChromosomalModel class*

---

## Description

Package: aroma.core

**Class CopyNumberChromosomalModel**

**Object**

```

~~|
~~+--ChromosomalModel
~~~~~|
~~~~~+--CopyNumberChromosomalModel

```

**Directly known subclasses:**

[CbsModel](#), [CopyNumberSegmentationModel](#), [GladModel](#), [HaarSegModel](#), [RawCopyNumberModel](#)

public abstract static class **CopyNumberChromosomalModel**  
 extends [ChromosomalModel](#)

This *abstract* class represents a copy-number model.

## Usage

```
CopyNumberChromosomalModel(cesTuple=NULL, refTuple=NULL, calculateRatios=TRUE, tags="*",
  genome="Human", chromosomes=NULL, maxNAFraction=1/5, ...)
```

## Arguments

|                 |                                                                                                                                                              |
|-----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------|
| cesTuple        | A <a href="#">CopyNumberDataSetTuple</a> .                                                                                                                   |
| refTuple        | An optional <a href="#">CopyNumberDataFile</a> , or <a href="#">CopyNumberDataSet</a> or <a href="#">CopyNumberDataSetTuple</a> for pairwise comparisons.    |
| calculateRatios | A <a href="#">logical</a> specifying whether ratios should be calculated relative to the reference. If <a href="#">FALSE</a> , argument refTuple is ignored. |
| tags            | A <a href="#">character vector</a> of tags.                                                                                                                  |
| genome          | A <a href="#">character</a> string specifying what genome is process.                                                                                        |
| chromosomes     | (optional) A <a href="#">vector</a> specifying which chromosomes to process.                                                                                 |

maxNAFraction A `double` in [0,1] indicating how many non-finite signals are allowed in the sanity checks of the data.

... Optional arguments that may be used by some of the subclass models.

## Fields and Methods

### Methods:

|                                    |                                 |
|------------------------------------|---------------------------------|
| <code>extractRawCopyNumbers</code> | Extracts relative copy numbers. |
| <code>fit</code>                   | Fits the model.                 |
| <code>getChromosomeLength</code>   | -                               |
| <code>getNames</code>              | -                               |
| <code>getReference</code>          | -                               |
| <code>isPaired</code>              | -                               |

### Methods inherited from ChromosomalModel:

`as.character`, `fit`, `getAlias`, `getAromaGenomeTextFile`, `getAsteriskTags`, `getChipType`, `getChipTypes`, `getChromosomes`, `getFullName`, `getFullNames`, `getGenome`, `getGenomeData`, `getGenomeFile`, `getListOfAromaUgpFiles`, `getName`, `getNames`, `getParentPath`, `getPath`, `getReportPath`, `getRootPath`, `getSetTuple`, `getSets`, `getTags`, `indexOf`, `nbrOfArrays`, `nbrOfChipTypes`, `setChromosomes`, `setGenome`

### 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`, `names`, `objectSize`, `print`, `save`, `asThis`

## Requirements

This class requires genome information annotation files for every chip type.

## Author(s)

Henrik Bengtsson

---

CopyNumberSegmentationModel

*The CopyNumberSegmentationModel class*

---

## Description

Package: aroma.core

**Class CopyNumberSegmentationModel**

`Object`

~~|

```

~+---ChromosomalModel
~~~~~|
~~~~~+---CopyNumberChromosomalModel
~~~~~|
~~~~~+---CopyNumberSegmentationModel

```

**Directly known subclasses:**

[CbsModel](#), [GladModel](#), [HaarSegModel](#)

public abstract static class **CopyNumberSegmentationModel**  
 extends [CopyNumberChromosomalModel](#)

This *abstract* class represents a copy-number segmentation model.

**Usage**

```
CopyNumberSegmentationModel(...)
```

**Arguments**

... Arguments passed to constructor [CopyNumberChromosomalModel](#).

**Fields and Methods****Methods:**

|                              |                 |
|------------------------------|-----------------|
| <a href="#">fit</a>          | Fits the model. |
| <a href="#">getFullNames</a> | -               |
| <a href="#">getRegions</a>   | -               |
| <a href="#">getTags</a>      | -               |
| <a href="#">plot</a>         | -               |
| <a href="#">writeRegions</a> | -               |

**Methods inherited from CopyNumberChromosomalModel:**

[as.character](#), [calculateChromosomeStatistics](#), [calculateRatios](#), [estimateSds](#), [extractRawCopyNumbers](#), [fit](#), [getChromosomeLength](#), [getDataFileMatrix](#), [getMaxNAFraction](#), [getNames](#), [getOptionalArguments](#), [getPairedNames](#), [getRefSetTuple](#), [getReference](#), [getReferenceSetTuple](#), [isPaired](#), [newPlot](#), [plotAxesLayers](#), [plotChromosomesLayers](#), [plotCytobandLayers](#), [plotFitLayers](#), [plotGridHorizontalLayers](#), [plotRawCopyNumbers](#), [plotSampleLayers](#), [setReference](#)

**Methods inherited from ChromosomalModel:**

[as.character](#), [fit](#), [getAlias](#), [getAromaGenomeTextFile](#), [getAsteriskTags](#), [getChipType](#), [getChipTypes](#), [getChromosomes](#), [getFullName](#), [getFullNames](#), [getGenome](#), [getGenomeData](#), [getGenomeFile](#), [getListOfAromaUgpFiles](#), [getName](#), [getNames](#), [getParentPath](#), [getPath](#), [getReportPath](#), [getRootPath](#), [getSetTuple](#), [getSets](#), [getTags](#), [indexOf](#), [nbrOfArrays](#), [nbrOfChipTypes](#), [setChromosomes](#), [setGenome](#)

**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, names, objectSize, print, save, asThis

### Author(s)

Henrik Bengtsson

---

doCBS

*Performs Circular Binary Segmentation (CBS) on a data set*

---

### Description

Performs Circular Binary Segmentation (CBS) on a data set for one or more chip types.

### Usage

```
## Default S3 method:
doCBS(dataSet, tags=NULL, chipTypes, arrays=NULL, ..., verbose=FALSE)
## S3 method for class 'CopyNumberDataSet'
doCBS(ds, arrays=NULL, ...)
## S3 method for class 'CopyNumberDataSetTuple'
doCBS(dsTuple, arrays=NULL, ..., verbose=FALSE)
```

### Arguments

|                      |                                                                                                                                                    |
|----------------------|----------------------------------------------------------------------------------------------------------------------------------------------------|
| ds, dsTuple, dataSet | A <a href="#">CopyNumberDataSet</a> , a <a href="#">CopyNumberDataSetTuple</a> or a <a href="#">character</a> string with the name of one of them. |
| tags                 | An optional <a href="#">character vector</a> of data set tags (only when <code>dataSet</code> is specified).                                       |
| chipTypes            | A <a href="#">character vector</a> specifying the chip types for the different data sets (only when <code>dataSet</code> is specified).            |
| arrays               | An optional <a href="#">vector</a> specifying the subset of arrays to process.                                                                     |
| ...                  | Additional arguments passed to <a href="#">CbsModel</a> and its <code>fit()</code> method.                                                         |
| verbose              | A <a href="#">logical</a> or <a href="#">Verbose</a> .                                                                                             |

### Value

Returns the output dataset of [CbsModel](#).

### Author(s)

Henrik Bengtsson

**Examples**

```
## Not run:

dataSet <- "GSE20939"
tags <- "ACC,-XY,BPN,-XY,RMA,FLN,-XY"
chipTypes <- c("Mapping250K_Nsp", "Mapping250K_Sty")
fit <- doCBS(dataSet, tags=tags, chipTypes=chipTypes, verbose=-10)
print(fit)

## End(Not run)
```

---

Explorer

*The Explorer class*

---

**Description**

Package: aroma.core

**Class Explorer****Object**

~~|

~~+--Explorer

**Directly known subclasses:**

[ChromosomeExplorer](#)

public abstract static class **Explorer**  
extends [Object](#)

**Usage**

```
Explorer(tags="*", version="0", ...)
```

**Arguments**

|         |                                                                            |
|---------|----------------------------------------------------------------------------|
| tags    | A <a href="#">character vector</a> of tags to be added to the output path. |
| version | An optional <a href="#">character</a> string.                              |
| ...     | Not used.                                                                  |

**Fields and Methods****Methods:**

|                          |                                                                    |
|--------------------------|--------------------------------------------------------------------|
| <code>display</code>     | Displays the explorer in the default browser.                      |
| <code>getFullName</code> | -                                                                  |
| <code>getName</code>     | Gets the name of the explorer.                                     |
| <code>getNames</code>    | Gets the names of the input samples.                               |
| <code>getPath</code>     | Gets the path of the output directory.                             |
| <code>getRootPath</code> | Gets the root path of the output directory.                        |
| <code>getTags</code>     | Gets the tags of the explorer.                                     |
| <code>getVersion</code>  | -                                                                  |
| <code>nbrOfArrays</code> | Gets the total number of arrays.                                   |
| <code>process</code>     | Generates image files, scripts and dynamic pages for the explorer. |
| <code>setArrays</code>   | Sets the arrays.                                                   |

#### 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, names, objectSize, print, save, asThis

#### Output directory structure

The *main directory* of an Explorer report is `reports/<name>/<subname>/`. The `<name>` is typically the same as the name of the input data set, and the `<subname>` is typically the tags of ditto. This main directory is where main HTML document is stored.

For each chip type, real or "virtual" (combined), there is a subdirectory with the same name as the chip type, i.e. `reports/<name>/<subname>/<chiptype>/`.

For each chip type directory, there are set of subdirectories each specifying a so called *image layer*, e.g. an image layer showing the raw data, another containing the estimates of a model fit and so on. Path format: `reports/<name>/<subname>/<chiptype>/<image layer>/`. In this directory all image files are stored, e.g. PNG files.

In some cases one do not want to all input tags to become part of the subname, but instead for instance use those to name the image layer(s). In such cases one has to override the default names.

#### Author(s)

Henrik Bengtsson

---

GladModel

*The GladModel class*

---

#### Description

Package: aroma.core

**Class GladModel**

`Object`

~~|

```

~+---ChromosomalModel
~~~~~|
~~~~~+---CopyNumberChromosomalModel
~~~~~|
~~~~~+---CopyNumberSegmentationModel
~~~~~|
~~~~~+---GladModel

```

### Directly known subclasses:

```

public static class GladModel
extends CopyNumberSegmentationModel

```

This class represents the Gain and Loss Analysis of DNA regions (GLAD) model [1]. This class can model chip-effect estimates obtained from multiple chip types, and not all samples have to be available on all chip types.

### Usage

```
GladModel(cesTuple=NULL, ...)
```

### Arguments

```

cesTuple      A CopyNumberDataSetTuple.
...           Arguments passed to the constructor of CopyNumberSegmentationModel.

```

### Details

Data from multiple chip types are combined "as is". This is based on the assumption that the relative chip effect estimates are non-biased (or at the equally biased across chip types). Note that in GLAD there is no way to down weight certain data points, which is why we can control for differences in variance across chip types.

### Fields and Methods

#### Methods:

```
writeRegions -
```

#### Methods inherited from *CopyNumberSegmentationModel*:

```
fit, getAsteriskTags, getFitFunction, getFullNames, getRegions, getTags, plot, plotCopyNumber-RegionLayers, writeRegions
```

**Methods inherited from CopyNumberChromosomalModel:**

as.character, calculateChromosomeStatistics, calculateRatios, estimateSds, extractRawCopyNumbers, fit, getChromosomeLength, getDataFileMatrix, getMaxNAFraction, getNames, getOptionalArguments, getPairedNames, getRefSetTuple, getReference, getReferenceSetTuple, isPaired, newPlot, plotAxesLayers, plotChromosomesLayers, plotCytobandLayers, plotFitLayers, plotGridHorizontalLayers, plotRawCopyNumbers, plotSampleLayers, setReference

**Methods inherited from ChromosomalModel:**

as.character, fit, getAlias, getAromaGenomeTextFile, getAsteriskTags, getChipType, getChipTypes, getChromosomes, getFullName, getFullNames, getGenome, getGenomeData, getGenomeFile, getListOfAromaUgpFiles, getName, getNames, getParentPath, getPath, getReportPath, getRootPath, getSetTuple, getSets, getTags, indexOf, nbrOfArrays, nbrOfChipTypes, setChromosomes, setGenome

**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, names, objectSize, print, save, asThis

**Benchmarking**

In high-density copy numbers analysis, the most time consuming step is fitting the GLAD model. The complexity of the model grows more than linearly (squared? exponentially?) with the number of data points in the chromosome and sample being fitted. This is why it take much more than twice the time to fit two chip types together than separately.

**Author(s)**

Henrik Bengtsson

**References**

[1] Hupe P et al. *Analysis of array CGH data: from signal ratio to gain and loss of DNA regions*. Bioinformatics, 2004, 20, 3413-3422.

**See Also**

[CopyNumberSegmentationModel](#).

---

HaarSegModel

*The HaarSegModel class*

---

**Description**

Package: aroma.core

**Class HaarSegModel**

[Object](#)

~~|



```

~~+---ChromosomalModel
~~~~~|
~~~~~+---CopyNumberChromosomalModel
~~~~~|
~~~~~+---CopyNumberSegmentationModel
~~~~~|
~~~~~+---HaarSegModel

```

### Directly known subclasses:

```

public static class HaarSegModel
extends CopyNumberSegmentationModel

```

This class represents the Haar wavelet-based segmentation (HaarSeg) model [1].

### Usage

```
HaarSegModel(cesTuple=NULL, ..., breaksFdrQ=1e-04)
```

### Arguments

|            |                                                                             |
|------------|-----------------------------------------------------------------------------|
| cesTuple   | A <i>CopyNumberDataSetTuple</i> .                                           |
| breaksFdrQ | Default tuning parameters specific to the HaarSeg algorithm.                |
| ...        | Arguments passed to the constructor of <i>CopyNumberSegmentationModel</i> . |

### Fields and Methods

#### Methods:

*No methods defined.*

#### Methods inherited from *CopyNumberSegmentationModel*:

fit, getAsteriskTags, getFitFunction, getFullNames, getRegions, getTags, plot, plotCopyNumberRegionLayers, writeRegions

#### Methods inherited from *CopyNumberChromosomalModel*:

as.character, calculateChromosomeStatistics, calculateRatios, estimateSds, extractRawCopyNumbers, fit, getChromosomeLength, getDataFileMatrix, getMaxNAFraction, getNames, getOptionalArguments, getPairedNames, getRefSetTuple, getReference, getReferenceSetTuple, isPaired, newPlot, plotAxesLayers, plotChromosomesLayers, plotCytobandLayers, plotFitLayers, plotGridHorizontalLayers, plotRawCopyNumbers, plotSampleLayers, setReference

#### Methods inherited from *ChromosomalModel*:

as.character, fit, getAlias, getAromaGenomeTextFile, getAsteriskTags, getChipType, getChipTypes, getChromosomes, getFullName, getFullNames, getGenome, getGenomeData, getGenomeFile, getListOfAromaUgpFiles, getName, getNames, getParentPath, getPath, getReportPath, getRootPath, getSetTuple, getSets, getTags, indexOf, nbrOfArrays, nbrOfChipTypes, setChromosomes, setGenome

#### 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, names, objectSize, print, save, asThis

### Author(s)

Henrik Bengtsson

### References

[1] Ben-Yaacov E. and Eldar YC. *A fast and flexible method for the segmentation of aCGH data*, Bioinformatics, 2008. <https://www.weizmann.ac.il/math/yonina/software-hardware/software/haarseg-fast-and-flexible-microarray-segmentation>

### See Also

[CopyNumberSegmentationModel](#).

---

NonPairedPSCNData      *The NonPairedPSCNData class*

---

### Description

Package: aroma.core

#### Class NonPairedPSCNData

```
data.frame
~~|
~~+--RichDataFrame
~~~~~|
~~~~~+--RawGenomicSignals
~~~~~|
~~~~~+--AbstractCNData
~~~~~|
~~~~~+--AbstractPSCNData
~~~~~|
~~~~~+--NonPairedPSCNData
```

#### Directly known subclasses:

```
public class NonPairedPSCNData
extends AbstractPSCNData
```

A NonPairedPSCNData object holds parent-specific copy number data. Two NonPairedPSCNData objects for a matched tumor-normal pair can be combined into a PairedPSCNData object.

**Usage**

```
NonPairedPSCNData(chromosome=NULL, x=NULL, isSNP=NULL, mu=NULL, C=NULL, beta=NULL, ...)
```

**Arguments**

|            |                                                                                                                                                                                                                                                                     |
|------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| C          | A <b>numeric vector</b> of J tumor total copy number (TCN) ratios in $[0, +\text{Inf})$ (due to noise, small negative values are also allowed). The TCN ratios are typically scaled such that copy-neutral diploid loci have a mean of two.                         |
| beta       | A <b>numeric vector</b> of J tumor allele B fractions (BAFs) in $[0, 1]$ (due to noise, values may be slightly outside as well) or <b>NA</b> for non-polymorphic loci.                                                                                              |
| mu         | An optional <b>numeric vector</b> of J genotype calls in $\{0, 1/2, 1\}$ for AA, AB, and BB, respectively, and <b>NA</b> for non-polymorphic loci. If not given, they are estimated from the normal BAFs using <code>callNaiveGenotypes</code> as described in [2]. |
| isSNP      | An optional <b>logical vector</b> of length J specifying whether each locus is a SNP or not (non-polymorphic loci).                                                                                                                                                 |
| chromosome | (Optional) An <b>integer</b> scalar (or a <b>vector</b> of length J), which can be used to specify which chromosome each locus belongs to in case multiple chromosomes are segments. This argument is also used for annotation purposes.                            |
| x          | Optional <b>numeric vector</b> of J genomic locations. If <b>NULL</b> , index locations 1:J are used.                                                                                                                                                               |
| ...        | Optional named locus-specific signal <b>vectors</b> of length J.                                                                                                                                                                                                    |

**Fields and Methods****Methods:**

```
as -
as.NonPairedPSCNData -
as.PairedPSCNData -
callNaiveGenotypes -
callSegmentationOutliers -
dropSegmentationOutliers -
getSignalColumnNames -
segmentByCBS -
```

**Methods inherited from AbstractPSCNData:**

```
callSNPs, getSNPFields
```

**Methods inherited from AbstractCNData:**

```
findLargeGaps, getChipType, getLocusData, getPlatform, hasKnownPositions, orderAlongGenome,
setChipType, setPlatform
```

**Methods inherited from RawGenomicSignals:**

```
*, +, -, addBy, append, applyBinaryOperator, as.character, as.data.frame, assertOneChromosome,
```

binnedSmoothing, binnedSmoothingByField, clearCache, clone, divideBy, drawDensity, estimateStandardDeviation, extractChromosome, extractChromosomes, extractDataForSegmentation, extractRegion, extractRegions, extractSubset, gaussianSmoothing, getBasicField, getCXY, getChromosome, getChromosomes, getDefaultLocusFields, getLocusFields, getPositions, getSigma, getSignalColumnName, getSignalColumnNames, getSignals, getWeights, getXScale, getXY, getYScale, hasWeights, kernelSmoothing, lines, multiplyBy, nbrOfChromosomes, nbrOfLoci, plot, points, print, segmentByCBS, segmentByGLAD, segmentByHaarSeg, segmentByMPCBS, setBasicField, setSigma, setSignals, setWeights, setXScale, setYScale, signalRange, sort, subtractBy, xMax, xMin, xRange, xSeq, yMax, yMin, yRange

**Methods inherited from RichDataFrame:**

\$, \$<-, [, [[, [[<-, as.data.frame, as.list, dim, dropVirtualColumn, getColumnNames, getColumnNamesTranslator, getFullName, getName, getTags, getVirtualColumn, getVirtualColumnFunction, getVirtualColumnNames, hasColumn, hasColumns, hasVirtualColumn, hasVirtualColumns, length, names, newInstance, print, rbind, setAttributes, setColumnNamesMap, setColumnNamesTranslator, setName, setTags, setVirtualColumn, subset, translateColumnNames

**Methods inherited from data.frame:**

\$<-, data.frame-method, \$<-, Math, Ops,nonStructure,vector-method, Ops,structure,vector-method, Ops,vector,nonStructure-method, Ops,vector,structure-method, Ops, Summary, [, [<-, data.frame-method, [<-, [[, [[<-, data.frame-method, [[<-, aggregate, anyDuplicated, anyNA, as.NonPairedPSCNData, as.PairedPSCNData, as.data.frame, as.list, as.matrix, as.vector, attachLocally, by, callSegmentationOutliers, cbind, coerce, ANY, list-method, coerce, oldClass, S3-method, dim, dimnames, dimnames<-, dropSegmentationOutliers, droplevels, duplicated, edit, findLargeGaps, format, formula, head, initialize, oldClass-method, is.na, merge, na.exclude, na.omit, plot, plotDensity, print, prompt, rbind, row.names, row.names<-, rowsum, segmentByCBS, segmentByPairedPSCBS, show, oldClass-method, slotsFromS3, data.frame-method, sort\_by, split, split<-, stack, str, subset, summary, t, tail, transform, type.convert, unique, unstack, unwrap, within, wrap, writeDataFrame, xfrm

**Author(s)**

Henrik Bengtsson

---

PairedPSCNData

*The PairedPSCNData class*

---

**Description**

Package: aroma.core

**Class PairedPSCNData**

```
data.frame
~~|
~~+--RichDataFrame
~~~~~|
~~~~~+--RawGenomicSignals
~~~~~|
~~~~~+--AbstractCNData
```

```

~~~~~|
~~~~~+---AbstractPSCNData
~~~~~|
~~~~~+---PairedPSCNData

```

### Directly known subclasses:

```

public class PairedPSCNData
  extends AbstractPSCNData

```

A PairedPSCNData object holds paired tumor-normal parent-specific copy number data.

### Usage

```

PairedPSCNData(chromosome=NULL, x=NULL, isSNP=NULL, muN=NULL, CT=NULL, betaT=NULL,
  CN=NULL, betaN=NULL, ...)

```

### Arguments

|            |                                                                                                                                                                                                                                                                                          |
|------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| CT         | A <a href="#">numeric vector</a> of J tumor total copy number (TCN) ratios in $[0, +\text{Inf})$ (due to noise, small negative values are also allowed). The TCN ratios are typically scaled such that copy-neutral diploid loci have a mean of two.                                     |
| CN         | An optional <a href="#">numeric vector</a> of J normal TCN ratios.                                                                                                                                                                                                                       |
| betaT      | A <a href="#">numeric vector</a> of J tumor allele B fractions (BAFs) in $[0,1]$ (due to noise, values may be slightly outside as well) or <a href="#">NA</a> for non-polymorphic loci.                                                                                                  |
| betaN      | A <a href="#">numeric vector</a> of J matched normal BAFs in $[0,1]$ (due to noise, values may be slightly outside as well) or <a href="#">NA</a> for non-polymorphic loci.                                                                                                              |
| muN        | An optional <a href="#">numeric vector</a> of J genotype calls in $\{0, 1/2, 1\}$ for AA, AB, and BB, respectively, and <a href="#">NA</a> for non-polymorphic loci. If not given, they are estimated from the normal BAFs using <a href="#">callNaiveGenotypes</a> as described in [2]. |
| isSNP      | An optional <a href="#">logical vector</a> of length J specifying whether each locus is a SNP or not (non-polymorphic loci).                                                                                                                                                             |
| chromosome | (Optional) An <a href="#">integer</a> scalar (or a <a href="#">vector</a> of length J), which can be used to specify which chromosome each locus belongs to in case multiple chromosomes are segments. This argument is also used for annotation purposes.                               |
| x          | Optional <a href="#">numeric vector</a> of J genomic locations. If <a href="#">NULL</a> , index locations 1 : J are used.                                                                                                                                                                |
| ...        | Optional named locus-specific signal <a href="#">vectors</a> of length J.                                                                                                                                                                                                                |

### Fields and Methods

#### Methods:

```

as -
as.PairedPSCNData -
callNaiveGenotypes -
callSegmentationOutliers -
dropSegmentationOutliers -
getSignalColumnNames -
getTCNs -
getTotalCopyNumbers -
normalizeTumorBoost -
plotTracks -
segmentByCBS -
segmentByPairedPSCBS -

```

#### Methods inherited from AbstractPSCNData:

callSNPs, getSNPFields

#### Methods inherited from AbstractCNData:

findLargeGaps, getChipType, getLocusData, getPlatform, hasKnownPositions, orderAlongGenome, setChipType, setPlatform

#### Methods inherited from RawGenomicSignals:

\*, +, -, addBy, append, applyBinaryOperator, as.character, as.data.frame, assertOneChromosome, binnedSmoothing, binnedSmoothingByField, clearCache, clone, divideBy, drawDensity, estimateStandardDeviation, extractChromosome, extractChromosomes, extractDataForSegmentation, extractRegion, extractRegions, extractSubset, gaussianSmoothing, getBasicField, getCXY, getChromosome, getChromosomes, getDefaultLocusFields, getLocusFields, getPositions, getSigma, getSignalColumnName, getSignalColumnNames, getSignals, getWeights, getXScale, getXY, getYScale, hasWeights, kernelSmoothing, lines, multiplyBy, nbrOfChromosomes, nbrOfLoci, plot, points, print, segmentByCBS, segmentByGLAD, segmentByHaarSeg, segmentByMPCBS, setBasicField, setSigma, setSignals, setWeights, setXScale, setYScale, signalRange, sort, subtractBy, xMax, xMin, xRange, xSeq, yMax, yMin, yRange

#### Methods inherited from RichDataFrame:

\$/, \$<-, [, [[, [[<-, as.data.frame, as.list, dim, dropVirtualColumn, getColumnNames, getColumnNamesTranslator, getFullName, getName, getTags, getVirtualColumn, getVirtualColumnFunction, getVirtualColumnNames, hasColumn, hasColumns, hasVirtualColumn, hasVirtualColumns, length, names, newInstance, print, rbind, setAttributes, setColumnNamesMap, setColumnNamesTranslator, setName, setTags, setVirtualColumn, subset, translateColumnNames

#### Methods inherited from data.frame:

\$<- ,data.frame-method, \$<- , Math, Ops,nonStructure,vector-method, Ops,structure,vector-method, Ops,vector,nonStructure-method, Ops,vector,structure-method, Ops, Summary, [, [<- ,data.frame-method, [<- , [[, [[<- ,data.frame-method, [[<- , aggregate, anyDuplicated, anyNA, as.NonPairedPSCNData, as.PairedPSCNData, as.data.frame, as.list, as.matrix, as.vector, attachLocally, by, callSegmentationOutliers, cbind, coerce,ANY,list-method, coerce,oldClass,S3-method, dim, dimnames, dimnames<- , dropSegmentationOutliers, droplevels, duplicated, edit, findLargeGaps, format, formula, head, initialize,oldClass-method, is.na, merge, na.exclude, na.omit, plot, plotDensity, print, prompt, rbind, row.names, row.names<- , rowsum, segmentByCBS, segmentByPairedPSCBS, show,oldClass-method, slotsFromS3,data.frame-method, sort\_by, split, split<- , stack, str, subset, summary, t, tail, transform, type.convert, unique, unstack, unwrap, within, wrap, writeDataFrame, xfrm

**Author(s)**

Henrik Bengtsson

---

ParametersInterface    *The ParametersInterface class interface*

---

**Description**

Package: aroma.core

**Class ParametersInterface**[Interface](#)

~~|

~~+--ParametersInterface

**Directly known subclasses:**[AromaTransform](#)

```
public class ParametersInterface
  extends Interface
```

**Usage**

```
ParametersInterface(...)
```

**Arguments**

```
...                    Not used.
```

**Fields and Methods****Methods:**

```
getParameters            -
getParametersAsString   -
```

**Methods inherited from Interface:**

extend, print, uses

**Author(s)**

Henrik Bengtsson

---

RawAlleleBFractions    *The RawAlleleBFractions class*

---

## Description

Package: aroma.core

### Class RawAlleleBFractions

```
data.frame
~~|
~~+--RichDataFrame
~~~~~|
~~~~~+--RawGenomicSignals
~~~~~|
~~~~~+--RawAlleleBFractions
```

### Directly known subclasses:

[RawMirroredAlleleBFractions](#), [SegmentedAlleleBFractions](#)

```
public class RawAlleleBFractions
extends RawGenomicSignals
```

## Usage

```
RawAlleleBFractions(...)
```

## Arguments

...                    Arguments passed to [RawGenomicSignals](#).

## Fields and Methods

### Methods:

```
extractRawMirroredAlleleBFractions -
plot -
```

### Methods inherited from RawGenomicSignals:

\*, +, -, addBy, append, applyBinaryOperator, as.character, as.data.frame, assertOneChromosome, binnedSmoothing, binnedSmoothingByField, clearCache, clone, divideBy, drawDensity, estimateStandardDeviation, extractChromosome, extractChromosomes, extractDataForSegmentation, extractRegion, extractRegions, extractSubset, gaussianSmoothing, getBasicField, getCXY, getChromosome, getChromosomes, getDefaultLocusFields, getLocusFields, getPositions, getSigma, getSignalColumnName, getSignalColumnNames, getSignals, getWeights, getXScale, getXY, getYScale,



hasWeights, kernelSmoothing, lines, multiplyBy, nbrOfChromosomes, nbrOfLoci, plot, points, print, segmentByCBS, segmentByGLAD, segmentByHaarSeg, segmentByMPCBS, setBasicField, setSigma, setSignals, setWeights, setXScale, setYScale, signalRange, sort, subtractBy, xMax, xMin, xRange, xSeq, yMax, yMin, yRange

**Methods inherited from RichDataFrame:**

\$, \$<-, [, [[, [[<-, as.data.frame, as.list, dim, dropVirtualColumn, getColumnNames, getColumnNamesTranslator, getFullName, getName, getTags, getVirtualColumn, getVirtualColumnFunction, getVirtualColumnNames, hasColumn, hasColumns, hasVirtualColumn, hasVirtualColumns, length, names, newInstance, print, rbind, setAttributes, setColumnNamesMap, setColumnNamesTranslator, setName, setTags, setVirtualColumn, subset, translateColumnNames

**Methods inherited from data.frame:**

\$<- ,data.frame-method, \$<-, Math, Ops,nonStructure,vector-method, Ops,structure,vector-method, Ops,vector,nonStructure-method, Ops,vector,structure-method, Ops, Summary, [, [<- ,data.frame-method, [<-, [[, [[<- ,data.frame-method, [[<-, aggregate, anyDuplicated, anyNA, as.NonPairedPSCNData, as.PairedPSCNData, as.data.frame, as.list, as.matrix, as.vector, attachLocally, by, callSegmentationOutliers, cbind, coerce,ANY,list-method, coerce,oldClass,S3-method, dim, dimnames, dimnames<- , dropSegmentationOutliers, droplevels, duplicated, edit, findLargeGaps, format, formula, head, initialize,oldClass-method, is.na, merge, na.exclude, na.omit, plot, plotDensity, print, prompt, rbind, row.names, row.names<-, rowsum, segmentByCBS, segmentByPairedPSCBS, show,oldClass-method, slotsFromS3,data.frame-method, sort\_by, split, split<-, stack, str, subset, summary, t, tail, transform, type.convert, unique, unstack, unwrap, within, wrap, writeDataFrame, xfrm

**Author(s)**

Henrik Bengtsson

---

RawCopyNumberModel      *The RawCopyNumberModel class*

---

**Description**

Package: aroma.core

**Class RawCopyNumberModel**

Object

```

~|
~+--ChromosomalModel
~~~~~|
~~~~~+--CopyNumberChromosomalModel
~~~~~|
~~~~~+--RawCopyNumberModel

```

**Directly known subclasses:**

public abstract static class **RawCopyNumberModel**  
 extends *CopyNumberChromosomalModel*

This class represents an identity copy-number model which returns the input as is.

### Usage

```
RawCopyNumberModel(...)
```

### Arguments

... Passed to the constructor of the superclass.

### Fields and Methods

#### Methods:

*No methods defined.*

#### Methods inherited from CopyNumberChromosomalModel:

as.character, calculateChromosomeStatistics, calculateRatios, estimateSds, extractRawCopyNumbers, fit, getChromosomeLength, getDataFileMatrix, getMaxNAFraction, getNames, getOptionalArguments, getPairedNames, getRefSetTuple, getReference, getReferenceSetTuple, isPaired, newPlot, plotAxesLayers, plotChromosomesLayers, plotCytobandLayers, plotFitLayers, plotGridHorizontalLayers, plotRawCopyNumbers, plotSampleLayers, setReference

#### Methods inherited from ChromosomalModel:

as.character, fit, getAlias, getAromaGenomeTextFile, getAsteriskTags, getChipType, getChipTypes, getChromosomes, getFullName, getFullNames, getGenome, getGenomeData, getGenomeFile, getListOfAromaUgpFiles, getName, getNames, getParentPath, getPath, getReportPath, getRootPath, getSetTuple, getSets, getTags, indexOf, nbrOfArrays, nbrOfChipTypes, setChromosomes, setGenome

#### 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, names, objectSize, print, save, asThis

### Author(s)

Henrik Bengtsson

**Description**

Package: aroma.core

**Class RawCopyNumbers**

```
data.frame
~~|
~~+--RichDataFrame
~~~~~|
~~~~~+--RawGenomicSignals
~~~~~|
~~~~~+--RawCopyNumbers
```

**Directly known subclasses:**

[SegmentedCopyNumbers](#)

```
public class RawCopyNumbers
extends RawGenomicSignals
```

**Usage**

```
RawCopyNumbers(cn=NULL, ...)
```

**Arguments**

cn                   A [numeric vector](#) of length J specifying the copy number at each loci.  
...                   Arguments passed to [RawGenomicSignals](#).

**Fields and Methods****Methods:**

```
cnRange                   -
extractRawCopyNumbers   -
getSignals               -
plot                      -
```

**Methods inherited from RawGenomicSignals:**

\*, +, -, addBy, append, applyBinaryOperator, as.character, as.data.frame, assertOneChromosome, binnedSmoothing, binnedSmoothingByField, clearCache, clone, divideBy, drawDensity, estimateStandardDeviation, extractChromosome, extractChromosomes, extractDataForSegmentation, extractRegion, extractRegions, extractSubset, gaussianSmoothing, getBasicField, getCXY, getChromosome, getChromosomes, getDefaultLocusFields, getLocusFields, getPositions, getSigma, getSignalColumnName, getSignalColumnNames, getSignals, getWeights, getXScale, getXY, getYScale, hasWeights, kernelSmoothing, lines, multiplyBy, nbrOfChromosomes, nbrOfLoci, plot, points,

print, segmentByCBS, segmentByGLAD, segmentByHaarSeg, segmentByMPCBS, setBasicField, setSigma, setSignals, setWeights, setXScale, setYScale, signalRange, sort, subtractBy, xMax, xMin, xRange, xSeq, yMax, yMin, yRange

**Methods inherited from RichDataFrame:**

\$, \$<-, [, [[, [[<-, as.data.frame, as.list, dim, dropVirtualColumn, getColumnNames, getColumnNamesTranslator, getFullName, getName, getTags, getVirtualColumn, getVirtualColumnFunction, getVirtualColumnNames, hasColumn, hasColumns, hasVirtualColumn, hasVirtualColumns, length, names, newInstance, print, rbind, setAttributes, setColumnNamesMap, setColumnNamesTranslator, setName, setTags, setVirtualColumn, subset, translateColumnNames

**Methods inherited from data.frame:**

\$<-,data.frame-method, \$<-, Math, Ops,nonStructure,vector-method, Ops,structure,vector-method, Ops,vector,nonStructure-method, Ops,vector,structure-method, Ops, Summary, [, [<-,data.frame-method, [<-, [[, [[<-,data.frame-method, [[<-, aggregate, anyDuplicated, anyNA, as.NonPairedPSCNData, as.PairedPSCNData, as.data.frame, as.list, as.matrix, as.vector, attachLocally, by, callSegmentationOutliers, cbind, coerce,ANY,list-method, coerce,oldClass,S3-method, dim, dimnames, dimnames<-, dropSegmentationOutliers, droplevels, duplicated, edit, findLargeGaps, format, formula, head, initialize,oldClass-method, is.na, merge, na.exclude, na.omit, plot, plotDensity, print, prompt, rbind, row.names, row.names<-, rowsum, segmentByCBS, segmentByPairedPSCBS, show,oldClass-method, slotsFromS3,data.frame-method, sort\_by, split, split<-, stack, str, subset, summary, t, tail, transform, type.convert, unique, unstack, unwrap, within, wrap, writeDataFrame, xfrm

**Author(s)**

Henrik Bengtsson

**Examples**

```
# -----
# Simulating copy-number data
# -----
# Number of loci
J <- 1000

mu <- double(J)
mu[200:300] <- mu[200:300] + 1
mu[650:800] <- mu[650:800] - 1
eps <- rnorm(J, sd=1/2)
y <- mu + eps
x <- sort(runif(length(y), max=length(y)))

cn <- RawCopyNumbers(y, x)
print(cn)

cn2 <- extractSubset(cn, subset=xSeq(cn, by=5))
print(cn2)

# -----
# Plot along genome
# -----
```

```

plot(cn, ylim=c(-3,3))
title(main="Complete and subsetting loci")
points(cn2, col="red", pch=176, cex=2)

legend("topright", pch=c(19,176), col=c("#999999", "red"),
      sprintf(c("raw [n=%d]", "every 5th [n=%d]"),
            c(nbrOfLoci(cn), nbrOfLoci(cn2))), bty="n")

# -----
# Binned smoothing
# -----
plot(cn, col="#999999", ylim=c(-3,3))
title(main="Binned smoothing")

cnSa <- binnedSmoothing(cn, by=3)
lines(cnSa, col="blue")
points(cnSa, col="blue")

cnSb <- binnedSmoothing(cn, by=9)
lines(cnSb, col="red")
points(cnSb, col="red")

legend("topright", pch=19, col=c("#999999", "blue", "red"),
      sprintf(c("raw [n=%d]", "Bin(w=3) [n=%d]", "Bin(w=9) [n=%d]"),
            c(nbrOfLoci(cn), nbrOfLoci(cnSa), nbrOfLoci(cnSb))), bty="n")

# -----
# Binned smoothing (by count)
# -----
plot(cn, col="#999999", ylim=c(-3,3))
title(main="Binned smoothing (by count)")

cnSa <- binnedSmoothing(cn, by=3, byCount=TRUE)
lines(cnSa, col="blue")
points(cnSa, col="blue")

cnSb <- binnedSmoothing(cn, by=9, byCount=TRUE)
lines(cnSb, col="red")
points(cnSb, col="red")

legend("topright", pch=19, col=c("#999999", "blue", "red"),
      sprintf(c("raw [n=%d]", "Bin0(w=3) [n=%d]", "Bin0(w=9) [n=%d]"),
            c(nbrOfLoci(cn), nbrOfLoci(cnSa), nbrOfLoci(cnSb))), bty="n")

# -----
# Kernel smoothing (default is Gaussian)
# -----
plot(cn, col="#999999", ylim=c(-3,3))
title(main="Kernel smoothing w/ Gaussian kernel")

```

```

cnSa <- kernelSmoothing(cn, h=2)
points(cnSa, col="blue")

cnSb <- kernelSmoothing(cn, h=5)
points(cnSb, col="red")

legend("topright", pch=19, col=c("#999999", "blue", "red"),
      sprintf(c("raw [n=%d]", "N(.,sd=2) [n=%d]", "N(.,sd=5) [n=%d]"),
            c(nbrOfLoci(cn), nbrOfLoci(cnSa), nbrOfLoci(cnSb))), bty="n")

# - - - - -
# Kernel smoothing
# - - - - -
plot(cn, col="#999999", ylim=c(-3,3))
title(main="Kernel smoothing w/ uniform kernel")

xOut <- xSeq(cn, by=10)
cnSa <- kernelSmoothing(cn, xOut=xOut, kernel="uniform", h=2)
lines(cnSa, col="blue")
points(cnSa, col="blue")

cnSb <- kernelSmoothing(cn, xOut=xOut, kernel="uniform", h=5)
lines(cnSb, col="red")
points(cnSb, col="red")

legend("topright", pch=19, col=c("#999999", "blue", "red"),
      sprintf(c("raw [n=%d]", "U(w=2) [n=%d]", "U(w=5) [n=%d]"),
            c(nbrOfLoci(cn), nbrOfLoci(cnSa), nbrOfLoci(cnSb))), bty="n")

```

---

RawGenomicSignals      *The RawGenomicSignals class*

---

## Description

Package: aroma.core

### Class RawGenomicSignals

```

data.frame
~~|
~~+--RichDataFrame
~~~~~|
~~~~~+--RawGenomicSignals

```

#### Directly known subclasses:

[AbstractCNData](#), [AbstractPSCNData](#), [NonPairedPSCNData](#), [PairedPSCNData](#), [RawAlleleBFractions](#), [RawCopyNumbers](#), [RawMirroredAlleleBFractions](#), [RawSequenceReads](#), [SegmentedAlleleBFractions](#), [SegmentedCopyNumbers](#)

```
public class RawGenomicSignals
  extends RichDataFrame
```

### Usage

```
RawGenomicSignals(y=NULL, x=NULL, w=NULL, chromosome=0L, name=NULL, ...)
```

### Arguments

|            |                                                                                                         |
|------------|---------------------------------------------------------------------------------------------------------|
| y          | A <a href="#">numeric vector</a> of length J specifying the signal at each locus.                       |
| x          | A (optional) <a href="#">numeric vector</a> of length J specifying the position of each locus.          |
| w          | A (optional) non-negative <a href="#">numeric vector</a> of length J specifying a weight of each locus. |
| chromosome | An (optional) <a href="#">integer</a> specifying the chromosome for these genomic signals.              |
| name       | An (optional) <a href="#">character</a> string specifying the sample name.                              |
| ...        | Not used.                                                                                               |

### Fields and Methods

#### Methods:

|                           |   |
|---------------------------|---|
| *                         | - |
| +                         | - |
| -                         | - |
| addBy                     | - |
| append                    | - |
| as.data.frame             | - |
| binnedSmoothing           | - |
| divideBy                  | - |
| drawDensity               | - |
| estimateStandardDeviation | - |
| gaussianSmoothing         | - |
| getChromosome             | - |
| getChromosomes            | - |
| getPositions              | - |
| getSigma                  | - |
| getSignals                | - |
| getWeights                | - |
| getXScale                 | - |
| getYScale                 | - |
| hasWeights                | - |
| kernelSmoothing           | - |
| lines                     | - |
| multiplyBy                | - |
| nrOfChromosomes           | - |
| nrOfLoci                  | - |

|                  |   |
|------------------|---|
| plot             | - |
| points           | - |
| segmentByCBS     | - |
| segmentByGLAD    | - |
| segmentByHaarSeg | - |
| segmentByMPCBS   | - |
| setSigma         | - |
| setSignals       | - |
| setWeights       | - |
| setXScale        | - |
| setYScale        | - |
| signalRange      | - |
| sort             | - |
| subtractBy       | - |
| xMax             | - |
| xMin             | - |
| xRange           | - |
| xSeq             | - |
| yMax             | - |
| yMin             | - |
| yRange           | - |

**Methods inherited from RichDataFrame:**

`$`, `$<-`, `[`, `[[`, `[[<-`, `as.data.frame`, `as.list`, `dim`, `dropVirtualColumn`, `getColumnNames`, `getColumnNamesTranslator`, `getFullName`, `getName`, `getTags`, `getVirtualColumn`, `getVirtualColumnFunction`, `getVirtualColumnNames`, `hasColumn`, `hasColumns`, `hasVirtualColumn`, `hasVirtualColumns`, `length`, `names`, `newInstance`, `print`, `rbind`, `setAttributes`, `setColumnNamesMap`, `setColumnNamesTranslator`, `setName`, `setTags`, `setVirtualColumn`, `subset`, `translateColumnNames`

**Methods inherited from data.frame:**

`$<-`, `data.frame-method`, `$<-`, `Math`, `Ops`, `nonStructure`, `vector-method`, `Ops`, `structure`, `vector-method`, `Ops`, `vector`, `nonStructure-method`, `Ops`, `vector`, `structure-method`, `Ops`, `Summary`, `[`, `[<-`, `data.frame-method`, `[<-`, `[[`, `[[<-`, `data.frame-method`, `[[<-`, `aggregate`, `anyDuplicated`, `anyNA`, `as.NonPairedPSCNData`, `as.PairedPSCNData`, `as.data.frame`, `as.list`, `as.matrix`, `as.vector`, `attachLocally`, `by`, `callSegmentationOutliers`, `cbind`, `coerce`, `ANY`, `list-method`, `coerce`, `oldClass`, `S3-method`, `dim`, `dimnames`, `dimnames<-`, `dropSegmentationOutliers`, `droplevels`, `duplicated`, `edit`, `findLargeGaps`, `format`, `formula`, `head`, `initialize`, `oldClass-method`, `is.na`, `merge`, `na.exclude`, `na.omit`, `plot`, `plotDensity`, `print`, `prompt`, `rbind`, `row.names`, `row.names<-`, `rowsum`, `segmentByCBS`, `segmentByPairedPSCBS`, `show`, `oldClass-method`, `slotsFromS3`, `data.frame-method`, `sort_by`, `split`, `split<-`, `stack`, `str`, `subset`, `summary`, `t`, `tail`, `transform`, `type.convert`, `unique`, `unstack`, `unwrap`, `within`, `wrap`, `writeDataFrame`, `xtfrm`

**Author(s)**

Henrik Bengtsson



---

 RawMirroredAlleleBFractions

*The RawMirroredAlleleBFractions class*


---

## Description

Package: aroma.core

### Class RawMirroredAlleleBFractions

data.frame

~~|

~~+--[RichDataFrame](#)

~~~~~|

~~~~~+--[RawGenomicSignals](#)

~~~~~|

~~~~~+--[RawAlleleBFractions](#)

~~~~~|

~~~~~+--RawMirroredAlleleBFractions

### Directly known subclasses:

public class **RawMirroredAlleleBFractions**

extends [RawAlleleBFractions](#)

## Usage

```
RawMirroredAlleleBFractions(...)
```

## Arguments

... Arguments passed to [RawAlleleBFractions](#).

## Fields and Methods

### Methods:

plot -

### Methods inherited from RawAlleleBFractions:

extractRawMirroredAlleleBFractions, plot

### Methods inherited from RawGenomicSignals:

\*, +, -, addBy, append, applyBinaryOperator, as.character, as.data.frame, assertOneChromosome,

binnedSmoothing, binnedSmoothingByField, clearCache, clone, divideBy, drawDensity, estimateStandardDeviation, extractChromosome, extractChromosomes, extractDataForSegmentation, extractRegion, extractRegions, extractSubset, gaussianSmoothing, getBasicField, getCXY, getChromosome, getChromosomes, getDefaultLocusFields, getLocusFields, getPositions, getSigma, getSignalColumnName, getSignalColumnNames, getSignals, getWeights, getXScale, getXY, getYScale, hasWeights, kernelSmoothing, lines, multiplyBy, nbrOfChromosomes, nbrOfLoci, plot, points, print, segmentByCBS, segmentByGLAD, segmentByHaarSeg, segmentByMPCBS, setBasicField, setSigma, setSignals, setWeights, setXScale, setYScale, signalRange, sort, subtractBy, xMax, xMin, xRange, xSeq, yMax, yMin, yRange

**Methods inherited from RichDataFrame:**

\$, \$<-, [, [[, [[<-, as.data.frame, as.list, dim, dropVirtualColumn, getColumnNames, getColumnNamesTranslator, getFullName, getName, getTags, getVirtualColumn, getVirtualColumnFunction, getVirtualColumnNames, hasColumn, hasColumns, hasVirtualColumn, hasVirtualColumns, length, names, newInstance, print, rbind, setAttributes, setColumnNamesMap, setColumnNamesTranslator, setName, setTags, setVirtualColumn, subset, translateColumnNames

**Methods inherited from data.frame:**

\$<- ,data.frame-method, \$<-, Math, Ops,nonStructure,vector-method, Ops,structure,vector-method, Ops,vector,nonStructure-method, Ops,vector,structure-method, Ops, Summary, [, [<- ,data.frame-method, [<-, [[, [[<- ,data.frame-method, [[<-, aggregate, anyDuplicated, anyNA, as.NonPairedPSCNData, as.PairedPSCNData, as.data.frame, as.list, as.matrix, as.vector, attachLocally, by, callSegmentationOutliers, cbind, coerce,ANY,list-method, coerce,oldClass,S3-method, dim, dimnames, dimnames<- , dropSegmentationOutliers, droplevels, duplicated, edit, findLargeGaps, format, formula, head, initialize,oldClass-method, is.na, merge, na.exclude, na.omit, plot, plotDensity, print, prompt, rbind, row.names, row.names<-, rowsum, segmentByCBS, segmentByPairedPSCBS, show,oldClass-method, slotsFromS3,data.frame-method, sort\_by, split, split<-, stack, str, subset, summary, t, tail, transform, type.convert, unique, unstack, unwrap, within, wrap, writeDataFrame, xfrm

**Author(s)**

Henrik Bengtsson

---

RawSequenceReads      *The RawSequenceReads class*

---

**Description**

Package: aroma.core

**Class RawSequenceReads**

```
data.frame
~~|
~~+--RichDataFrame
~~~~~|
~~~~~+--RawGenomicSignals
~~~~~|
```

```
~~~~~+--RawSequenceReads
```

### Directly known subclasses:

```
public class RawSequenceReads
  extends RawGenomicSignals
```

### Usage

```
RawSequenceReads(x=NULL, y=rep(1L, length(x)), ...)
```

### Arguments

x            An [integer vector](#) of length J specifying the read positions.

y            An (optional) [integer vector](#) of length J specifying the number of reads at each position. Default is one read per position.

...          Arguments passed to [RawGenomicSignals](#).

### Fields and Methods

#### Methods:

```

binnedSums            -
extractRawCopyNumbers -
nbrOfReads            -
plot                   -

```

#### Methods inherited from [RawGenomicSignals](#):

\*, +, -, addBy, append, applyBinaryOperator, as.character, as.data.frame, assertOneChromosome, binnedSmoothing, binnedSmoothingByField, clearCache, clone, divideBy, drawDensity, estimateStandardDeviation, extractChromosome, extractChromosomes, extractDataForSegmentation, extractRegion, extractRegions, extractSubset, gaussianSmoothing, getBasicField, getCXY, getChromosome, getChromosomes, getDefaultLocusFields, getLocusFields, getPositions, getSigma, getSignalColumnName, getSignalColumnNames, getSignals, getWeights, getXScale, getX, getYScale, hasWeights, kernelSmoothing, lines, multiplyBy, nbrOfChromosomes, nbrOfLoci, plot, points, print, segmentByCBS, segmentByGLAD, segmentByHaarSeg, segmentByMPCBS, setBasicField, setSigma, setSignals, setWeights, setXScale, setYScale, signalRange, sort, subtractBy, xMax, xMin, xRange, xSeq, yMax, yMin, yRange

#### Methods inherited from [RichDataFrame](#):

\$, \$<-, [, [[, [[<-, as.data.frame, as.list, dim, dropVirtualColumn, getColumnNames, getColumnNamesTranslator, getFullName, getName, getTags, getVirtualColumn, getVirtualColumnFunction, getVirtualColumnNames, hasColumn, hasColumns, hasVirtualColumn, hasVirtualColumns, length, names, newInstance, print, rbind, setAttributes, setColumnNamesMap, setColumnNamesTranslator, setName, setTags, setVirtualColumn, subset, translateColumnNames

**Methods inherited from data.frame:**

\$<-,data.frame-method, \$<-, Math, Ops,nonStructure,vector-method, Ops,structure,vector-method, Ops,vector,nonStructure-method, Ops,vector,structure-method, Ops, Summary, [, [<-,data.frame-method, [<-, [[, [[<-,data.frame-method, [[<-, aggregate, anyDuplicated, anyNA, as.NonPairedPSCNData, as.PairedPSCNData, as.data.frame, as.list, as.matrix, as.vector, attachLocally, by, callSegmentationOutliers, cbind, coerce,ANY,list-method, coerce,oldClass,S3-method, dim, dimnames, dimnames<-, dropSegmentationOutliers, droplevels, duplicated, edit, findLargeGaps, format, formula, head, initialize,oldClass-method, is.na, merge, na.exclude, na.omit, plot, plotDensity, print, prompt, rbind, row.names, row.names<-, rowsum, segmentByCBS, segmentByPairedPSCBS, show,oldClass-method, slotsFromS3,data.frame-method, sort\_by, split, split<-, stack, str, subset, summary, t, tail, transform, type.convert, unique, unstack, unwrap, within, wrap, writeDataFrame, xfrm

**Author(s)**

Henrik Bengtsson

---

SegmentedAlleleBFractions

*The SegmentedAlleleBFractions class*

---

**Description**

Package: aroma.core

**Class SegmentedAlleleBFractions**

```
data.frame
~~|
~~+--RichDataFrame
~~~~~|
~~~~~+--RawGenomicSignals
~~~~~|
~~~~~+--RawAlleleBFractions
~~~~~|
~~~~~+--SegmentedGenomicSignalsInterface
~~~~~|
~~~~~+--SegmentedAlleleBFractions
```

**Directly known subclasses:**

```
public class SegmentedAlleleBFractions
extends SegmentedGenomicSignalsInterface
```

**Usage**

```
SegmentedAlleleBFractions(..., states=NULL)
```

**Arguments**

- ... Arguments passed to [RawAlleleBFractions](#).
- states A [function](#) returning the copy-number states given a [vector](#) of locus positions.

**Fields and Methods****Methods:**

*No methods defined.*

**Methods inherited from SegmentedGenomicSignalsInterface:**

as.data.frame, binnedSmoothingByState, extractSubsetByState, findChangePointsByState, getStateColorMap, getStateColors, getStates, getUniqueStates, getVirtualField, getVirtualLocusFields, kernelSmoothingByState, plot, points, setStateColorMap, setStates

**Methods inherited from RawAlleleBFractions:**

extractRawMirroredAlleleBFractions, plot

**Methods inherited from RawGenomicSignals:**

\*, +, -, addBy, append, applyBinaryOperator, as.character, as.data.frame, assertOneChromosome, binnedSmoothing, binnedSmoothingByField, clearCache, clone, divideBy, drawDensity, estimateStandardDeviation, extractChromosome, extractChromosomes, extractDataForSegmentation, extractRegion, extractRegions, extractSubset, gaussianSmoothing, getBasicField, getCX, getChromosome, getChromosomes, getDefaultLocusFields, getLocusFields, getPositions, getSigma, getSignalColumnName, getSignalColumnNames, getSignals, getWeights, getXScale, getXY, getYScale, hasWeights, kernelSmoothing, lines, multiplyBy, nbrOfChromosomes, nbrOfLoci, plot, points, print, segmentByCBS, segmentByGLAD, segmentByHaarSeg, segmentByMPCBS, setBasicField, setSigma, setSignals, setWeights, setXScale, setYScale, signalRange, sort, subtractBy, xMax, xMin, xRange, xSeq, yMax, yMin, yRange

**Methods inherited from RichDataFrame:**

\$, \$<-, [, [[, [[<-, as.data.frame, as.list, dim, dropVirtualColumn, getColumnNames, getColumnNamesTranslator, getFullName, getName, getTags, getVirtualColumn, getVirtualColumnFunction, getVirtualColumnNames, hasColumn, hasColumns, hasVirtualColumn, hasVirtualColumns, length, names, newInstance, print, rbind, setAttributes, setColumnNamesMap, setColumnNamesTranslator, setName, setTags, setVirtualColumn, subset, translateColumnNames

**Methods inherited from data.frame:**

\$<-,data.frame-method, \$<-, Math, Ops,nonStructure,vector-method, Ops,structure,vector-method, Ops,vector,nonStructure-method, Ops,vector,structure-method, Ops, Summary, [, [<-,data.frame-method, [<-, [, [[<-,data.frame-method, [[<-, aggregate, anyDuplicated, anyNA, as.NonPairedPSCNData, as.PairedPSCNData, as.data.frame, as.list, as.matrix, as.vector, attachLocally, by, callSegmentationOutliers, cbind, coerce,ANY,list-method, coerce,oldClass,S3-method, dim, dimnames, dimnames<-, dropSegmentationOutliers, droplevels, duplicated, edit, findLargeGaps, format, formula, head, initialize,oldClass-method, is.na, merge, na.exclude, na.omit, plot, plotDensity, print, prompt, rbind, row.names, row.names<-, rowsum, segmentByCBS, segmentByPairedPSCBS, show,oldClass-method, slotsFromS3,data.frame-method, sort\_by, split, split<-, stack, str, subset, summary, t, tail, transform, type.convert, unique, unstack, unwrap, within, wrap, writeDataFrame, xfrm

**Author(s)**

Henrik Bengtsson

---

SegmentedCopyNumbers    *The SegmentedCopyNumbers class*

---

## Description

Package: aroma.core

### Class SegmentedCopyNumbers

data.frame

~~|

~~+--RichDataFrame

~~~~~|

~~~~~+--RawGenomicSignals

~~~~~|

~~~~~+--RawCopyNumbers

~~~~~|

~~~~~+--SegmentedGenomicSignalsInterface

~~~~~|

~~~~~+--SegmentedCopyNumbers

### Directly known subclasses:

public class **SegmentedCopyNumbers**

extends [SegmentedGenomicSignalsInterface](#)

## Usage

```
SegmentedCopyNumbers(..., states=NULL)
```

## Arguments

...                    Arguments passed to [RawCopyNumbers](#).

states                A [function](#) returning the copy-number states given a [vector](#) of locus positions.

## Fields and Methods

### Methods:

*No methods defined.*

### Methods inherited from [SegmentedGenomicSignalsInterface](#):

as.data.frame, binnedSmoothingByState, extractSubsetByState, findChangePointsByState, getStateColorMap, getStateColors, getStates, getUniqueStates, getVirtualField, getVirtualLocusFields, kernelSmoothingByState, plot, points, setStateColorMap, setStates

### Methods inherited from [RawCopyNumbers](#):

cnRange, extractRawCopyNumbers, getCNs, getCn, getSignals, plot

**Methods inherited from RawGenomicSignals:**

\*, +, -, addBy, append, applyBinaryOperator, as.character, as.data.frame, assertOneChromosome, binnedSmoothing, binnedSmoothingByField, clearCache, clone, divideBy, drawDensity, estimateStandardDeviation, extractChromosome, extractChromosomes, extractDataForSegmentation, extractRegion, extractRegions, extractSubset, gaussianSmoothing, getBasicField, getCX, getChromosome, getChromosomes, getDefaultLocusFields, getLocusFields, getPositions, getSigma, getSignalColumnName, getSignalColumnNames, getSignals, getWeights, getXScale, getXY, getYScale, hasWeights, kernelSmoothing, lines, multiplyBy, nbrOfChromosomes, nbrOfLoci, plot, points, print, segmentByCBS, segmentByGLAD, segmentByHaarSeg, segmentByMPCBS, setBasicField, setSigma, setSignals, setWeights, setXScale, setYScale, signalRange, sort, subtractBy, xMax, xMin, xRange, xSeq, yMax, yMin, yRange

**Methods inherited from RichDataFrame:**

\$, \$<-, [, [[, [[<-, as.data.frame, as.list, dim, dropVirtualColumn, getColumnNames, getColumnNamesTranslator, getFullName, getName, getTags, getVirtualColumn, getVirtualColumnFunction, getVirtualColumnNames, hasColumn, hasColumns, hasVirtualColumn, hasVirtualColumns, length, names, newInstance, print, rbind, setAttributes, setColumnNamesMap, setColumnNamesTranslator, setName, setTags, setVirtualColumn, subset, translateColumnNames

**Methods inherited from data.frame:**

\$<-,.data.frame-method, \$<-, Math, Ops,nonStructure,vector-method, Ops,structure,vector-method, Ops,vector,nonStructure-method, Ops,vector,structure-method, Ops, Summary, [, [<-,.data.frame-method, [<-, [[, [[<-,.data.frame-method, [[<-, aggregate, anyDuplicated, anyNA, as.NonPairedPSCNData, as.PairedPSCNData, as.data.frame, as.list, as.matrix, as.vector, attachLocally, by, callSegmentationOutliers, cbind, coerce,ANY,list-method, coerce,oldClass,S3-method, dim, dimnames, dimnames<-, dropSegmentationOutliers, droplevels, duplicated, edit, findLargeGaps, format, formula, head, initialize,oldClass-method, is.na, merge, na.exclude, na.omit, plot, plotDensity, print, prompt, rbind, row.names, row.names<-, rowsum, segmentByCBS, segmentByPairedPSCBS, show,oldClass-method, slotsFromS3,.data.frame-method, sort\_by, split, split<-, stack, str, subset, summary, t, tail, transform, type.convert, unique, unstack, unwrap, within, wrap, writeDataFrame, xfrm

**Author(s)**

Henrik Bengtsson

**Examples**

```
# -----
# Simulating copy-number data
# -----
# True CN states
stateFcn <- function(x, ...) {
  states <- integer(length(x))
  states[200 <= x & x <= 300] <- -1L
  states[650 <= x & x <= 800] <- +1L
  states
}

# Number of loci
J <- 1000

y <- rnorm(J, sd=1/2)
```

```

x <- 1:length(y)
for (state in c(-1,+1)) {
  idxs <- (stateFcn(x) == state)
  y[idxs] <- y[idxs] + state
}

cn <- SegmentedCopyNumbers(y, x, states=stateFcn)
print(cn)

# -----
# Subsetting
# -----
plot(cn, ylim=c(-4,4))
title("Copy numbers annotated by state (and subset by state)")

cnS <- extractSubsetByState(cn, states=c(0,+1L))
print(cnS)
points(cnS, pch=21, cex=1.2, lwd=2, col="purple")

legend("topright", pch=c(19, 21), col=c("#999999", "purple"),
       sprintf(c("raw [n=%d]", "CN in {0,1} [n=%d]"),
               c(nbrOfLoci(cn), nbrOfLoci(cnS))), bty="n")

# -----
# Kernel smoothing stratified by state
# -----
plot(cn, col="#999999", ylim=c(-3,3))
title(main="Kernel smoothing stratified by state w/ Gaussian kernel")

cnSa <- kernelSmoothingByState(cn, h=2)
points(cnSa, col="blue")

cnSb <- kernelSmoothingByState(cn, h=5)
points(cnSb, col="red")

legend("topright", pch=19, col=c("#999999", "blue", "red"),
       sprintf(c("raw [n=%d]", "N(.,sd=2) [n=%d]", "N(.,sd=5) [n=%d]"),
               c(nbrOfLoci(cn), nbrOfLoci(cnSa), nbrOfLoci(cnSb))), bty="n")

# -----
# Binned smoothing stratified by state
# -----
plot(cn, col="#999999", ylim=c(-3,3))
title(main="Binned smoothing stratified by state")

cnSa <- binnedSmoothingByState(cn, by=3, verbose=-1)
lines(cnSa, col="blue")
points(cnSa, col="blue")

cnSb <- binnedSmoothingByState(cn, by=9, verbose=-1)

```



```

lines(cnSb, col="red")
points(cnSb, col="red")

legend("topright", pch=19, col=c("#999999", "blue", "red"),
       sprintf(c("raw [n=%d]", "Bin(w=3) [n=%d]", "Bin(w=9) [n=%d]"),
              c(nbrOfLoci(cn), nbrOfLoci(cnSa), nbrOfLoci(cnSb))), bty="n")

```

---

SegmentedGenomicSignalsInterface

*The SegmentedGenomicSignalsInterface class interface*


---

## Description

Package: aroma.core

### Class SegmentedGenomicSignalsInterface

#### Interface

~~|

~~+--SegmentedGenomicSignalsInterface

#### Directly known subclasses:

[SegmentedAlleleBFractions](#), [SegmentedCopyNumbers](#)

```

public class SegmentedGenomicSignalsInterface
extends Interface

```

## Usage

```
SegmentedGenomicSignalsInterface(...)
```

## Arguments

... Not used.

## Fields and Methods

### Methods:

```

as.data.frame          -
binnedSmoothingByState -
extractSubsetByState  -
findChangePointsByState -
getStateColorMap      -
getStateColors         -
getStates              -

```

|                        |   |
|------------------------|---|
| getUniqueStates        | - |
| kernelSmoothingByState | - |
| plot                   | - |
| points                 | - |
| setStateColorMap       | - |
| setStates              | - |

**Methods inherited from Interface:**

extend, print, uses

**Author(s)**

Henrik Bengtsson

---

UnitAnnotationDataFile

*The UnitAnnotationDataFile interface class*

---

**Description**

Package: aroma.core

**Class UnitAnnotationDataFile**

[Interface](#)

~~|

~~+--UnitAnnotationDataFile

**Directly known subclasses:**

[AromaUfiFile](#), [AromaUgcFile](#), [AromaUgpFile](#), [AromaUnitChromosomeTabularBinaryFile](#), [AromaUnitGcContentFile](#), [AromaUnitTabularBinaryFile](#), [AromaUnitTypesFile](#), [TextUnitNamesFile](#), [UnitNamesFile](#), [UnitTypesFile](#)

public static class **UnitAnnotationDataFile**

extends [Interface](#)

A UnitAnnotationDataFile provides methods for querying certain types of chip type annotation data by units.

**Usage**

UnitAnnotationDataFile(...)

**Arguments**

... Arguments passed to [Interface](#).

**Methods****Methods:**

|                 |   |
|-----------------|---|
| byChipType      | - |
| getAromaUflFile | - |
| getAromaUgpFile | - |
| getChipType     | - |
| getPlatform     | - |
| nbrOfUnits      | - |

**Methods inherited from Interface:**

extend, print, uses

**Author(s)**

Henrik Bengtsson

---

UnitNamesFile

*The UnitNamesFile interface class*


---

**Description**

Package: aroma.core

**Class UnitNamesFile****Interface**

```

~~|
~~+--UnitAnnotationDataFile
~~~~~|
~~~~~+--UnitNamesFile

```

**Directly known subclasses:**

[TextUnitNamesFile](#)

public abstract static class **UnitNamesFile**

extends [UnitAnnotationDataFile](#)

A UnitNamesFile provides methods for querying the unit names of a given chip type.

**Usage**

```
UnitNamesFile(...)
```

**Arguments**

... Arguments passed to [UnitAnnotationDataFile](#).

**Methods****Methods:**

```

getUnitNames -
indexOf      -
nbrOfUnits   -

```

**Methods inherited from UnitAnnotationDataFile:**

byChipType, getAromaUflFile, getAromaUgpFile, getChipType, getDefaultExtension, getPlatform, nbrOfUnits

**Methods inherited from Interface:**

extend, print, uses

**Author(s)**

Henrik Bengtsson

---

UnitTypesFile

*The UnitTypesFile interface class*


---

**Description**

Package: aroma.core

**Class UnitTypesFile****Interface**

```

~~|
~~+--UnitAnnotationDataFile
~~~~~|
~~~~~+--UnitTypesFile

```

**Directly known subclasses:**

[AromaUnitTypesFile](#)

```
public abstract static class UnitTypesFile
```

```
extends UnitAnnotationDataFile
```

A UnitTypesFile provides methods for querying the unit types of a given chip type, e.g. genotyping or copy-number unit, exon unit etc.

**Usage**

UnitTypesFile(...)

**Arguments**

... Arguments passed to [UnitAnnotationDataFile](#).

**Methods**

**Methods:**

getUnitTypes -  
nbrOfUnits -

**Methods inherited from UnitAnnotationDataFile:**

byChipType, getAromaUflFile, getAromaUgpFile, getChipType, getDefaultExtension, getPlatform, nbrOfUnits

**Methods inherited from Interface:**

extend, print, uses

**The aroma unit-type map**

unknown=0, expression=1, genotyping=2, resequencing=3, tag=4, copynumber=5, genotypingcontrol=6, expressioncontrol=7

**Author(s)**

Henrik Bengtsson

# Index

- \* **array**
  - colBinnedSmoothing.matrix, [68](#)
  - colKernelSmoothing.matrix, [71](#)
- \* **classes**
  - AbstractCNData, [4](#)
  - AbstractPSCNData, [6](#)
  - AromaCellCpgFile, [7](#)
  - AromaCellPositionFile, [8](#)
  - AromaCellTabularBinaryFile, [9](#)
  - AromaGenomeTextFile, [11](#)
  - AromaMicroarrayDataFile, [14](#)
  - AromaMicroarrayDataSet, [15](#)
  - AromaMicroarrayDataSetTuple, [17](#)
  - AromaMicroarrayTabularBinaryFile, [18](#)
  - AromaPlatform, [21](#)
  - AromaPlatformInterface, [22](#)
  - AromaRepository, [23](#)
  - AromaTabularBinaryFile, [25](#)
  - AromaTabularBinarySet, [27](#)
  - AromaTransform, [29](#)
  - AromaUnitCallFile, [30](#)
  - AromaUnitCallSet, [32](#)
  - AromaUnitFracBCnBinaryFile, [34](#)
  - AromaUnitFracBCnBinarySet, [37](#)
  - AromaUnitGenotypeCallFile, [39](#)
  - AromaUnitGenotypeCallSet, [41](#)
  - AromaUnitPscnBinaryFile, [43](#)
  - AromaUnitPscnBinarySet, [46](#)
  - AromaUnitSignalBinaryFile, [48](#)
  - AromaUnitSignalBinarySet, [50](#)
  - AromaUnitTabularBinaryFile, [52](#)
  - AromaUnitTotalCnBinaryFile, [54](#)
  - AromaUnitTotalCnBinarySet, [57](#)
  - AromaUnitTypesFile, [59](#)
  - BinnedScatter, [61](#)
  - CbsModel, [63](#)
  - ChromosomalModel, [65](#)
  - ChromosomeExplorer, [66](#)
  - CopyNumberChromosomalModel, [73](#)
  - CopyNumberSegmentationModel, [74](#)
  - Explorer, [77](#)
  - GladModel, [78](#)
  - HaarSegModel, [80](#)
  - NonPairedPSCNData, [82](#)
  - PairedPSCNData, [84](#)
  - ParametersInterface, [87](#)
  - RawAlleleBFractions, [88](#)
  - RawCopyNumberModel, [89](#)
  - RawCopyNumbers, [90](#)
  - RawGenomicSignals, [94](#)
  - RawMirroredAlleleBFractions, [97](#)
  - RawSequenceReads, [98](#)
  - SegmentedAlleleBFractions, [100](#)
  - SegmentedCopyNumbers, [102](#)
  - SegmentedGenomicSignalsInterface, [105](#)
  - UnitAnnotationDataFile, [106](#)
  - UnitNamesFile, [107](#)
  - UnitTypesFile, [108](#)
- \* **iteration**
  - colBinnedSmoothing.matrix, [68](#)
  - colKernelSmoothing.matrix, [71](#)
- \* **methods**
  - colBinnedSmoothing.matrix, [68](#)
  - colKernelSmoothing.matrix, [71](#)
- \* **package**
  - aroma.core-package, [3](#)
- \* **robust**
  - colBinnedSmoothing.matrix, [68](#)
  - colKernelSmoothing.matrix, [71](#)
- \* **univar**
  - colBinnedSmoothing.matrix, [68](#)
  - colKernelSmoothing.matrix, [71](#)
- \*colBinnedSmoothing, [72](#)
- \*colKernelSmoothing, [69](#)
  
- AbstractCNData, [4](#), [6](#), [82](#), [84](#), [94](#)
- AbstractPSCNData, [4](#), [6](#), [82](#), [85](#), [94](#)

- aroma.core (aroma.core-package), 3
- aroma.core-package, 3
- AromaCellCpgFile, 7, 9, 19, 22, 25
- AromaCellPositionFile, 8, 9, 19, 22, 25
- AromaCellSequenceFile, 9, 19, 22, 25
- AromaCellTabularBinaryFile, 8, 9, 19, 22, 25
- AromaGenomeTextFile, 11
- AromaMicroarrayDataFile, 14, 16
- AromaMicroarrayDataSet, 15, 15, 29
- AromaMicroarrayDataSetTuple, 17, 65
- AromaMicroarrayTabularBinaryFile, 9, 18, 22, 25, 52
- AromaPlatform, 21
- AromaPlatformInterface, 9, 19, 22, 30, 35, 39, 44, 48, 52, 55, 59
- AromaRepository, 23
- AromaTabularBinaryFile, 9, 10, 19, 21, 25, 27, 28, 30, 35, 39, 44, 48, 50, 52, 53, 55, 59
- AromaTabularBinarySet, 27, 33, 37, 41, 46, 50, 51, 57
- AromaTransform, 29, 87
- AromaUcscGenomeTextFile, 12
- AromaUflFile, 19, 22, 25, 53, 106
- AromaUgcFile, 19, 22, 25, 53, 106
- AromaUgpFile, 19, 22, 25, 53, 106
- AromaUnitCallFile, 22, 25, 30, 33, 39, 48
- AromaUnitCallSet, 27, 32, 42, 50
- AromaUnitChromosomeTabularBinaryFile, 19, 22, 25, 53, 106
- AromaUnitFracBCnBinaryFile, 22, 25, 34, 37, 48
- AromaUnitFracBCnBinarySet, 27, 37, 50
- AromaUnitGcContentFile, 19, 22, 25, 53, 106
- AromaUnitGenotypeCallFile, 22, 25, 31, 39, 42, 48
- AromaUnitGenotypeCallSet, 27, 33, 41, 50
- AromaUnitPscnBinaryFile, 22, 25, 43, 46, 48
- AromaUnitPscnBinarySet, 27, 46, 50
- AromaUnitSignalBinaryFile, 22, 25, 31, 35, 39, 44, 48, 51, 55, 59
- AromaUnitSignalBinarySet, 27, 33, 37, 42, 46, 50, 57, 58
- AromaUnitTabularBinaryFile, 11, 19, 22, 25, 35, 39, 40, 52, 60, 106
- AromaUnitTotalCnBinaryFile, 22, 25, 48, 54, 57
- AromaUnitTotalCnBinarySet, 27, 50, 57
- AromaUnitTotalCnBinarySetTuple, 17
- AromaUnitTypesFile, 22, 25, 48, 59, 106, 108
- BinnedScatter, 61
- binnedSmoothing
  - (colBinnedSmoothing.matrix), 68
- CacheKeyInterface, 9, 11, 14, 19, 25, 30, 35, 39, 43, 48, 52, 55, 59
- callNaiveGenotypes, 83, 85
- CbsModel, 63, 65, 73, 75, 76
- character, 4, 29, 65, 72, 73, 76, 77, 95
- ChromosomalModel, 63, 65, 73, 75, 79, 81, 89
- ChromosomeExplorer, 66, 77
- colBinnedSmoothing
  - (colBinnedSmoothing.matrix), 68
- colBinnedSmoothing.matrix, 68
- colKernelSmoothing
  - (colKernelSmoothing.matrix), 71
- colKernelSmoothing.matrix, 71
- ColumnNamesInterface, 9, 11, 19, 25, 30, 34, 39, 43, 48, 52, 55, 59
- CopyNumberChromosomalModel, 63, 65, 67, 68, 73, 75, 79, 81, 89, 90
- CopyNumberDataFile, 44, 55, 73
- CopyNumberDataSet, 46, 57, 73, 76
- CopyNumberDataSetTuple, 64, 73, 76, 79, 81
- CopyNumberSegmentationModel, 63–65, 73, 74, 79–82
- display, 78
- doCBS, 76
- double, 74
- downloadChipTypeFile, 24
- Explorer, 67, 77
- extractRawCopyNumbers, 74
- FALSE, 73
- FileCacheKeyInterface, 9, 11, 12, 14, 19, 25, 30, 35, 39, 43, 48, 52, 55, 59
- fit, 74, 75
- FullNameInterface, 9, 11, 14, 16, 17, 19, 25, 27, 30, 33, 34, 37, 39, 41, 43, 46, 48, 50, 52, 55, 57, 59
- function, 4, 69, 72, 101, 102

- GenericDataFile, [9](#), [11](#), [14](#), [19](#), [25](#), [27](#), [30](#),  
[34](#), [39](#), [43](#), [48](#), [52](#), [55](#), [59](#)
- GenericDataFileSet, [16](#), [27](#), [28](#), [33](#), [37](#), [41](#),  
[46](#), [50](#), [57](#)
- GenericDataFileSetList, [17](#), [18](#)
- GenericTabularFile, [9](#), [11](#), [19](#), [25](#), [30](#), [35](#),  
[39](#), [43](#), [48](#), [52](#), [55](#), [59](#)
- GenericTabularFileSet, [27](#), [33](#), [37](#), [41](#), [46](#),  
[50](#), [57](#)
- getAromaPlatform, [23](#)
- getChipType, [23](#), [66](#)
- getChromosomes, [66](#), [67](#)
- getFullName, [29](#)
- getInputDataSet, [29](#)
- getModel, [67](#)
- getName, [30](#), [78](#)
- getNames, [66](#), [78](#)
- getOutputDataSet, [30](#)
- getPath, [30](#), [78](#)
- getPlatform, [23](#)
- getRootPath, [30](#), [78](#)
- getTags, [30](#), [78](#)
- GladModel, [65](#), [73](#), [75](#), [78](#)
  
- HaarSegModel, [65](#), [73](#), [75](#), [80](#)
  
- Inf, [72](#), [83](#), [85](#)
- integer, [4](#), [6](#), [64](#), [67](#), [83](#), [85](#), [95](#), [99](#)
- Interface, [22](#), [23](#), [87](#), [105](#)–[108](#)
- isDone, [30](#)
  
- kernelSmoothing  
    (`colKernelSmoothing.matrix`), [71](#)
  
- list, [4](#), [16](#), [28](#), [62](#)
- logical, [6](#), [73](#), [76](#), [83](#), [85](#)
  
- matrix, [62](#), [68](#), [69](#), [72](#)
  
- NA, [6](#), [69](#), [83](#), [85](#)
- nbrOfArrays, [66](#), [78](#)
- nbrOfChipTypes, [18](#), [66](#)
- NonPairedPSCNData, [4](#), [6](#), [82](#), [94](#)
- NULL, [4](#), [6](#), [83](#), [85](#)
- numeric, [4](#), [6](#), [62](#), [68](#), [69](#), [72](#), [83](#), [85](#), [91](#), [95](#)
  
- Object, [9](#), [11](#), [14](#), [16](#), [17](#), [19](#), [21](#), [23](#)–[25](#), [27](#),  
[29](#), [30](#), [33](#), [34](#), [37](#), [39](#), [41](#), [43](#), [46](#), [48](#),  
[50](#), [52](#), [54](#), [57](#), [59](#), [63](#), [65](#), [66](#), [73](#), [74](#),  
[77](#), [78](#), [80](#), [89](#)
  
- PairedPSCNData, [4](#), [6](#), [82](#), [84](#), [94](#)
- ParametersInterface, [29](#), [87](#)
- png2, [68](#)
- process, [30](#), [67](#), [78](#)
  
- RawAlleleBFractions, [88](#), [94](#), [97](#), [100](#), [101](#)
- RawCopyNumberModel, [65](#), [73](#), [89](#)
- RawCopyNumbers, [90](#), [94](#), [102](#)
- RawGenomicSignals, [4](#), [6](#), [82](#), [84](#), [88](#), [91](#), [94](#),  
[97](#)–[100](#), [102](#)
- RawMirroredAlleleBFractions, [88](#), [94](#), [97](#)
- RawSequenceReads, [94](#), [98](#)
- readFooter, [26](#)
- RichDataFrame, [4](#), [6](#), [82](#), [84](#), [88](#), [91](#), [94](#), [95](#),  
[97](#), [98](#), [100](#), [102](#)
  
- segmentByCBS, [64](#)
- SegmentedAlleleBFractions, [88](#), [94](#), [100](#),  
[105](#)
- SegmentedCopyNumbers, [91](#), [94](#), [102](#), [105](#)
- SegmentedGenomicSignalsInterface, [100](#),  
[102](#), [105](#)
- seq, [69](#)
- setArrays, [67](#), [78](#)
  
- TabularTextFile, [11](#), [12](#)
- TextUnitNamesFile, [106](#), [107](#)
- TRUE, [69](#), [72](#)
  
- UnitAnnotationDataFile, [52](#), [53](#), [59](#), [106](#),  
[107](#)–[109](#)
- UnitNamesFile, [106](#), [107](#)
- UnitTypesFile, [59](#), [60](#), [106](#), [108](#)
  
- vector, [4](#), [6](#), [29](#), [65](#), [67](#)–[69](#), [72](#), [73](#), [76](#), [77](#), [83](#),  
[85](#), [91](#), [95](#), [99](#), [101](#), [102](#)
- Verbose, [24](#), [69](#), [72](#), [76](#)
  
- writeDataFrame, [49](#), [51](#), [53](#)
- writeFooter, [26](#)