Package ‘beadarraySNP’

January 30, 2017

Type Package
Title Normalization and reporting of Illumina SNP bead arrays
Version 1.40.0
Date 2014-10-10
Author Jan Oosting
Maintainer Jan Oosting <j.oosting@lumc.nl>
Depends methods, Biobase (>= 2.14), quantsmooth
Suggests aCGH, affy, limma, snapCGH, beadarray, DNAcopy
Description Importing data from Illumina SNP experiments and performing copy number calculations and reports.
License GPL-2
biocViews CopyNumberVariation, SNP, GeneticVariability, TwoChannel, Preprocessing, DataImport

NeedsCompilation no

R topics documented:

alterCN ........................................... 2
backgroundCorrect.SNP ........................................... 3
backgroundEstimate ........................................... 4
BeadstudioQC ........................................... 5
calculateLOH ........................................... 5
calculateQCarray ........................................... 6
compareGenotypes ........................................... 7
copynumberConversion ........................................... 8
createCNSummary ........................................... 9
dist.GT ........................................... 10
GenomicReports ........................................... 11
GetBeadStudioSampleNames ........................................... 12
getDNAindex ........................................... 13
heterozygosity ........................................... 14
heterozygousSNPs ........................................... 15
Illumina Genomic data ........................................... 16
interactiveCNselect ........................................... 16
normalizeBetweenAlleles.SNP ........................................... 17
normalizeBetweenSubsamples.SNP ........................................... 18
Description

Changes one of the levels of a cn.sum data structure

Usage

```r
alterCN(cn.sum, opa, value, updown)
```

Arguments

- `cn.sum`: cn.sum structure to change
- `opa`: opa panel within the structure
- `value`: the predicted value to change
- `updown`: the value has a higher (TRUE) or lower (FALSE) cn value

Details

The state in the cn.sum structure that has a predicted value of value will have it’s associated inferred copy number increased (updown is TRUE) or decreased (updown is FALSE). The function makes sure that the copynumber values within a OPA panel have the same order as the predicted values.
Value

a new cn.sum data structure

Author(s)

Jan Oosting

See Also

interactiveCNselect, createCNSummary, setRealCN

backgroundCorrect.SNP  Background correction

Description

Perform background correction on Illumina Golden Gate bead arrays

Usage

backgroundCorrect.SNP(object, method=c("none", "subtract", "half", "minimum", "edwards", "normexp", "rma"), offset = 0)

Arguments

object  SnpSetIllumina object
method  character, method of correction
offset  numeric, constant to add after correction

Details

Code has been ported from the limma package. The matrices Gb and Rb should be available in the arrayData slot of the object.

Value

This function returns an SnpSetIllumina object with background corrected values in the G and R.

Author(s)

Jan Oosting, based on limma package by G. Smyth

See Also

SnpSetIllumina-class, backgroundCorrect, backgroundEstimate, normalizeBetweenAlleles.SNP, normalizeWithinArrays.SNP

Examples

## Not run: data.bg<-backgroundCorrect.SNP(data.raw,"subtract")
backgroundEstimate

Estimate background intensities from foreground intensity

Description

Background intensity from Illumina Golden Gate bead arrays are estimated based on several data models.

Usage

```r
backgroundEstimate(object, method=c("minimum", "mode", "intmin", "anglemode"), maxmode=3000, bincount=40, maxangle=0.3, subsample="OPA")
```

Arguments

- `object`: SnpSetIllumina object
- `method`: character, data model to use
- `maxmode`: numeric, maximum intensity for mode for method="mode"
- `bincount`: numeric, for method="intmin", see details
- `maxangle`: numeric in radians, maximum theta for mode for method="anglemode"
- `subsample`: factor or column name in featureData slot

Details

The Illumina software does not provide background values in the output. Some models can be used to estimate background from the raw data intensities.

- **minimum**: The allele specific minimum intensity is used.
- **mode**: This model assumes that the first mode of the density of the intensities is determined by the zero-allele in the data, see ref. The signal intensity of the zero-allele should be zero, therefore this is considered the background value.
- **intmin**: This model assumes there is crosstalk between the alleles, and background increases with the intensity of the other allele. The range between 0 and the maximum of the other allele is divided in `bincount` bins, and the minimum for this allele is determined for probes where the other allele falls in a bin. A linear fit is determined through the minimum values to obtain a gradually increasing value.
- **anglemode**: This model finds the density modes closest to 0 and π/2 for polar transformed intensities, and uses this to determine background.

Value

This function returns an SnpSetIllumina object. The Rb and Gb matrices in the assayData slot contain estimated background values.

Author(s)

Jan Oosting

See Also

SnpSetIllumina-class, backgroundCorrect.SNP
**BeadstudioQC**

**Quality control of Beadstudio report files**

**Description**

When data has been imported using a Beadstudio samplesheet and reportfile, these functions can be used to generate quality measures.

**Usage**

```r
BeadstudioQC(object, QClist = list(), arrayType = "Sentrix96")
pdfBeadstudioQC(QClist, basename = "beadstudio", by = 10)
```

**Arguments**

- `object` : SnpSetIllumina object.
- `QClist` : list, result of previous call to `BeadstudioQC`.
- `arrayType` : character, type of array.
- `basename` : character, prefix for PDF files. This name will be added before the Barcode of the chip.
- `by` : integer, number of samples in barplot, see `reportSamplePanelQC`.

**Value**

The `BeadstudioQC` function generates a list of `QCIlumina` objects. The `pdfBeadstudioQC` function generates a pdf-file for each `QC Illumina` object in the list.

**Author(s)**

J. Oosting

**See Also**

- `pdfQC, calculateQCarray`
Arguments

object SnpSetIllumina object
grouping Factor to show which samples belong together (are of the same individual)
NorTum character vector or factor. Elements containing "N" are considered to be the normal sample
min.intensity numeric
use.homozygous.avg logical
... extra arguments for link(heterozygousSNPs)

Details

The heterozygous SNPs of the normal sample are inspected for changes. SNPs where the genotype of the test sample are homozygous are set to TRUE.

Value

For calculateLOH a SnpSetIllumina object with loh and nor.gt matrices in assayData. loh is a logical matrix, and nor.gt is a character matrix containing the genotypes of the corresponding normal sample. For calculateLair a SnpSetIllumina object with lair matrix in assayData. lair is the lesser allele intensity ratio. If a corresponding normal sample is found, it is taken as reference. Else the genotypes of normal samples are taken as a reference.

Author(s)

Jan Oosting

See Also

SnpSetIllumina-class

---

**calculateQArray**

Retrieve QC information from a SnpSetIllumina object

Description

Retrieves QC and identifying information of Illumina Sentrix arrays.

Usage

calculateQArray(object, QCobject = NULL, arrayType="Sentrix96")

Arguments

object SnpSetIllumina object. Should contain information of a single Sentrix array and a single type of OPA panel
QCobject QC1llumina-class object: If set the information in the object is amended with data from the SnpSetIllumina object
arrayType character, see arrayType
**compareGenotypes**

**Details**

Sample summary values are mapped to the physical layout of the Sentrix array using the Row and Col columns of the phenoData slot. These will be available when `read.SnpSetIllumina` is used to create SnpSetIllumina objects. Use successive calls to `calculateQCarray` to process Sentrix arrays with multiple probe panels. If data is read using a samplesheet that defines manifest files it is possible to handle data with multiple manifests and/or multiple Sentrix arrays.

**Value**

A QCIllumina object, when multiple arrays were combined a list of QCIllumina objects.

**Author(s)**

Jan Oosting

**See Also**

`link{QCIllumina-class}`, `link{reportSamplePanelQC}`, `link{plotQC}`

**Examples**

```r
## Not run: QC<-calculateQCarray(data.raw1)
## Not run: QC<-calculateQCarray(data.raw2,QC)
```

---

**compareGenotypes**  
*Compare genotypes*

**Description**

Pairwise comparison of genotypes between unaffected and affected tissue from the same subject.

**Usage**

```r
compareGenotypes(genotypeT, genotypeN)
```

**Arguments**

- `genotypeT` character or logical vector, genotypes of affected tissue
- `genotypeN` character or logical vector with same length as genotypeT, genotypes of unaffected, normal tissue

**Details**

Heterozygous probes have one the following values. TRUE, 'H' or 'AB'. All other values are considered homozygous. The primary purpose of the method is to find probes with loss of heterozygosity (LOH), where the unaffected probe is heterozygous and the affected is called homozygous.
Value

A vector with the same length as the arguments where each element can have one of four values

' u' Uninformative: both affected and normal are homozygous
'i' Informative: both affected and unaffected heterozygous
'l' Loss: unaffected heterozygous, affected homozygous
'a' Artefact: unaffected homozygous, affected heterozygous

Author(s)

Jan Oosting

See Also

heterozygousSNPs

Examples

data(chr17.260)
compareGenotypes(exprs(chr17.260)[,"514TV"],exprs(chr17.260)[,"514NP"])

Description

SnpSetIllumina objects are converted to other objects for numerical analysis

Usage

    convert2aCGH(object, normalizedTo=2, doLog=TRUE, organism="hsa")
    convert2SegList(object, normalizedTo=2, doLog=TRUE, organism="hsa")

Arguments

object SnpSetIllumina object
normalizedTo numeric, 'normal' copynumber data value for object
doLog logical, perform logarithmic transformation (log2)
organism character, organism used in object. Currently 'hsa' and 'mmu' are recognized. Used to convert sex chromosomes to their proper numerical representation

Details

These functions produce objects that can be used by the analysis functions in the aCGH or snapCGH packages. The SnpSetIllumina intensity values are stored in a linear scale. Both types of objects assume a logarithmic scale, so by default the values are transformed to a log2 scale centered around 0.
createCNSummary

Value

convert2aCGH returns a aCGH object as used in the aCGH package. convert2SegList returns a SegList object as used in the snapCGH package.

Author(s)

Jan Oosting

See Also

SnpSetIllumina-class, aCGH-class, SegList-class

createCNSummary Summarization of Copy number states

Description

Create a summary object of the genomic copy number states in a sample of segmented data

Usage

createCNSummary(object, sample, dnaIndex=1, subsample = "OPA")

Arguments

object SNPSetIllumina object after segmentation segmentate
sample SampleName or index of the sample for which to create the summary
dnaIndex Measured DNA index of the sample
subsample factor or column name in featureData slot

Details

The segments within a sample are assigned a copy number value. When the inferred slot in assayData is empty, all segments will be set to 2. Otherwise the values are recovered from the inferred slot. Gender is taken into account for the sex chromosomes.

Value

list with the following elements
dnaIndex same as parameter dnaIndex
CN.total.nrm Total expected copynumber for a 'normal' specimen ~ 2*featurecount
states data.frame with columns opa, count, intensity, copynumber

This list can be used as the cn.sum argument for plotGoldenGate4OPA, alterCN, getDNAindex and setRealCN

Author(s)

Jan Oosting
**Description**

Calculate distance matrix based of differences in genotype calls

**Usage**

```
dist.GT(object)
```

**Arguments**

- `object`: SnpSetIllumina object

**Details**

Calculates distances between samples as percentage of differences in genotype

**Value**

‘dist.GT’ returns an object of class ‘dist’

**Author(s)**

Jan Oosting

**See Also**

- `dist`, `hclust`
GenomicReports

Genomic reports

Description

Create reports for all samples in a dataset.

Usage

reportChromosomesSmoothCopyNumber(snpdata, grouping, normalizedTo=2, smooth.lambda=2, ridge.kappa=0, plotLOH=c("none", "marker", "line", "NorTum"), sample.colors = NULL, ideo.bleach=0.25, ...) reportSamplesSmoothCopyNumber(snpdata, grouping, normalizedTo=2, smooth.lambda=2, ridge.kappa=0, plotLOH=c("none", "marker", "line", "NorTum"), sample.colors=0, ...) reportGenomeGainLossLOH(snpdata, grouping, plotSampleNames=FALSE, distance.min, upcolor="red", downcolor="blue", lohcolor="grey", hetcolor="lightgrey", lohwidth=1, segment=101, orientation=c("V","H"), ...) reportChromosomeGainLossLOH(snpdata, grouping, plotSampleNames=FALSE, distance.min, upcolor="red", downcolor="blue", lohcolor="grey", hetcolor="lightgrey", proportion=0.2, plotLOH=TRUE, segment=101, ...) reportGenomeIntensityPlot(snpdata, normalizedTo=NULL, subsample=NULL, smoothing=c("mean", "quant"), dot.col="black", smooth.col="red", ...) 

Arguments

snpdata SnpSetIllumina object.

grouping factor, elements with same value are plotted together. Defaults to groups of 4 in order of the samples in the object.

normalizedTo numeric, a horizontal line is drawn at this position.

smooth.lambda smoothing parameter for quantsmooth.

ridge.kappa smoothing parameter for quantsmooth.

plotLOH indicate regions or probes with LOH, see details.

sample.colors vector of color.

plotSampleNames logical.

sizeSampleNames numeric, margin size for sample names.

distance.min numerical.

upcolor color.

downcolor color.

lohcolor color.

hetcolor color.

lohwidth numerical, relative width of the LOH part of the sample

segment integer.

orientation ["V","H"], vertical or horizontal orientation of plot.
proportion  numerical, proportion of the plot to use for idiogram annotation
subsample  character, or factor of length of features
smoothing  Type of smoothing per chromosome.
dot.col  color.
smooth.col  color.
ideo.bleach  numeric [0,1]
... arguments are forwarded to plot or getChangedRegions.

Details
The first function creates plots for each group and each chromosome in the dataset. The second function creates full genome plot for each group in the dataset. Beware that a lot of plots can be created, and usually you should prepare for that, by redirecting the plots to pdf or functions that create picture files like jpg, png, bmp.

Value
These functions are executed for their side effects

Author(s)
Jan Oosting

See Also
quantsmooth, prepareGenomePlot, pdfChromosomesSmoothCopyNumber, pdfSamplesSmoothCopyNumber

Examples
data(chr17.260)
chr17nrm <- standardNormalization(chr17.260)
par(mfrow = c(4,2), mar = c(2,4,2,1))
reportChromosomesSmoothCopyNumber(chr17nrm, grouping=pData(chr17.260)$Group,smooth.lambda = 4)

GetBeadStudioSampleNames

Extract samplenames from a report file

Description
Extract the samplenames from a report file that was created as a final report from Illumina Beadstudio

Usage
GetBeadStudioSampleNames(reportfile)

Arguments
reportfile character, name of report file
getDNAindex

Details
This function will read the report file, and extract the sample names from the Sample ID column

Value
character vector

Author(s)
Jan Oosting

See Also
read.SnpSetIllumina

getDNAindex Calculate the DNA index based on assigned copy number values to probes

Description
Calculate the DNA index based on assigned copy number values to probes

Usage
getDNAindex(cn.sum)

Arguments
cn.sum list with elements dnaIndex, CN.total.nrm, states, see createCNSummary

Value
scalar. DNA index of an unaffected sample is 1

Author(s)
Jan Oosting

See Also
createCNSummary, plotGoldenGate4OPA
### Description

Analyze affected material without corresponding unaffected material in order to find regions that contain stretches of homozygous SNPs as an indication of loss of heterozygosity (LOH).

### Usage

```r
heterozygosity(genotype, decay = 0.8, threshold = 0.1)
```

### Arguments

- **genotype**: character or logical vector, genotypes of affected tissue
- **decay**: numeric in range (0,1)
- **threshold**: numeric in range (0,1)

### Details

The method calculates how long the stretch of homozygous SNPs is for each element. `decay` and `threshold` can be set to skip individual heterozygous probes in a longer stretch of homozygous probes. The default setting tolerate 1 erroneous heterozygous SNP every 10 homozygous SNPs. Set `threshold` at 1 to stop discarding heterozygous SNPs.

### Value

A numeric vector with the same length as genotype is returned. Higher values, of 15 and higher, indicate regions of LOH.

### Author(s)

Jan Oosting

### See Also

- `compareGenotypes`
- `heterozygousSNPs`

### Examples

```r
data(chr17.260)
plot(heterozygosity(exprs(chr17.260)[,"514Tv"]))
```
heterozygousSNPs

| heterozygousSNPs | Retrieve heterozygous SNPs |

Description

Heterozygous SNPs are determined based on quality score criteria

Usage

```r
heterozygousSNPs(object, threshold=0.9, useQuality=TRUE, relative=TRUE, percentile=FALSE)
```

Arguments

- **object**: class SnpSetIllumina
- **threshold**: numeric (0:1) minimum quality score to be called heterozygous
- **useQuality**: logical, use quality score
- **relative**: logical, use quality score relative to GTS, see details
- **percentile**: logical, use percentage of probes above threshold

Details

This function presumes that the specificity for determining heterozygity is more important than the sensitivity, and will therefore only call probes heterozygous if that can be done with high certainty. The Illumina genotyping software calculates two quality measures: gen train score (GTS) and gen call score (GCS). The GTS is a measure for how well clusters can be recognized in a training set. This value is probe specific, and the same for all samples in an experiment. The GCS is a probe-specific, sample specific value that measures how close a probe in a sample is to the clusters determined in the training step. This value is always lower than the GTS for a probe.

`read.SnpSetIllumina` will put GCS into the `callProbability` element of the `assaydata` slot and the GTS into the `featureData` slot. The function uses these locations to retrieve the necessary information.

If `relative` is `FALSE` then the raw GCS values are compared to the threshold. In this case a threshold of around 0.5 should be used. If `relative` is `TRUE` then GCS/GTS is compared to the threshold and threshold should be around 0.9.

With `percentile=TRUE` the threshold quantile is calculated for each sample, and only probes with higher scores can be called heterozygous. A threshold of around 0.2 seems to work fine usually.

Value

This function returns a logical matrix with same dimensions as `object`.

Note

The purpose of the function is to separate heterozygous probes from non-heterozygous probes. In tumor samples the determination of the genotype can be difficult, because of aneuploidy and the fact that a sample is often a mixture of normal and tumor cells.

Author(s)

Jan Oosting
See Also

SnpSetIllumina-class

Examples

```r
data(chr17.260)
plot(heterozygosity(heterozygousSNPs(chr17.260[,"514TV"])), col="red", pch="x")
points(heterozygosity(exprs(chr17.260)[,"514TV"]))
```

Illumina Genomic data  Illumina example data

Description

These datasets are subsets of an experiment to test the applicability of paraffin embedded material in Illumina SNP arrays

Usage

```r
data(chr17.260)
data(QC.260)
```

Format

chr17.260 is a SnpSetIllumina object with data from chromosome 17 of 24 samples. QC.260 is a QCllumina object with summary data of 96 samples of a single SAM array

interactiveCNselect  Interactive assignment of copynumbers to genomic segments

Description

This function plots the genomic view of a sample, and allows the assignment of a discrete copy number to each segment

Usage

```r
interactiveCNselect(object, sample = 1, dnaIndex)
```

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>object</td>
<td>class SnpSetIllumina after segmentation</td>
</tr>
<tr>
<td>sample</td>
<td>Sample identifier within object</td>
</tr>
<tr>
<td>dnaIndex</td>
<td>numeric, measured DNA index of the sample (1=normal)</td>
</tr>
</tbody>
</table>
normalizeBetweenAlleles.SNP

**Details**

The user can interactively assign discrete, integer copy number values to each segment. This is done by either clicking in the lower part of a panel to decrease the copy number, or in the higher part of a panel to increase the copy number. The order of copy number values is always maintained; a segment with a lower raw value cannot get a higher copy number assigned than a segment with a higher raw copy number value.

**Value**

list, see `createCNSummary`

**Author(s)**

Jan Oosting

**See Also**

`segmentate`, `alterCN`, `plotGoldenGate4OPA` `createCNSummary`

---

**Description**

Perform between Allele normalization on Illumina Golden Gate bead arrays

**Usage**

```r
normalizeBetweenAlleles.SNP(object, method=c("quantile"), subsample="OPA")
```

**Arguments**

- `object`: class SnpSetIllumina
- `method`: char, type of normalization
- `subsample`: factor with length number of features in object or char, column name in featureData slot

**Details**

This function performs a quantile normalization between the Red and Green channels for each sample. The rationale for this procedure stems from the fact that the allele frequencies within each channel are always very similar, even in the presence of genomic abnormalities.

**Value**

This function returns an SnpSetIllumina object.

**Author(s)**

Jan Oosting
normalizeBetweenSubsamples.SNP

See Also
SnpSetIllumina-class, normalizeWithinArrays.SNP, backgroundCorrect.SNP

Examples

data(chr17.260)
data.nrm<-normalizeBetweenAlleles.SNP(chr17.260)

normalizeBetweenSubsamples.SNP

Normalization between subsamples

Description
Quantile normalization between subsamples within a single SnpSetIllumina object

Usage
normalizeBetweenSubsamples.SNP(object, subsample = "OPA")

Arguments

object
class SnpSetIllumina

subsample
factor with length number of features in object or char, column name in featureData slot

Details
Perform quantile normalization of the red and green channel between subsamples. This can be used in situations where multiple different assays that cover the same genomic regions (or whole genome) have been done on the same biological specimen. This function was introduced for version 5 Golden Gate Linkage analysis that consist of 4 assays of ~ 1500 probes. Where previous versions of this assay each targeted a number of chromosomes, in version 5 each assay covers the whole genome.

Value
This function returns an SnpSetIllumina object.

Author(s)
Jan Oosting

See Also
SnpSetIllumina-class, normalizeBetweenAlleles.SNP, normalizeWithinArrays.SNP, backgroundCorrect.SNP

Examples

data(chr17.260)
data.nrm<-normalizeBetweenSubsamples.SNP(chr17.260)
normalizeLoci.SNP

Description

Perform locus normalization on Illumina Golden Gate bead arrays

Usage

normalizeLoci.SNP(object, method=c("normals","paired","alleles"), NorTum="NorTum", Gender="Gender", Subject="Subject", normalizeTo=2, trig=FALSE)

Arguments

- **object**: object class SnpSetIllumina
- **method**: character. If "normals" then all normal samples in the dataset are used as the invariant set. If "paired" then affected samples are normalized to their paired normal samples. "alleles" fits a linear model between the B-allele ratio and the signal intensity and normalizes for that
- **NorTum**: logical or character vector or name of column in pData slot. depicts the normal, unaffected samples in the dataset. In a character vector these should have the value "N"
- **Gender**: logical or character vector or name of column in pData slot. depicts the female samples in the dataset and is used to normalize the sex chromosomes. In a character vector these should have value "F"
- **Subject**: factor or name of or column in pData slot. This factor is used to pair the samples when method is "paired"
- **normalizeTo**: numeric. The average copy number of the sample.
- **trig**: Logical, use geometric distance of intensity. Otherwise use addition of intensities

Details

This function is usually performed in the last step of normalization in order to obtain calculated copy numbers.

Value

This function returns an SnpSetIllumina object.

Author(s)

Jan Oosting

See Also

SnpSetIllumina, normalizeWithinArrays.SNP, normalizeBetweenAlleles.SNP
**Examples**

```r
data(chr17.260)
data.nrm<-normalizeLoci.SNP(chr17.260)
```

**normalizeWithinArrays.SNP**

*Within Array normalization*

**Description**

Perform within array normalization on Illumina Golden Gate bead arrays.

**Usage**

```r
normalizeWithinArrays.SNP(object, callscore=0.5, normprob=0.5, quantilepersample=FALSE, relative=FALSE, fixed=FALSE, useAll=FALSE, subsample="OPA", Q.scores="callProbability")
```

**Arguments**

- **object**: class SnpSetIllumina.
- **callscore**: numeric with range 0:1, threshold for probe inclusion.
- **normprob**: numeric with range 0:1, target quantile for normalization. The default is to divide the sample intensities by the median of the selected subset.
- **quantilepersample**: logical. If TRUE then the threshold is determined for each sample, else it is experiment wide. This is only relevant when fixed is FALSE.
- **relative**: logical. If TRUE then the ratio of GCS and GTS is used, else only the GCS is used as the quality score.
- **fixed**: logical. If TRUE then callscore is the fixed threshold for the quality score, else the probes above the quantile callscore are used.
- **useAll**: logical. If TRUE then all probes in the dataset are eligible as the invariant set, else only the heterozygous SNPs.
- **subsample**: factor or column name in featureData slot, the levels of the factor are treated separately.
- **Q.scores**: name of assayData() element, or numeric matrix of appropriate size. Quality scores to select high quality SNPs

**Details**

The function uses high quality heterozygous SNPs as an invariant set with the assumption that these have the highest probability of coming from unaffected regions of the genome. Most of the arguments are used to determine the quality of the call.

**Value**

This function returns a SnpSetIllumina object.
Author(s)
Jan Oosting

See Also
SnpSetIllumina, normalizeLoci.SNP, backgroundCorrect.SNP, normalizeBetweenAlleles.SNP

Examples
```r
data(chr17.260)
data.nrm <- normalizeWithinArrays.SNP(chr17.260)

pdfChromosomesSmoothCopyNumber(object, filename, ...)
```

Description
Functions that help create pdf reports

Usage
```
pdfChromosomesSmoothCopyNumber(object, filename, ...)  
pdfSamplesSmoothCopyNumber(object, filename, ...)  
pdfChromosomeGainLossLOH(object, filename, ...)
```

Arguments
```
object SnpSetIllumina object
filename filename of output pdf file
... arguments for report functions
```

Details
These functions set up and perform reporting to pdf files.

Value
This function is used for its side effects

Author(s)
Jan Oosting

See Also
```
reportChromosomesSmoothCopyNumber, reportSamplesSmoothCopyNumber, reportChromosomeGainLossLOH
```

Examples
```r
# Not run: data(chr17.260)
# Not run: data.nrm<-standardNormalization(chr17.260)
# Not run: pdfChromosomesSmoothCopyNumber(data.nrm, "Chr17.pdf", grouping=pData(data.nrm)$Group, smooth.lambda=4)
```
pdfQC  

Description
Create PDF file with experimental quality control plots

Usage
pdfQC(object, filename = "arrayQC.pdf", by = 10)

Arguments
- object: QCIllumina object, or list of QCIllumina objects
- filename: character, output pdf filename
- by: number of samples in barplot, see reportSamplePanelQC

Details
This function creates a PDF file with QC information. The first page contains 8 plotQC panels showing the spatial distribution of intensities on a SAM plate. The following page(s) contain the output of reportSamplePanelQC

Value
A PDF file is produced

Author(s)
Jan Oosting

See Also
plotQC, reportSamplePanelQC, QCIllumina-class

plotGoldenGate4OPA  

Description
Plots a full genome view based on 4 subsamples of Illumina Golden Gate data

Usage
plotGoldenGate4OPA(object, cn.sum = NULL, sample = 1, plotRaw = FALSE, main = NULL, interact = FALSE)
plotGenomePanels(object, cn.sum = NULL, sample = 1, plotRaw = FALSE, main = NULL, interact = FALSE)
plotQC

Arguments

object class SnpSetIllumina
cn.sum list containing genomic states, see createCNSummary
sample identifier to select the sample within the object
plotRaw logical, plot raw data points
main character, Title of plot
interact logical, plot should be usable for interactive copy number determination interactiveCNselect
allLair logical, TRUE: plot all LAIR values, FALSE: only plot LAIR values from probes that are heterozygous in the paired normal sample
panels list, vectors of chromosomes for each panel
... extra arguments are forwarded to plot

Details

prepare interactive selection

Value

list, see createCNSummary

Author(s)

Jan Oosting

See Also

segmentate, alterCN, interactiveCNselect createCNSummary

plotQC Spatial plots of array QC information

Description

Plots array wide summary information using the layout of the physical medium

Usage

plotQC(object, type)

Arguments

object object that contains QC information. e.g. QC-Illumina-class
type character, the type of information to plot, currently the following types are supported: "intensityMed", "greenMed", "redMed", "validn", "annotation" and "samples"

Value

The function is used for its side effects
PolarTransforms

Author(s)
Jan Oosting

See Also
pdfQC, reportSamplePanelQC

Examples
```r
data(QC.260)
plotQC(QC.260,"greenMed")
```

Description
Perform polar transforms on Illumina Golden Gate bead arrays

Usage
```r
RG2polar(object, trig = FALSE)
polar2RG(object, trig = FALSE)
```

Arguments
- **object**: SnpSetIllumina object
- **trig**: Logical, use geometric distance intensity. Otherwise use addition of intensities

Details
- **RG2polar** transforms the \( R \) and \( G \) matrices to \( \theta \) and intensity matrices. Note that the intensity value is the sum of \( R \) and \( G \) and not the geometric distance to the origin.
- **polar2RG** performs the reverse transformation

Value
This function returns an SnpSetIllumina object.

Author(s)
Jan Oosting

See Also
SnpSetIllumina-class

Examples
```r
data(chr17.260)
data.polar <- RG2polar(chr17.260)
plot(assayData(data.polar)$theta, assayData(data.polar)$intensity)
```
Description

These generic functions set and retrieve properties of quality control objects like `QCIllumina-class`.

Usage

```r
arrayType(object)
arrayType(object) <- value
arrayID(object)
arrayID(object) <- value
```

Arguments

- `object`: Object, possibly derived from class `QCIllumina-class`.
- `value`: character.

Details

Currently the following types of arrays are supported:
- "Sentrix96": Sentrix array, 12 columns, 8 rows
- "Sentrix16": Sentrix array, 2 columns, 8 rows
- "Slide12": Slide with 12 samples

Value

`arrayType` and `arrayID` return a character value.

Author(s)

Jan Oosting

QCIllumina-class

Class "QCIllumina"

Description

Container of QC information on arrays that contain multiple samples.

Objects from the Class

Objects can be created by calls of the form `new("QCIllumina", arrayType, arrayID, intensityMed, greenMed, redMed, intensityMode, greenMode, redMode, validn, annotation, samples)` but are usually created by `calculateQCarray`. 
Slots

arrayType: character, Type of array. See arrayType
arrayID: character, Array ID
intensityMed: numeric matrix, Median of intensity of samples
greenMed: numeric matrix, Median of green values
redMed: numeric matrix, Median of red values
callrate: numeric matrix, callrate of genotyping
hetPerc: numeric matrix, Percentage of heterozygotes
ptpdiff: numeric matrix, point-to-point difference, local estimate of variability
validn: numeric matrix, Number of valid probe values in samples
annotation: character matrix, Annotation of samples
samples: character matrix, Sample IDs

Methods

arrayID signature(object = "QCIllumina"): Returns type of array
arrayID<- signature(object = "QCIllumina"): Sets type of array. Currently only "Sentrix" is supported
arrayType signature(object = "QCIllumina"): Returns ID of array
arrayType<- signature(object = "QCIllumina"): Sets ID/Barcode of array
initialize signature(.Object = "QCIllumina")
plotQC signature(object = "QCIllumina") character: plots spatial overview of QC information, type is one of c("intensityMed", "greenMed", "redMed", "validn", "annotation", "samples")

Author(s)

Jan Oosting

See Also

calculateQCarray

Description

A SnpSetIllumina object is created from the textfiles created by the Illumina GenCall or BeadStudio software.

Usage

read.SnpSetIllumina(samplesheet, manifestpath=NULL, reportpath=NULL, rawdatapath=NULL, reportfile=NULL, briefOPAinfo=TRUE, readTIF=FALSE, nochecks=FALSE, sepreport="\t", essentialOnly=FALSE, ...)
**Arguments**

- **samplesheet**: a data.frame or filename, contains the sample sheet
- **manifestpath**: a character string for the path containing the manifests / OPA definition files, defaults to path of samplesheet
- **reportpath**: a character string for the path containing the report files, defaults to path of samplesheet
- **rawdatapath**: a character string for the path containing the intensity data files, defaults to path of samplesheet
- **reportfile**: a character string for the name of BeadStudio reportfile
- **briefOPAinfo**: logical, if TRUE then only the SNP name, Illumina code, chromosome and basepair position are put into the `featureData` slot of the result, else all information from the OPA file is put into the `featureData` slot
- **readTIF**: logical, uses `beadarray` package and raw TIF files to read data
- **nochecks**: logical, limited validity checks on beadstudio report files. See details
- **separator**: character, field separator character for beadstudio report files
- **essentialOnly**: logical, if TRUE then only the essential columns from a reportfile are included into the result. See details
- ... arguments are forwarded to `readIllumina` and can be used to perform bead-level normalization

**Details**

The text files from Illumina software are imported to a `SnpSetIllumina` object. Both result files from GenCall and BeadStudio can be used. In both cases the sample sheets from the experiments are used to select the proper data from the report or data files. The following columns from the sample sheet file are used for this purpose: 'Sample_Name', 'Sentrix_Position', and 'Pool_ID'. The values in columns 'Sample_Plate', 'Pool_ID', and 'Sentrix_ID' should be the same for all samples in the file, as this is the case for processed experiments. The contents of the sample sheet are put into the `phenoData` slot.

Zero values in the raw data signals are set to NA

Ideally the OPA manifest file containing SNP annotation should be available, these files are provided by Illumina. Columns 'IllCode', 'CHR', and 'MapInfo' are put into the `featureData` slot.

**GenCall Data**

In order to process experiments that were genotyped using the GenCall software, the arrays should be scanned with the setting `<SaveTextFiles>true</SaveTextFiles>` in the Illumina configuration file `Settings.XML`. 3 Types of files need to be present in the same folder: The sample sheet, `.csv` files containing signal intensity data, and the report file that contains the genotype information. For each sample in the sample sheet there should be a `.csv` file with the following file mask: `[sam_id]_R00[yy]_C00[xx].csv`, where `sam_id` is the Illumina ID for the SAM, and `xx` and `yy` are the column and row number respectively. From the report files the file with mask `[Pool_ID]_LocusByDNA[_ExpName].csv` is used. 'Pool_ID' is the OPA panel used, and '_ExpName' is optional.

**BeadStudio Data**

To process experiments that were processed with BeadStudio, only two files are needed. The sample sheet and the Final Report file. The sample sheet must contain the same columns as for GenCall, the report file should contain the following columns: 'SNP Name', 'Sample ID', 'GC Score', 'Allele1 - AB', 'Allele2 - AB', 'GT Score', 'X Raw', and 'Y Raw'. 'SNP Name' and
'Sample ID' are used to form rows and columns in the experimental data, 'GC Score' is put in the callProbability matrix, 'Allele1 - AB' and 'Allele2 - AB' are combined into the call matrix, 'GT Score' is added to the featureData slot, 'X Raw' is put in the R matrix and 'Y Raw' in the G matrix. Other columns in the report files are added as matrices in the assayData slot, or columns in the featureData slot if values are identical for all samples in the reportfile. When nochecks is TRUE then only the 'SNP Name' and 'Sample ID' columns are required. The resulting object is now of class MultiSet.

Sample sheets

To help generate a sample sheet for BeadStudio data a Sample_Map.txt file can be converted to a sample sheet with the Sample_Map2Samplesheet function. For Beadstudio reportfiles it is also possible to set samplesheet=NULL. In this case the phenoData slot will be fabricated from the sample names in the reportfile.

Manifest/OPA/annotation files

For BeadStudio reportfiles it is not necessary to have a Manifest file if the columns 'Chr' and 'Position' are available in the report file. Currently this is the only way to import data from Infinium arrays, because Illumina does not supply Manifest files for these arrays.

Value

This function returns an SnpSetIllumina object, or a MultiSet object when nochecks is TRUE.

Author(s)

Jan Oosting

See Also

SnpSetIllumina-class, Sample_Map2Samplesheet, readIllumina

Examples

# read a SnpSetIllumina object using example textfiles in data directory
datadir <- system.file("testdata", package="beadarraySNP")
SNPdata <- read.SnpSetIllumina(paste(datadir,"4samples_opa4.csv",sep="/"),datadir)

removeLowQualityProbes

Quality control of SnpSetIllumina objects

Description

Remove probes form a SnpSetIllumina object that show a low quality throughout the experiment

Usage

removeLowQualityProbes(object, cutoff = 0.25)

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>object</td>
<td>SnpSetIllumina object</td>
</tr>
<tr>
<td>cutoff</td>
<td>numeric</td>
</tr>
</tbody>
</table>
Details

Probes that have a median value below cutoff * median value for the whole experiment are deleted from the object.

Value

SnpSetIllumina object

Author(s)

Jan Oosting

---

**removeLowQualitySamples**

*Quality control of SnpSetIllumina objects*

Description

Remove samples from a SnpSetIllumina object that show a low quality

Usage

`removeLowQualitySamples(object, min.intensity = 1500, min.gt = 100, subsample = "OPA")`

Arguments

- **object** `SnpSetIllumina-class` object
- **min.intensity** numeric. Samples that show a median intensity below this value in either Red or Green channel are removed
- **min.gt** numeric. Samples that have less than this amount of valid genotypes are removed
- **subsample** factor or column name in featureData slot of object

Value

This function returns an SnpSetIllumina object.

Author(s)

Jan Oosting

Examples

```r
data(chr17.260)
chr17.260<-removeLowQualitySamples(chr17.260,min.gt=10)
```
renameOPA  

*Change the linkage panel in a dataset*

**Description**

Change the linkage panel in a dataset

**Usage**

```r
renameOPA(snpdata, newOPA)
```

**Arguments**

- `snpdata`  
  SnpSetIllumina object
- `newOPA`  
  character, new linkage panel

**Details**

In order to combine different versions of the linkage panels, this function makes it possible to map the equivalent SNPs in both datasets.

**Value**

SnpSetIllumina object

**Author(s)**

Jan Oosting

---

reportGenotypeSegmentation  

*plot genomic view*

**Description**

Create a figure that can be used for interactive work

**Usage**

```r
reportGenotypeSegmentation(object, plotRaw = TRUE, subsample = NULL, panels = 0, minProbes = 10, maxY = 2, ...)
```

**Arguments**

- `object`  
  SnpSetIllumina object after segmentation
- `plotRaw`  
  logical
- `subsample`  
  factor
- `panels`  
  number of panels on a page
- `minProbes`  
  minimum number of probes for a chromosome within a panel
- `maxY`  
  maximum value on vertical scale within panels
- `...`  
  arguments are forwarded to `plot`
reportSamplePanelQC-methods

Value
this function is used for its side effects

Author(s)
Jan Oosting

Description
Show raw intensity values for green and red channel for all samples in an experiment

Usage
reportSamplePanelQC(object, by=10, legend=TRUE, ...)

Arguments
object
QCIllumina object
by
numeric, number of samples in each plot
legend
logical, create a final plot with a common legend for the barplots
... arguments are forwarded to barplot

Examples
data(QC.260)
par(mfrow=c(2,2))
reportSamplePanelQC(QC.260,by=8)

Sample_Map2Samplesheet

Convert Beadstudio Sample Map file to samplesheet

Description
Create a samplesheet that can be used to import Illumina beadstudio data

Usage
Sample_Map2Samplesheet(samplemapfile, saveas = "")

Arguments
samplemapfile character, name of the SampleMap file
saveas character, optional, name of samplesheet file that can be used directly by read.SnpSetIllumina
segmentate

Details

During the creation of a final report file from Beadstudio, there is an option to create Map files. The Sample_Map.txt file can be used to create an initial samplesheet for use in the read.SnpSetIllumina function.

Value

A data.frame with the samplesheet.

Author(s)

J. Oosting

See Also

read.SnpSetIllumina

segmentate

Segmentation for SnpSetIllumina objects

Description

Use snapCGH package to perform segmentation.

Usage

segmentate(object, method = c("DNACopy", "HMM", "BioHMM", "GLAD"), normalizedTo = 2, doLog = TRUE, doMerge = FALSE, useLair = FALSE, subsample = "OPA", alpha = 0.01)

Arguments

object: class SnpSetIllumina
method: char, type of segmentation
normalizedTo: numeric
doLog: logical, perform transformation before segmentation, see convert2seglist
doMerge: logical, perform merging of close states
useLair: logical, Also segmentate on lair
subsample: factor
alpha: numeric, probability threshold to distinguish segments

Value

SnpSetIllumina object with elements observed, states and predicted set in the AssayData slot.

Author(s)

Jan Oosting
setRealCN

Integrate state information into SNP object

Description
Set calculated values of copy numbers in inferred element of AssayData slot

Usage
setRealCN(object, sample, cn.sum, subsample="OPA")

Arguments
- object: class SnpSetIllumina
- sample: sample identifier
- cn.sum: list, see createCNSummary
- subsample: "OPA"

Value
SnpSetIllumina object with inferred element of AssayData slot set

Author(s)
Jan Oosting

See Also
segmentate, alterCN, plotGoldenGate4OPA createCNSummary

smoothed.intensity
Smooth intensity data

Description
Create a table of smoothe intensity values

Usage
smoothed.intensity(snpdata, smooth.lambda = 4, tau = 0.5)

Arguments
- snpdata: SnpSetIllumina object
- smooth.lambda: smoothing parameter
- tau: quantile to smooth

Value
Numerical matrix with same dimensions as data
SnpSetIllumina

Author(s)
Jan Oosting

See Also
SnpSetIllumina-class

SnpSetIllumina  
Class to Contain Objects Describing High-Throughput SNP Assays.

Description
Container for high-throughput assays and experimental metadata. SnpSetIllumina class is derived from eSet, and requires matrices R, G, call, callProbability as assay data members. It supports featureData. Several visualization methods use columns CHR and MapInfo. The CHR column is used to handle sex chromosomes in a specific way. The OPA column is the default way to specify subsamples.

Extends
Directly extends class eSet.

Creating Objects
ew('SnpSetIllumina', phenoData = [AnnotatedDataFrame], experimentData = [MIAME], annotation = [character], call = [matrix], callProbability = [matrix], G = [matrix], R = [matrix], featureData = [data.frameOrNULL], ...)

SnpSetIllumina instances are usually created through new("SnpSetIllumina", ...). Arguments to new include call (a matrix of genotypic calls, with features (SNPs) corresponding to rows and samples to columns), callProbability, G, R, phenoData, experimentData, and annotation. phenoData, experimentData, and annotation can be missing, in which case they are assigned default values.

Slots
Inherited from Biobase:eSet:

assayData: Contains matrices with equal dimensions, and with column number equal to nrow(phenoData). assayData must contain a matrix call with rows representing features (e.g., SNPs) and columns representing samples, a matrix callProbability describing the certainty of the call, and matrices R and G to describe allele specific intensities. The contents of these matrices are not enforced by the class. The assayData matrices Gb, Rb, intensity, theta are optional, but are either results or input for several methods of the class. Additional matrices of identical size may also be included in assayData. Class: AssayData.

phenoData: See eSet.

experimentData: See eSet.

annotation: See eSet.

featureData: annotation for SNPs, usually will contain a CHR and a MapInfo column for genomic localization.
Methods

Class-specific methods:

eprs(SnpSetIllumina), eprs(SnpSetIllumina, matrix) <- Access and set elements named call in the AssayData slot.

combine(SnpSetIllumina, SnpSetIllumina): performs union-like combination in both dimensions of SnpSetIllumina objects.

efData(SnpSetIllumina), efData(SnpSetIllumina, data.frame) <- Access and set the pData in the featureData slot.

calculateGSR(SnpSetIllumina) calculate ratio of Gentrain score and Gencall score. Creates GSR matrix in assayData. Should be performed before combining datasets.

calculateSmooth(object, smoothType) calculate smoothed data, creates smoothed matrix in assayData. smoothType can only be "quantsmooth" at the moment.

sortGenomic(SnpSetIllumina) order the data by chromosome and position on the chromosome.

Derived from eSet:

sampleNames(SnpSetIllumina) and sampleNames(SnpSetIllumina) <- See eSet.

featureNames(SnpSetIllumina), featureNames(SnpSetIllumina, value) <- See eSet.

dims(SnpSetIllumina): See eSet.

phenoData(SnpSetIllumina), phenoData(SnpSetIllumina, value) <- See eSet.

varLabels(SnpSetIllumina), varLabels(SnpSetIllumina, value) <- See eSet.

varMetadata(SnpSetIllumina), varMetadata(SnpSetIllumina, value) <- See eSet.

pData(SnpSetIllumina), pData(SnpSetIllumina, value) <- See eSet.

varMetadata(SnpSetIllumina), varMetadata(SnpSetIllumina, value) <- See eSet.

experimentData(SnpSetIllumina), experimentData(SnpSetIllumina, value) <- See eSet.

pubMedIds(SnpSetIllumina), pubMedIds(SnpSetIllumina, value) <- See eSet.

abstract(SnpSetIllumina): See eSet.

annotation(SnpSetIllumina), annotation(SnpSetIllumina, value) <- See eSet.

storageMode(eSet), storageMode(eSet, character) <- See eSet.

featureData(SnpSetIllumina), featureData(SnpSetIllumina, AnnotatedDataFrame) <- See eSet.

object[(index)]: Conducts subsetting of matrices and phenoData and featureData components.

Standard generic methods:

initialize(SnpSetIllumina): Object instantiation, used by new; not to be called directly by the user.

validObject(SnpSetIllumina): Validity-checking method, ensuring that call, callProbability, G, and R are members of assayData. checkValidity(SnpSetIllumina) imposes this validity check, and the validity checks of Biobase::class.eSet.

show(SnpSetIllumina) See eSet.

dim(SnpSetIllumina), ncol See eSet.

SnpSetIllumina[(index)]: See eSet.

SnpSetIllumina$, SnpSetIllumina$<- See eSet.
Author(s)
J. Oosting, based on Biobase eSet class

See Also
eSet

SnpSetSegments-class  Class “SnpSetSegments”

Description
The SnpSetSegments class is a direct descendant of the SnpSetIllumina class, with an extra slot to define the genomic segments in each sample.

Objects from the Class
Objects can be created by calls of the form new("SnpSetSegments", assayData, phenoData, experimentData, annotation, protocolData, call, callProbability, G, R, cn.segments, featureData, extraData, ...).

Slots
- cn.segments: Object of class "list"
- assayData: Object of class "AssayData" see "SnpSetIllumina"
- phenoData: Object of class "AnnotatedDataFrame" see "SnpSetIllumina"
- featureData: Object of class "AnnotatedDataFrame" see "SnpSetIllumina"
- experimentData: Object of class "MIAME" see "SnpSetIllumina"
- annotation: Object of class "character" see "SnpSetIllumina"
- protocolData: Object of class "AnnotatedDataFrame" see "SnpSetIllumina"
- __classVersion__: Object of class "Versions" "VersionedBiobase"

Extends

Methods
- cn.segments signature(object = "SnpSetSegments"): ...
- cn.segments<- signature(object = "SnpSetSegments", value = "list"): ...
- initialize signature(.Object = "SnpSetSegments"): ...

Note
This class is under development, and not usable in the current form

Author(s)
Jan Oosting
standardNormalization

References
Corver et.al. Can Res dec 2008

See Also
segmentate

Examples
showClass("SnpSetSegments")

standardNormalization  Default complete normalization

Description
Performs all steps in normalization at best settings as determined in ref.

Usage
standardNormalization(snpdata)

Arguments
snpdata SnpSetIllumina object with raw data

Details
The function performs in the following steps
snpdata<-normalizeBetweenAlleles.SNP(snpdata)
snpdata<-normalizeWithinArrays.SNP(snpdata,callscore = 0.8, relative = TRUE, fixed = FALSE, quantilepersample = TRUE)
snpdata<-normalizeLoci.SNP(snpdata,normalizeTo = 2)

Value
A SnpSetIllumina object with the G, R and intensity elements in assayData normalized to obtain values close to 2 on a linear scale for unaffected material.

Author(s)
Jan Oosting

See Also
normalizeBetweenAlleles.SNP,normalizeWithinArrays.SNP,normalizeLoci.SNP

Examples
data(chr17.260)
data.nrm<-standardNormalization(chr17.260)
Index

*Topic **classes**
  QCIllumina-class, 25
  SnpSetIllumina, 34
  SnpSetSegments-class, 36
*Topic **datasets**
  Illumina Genomic data, 16
*Topic **file**
  read.SnpSetIllumina, 26
*Topic **hplot**
  BeadstudioQC, 5
  pdfChromosomesSmoothCopyNumber, 21
  pdfQC, 22
  plotGoldenGate4OPA, 22
  plotQC, 23
  reportGenotypeSegmentation, 30
  reportSamplePanelQC-methods, 31
*Topic **iplot**
  interactiveCNselect, 16
  plotGoldenGate4OPA, 22
*Topic **manip**
  alterCN, 2
  backgroundCorrect.SNP, 3
  backgroundEstimate, 4
  BeadstudioQC, 5
  calculateLOH, 5
  calculateQCaray, 6
  compareGenotypes, 7
  copynumberConversion, 8
  createCNSummary, 9
  dist.GT, 10
  GenomicReports, 11
  GetBeadStudioSampleNames, 12
  getDNAindex, 13
  heterozygosity, 14
  heterozygousSNPs, 15
  interactiveCNselect, 16
  normalizeBetweenAlleles.SNP, 17
  normalizeBetweenSubsamples.SNP, 18
  normalizeLocis.SNP, 19
  normalizeWithinArrays.SNP, 20
  PolarTransforms, 24
  QCaccessors, 25
  read.SnpSetIllumina, 26
  removeLowQualityProbes, 28
  removeLowQualitySamples, 29
  renameOPA, 30
  Sample_Map2Samplesheet, 31
  segmentate, 32
  setRealCN, 33
  smoothed.intensity, 33
  standardNormalization, 37
*Topic **methods**
  reportSamplePanelQC-methods, 31
  [,SnpSetIllumina-method (SnpSetIllumina), 34
  alterCN, 2, 10, 17, 23, 33
  arrayID (QCaccessors), 25
  arrayID, QCIllumina-method (QCIllumina-class), 25
  arrayID<- (QCaccessors), 25
  arrayID<-,QCIllumina-method (QCIllumina-class), 25
  arrayType, 6, 26
  arrayType (QCaccessors), 25
  arrayType, QCIllumina-method (QCIllumina-class), 25
  arrayType<-,QCIllumina-method (QCIllumina-class), 25
  AssayData, 34
  backgroundCorrect, 3
  backgroundCorrect.SNP, 3, 4, 18, 21
  backgroundEstimate, 3, 4
  BeadstudioQC, 5
  calculateGSR (SnpSetIllumina), 34
  calculateGSR, SnpSetIllumina-method (SnpSetIllumina), 34
  calculateLair (calculateLOH), 5
  calculateLOH, 5
  calculateQCaray, 5, 6, 25, 26
  calculateSmooth (SnpSetIllumina), 34
  calculateSmooth, SnpSetIllumina,character-method (SnpSetIllumina), 34
  chr17.260 (Illumina Genomic data), 16
<table>
<thead>
<tr>
<th>Class/Method/Function</th>
<th>Page Numbers</th>
</tr>
</thead>
<tbody>
<tr>
<td>class:SnpSetIllumina</td>
<td>34</td>
</tr>
<tr>
<td>cn.segments(SnpSetSegments-class)</td>
<td>36</td>
</tr>
<tr>
<td>cn.segments(SnpSetSegments-method)</td>
<td>36</td>
</tr>
<tr>
<td>cn.segments&lt;- (SnpSetSegments-class)</td>
<td>36</td>
</tr>
<tr>
<td>cn.segments&lt;- (SnpSetSegments-list-method)</td>
<td>36</td>
</tr>
<tr>
<td>combine,SnpSetIllumina,ANY-method</td>
<td>34</td>
</tr>
<tr>
<td>combine,SnpSetIllumina,SnpSetIllumina-method</td>
<td>34</td>
</tr>
<tr>
<td>compareGenotypes</td>
<td>7, 14</td>
</tr>
<tr>
<td>convert2aCGH(copynumberConversion)</td>
<td>8</td>
</tr>
<tr>
<td>convert2SegList(copynumberConversion)</td>
<td>8</td>
</tr>
<tr>
<td>copynumberConversion</td>
<td>8</td>
</tr>
<tr>
<td>createCNSummary</td>
<td>3, 9, 13, 17, 23, 33</td>
</tr>
<tr>
<td>dist</td>
<td>10</td>
</tr>
<tr>
<td>dist.GT</td>
<td>10</td>
</tr>
<tr>
<td>eSet</td>
<td>34–36</td>
</tr>
<tr>
<td>exprs,SnpSetIllumina-method</td>
<td>34</td>
</tr>
<tr>
<td>exprs&lt;-SnpSetIllumina, matrix-method</td>
<td>34</td>
</tr>
<tr>
<td>fData,SnpSetIllumina-method</td>
<td>34</td>
</tr>
<tr>
<td>fData&lt;-SnpSetIllumina.data.frame-method</td>
<td>34</td>
</tr>
<tr>
<td>featureData,SnpSetIllumina-method</td>
<td>34</td>
</tr>
<tr>
<td>featureData&lt;-SnpSetIllumina-method</td>
<td>34</td>
</tr>
<tr>
<td>.GenomicReports, GetBeadStudioSampleNames, getDNAindex</td>
<td>11, 12</td>
</tr>
<tr>
<td>hclust</td>
<td>10</td>
</tr>
<tr>
<td>heterozygosity</td>
<td>14</td>
</tr>
<tr>
<td>heterozygousSNPs</td>
<td>8, 14, 15</td>
</tr>
<tr>
<td>Illumina Genomic data</td>
<td>16</td>
</tr>
<tr>
<td>initialize,QCIIllumina-method</td>
<td>25</td>
</tr>
<tr>
<td>initialize,SnpSetIllumina-method</td>
<td>34</td>
</tr>
<tr>
<td>initialize,SnpSetSegments-method</td>
<td>34</td>
</tr>
<tr>
<td>initialize,SnpSetSegments-method</td>
<td>36</td>
</tr>
<tr>
<td>interactiveCNselect</td>
<td>3, 16, 23</td>
</tr>
<tr>
<td>MultiSet, normalizeBetweenAlleles.SNP, normalizeBetweenSubsamples.SNP, normalizeLoci.SNP, normalizeWithinArrays.SNP</td>
<td>34, 36, 37</td>
</tr>
<tr>
<td>normalizeChromosomeGainLossLOH (pdfChromosomesSmoothCopyNumber)</td>
<td>21</td>
</tr>
<tr>
<td>plotGenomePanel</td>
<td>10, 13, 17, 22, 33</td>
</tr>
<tr>
<td>plotQC</td>
<td>22, 23</td>
</tr>
<tr>
<td>qsar2RG(PolarTransforms)</td>
<td>24</td>
</tr>
<tr>
<td>prepareGenomePlot</td>
<td>12</td>
</tr>
<tr>
<td>QC.260(Illumina Genomic data)</td>
<td>16</td>
</tr>
<tr>
<td>QCaccessors</td>
<td>25</td>
</tr>
<tr>
<td>QCIIllumina</td>
<td>5</td>
</tr>
<tr>
<td>QCIIllumina(QCIIllumina-class)</td>
<td>25</td>
</tr>
<tr>
<td>QCIIllumina-class</td>
<td>25</td>
</tr>
<tr>
<td>quantsmooth</td>
<td>11, 12</td>
</tr>
<tr>
<td>read.SnpSetIllumina</td>
<td>7, 13, 26, 31, 32</td>
</tr>
<tr>
<td>readIllumina</td>
<td>27, 28</td>
</tr>
<tr>
<td>removeLowQualityProbes</td>
<td>28</td>
</tr>
<tr>
<td>removeLowQualitySamples</td>
<td>29</td>
</tr>
<tr>
<td>renameOA</td>
<td>30</td>
</tr>
<tr>
<td>reportChromosomeGainLossLOH</td>
<td>21</td>
</tr>
<tr>
<td>reportChromosomeSampleCopyNumber</td>
<td>21</td>
</tr>
<tr>
<td>reportChromosomeSmoothCopyNumber</td>
<td>21</td>
</tr>
<tr>
<td>reportGenomeGainLossLOH</td>
<td>(GenomicReports), 11</td>
</tr>
<tr>
<td>reportGenomeSampleCopyNumber</td>
<td>(GenomicReports), 11</td>
</tr>
<tr>
<td>reportGenomeIntensityPlot</td>
<td>(GenomicReports), 11</td>
</tr>
<tr>
<td>reportGenotypeTypeSegment</td>
<td>30</td>
</tr>
<tr>
<td>reportSamplePanelQC</td>
<td>5, 22, 24</td>
</tr>
</tbody>
</table>
reportSamplePanelQC
  (reportSamplePanelQC-methods), 31
reportSamplePanelQC,QCIllumina-method
  (reportSamplePanelQC-methods), 31
reportSamplePanelQC-methods, 31
reportSamplesSmoothCopyNumber, 21
reportSamplesSmoothCopyNumber
  (GenomicReports), 11
RG2polar (PolarTransforms), 24
Sample_Map2Samplesheet, 28, 31
segmentate, 9, 10, 17, 23, 32, 33, 37
setRealCN, 3, 33
smoothed.intensity, 33
SnpSetIllumina, 19, 21, 34, 36
SnpSetIllumina-class (SnpSetIllumina), 34
SnpSetSegments-class, 36
sortGenomic (SnpSetIllumina), 34
sortGenomic, SnpSetIllumina-method
  (SnpSetIllumina), 34
standardNormalization, 37
Versioned, 36
VersionedBiobase, 36