Package 'VanillaICE'

September 24, 2012

Version 1.18.0

Title A Hidden Markov Model for high throughput genotyping arrays

Author
Robert Scharpf <rscharpf@jhsph.edu>, Kevin Scharpf, and Ingo Ruczinski <ingo@jhsph.edu>

Maintainer Robert Scharpf <rscharpf@jhsph.edu>

Depends R (>= 2.14.0)

Imports stats, utils, methods, Biobase, oligo-
Classes (>= 1.17.36),lattice, IRanges (>= 1.13.22), grid, msm, crlmm (>= 1.13.12),foreach

Suggests genomewidesnp6Crlmm, hapmapsnp6, RColorBrewer, genefilter,RSQLite, fore-
ach, RUnit, pd.mapping50k.hind240, SNPchip (>=
2.0.3)

Enhances DNAcopy, crlmm

Description Hidden Markov Models for characterizing chromosomal alterations in high through-
put SNP arrays

License LGPL-2

LazyLoad yes

Collate AllGenerics.R AllClasses.R methods-CopyNumberSet.R
methods-oligoSnpSet.R methods-BeadStudioSet.R
methods-RangedDataHMM.R methods-HmmOptionList.R methods-CNSet.R
methods-matrix.R methods-SnpSet.R methods-Vit.R methods-Viterbi.R methods-Viterbi2.R
methods-GenomeAnnotatedDataFrame.R methods-BeadStudioSetList.R
hmm-methods.R functions.R utils.R zzz.R

biocViews Bioinformatics, CopyNumberVariants, SNP, GeneticVariability,Visualization

R topics documented:

`BeadStudioSet` *Constructor for BeadStudioSet class*

Description

Constructs an instance of BeadStudioSet from a list of files containing log R ratios and B allele frequencies.

Usage

```
BeadStudioSet(filenames, lrr.colname = "Log.R.Ratio", baf.colname = "B.Allele", sep = "\t", head
```

Arguments

`filenames`	character string providing the names of the BeadStudio files, including the complete path if not in the working directory.
`lrr.colname`	character string providing the column header for log R ratios
`baf.colname`	character string providing the column header for log R ratios
`sep`	field delimiter in the BeadStudio files. See `read.table`
`header`	logical: whether the files contain a header.
`colClasses`	See `read.table`.
`universe`	character string indicating which genome build to use. This option is not currently available.
`annotationPkg`	character string providing the name of the annotation package.
`chromosome`	integer vector indicating which chromosomes to include in the BeadStudioSet. E.g., 1:23 for autosomes and chromosome X
`...`	Additional arguments to `read.bsfiles`.

Value

An object of class `BeadStudioSet`

Author(s)

R. Scharpf

See Also

read.bsfiles, BeadStudioSet

Examples

```
path <- system.file("extdata", package="VanillaICE")
fname <- file.path(path, "LRRandBAF.txt")
bsSet <- BeadStudioSet(fname, annotationPkg="genomewidesnp6Crlmm")
```

BeadStudioSetList *Constructor for BeadStudioSetList class.*

Description

Reads processed files containing log R Ratios and B allele frequencies and construct a BeadStudioList object.

Usage

```
BeadStudioSetList(fnames, annotationPkg, genomeBuild = "hg19", outdir = ldPath(), sampleIds, phe
```

Arguments

fnames	character vector containing the complete path to files containing log R ratios and BAFs.
annotationPkg	character string indicating the name of the annotation package.
genomeBuild	character string indicating which genome build
outdir	character string indicating where to store ff files for storing the log R ratios and B allele frequencies. Ignored if the ff package is not loaded.
sampleIds	character vector of sample identifiers. If missing, basename(fnames) is used.
phenoData	An AnnotatedDataFrame containing covariates for the samples.
...	Additional arguments passed to the initialization method for BeadStudioSetList.

Value

A BeadStudioSetList.

Author(s)

R. Scharpf

See Also

BeadStudioSet, BeadStudioSetList

Examples

```
new("BeadStudioSetList")
```

BeadStudioSetList-class
BeadStudioSetList class

Description

Container for log R ratios and B allele frequencies stored by chromosome.

Objects from the Class

Objects can be created by calls of the form `new("BeadStudioSetList", assayDataList, logRRatio, BAF, featur`

Slots

`assayDataList`: Object of class `"AssayData"` ~~

`phenoData`: Object of class `"AnnotatedDataFrame"` ~~

`featureDataList`: Object of class `"list"` ~~

`chromosome`: Object of class `"integer"` ~~

`annotation`: Object of class `"character"` ~~

`genomeBuild`: Object of class `"character"` ~~

Methods

[`signature(x = "BeadStudioSetList")`: ...

`object[[i]]`: Returns an object of class `BeadStudioSet` from the ith element in `object`.

$ `signature(x = "BeadStudioSetList")`: ...

`object$name <- value`: assign vector `value` to variable name `name` of `phenoData`

`hmm(object, ...)`: Fits HMM to BeadStudioSetList object. Additional arguments can be passed to `hmmBeadStudioSetList`.

length `signature(x = "BeadStudioSetList")`: ...

Author(s)

R. Scharpf

See Also

`BeadStudioSetList`

Examples

```
new("BeadStudioSetList")
```

centerAutosomesAt *Center estimates of copy number for autosomes.*

Description

Center estimates of copy number for autosomes.

Usage

```
centerAutosomesAt(x, at, ...)
```

Arguments

x	A oligoSnpSet or CopyNumberSet object.
at	numeric. Value at which to center the copy number estimates (e.g., 2).
...	Ignored

Details

The function sweeps out the column median of the autosomal copy number estimates, and adds back a constant given by at.

Value

A object of the same class as x with centered copy number estimates for the autosomal chromosomes. Chromosomes X and Y are not centered.

Author(s)

R. Scharpf

See Also

link{sweep}

copyNumberLimits *Constraints for updating the means for the copy number states in the hidden markov model.*

Description

Constraints for updating the means for the copy number states in the hidden markov model.

Usage

```
copyNumberLimits(is.log)
```

Arguments

is.log	logical: whether the copy number estimates are on the log scale

Details

Not indented to be called directly – used by packages that depend on VanillaICE.

Value

A numeric vector of length 2 giving the lower and upper bounds for the copy number estimates.

Author(s)

R. Scharpf

---

hmm-methods *Hidden Markov Model methods*

Description

Hidden Markov Model methods in package **VanillaICE**

Methods

The hmm method is defined for several classes of containers of preprocessed and normalized SNP array data. The most common containers for use with genotyping platforms are the BeadStudioSet and oligoSnpSet classes. The primary difference between these two containers are the requirements for the assay data elements. A BeadStudioSet object must have assay data elements "lrr" (log R ratios) and "baf" (B allele frequencies). Genotype calls are optional for the BeadStudioSet object. As the name implies, the BeadStudioSet container would typically be generated as part of a pipeline to process data from Illumina array platforms. By contrast, the oligoSnpSet object has required assay data elements "call" (genotype calls), "callProbability" (genotype confidence scores), "copyNumber", and "cnConfidence". As B allele frequencies are perhaps more informative than the genotype calls for distinguishing copy number states (particularly amplifications), an assay data element named "baf" can be included in the assay data for an oligoSnpSet object. The presence of a "baf" element in the assay data of an oligoSnpSet has implications on the particular HMM fit to identify the CNV boundaries (as discussed below).

A hidden Markov model for the BeadStudioSet class. The assay data are log R ratios and B allele frequencies. See hmmBeadStudioSet for additional arguments that can be passed through the ... operator.

signature(object = "BeadStudioSet", ...) signature(object = "SnpSet", ...) A hidden Markov model for the SnpSet class. The assay data are diallelic genotype calls represented as integers (1=AA, 2=AB, 3=BB). See hmmSnpSet for additional arguments that can be passed through the ... operator.

signature(object = "CNSet", ...) A hidden Markov model for the CNSet class. The CNSet instance is first coerced to an object of class oligoSnpSet containing estimates of total copy number and B allele frequencies. See hmmBeadStudioSet for additional arguments that can be passed through the ... operator. For large data sets, the initial coercion to the oligoSnpSet class can be very expensive in terms of I/O and require a large amount of RAM. Users with large data sets may prefer to coerce selected samples (e.g., the set of samples belonging to a given batch) to an oligoSnpSet object, and then fit the hmm on the oligoSnpSet object directly. This approach is illustrated in the crlmmDownstream vignette.

`signature(object = "CopyNumberSet", ...)` A hidden Markov model for the `CopyNumberSet` class. The assay data are estimates of total copy number. This method should not be used for arrays with genotype information as the genotypes / B allele frequencies are informative for copy number inference.

`signature(object = "oligoSnpSet", ...)` A hidden Markov model for the `oligoSnpSet` class. If "baf" is included among the assay data elements, the `hmmBeadStudioSet` HMM is implemented. Otherwise, the `hmmOligoSnpSet` is implemented.

See Also

`hmmBeadStudioSet`, `hmmOligoSnpSet`, `hmmSnpSet`

`hmmBeadStudioSet` *HMM functions for oligoSnpSet and BeadStudioSet containers*

Description

HMM functions for `oligoSnpSet` and `BeadStudioSet` containers. These functions are exported in the package's namespace to provide documentation of arguments that can be passed from the `hmm` method for these containers. The `hmmBeadStudioSet` function is always called when the object passed to the `hmm` method is a `BeadStudioSet`. By contrast, the `hmm` method for `oligoSnpSet` objects will only call the `hmmOligoSnpSet` function if B allele frequences (assay data element "baf") is not included in the list of assay data elements. Specifically, if assay data element "baf" is in the list of assay data elements of a `oligoSnpSet` container, the `hmm` method for the `oligoSnpSet` class calls the `hmmBeadStudioSet` function.

Usage

```
hmmBeadStudioSet(object, cnStates = logCnStates(), normalIndex = 3L,rohIndex = normalIndex + 1L,
hmmOligoSnpSet(object, cnStates = c(0, 1, 2, 2, 3, 4), normalIndex = 3L, rohIndex = normalIndex
```

Arguments

`object`	A `oligoSnpSet` or `BeadStudioSet`.
`cnStates`	A vector of starting values (numeric) specifying the means of the Normal distribution assumed for latent copy numbers. The means must be specified for states homozygous deletion (zero copies), hemizygous deletion (1 copy), normal (2 copies), normal and no heterozygotes (2 copies), single copy duplication (3 copies), and two+ copy duplication (4+ copies). The starting values are updated via EM.
`normalIndex`	Integer indicating the state index for diploid copy number. This should nearly always be '3' if the 6-state HMM (see `cnStates`) is fit as recommended.
`rohIndex`	The state index (integer) for diploid copy number without heterozygotes. This should be '4'. It is important to include this state even if homozygous regions with diploid copy number are not of interest.
`prOutlierCN`	The probability that a copy number estimate is an outlier. This is an initial estimate that is updated for each copy number state via EM.
`prOutlierBAF`	The probability that a B allele frequency is an outlier. This is an initial estimate that is updated for each copy number state via EM.

p.hom

TAUP — Scalar for the transition probability matrix. Larger values discourage transitions from the normal state. (The transition probabilities are a function of the distance between adjacent markers. These probabilities are not updated as part of the EM step.)

is.log — A logical indicating whether the copy number estimates are on the log scale. Note that the assay data elements in oligoSnpSet and BeadStudioSet should be represented as integers (copy number or relative copy number * 100). If is.log is TRUE, we assume that after division by 100 the assay data element containing the copy numbers (or relative copy numbers) is on the log-scale. The scale has implications on what is considered to be extreme.

initialProb — Vector of initial state probabilities. This is required to be the same length as cnStates.

center — Whether to center the copy number for each chromosomal arm at the theoretical mean for the diploid copy number state. This may not be appropriate for some datasets (e.g., trisomy 21, cancer applications). A safer approach is to set this argument to FALSE and center all autosomes at the theoretical mean for two copies prior to fitting the HMM.

reestimation — Logical, but currently ignored.

nupdates — The maximum number of reestimation steps for updating the mean, variance, and outlier probabilities of the Gaussian-Uniform mixture for each copy number state.

tolerance — If the difference in the log likelihood between successive EM updates is less than tolerance, the number of updates can be less than nupdates.

... — Presently ignored

Value

A RangedData-derived object.

Author(s)

R. Scharpf

See Also

hmmSnpSet

---

hmmResults *Example output from hmm*

Description

Example output from hmm method applied to simulated data.

Usage

```
data(hmmResults)
```

Format

A RangedDataHMM object.

Details

The results of a 6-state HMM fit to simulated copy number and genotype data.

See Also

xyplot

Examples

```
library(oligoClasses)
data(oligoSetExample, package="oligoClasses")
oligoSet <- oligoSet[chromosome(oligoSet) == 1, ]
hmmResults <- hmm(oligoSet)
state(hmmResults)
##
## Plotting ranges:
##
if(require(SNPchip) && require(IRanges)){
## Plot the data for the second range with a blue
## border, and frame the region by 10 Mb on each side
## of the state boundary.
##
xyplot(cn~x, oligoSet, range=hmmResults[2, ], frame=10e6,
       panel=xypanel, pch=21, cex=0.3,
       col.hom="royalblue", fill.hom="royalblue",
       col.het="red", fill.het="red", xlab="Mb",
       ylab=expression(log[2]("copy number")))
##  (Note that the formula cn~x is required at this time)
##
## Or, plot each range in its own panel with a frame
##  of 2e6 bases.  (Again, the formula is a standard format
##  with cn, x, range, and id the only allowed terms) Because
##  these are all the ranges from one individual's chromosome,
##  the ranges are overlapping The range 'in focus' is
##  demarcated by vertical blue lines
xyplot(cn~x | range, oligoSet, range=hmmResults, frame=2e6,
       panel=xypanel,
       pch=21,
       cex=0.3,
       scales=list(x="free"),
       border="blue",
       col.hom="royalblue",
       col.het="salmon",
       col.np="grey",
       par.strip.text=list(cex=0.6),
       xlab="Mb",
       ylab=expression(log[2]("copy number")))
}
```

hmmSnpSet *Function for fitting a HMM to SnpSet containers*

Description

Function for fitting a HMM to SnpSet containers. This HMM uses only the genotypes to find regions of homozygosity. For copy number inference, see hmmBeadStudioSet and hmmOligoSnpSet.

Usage

```
hmmSnpSet(object, ICE = FALSE, chromosome = 1:22, normalIndex = 1L, rohIndex = normalIndex + 1L,
```

Arguments

object	A SnpSet.
ICE	Whether to use the genotype confidence scores when estimating the emission probabilities.
chromosome	Numeric vector indicating which chromosomes to fit for the HMM. See unique(chromosome(object for valid chromosomes.
normalIndex	Index for state with typical rate of heterozygosity.
rohIndex	Index for state with homozygous genotypes.
S	Integer indicating number of states (typically 2).
prGtHom	Numeric vector indicating the probability of a homozygous genotype for each of the hidden states. E.g., c(0.70, 0.99) for states corresponding to typical heterozygosity and homozygosity.
prGtMis	Numeric vector indicating the probability of a missing genotype for each hidden state. The default assumes that missing genotypes are equally probable in any of the hidden states.
prHetCalledHom	Numeric vector indicating the probability that a true heteroygous genotype is incorrectly called homozygous – one value for each hidden state.
prHetCalledHet	Numeric vector indicating the probability that a truly heterozygous genotype is correctly called heterozygous – one value for each hidden state.
prHomInNormal	The probability of a homozygous genotype in a region with typical heterozygosity.
prHomInRoh	The probability of a homozygous genotype in a region of homozygosity.
TAUP	scalar for defining transition probabilities. Larger values of TAUP discourage jumps between states.
...	Presently ignored

Value

A RangedData-derived class.

Author(s)

R. Scharpf

See Also

hmm, hmmBeadStudioSet, hmmOligoSnpSet

icePlatforms *List platforms for which ICE option is supported.*

Description

Lists platforms for which ICE option is supported.

Usage

```
icePlatforms()
```

Details

When procecessing genotypes with the **crlmm**, confidence scores for the diallelic genotype calls are available. One can estimate the emission probabilities for the crlmm diallelic genotypes using the confidence scores by setting the value of `ICE` to TRUE in the constructor for the `HmmOptionList` class. Currently, only certain platforms are supported for this option.

Value

A character vector of the annotation packages that are supported for the ICE option

Author(s)

R. Scharpf

References

Scharpf, RB et al., 2008, Annals of Applied Statistics

Examples

```
icePlatforms()
```

read.bsfiles *Read BeadStudio/GenomeStudio processed data.*

Description

Read BeadStudio/GenomeStudio processed data and return an array of log R ratios and B allele frequencies.

Usage

```
read.bsfiles(path = "", filenames, ext = "", row.names = 1, sep = "\t", lrr.colname = "Log.R.Rat
```

Arguments

path	character: path to plain text files containing BeadStudio processed data
filenames	character: name of file(s)
ext	character: filename extension
row.names	As in read.table. By default, the first column is assumed to be the feature identifiers.
sep	As in read.table.
lrr.colname	character: used to grep for the log R ratios in the header. E.g., `grep(lrr.colname, header)` should return a length 1 vector, where header is a vector of the column labels.
baf.colname	character: used to grep for the B allele frequency in the header. E.g., `grep(baf.colname, header)` return a length 1 vector, where header is a vector of the column labels.
drop	Logical: if TRUE, dimnames will not be returned
colClasses	Vector as in `read.table`. Note that if colClasses is not specified, the colClasses will be defined by reading in the first few rows. "NULL" will be assigned to all columns not containing B allele frequencies or log R ratios.
nrows	As in `read.table`.
...	Additional arguments passed to `read.table`.

Value

A 3 dimensional array: features x statistic (lrr or baf) x sample

Author(s)

R. Scharpf

See Also

`read.table`

Examples

```
path <- system.file("extdata", package="VanillaICE")
filename <- list.files(path, pattern="LRRandBAF", full.names=TRUE)
dat <- read.bsfiles(filenames=filename)
```

---

rescale *Rescale a numeric vector*

Description

Rescale a numeric vector

Usage

```
rescale(x, l, u)
```

Arguments

x	a numeric vector
l	numeric: lower limit of rescaled x.
u	numeric: upper limit of rescaled x.

Details

Not intended to be called directly, but used in packages that depend on **VanillaICE**

Value

numeric vector the same length as x with range [l, u].

Author(s)

R. Scharpf

---

robustSds *Calculate robust estimates of the standard deviation*

Description

Uses the median absolute deviation (MAD) to calculate robust estimates of the standard deviation

Usage

```
robustSds(x, takeLog = FALSE, ...)
```

Arguments

x	A matrix of copy number estimates. Rows are features, columns are samples.
takeLog	Whether to log-transform the copy number estimates before computing robust sds
...	additional arguments to rowMedians

Details

For matrices x with 4 or more samples, the row-wise MAD (SNP-specific sds) are scaled by sample MAD / median(sample MAD).

If the matrix has 3 or fewer samples, the MAD of the sample(s) is returned.

Value

Matrix of standard deviations.

Examples

```
data(locusLevelData, package="oligoClasses")
sds <- robustSds(locusLevelData[["copynumber"]]/100,
 takeLog=TRUE)
```

rowMAD — *Calculate the median absolute deviation for each row in a matrix.*

Description

Calculate the median absolute deviation for each row in a matrix.

Usage

```
rowMAD(x, y, ...)
```

Arguments

x	matrix
y	ignored
...	Addition arguments to function mad.

Value

A numeric vector of median absolute deviations.

Author(s)

R.Scharpf

See Also

mad

sd-methods — *Methods for estimating copy number standard deviations.*

Description

Estimate the standard deviation for CopyNumberSet and oligoSnpSet objects.

Usage

```
sd(x, na.rm=FALSE)
```

Arguments

x	A CopyNumberSet or oligoSnpSet
na.rm	Logical.

Details

The sd method for `CopyNumberSet` and `oligoSnpSet` objects retrieves the copy number confidence scores from the `cnConfidence` assay data element. The confidence matrix is a R x C matrix for an object with R features and C samples. Valid confidence estimates must be positive and not missing (not `NA`). If any elements in the confidence matrix are invalid, a robust estimate of the standard deviation is computed (described below). If all elements are valid, the standard deviation matrix is returned as 1 / confidence.

If any elements in the confidence matrix are invalid, the standard deviation for each marker and sample is calculated as follows. If autosomal markers are present, the standard deviation is estimated as the median absolute deviation across autosomal markers for each sample. This gives a vector of length C. The R x C standard deviation matrix is populated by row from the vector of length C (the standard deviation for each marker in a sample is given the same standard deviation). If autosomal markers are not present, the median absolute deviation across X-chromosome markers and Y-chromosome markers are estimated independently, providing to vectors of length C. The matrix of standard deviations for the X chromosome is populated by the C-length vector for the X-chromosome (by-row) and likewise for the Y chromosome.

Value

A matrix.

See Also

`mad`

Examples

```
library(oligoClasses)
data(oligoSetExample)
sds <- sd(oligoSet)
```

`Viterbi-methods` *Methods for Viterbi objects*

Description

Methods for Viterbi objects

Methods

In the following methods, `object` is of class `Viterbi` or `Viterbi2`.

`emission(object)`: Accessor for the emission probabilities.

`viterbi2Wrapper` *Wrapper function for fitting the viterbi algorithm*

Description

The viterbi algorithm, implemented in C, estimates the optimal state path as well as the forward and backward variables that are used for updating the mean and variances in a copy number HMM.

Usage

```
viterbi2Wrapper(r, b, gt, pos, is.snp, cnStates, chrom, prOutlierBAF = 0.001, p.hom = 0.05, TAUP
```

Arguments

`r`	`matrix` of copy number estimates.
`b`	`matrix` of B allele frequencies
`gt`	`matrix` of genotype calls (1=AA, 2=AB, 3=BB). Ignored unless `b` is missing.
`pos`	`integer` vector of genomic position along a chromosome.
`is.snp`	`indicator` for whether the marker is polymorphic. Must be the same length as the number of rows in `r` and `b`, and the same length as the vector `pos`.
`cnStates`	`numeric` vector for the initial copy number state means.
`chrom`	`integer`: the chromosome.
`prOutlierBAF`	`numeric`: initial probability for observing an outlier in the B allele frequencies.
`p.hom`	`numeric`: weight for observing homozygous genotypes. For value `0`, homozygous genotypes / B allele frequencies have the same emission probability in the 'normal' state as in the states hemizygous deletion and in copy-neutral region of homozygosity. Regions of homozygosity are common in normal genomes. For small values of `p.hom`, hemizygous deletions will only be called if the copy number estimates show evidence of a decrease from normal.
`TAUP`	`numeric`: scalar for the transition probability matrix. Larger values discourage transitions from the normal state.
`is.log`	`logical`: Whether the copy number estimates in the `r` matrix are on the log-scale.
`center`	`logical`: If TRUE, the copy number estimates for a chromosomal arm are re-centered such that the median value is the value specified for the mean of the normal copy number state.
`reestimation`	`logical`: if TRUE, the initial values provided for the mean of the copy number states will be reestimated as described previously (Rabiner, 1989) as a function of the forward and backward probabilities from the Viterbi algorithm and the mixture probabilities from the Normal-uniform mixture model for each state.
`limits`	`numeric` vector of length two specifying the range of the copy number estimates in `r`. Values of `r` outside of this range are truncated. See `copyNumberLimits`.
`initialProb`	`numeric` vector indicating the initial state probabilities for the hidden Markov model. The length of `initialProb` must be the same as the length of `cnStates`.

`normalIndex` `integer` specifying the index for the normal state. Note that states must be ordered by the mean of the copy number state. E.g., state 1 is homozygous deletion (0 copies), state 2 is hemizygous deletion (1 copy), normal (2 copies), ... In a 6-state HMM, normalIndex should be 3.

`rohIndex` `integer` specifying the index for copy-neutral region of homozygosity. In a 6-state HMM, the `rohIndex` should be 4.

`nupdates` `integer` specifying the maximum number of iterations for reestimating the mean and variance for each of the copy number states. The number of iterations may be fewer than `nupdates` if the difference in the log-likelihood between successive iterations is less than `tolerance`.

`tolerance` `numeric` value for indicating convergence of the log-likelihood. If the difference in the log-likelihood of the observed data given the HMM model at iteration i and i-1 is less than tolerance, no additional updates of model parameters using the EM algorithm is needed.

`returnViterbiObject`
`logical`: whether to return an object of class `Viterbi`. For internal use only.

`...` Additional arguments can be passed to the function `cnEmissionFromMatrix` and is currently for internal use only.

Details

This function is used by related packages extending **VanillaICE** and is not intended to be called directly by the user.

Value

A `RangedDataHMM` object if `returnViterbiObject` is FALSE.

Author(s)

R. Scharpf

Index