crlmm

November 11, 2009

R topics documented:

`celDates` *Extract dates from the cel file header*

Description

Extract dates from the cel file header.

Usage

```
celDates(celfiles)
```

Arguments

`celfiles`	CEL file names. Must specify the complete path.

Value

date-time class POSIXt

Author(s)

R. Scharpf

See Also

read.celfile.header, *POSIXt*

`cnrma` *Quantile normalizes the intensities for the nonpolymorphic probe to a HapMap reference distribution.*

Description

Quantile normalizes the intensities for the nonpolymorphic probe to a HapMap reference distribution.

Usage

```
cnrma(filenames, cdfName="genomewidesnp6", sns, seed = 1, verbose=FALSE)
```

Arguments

`filenames`	filenames with complete path
`cdfName`	Only 'genomewidesnp6' allowed
`sns`	the sample names. If missing, the basename of the filenames is used.
`seed`	seed for sampling the intensities to calculate skewness
`verbose`	logical

Value

A list. First element is the matrix of quantile-normalized intensities. The second element is the skew.

Note

Not tested

Author(s)

Rob Scharpf

Examples

```
library(genomewidesnp6Crlmm)
library(hapmapsnp6)
path <- system.file("celFiles", package="hapmapsnp6")
celFiles <- list.celfiles(path, full.names=TRUE)
cnrmaResult <- cnrma(celFiles, cdfName="genomewidesnp6")
```

`computeCopynumber` *Computes copy number*

Description

A function that transforms the quantile-normalized fluorescence intensities of the polymorphic and nonpolymorphic probes to a copy number scale.

Usage

```
computeCopynumber(chrom, A, B, calls, conf, NP, plate, MIN.OBS=1,
envir, P, DF.PRIOR = 50, CONF.THR = 0.99, bias.adj=FALSE,
priorProb, gender=NULL, SNR, SNRmin, seed=123, cdfName="genomewidesnp6",
verbose=TRUE, ...)
```

Arguments

`chrom`	Chromosome (an integer). Use 23 for X and 24 for Y.
`A`	The A allele intensities from `snprma`
`B`	The B allele intensities from `snprma`
`calls`	The genotype calls from `crlmm`
`conf`	The genotype confidence scores from `crlmm`
`NP`	The quantile normalized intensities of the nonpolymorphic probes
`plate`	The bach variable. Should be the same length as the number of columns in A
`MIN.OBS`	Integer: The minimum number of observations in a genotype cluster for which a SNP is deemed complete.
`envir`	An environment to save intermediate objects
`P`	Mainly for debugging a particular plate/batch.
`DF.PRIOR`	The degrees of freedom for the prior. Higher numbers will shrink the variance and correlation more.
`CONF.THR`	A threshold for the genotype confidence scores. Genotypes with scores below the threshold are ignored when computing SNP-specific within-genotype estimates of location and scale.
`bias.adj`	Logical: whether to adjust the location and scale parameters to account for biases due to common copy number variants. This is a SNP-specific adjustment. Parameters for background and slope must have already been estimated and available from the environment variable.
`priorProb`	Numerical vector of length 4. The prior probability of each copy number state (0, 1, 2, 3, and 4). The default is a uniform prior. Ignored if bias.adj=FALSE
`gender`	Gender of subjects. If not specified, we predict the gender from the X chromosome.
`SNR`	Signal to noise ratio from crlmm.
`SNRmin`	The minimum value for the SNR – we suggest 5. Samples with SNR below SNRmin are excluded.
`seed`	Seed used for random samples
`cdfName`	Annotation package
`verbose`	Logical: verbose output
`...`	Currently ignored

Details

Parameters for copy number are estimated using a linear model based on the diallelic genotype calls. No training data is used to estimate model parameters. Therefore, this function requires at least 10 samples to estimate copy number. For small sample sizes (e.g., 10 - 30 samples), this function will impute model parameters for a large number of loci and the precision of the estimates will be reduced.

Key assumption:

- we assume that the median copy number at any given locus is two for each batch. This assumption may not be appropriate for many datasets (e.g., a cancer dataset without normals processed in the same batch).

The developmental version of this package available from Bioconductor has many improvements to this function.

Value

All objects created by this function are stored in the environment passed to this function. In addition, each of the elements are specific to the chromosome(s) specified by the argument `chrom`. For instance the element `A` is the matrix of quantile-normalized intensities for the A-allele on chromosome(s) `chrom`. The element of this environment are as follows

`A`	Matrix of quantile-normalized intensities for the A-allele
`B`	Matrix of quantile-normalized intensities for the A-allele
`CA`	Copy number estimate for the A-allele (x 100)
`CB`	Copy number estimate for the B-allele (x 100)
`calls`	CRLMM genotype calls (AA=1, AB=2, BB=3)
`chrom`	Integer(s) indicating the chromosome(s)
`cnvs`	Names of the nonpolymorphic probes. These are the rownames of `NP` and `CT`.
`conf`	CRLMM confidence scores for the genotypes: 'round(-1000*log2(1-p))'
`corr`	Correlation of the A and B alleles for genotypes AB
`corrA.BB`	Correlation of A and B alleles for genotypes BB
`corrB.AA`	Correlation of A and B alleles for genotypes AA
`CT`	Copy number estimates for nonpolymorphic probe. See `cnvs` for the rownames.
`CT.sds`	Standard deviation estimates for `CT`
`npflags`	Flags for the nonpolymorphic probes.
`Ns`	The number of observations for each genotype/plate
`nuA`	Background/cross-hyb for the A allele (plate- and locus-specific)
`nuB`	Background/cross-hyb for the B allele (plate- and locus-specific)
`nuT`	Background for the nonpolymorphic probes (plate- and locus-specific)
`phiA`	Slope for the A allele (plate- and locus-specific)
`phiB`	Slope for the B allele (plate- and locus-specific)
`phiT`	Slope for the nonpolymorphic probes (plate- and locus-specific)
`plate`	Factor indicating batch (same length as number of cel files)
`sig2A`	Variance estimate for the A-allele signal (plate- and locus-specific)

sig2B	Variance estimate for the B-allele signal (plate- and locus-specific)
sig2T	Variance estimate for the nonpolymorphic signal (plate- and locus-specific)
snpflags	Flags for polymorphic probes
snps	Rownames for A, B, CA, CB, ...
sns	Sample names – the column names for A, B, ...
steps	Steps completed. For internal use.
tau2A	Variance estimate for the B-allele background/cross-hyb (plate- and locus-specific)
tau2B	Variance estimate for the B-allele background/cross-hyb (plate- and locus-specific)

Author(s)

Rob Scharpf

References

Nothing yet.

`crlmmIllumina` *Genotype Illumina Infinium II BeadChip data with CRLMM*

Description

This implementation of the CRLMM is especially designed for data from Illumina Infinium II BeadChips.

Usage

```
crlmmIllumina(RG, XY, stripNorm=TRUE, useTarget=TRUE,
      row.names=TRUE, col.names=TRUE,
      probs=c(1/3, 1/3, 1/3), DF=6, SNRMin=5,
      gender=NULL, seed=1, save.it=FALSE, load.it=FALSE,
      intensityFile, mixtureSampleSize=10^5,
      eps=0.1, verbose=TRUE, cdfName, sns, recallMin=10,
      recallRegMin=1000, returnParams=FALSE, badSNP=0.7)
```

Arguments

RG	NChannelSet containing R and G bead intensities
XY	NChannelSet containing X and Y bead intensities
stripNorm	'logical'. Should the data be strip-level normalized?
useTarget	'logical' (only used when stripNorm=TRUE). Should the reference HapMap intensities be used in strip-level normalization?
row.names	'logical'. Use rownames - SNP names?
col.names	'logical'. Use colnames - Sample names?
probs	'numeric' vector with priors for AA, AB and BB.
DF	'integer' with number of degrees of freedom to use with t-distribution.

SNRMin	'numeric' scalar defining the minimum SNR used to filter out samples.
gender	'integer' vector, with same length as 'filenames', defining sex. (1 - male; 2 - female)
seed	'integer' scalar for random number generator (used to sample mixtureSampleSize SNPs for mixture model.
save.it	'logical'. Save preprocessed data?
load.it	'logical'. Load preprocessed data to speed up analysis?
intensityFile	'character' with filename of preprocessed data to be saved/loaded.
mixtureSampleSize	'integer'. The number of SNP's to be used when fitting the mixture model.
eps	Minimum change for mixture model.
verbose	'logical'.
cdfName	'character' defining the chip annotation (manifest) to use ('human370v1c', human550v3b', 'human650v3a', 'human1mv1c', 'human370quadv3c', 'human610quadv1b', 'human660quadv1a' 'human1mduov3b')
sns	'character' vector with sample names to be used.
recallMin	'integer'. Minimum number of samples for recalibration.
recallRegMin	'integer'. Minimum number of SNP's for regression.
returnParams	'logical'. Return recalibrated parameters.
badSNP	'numeric'. Threshold to flag as bad SNP (affects batchQC)

Details

Note: The user should specify either the RG or XY intensities, not both. Alternatively if crlmmIllumina has been run already with save.it=TRUE, the preprocessed data can be loaded from file by specifying load.it=TRUE and intensityFile (RG or XY are not needed in this case).

Value

A SnpSet object which contains

calls	Genotype calls (1 - AA, 2 - AB, 3 - BB)
callProbability	confidence scores 'round(-1000*log2(1-p))'
SNPQC	SNP Quality Scores
batchQC	Batch Quality Scores

along with center and scale parameters when returnParams=TRUE in the featureData slot.

Author(s)

Matt Ritchie

References

Carvalho B, Bengtsson H, Speed TP, Irizarry RA. Exploration, normalization, and genotype calls of high-density oligonucleotide SNP array data. Biostatistics. 2007 Apr;8(2):485-99. Epub 2006 Dec 22. PMID: 17189563.

Carvalho B, Louis TA, Irizarry RA. Describing Uncertainty in Genome-wide Genotype Calling. (in prep)

Examples

```
## crlmmOut = crlmmIllumina(RG)
```

`crlmm-package` *Genotype Calling via CRLMM Algorithm*

Description

Faster implementation of CRLMM specific to SNP 5.0 and 6.0 arrays.

Details

Index:

```
crlmm-package           New implementation of the CRLMM Algorithm.
crlmm                   Genotype SNP 5.0 or 6.0 samples.
calls                   Accessor for genotype calls.
confs                   Accessor for confidences.
```

The 'crlmm' package reimplements the CRLMM algorithm present in the 'oligo' package. This implementation primes for efficient genotyping of samples on SNP 5.0 and SNP 6.0 Affymetrix arrays.

To use this package, the user must have additional data packages: 'genomewidesnp5Crlmm' - SNP 5.0 arrays 'genomewidesnp6Crlmm' - SNP 6.0 arrays

Author(s)

Rafael A Irizarry Maintainer: Benilton S Carvalho <bcarvalh@jhsph.edu>

References

Carvalho B, Louis TA, Irizarry RA. Describing Uncertainty in Genome-wide Genotype Calling. (in prep)

`crlmm` *Genotype oligonucleotide arrays with CRLMM*

Description

This is a faster and more efficient implementation of the CRLMM algorithm, especially designed for Affymetrix SNP 5 and 6 arrays (to be soon extended to other platforms).

Usage

```
crlmm(filenames, row.names=TRUE, col.names=TRUE,
      probs=c(1/3, 1/3, 1/3), DF=6, SNRMin=5,
      gender=NULL, save.it=FALSE, load.it=FALSE,
      intensityFile, mixtureSampleSize=10^5,
      eps=0.1, verbose=TRUE, cdfName, sns, recallMin=10,
      recallRegMin=1000, returnParams=FALSE, badSNP=0.7)
```

Arguments

filenames	'character' vector with CEL files to be genotyped.
row.names	'logical'. Use rownames - SNP names?
col.names	'logical'. Use colnames - Sample names?
probs	'numeric' vector with priors for AA, AB and BB.
DF	'integer' with number of degrees of freedom to use with t-distribution.
SNRMin	'numeric' scalar defining the minimum SNR used to filter out samples.
gender	'integer' vector, with same length as 'filenames', defining sex. (1 - male; 2 - female)
save.it	'logical'. Save preprocessed data?
load.it	'logical'. Load preprocessed data to speed up analysis?
intensityFile	'character' with filename to be saved/loaded - preprocessed data.
mixtureSampleSize	Number of SNP's to be used with the mixture model.
eps	Minimum change for mixture model.
verbose	'logical'.
cdfName	'character' defining the CDF name to use ('GenomeWideSnp5', 'GenomeWideSnp6')
sns	'character' vector with sample names to be used.
recallMin	Minimum number of samples for recalibration.
recallRegMin	Minimum number of SNP's for regression.
returnParams	'logical'. Return recalibrated parameters.
badSNP	'numeric'. Threshold to flag as bad SNP (affects batchQC)

Value

A SnpSet object.

calls	Genotype calls (1 - AA, 2 - AB, 3 - BB)
confs	Confidence scores 'round(-1000*log2(1-p))'
SNPQC	SNP Quality Scores
batchQC	Batch Quality Score
params	Recalibrated parameters

References

Carvalho B, Bengtsson H, Speed TP, Irizarry RA. Exploration, normalization, and genotype calls of high-density oligonucleotide SNP array data. Biostatistics. 2007 Apr;8(2):485-99. Epub 2006 Dec 22. PMID: 17189563.

Carvalho B, Louis TA, Irizarry RA. Describing Uncertainty in Genome-wide Genotype Calling. (in prep)

Examples

```
## this can be slow
if (require(genomewidesnp5Crlmm) & require(hapmapsnp5)){
  path <- system.file("celFiles", package="hapmapsnp5")

  ## the filenames with full path...
  ## very useful when genotyping samples not in the working directory
  cels <- list.celfiles(path, full.names=TRUE)
  (crlmmOutput <- crlmm(cels))
}
```

---

list.celfiles *List CEL files.*

Description

Function used to get a list of CEL files.

Usage

```
list.celfiles(...)
```

Arguments

...	Same arguments of list.files

Details

For the moment, this function returns only uncompressed CEL files (ie, no CEL.gz)

Value

Character vector with filenames.

Note

Quite often users want to use this function to pass filenames to other methods. In this situations, it is safer to use the argument 'full.names=TRUE'.

See Also

list.files

Examples

```
if (require(hapmapsnp5)){
  path <- system.file("celFiles", package="hapmapsnp5")

  ## only the filenames
  list.celfiles(path)

  ## the filenames with full path...
  ## very useful when genotyping samples not in the working directory
```

```
  list.celfiles(path, full.names=TRUE)
}else{
  ## this won't return anything
  ## if in the working directory there isn't any CEL
  list.celfiles(getwd())
}
```

`calls` *Accessors for Calls and Confidences on a SnpSet object*

Description

`calls` returns the genotype calls. CRLMM stores genotype calls as integers (1 - AA; 2 - AB; 3 - BB).

`confs` returns the confidences associated to the genotype calls. THe current implementation of CRLMM stores the confidences as integers by using the transformation:

conf = round(-1000*log2(1-p)),

where 'p' is the posterior probability of the call.

Usage

```
calls(x)
confs(x)
```

Arguments

`x`	SnpSet object

Value

Matrix of genotype calls or confidences.

Examples

```
set.seed(1)
theCalls <- matrix(sample(1:3, 20, rep=TRUE), nc=2)
p <- matrix(runif(20), nc=2)
theConfs <- round(-1000*log2(1-p))
obj <- new("SnpSet", call=theCalls, callProbability=theConfs)
calls(obj)
confs(obj)
```

`readIdatFiles` *Reads Idat Files from Infinium II Illumina BeadChips*

Description

Reads intensity information for each bead type from .idat files of Infinium II genotyping BeadChips

Usage

```
readIdatFiles(sampleSheet=NULL, arrayNames=NULL, ids=NULL, path=".",
              arrayInfoColNames=list(barcode="SentrixBarcode_A",
                                     position="SentrixPosition_A"),
              highDensity=FALSE, sep="_",
              fileExt=list(green="Grn.idat", red="Red.idat"),
              saveDate=FALSE)
```

Arguments

`sampleSheet`	`data.frame` containing Illumina sample sheet information (for required columns, refer to BeadStudio Genotyping guide - Appendix A).
`arrayNames`	character vector containing names of arrays to be read in. If `NULL`, all arrays that can be found in the specified working directory will be read in.
`ids`	vector containing ids of probes to be read in. If `NULL` all probes found on the first array are read in.
`path`	character string specifying the location of files to be read by the function
`arrayInfoColNames`	(used when `sampleSheet` is specified) list containing elements 'barcode' which indicates column names in the `sampleSheet` which contains the arrayNumber/barcode number and 'position' which indicates the strip number. In older style sample sheets, this information is combined (usually in a column named 'SentrixPosition') and this should be specified as `list(barcode=NULL, position="SentrixPosition")`
`highDensity`	logical (used when `sampleSheet` is specified). If `TRUE`, array extensions '_A', '_B' in sampleSheet are replaced with 'R01C01', 'R01C02' etc.
`sep`	character string specifying separator used in .idat file names.
`fileExt`	list containing elements 'Green' and 'Red' which specify the .idat file extension for the Cy3 and Cy5 channels.
`saveDate`	logical. Should the dates from each .idat be saved with sample information?

Details

The summarised Cy3 (G) and Cy5 (R) intensity, number of beads that were used in each channel and standard errors (all on the orginal scale) are read in from the .idat files.

Where available, a `sampleSheet` data.frame, in the same format as used by BeadStudio (columns 'Sample_ID', 'SentrixBarcode_A' and 'SentrixPosition_A' are required) which keeps track of sample information can be specified.

Thanks to Keith Baggerly who provided the code to read in the binary .idat files.

Value

NChannelSet with intensity data (R, G), number of beads (Rnb, Gnb) and standard errors (Rse, Gse) for each bead type.

Author(s)

Matt Ritchie

Examples

```
#RG = readIdatFiles()
```

---

snprma *Preprocessing tool for SNP arrays.*

Description

SNPRMA will preprocess SNP chips. The preprocessing consists of quantile normalization to a known target distribution and summarization to the SNP-Allele level.

Usage

```
snprma(filenames, mixtureSampleSize = 10^5, fitMixture = FALSE, eps = 0.1, verbo
```

Arguments

filenames	'character' vector with file names.
mixtureSampleSize	Sample size to be use when fitting the mixture model.
fitMixture	'logical'. Fit the mixture model?
eps	Stop criteria.
verbose	'logical'.
seed	Seed to be used when sampling.
cdfName	cdfName: 'GenomeWideSnp_5', 'GenomeWideSnp_6'
sns	Sample names.

Value

A	Summarized intensities for Allele A
B	Summarized intensities for Allele B
sns	Sample names
gns	SNP names
SNR	Signal-to-noise ratio
SKW	Skewness
mixtureParams	Parameters from mixture model
cdfName	Name of the CDF

Examples

```
if (require(genomewidesnp5Crlmm) & require(hapmapsnp5)){
  path <- system.file("celFiles", package="hapmapsnp5")

  ## the filenames with full path...
  ## very useful when genotyping samples not in the working directory
  cels <- list.celfiles(path, full.names=TRUE)
  snprmaOutput <- snprma(cels)
  snprmaOutput[["A"]][1:10,]
  snprmaOutput[["B"]][1:10,]
}
```

Index