Package 'XCIR'

October 14, 2021

Type Package

Title XCI-inference

Version 1.6.0

Author Renan Sauteraud, Dajiang Liu

Maintainer Renan Sauteraud <rxs575@psu.edu>

Description Models and tools for subject level analysis of X chromosome inactivation (XCI) and XCI-escape inference.

License GPL-2

LazyData TRUE

biocViews StatisticalMethod, RNASeq, Sequencing, Coverage

VignetteBuilder knitr

URL https://github.com/SRenan/XCIR

BugReports https://github.com/SRenan/XCIR/issues

Depends methods

Imports stats, utils, data.table, IRanges, VariantAnnotation, seqminer, ggplot2, biomaRt, readxl, S4Vectors

Suggests knitr, rmarkdown

RoxygenNote 6.1.1

git_url https://git.bioconductor.org/packages/XCIR

git_branch RELEASE_3_13

git_last_commit a1c5af3

git_last_commit_date 2021-05-19

Date/Publication 2021-10-14

R topics documented:

`XCIR-package` *Estimating inactivated X chromosome expression*

Description

Tools for the analysis of X chromosome inactivation (XCI) and XCI-escape inference.

Author(s)

Renan Sauteraud <rxs575@psu.edu>

`addAnno` *Read annotation file*

Description

Read a given annotation file and merge it with a data.table containing the relevant information to estimate inactivated X chromosome expression and filter out SNPs with low coverage.

Usage

```
addAnno(dt, seqm_annotate = TRUE, read_count_cutoff = 20,
  het_cutoff = 3, filter_pool_cutoff = 3, anno_file = NULL)
```

Arguments

dt	A data.table object.
seqm_annotate	A logical. If set to TRUE, the seqminer package will be used to annotate dt. If set to FALSE, this function is a simple read count filtering step.
read_count_cutoff	A numeric. Keep only SNPs that have at least that many reads.
het_cutoff	A numeric. Keep only SNPs that have at least that many reads on each allele.
filter_pool_cutoff	A numeric. Keep only SNPs that have at least that many reads on each allele across all samples. See details for more information.
anno_file	A character. The name of a file containing annotations.

Details

If the samples all have the same genotype (e.g: technical replicates), filter_pool_cutoff will sum counts across samples and preserve SNPs that pass the cutoff on both the reference and alternate alleles. This may lead to samples with 0 counts on either allele but will prevent removing heterozygous sites with lower coverage (especialliy in skewed samples). seqm_anno will call annotatePlain from the seqminer package. For convenience, seqminer's necessary annotation sources can be copied into XCIR's extdata folder. See ?annotatePlain for more information.

Value

A data.table object that contains allelic coverage, genotype and annotations at the covered SNPs.

See Also

annotatePlain

Examples

```
# Example workflow for documentation

vcff <- system.file("extdata/AD_example.vcf", package = "XCIR")
# Reading functions
vcf <- readRNASNPs(vcff)
vcf <- readVCF4(vcff)

# Annotation functions
# Using seqminer (requires additional annotation files)

anno <- addAnno(vcf)

# Using biomaRt
anno <- annotateX(vcf)
# Do not remove SNPs with 0 count on minor allele
anno0 <- annotateX(vcf, het_cutoff = 0)

# Summarise read counts per gene
```

```
# Assuming data is phased, reads can be summed across genes.
genic <- getGenicDP(anno, highest_expr = FALSE)
# Unphased data, select SNP with highest overall expression.
genic <- getGenicDP(anno, highest_expr = TRUE)
```

---

annotateX *Annotate*

Description

Map positions of SNPs to genes extracted from biomaRt

Usage

```
annotateX(xciObj, read_count_cutoff = 20, het_cutoff = 3,
  release = "hg19", verbose = FALSE)
```

Arguments

xciObj	A data.table. The data to be annotated must contain at least the 4 columns 'GENE', 'POS', 'AD_hap1', 'AD_hap2'. Additional columns will be preserved.
read_count_cutoff	A numeric. Keep only SNPs that have at least that many reads.
het_cutoff	A numeric. Keep only SNPs that have at least that many reads on each allele.
release	A character. Genome release name. Valid releases are "hg19", "hg38".
verbose	A logical. If set to TRUE, print additional information.

Value

A data.table. The input table annotated with gene symbols and filtered for read counts.

Examples

```
# Example workflow for documentation

vcff <- system.file("extdata/AD_example.vcf", package = "XCIR")
# Reading functions
vcf <- readRNASNPs(vcff)
vcf <- readVCF4(vcff)

# Annotation functions
# Using seqminer (requires additional annotation files)

anno <- addAnno(vcf)

# Using biomaRt
anno <- annotateX(vcf)
# Do not remove SNPs with 0 count on minor allele
```

```
anno0 <- annotateX(vcf, het_cutoff = 0)

# Summarise read counts per gene
# Assuming data is phased, reads can be summed across genes.
genic <- getGenicDP(anno, highest_expr = FALSE)
# Unphased data, select SNP with highest overall expression.
genic <- getGenicDP(anno, highest_expr = TRUE)
```

`betaBinomXI` *Fit mixture model*

Description

Fit a mixture model to estimate mosaicism and XCI-escape.

Usage

```
betaBinomXI(genic_dt, model = "AUTO", plot = FALSE, hist = FALSE,
  flag = 0, xciGenes = NULL, a0 = NULL, optimizer = c("nlminb",
  "optim"), method = NULL, limits = TRUE, debug = FALSE)
```

Arguments

`genic_dt`	A `data.table`. The table as outputted by `getGenicDP`.
`model`	A `character` indicating which model to use to estimate the mosaicism. Valid choices are "AUTO", "BB", "MM", "MM2", "MM3". See details.
`plot`	A `logical`. If set to TRUE, information about the training set and the skewing estimate will be plotted.
`hist`	A `logical`. If set to TRUE, an histogram of the skewing estimates will be displayed.
`flag`	A `numeric`. Specify how to handle convergence issues. See details.
`xciGenes`	A `character` or NULL. To be passed to `readXCI` to select the training set of inactivated genes.
`a0`	A `numeric` or NULL. Starting values for the optimization. This should not be used with more than one model as different models have different parameters. Leave NULL unless you know what you're doing.
`optimizer`	A `character`. The optimization function to use for minimization of the log-likelihood. Should be one of "nlminb" or "optim".
`method`	A `character`. The method to be passed to `optim` when it is the selected `optimizer`.
`limits`	A `logical`. If set to TRUE, the optimization will be constrained. Using upper bounds on the probability of sequencing error and escape in the training set ensures that the dominant mixture represents the skewing for inactivated genes.
`debug`	A `logical`. If set to TRUE, information about each iteration will be printed (Useful to identify problematic samples).

Details

The model determines the number of components used in the mixture model. By default, "AUTO" tries all combinations of mixtures and the best estimate is kept using backward selection based on AIC. BB is a simple beta-binomial. MM adds a binomial component to model the sequencing errors. MM2 jointly models the probability of misclasification in the training set. MM3 include all 3 components.

Flags in the output reports issues in convergence. If `flag` is set to 0, nothing is done. If set to 1, the model selection will avoid flagged models (will favor parcimonious models). If set to 2, calls for which the best selected model had convergence issue will be removed.

Value

A `data.table` with an entry per sample and per gene.

See Also

getGenicDP readXCI

Examples

```
library(data.table)
# Simulated data
dtf <- system.file("extdata/data2_vignette.tsv", package = "XCIR")
dt <- fread(dtf)
xcigf <- system.file("extdata/xcig_vignette.txt", package = "XCIR")
xcig <- readLines(xcigf)
# Run all models on the data
all <- betaBinomXI(dt, xciGenes = xcig)
# Simple BetaBinomial model and show histogram of skewing
bb <- betaBinomXI(dt, xciGenes = xcig, model = "BB", hist = TRUE)

# Plotting fits
stoshow <- paste0("sample", c(31, 33, 35, 40)) #interesting samples
plotQC(all[sample %in% stoshow], xcig = xcig)

# Summarizing results
# Sample information
samps <- sample_clean(all)
# Gene-level predictions
xcistates <- getXCIstate(all)
```

`betaParam` *Converting beta distribution parameters*

Description

Convert parameter values between different beta distribution parametrization

Usage

```
betaParam(alpha = NULL, beta = NULL, m = NULL, theta = NULL,
  mu = NULL, sigma2 = NULL)
```

Arguments

alpha	A numeric. First shape parameter
beta	A numeric.Second shape parameter
m	A numeric. Mode
theta	A numeric. Concentration
mu	A numeric. Mean
sigma2	A numeric. Variance

Details

This function needs two parameters that caracterise the beta distribution (alpha and beta, mode and concentration or mean and variance) and returns all parametrizations.

Value

A list with all equivalent formulations of the distribution.

Examples

```
betaParam(alpha = 5, beta = 5)
betaParam(m = 0.5, theta = 10)
betaParam(mu = 0.5, sigma2 = 0.02272727)
```

---

consensusXCI *XCI consensus*

Description

Read consensus & XCIR calls for all X-linked genes

Usage

```
consensusXCI(redownload = FALSE, simple = TRUE)
```

Arguments

redownload	A logical. If set to TRUE, the original supplementary file is redownloaded from PMC.
simple	A logical. If set to TRUE, minimal information is returned, only for genes with an available XCIR classification.

Details

The consensus is as published in Supplementary table S1 of Balaton et al. (Biol Sex Differ. 2015). doi: 10.1186/s13293-015-0053-7

Value

A `data.table` with the annotated X-linked genes.

Examples

```
consensusXCI(simple = TRUE)
```

`getGenicDP` *Get expression at the gene level*

Description

Calculate allele specific expression for each gene in each sample, either using only the most expressed SNP or using all SNPs (when phasing has been performed).

Usage

```
getGenicDP(dt_anno, highest_expr = TRUE, pool = FALSE,
  gender_file = NULL)
```

Arguments

`dt_anno`	A `data.table`. An annotated table of read counts for each SNP, as outputted by `addAnno`
`highest_expr`	A `logical`. If FALSE, all SNPs will be summed within each gene. This should only be set to FALSE when high quality phasing information is available. If set to TRUE, the highest expressed SNP (across both alleles) will be used instead.
`pool`	A `logical`. Only works when `highest_expr` is set to TRUE. If set to TRUE, the read counts are pooled accross all samples for each SNP. Only use this if the samples come from the same subject
`gender_file`	A `character` or NULL. Leave NULL if `dt_anno` already contains a gender column. The file must contain at least a "sample" and "gender" column with samples matching the samples in `dt_anno`.

Value

A `data.table`. That should be used as input for `betaBinomXI`.

See Also

betaBinomXI, addAnno

Examples

```
# Example workflow for documentation

vcff <- system.file("extdata/AD_example.vcf", package = "XCIR")
# Reading functions
vcf <- readRNASNPs(vcff)
vcf <- readVCF4(vcff)

# Annotation functions
# Using seqminer (requires additional annotation files)

anno <- addAnno(vcf)

# Using biomaRt
anno <- annotateX(vcf)
# Do not remove SNPs with 0 count on minor allele
anno0 <- annotateX(vcf, het_cutoff = 0)

# Summarise read counts per gene
# Assuming data is phased, reads can be summed across genes.
genic <- getGenicDP(anno, highest_expr = FALSE)
# Unphased data, select SNP with highest overall expression.
genic <- getGenicDP(anno, highest_expr = TRUE)
```

---

getXCIstate *Classify X-genes*

Description

Classify X-linked genes between Escape (E), Variable Escape (VE) and Silenced (S)

Usage

```
getXCIstate(xciObj)
```

Arguments

xciObj	A data.table. The table returned by betaBinomXI

Value

A data.table with genes and their XCI-state.

Examples

```
library(data.table)
# Simulated data
dtf <- system.file("extdata/data2_vignette.tsv", package = "XCIR")
dt <- fread(dtf)
```

```
xcigf <- system.file("extdata/xcig_vignette.txt", package = "XCIR")
xcig <- readLines(xcigf)
# Run all models on the data
all <- betaBinomXI(dt, xciGenes = xcig)
# Simple BetaBinomial model and show histogram of skewing
bb <- betaBinomXI(dt, xciGenes = xcig, model = "BB", hist = TRUE)

# Plotting fits
stoshow <- paste0("sample", c(31, 33, 35, 40)) #interesting samples
plotQC(all[sample %in% stoshow], xcig = xcig)

# Summarizing results
# Sample information
samps <- sample_clean(all)
# Gene-level predictions
xcistates <- getXCIstate(all)
```

`mart_genes` *biomaRt genes*

Description

Extract gene informations from biomaRt

Usage

```
mart_genes(release = "hg19", chr = "X")
```

Arguments

`release`	A `character`. Genome release name. Valid releases are "hg19", "hg38".
`chr`	A `character` or `NULL`. If specified, only the genes from the specified chromosomes will be returned.

Value

A `data.table` with the gene symbol, start and end position and matching ensembl transcripts.

Examples

```
#Chromosome X, hg19
egX <- mart_genes()
#Full genome, latest release
eg <- mart_genes("hg38")
```

plotBBCellFrac *Plot cell fraction estimates*

Description

Plot cell fraction estimates from list of known XCI genes

Usage

```
plotBBCellFrac(xci_dt, xcig = NULL, gene_names = "",
  color_col = NULL, xist = TRUE)
```

Arguments

xci_dt	A data.table. The data to be used for the estimate of skewing (i.e: limited to XCI genes).
xcig	A logical. If xci_dt was not subset for training genes only, setting xcig to TRUE will filter the data.
gene_names	A character. If left blank, only genes that are further than 20 to "all", all genes will be named. Set to "none" to remove all annotations. Alternately, a character vector can be passed to annotate specific genes of interest.
color_col	A character. One of the columns of xci_dt can be used to color genes.
xist	A logical. Set to TRUE to display XIST in addition to the training genes.

Details

This function is mostly used in betaBinomXI to ensure that the cell fraction is estimated properly. However, it can be used from the output of betaBinomXI to troubleshoot estimation issues.

Value

The plot object in class ggplot.

plotQC *Plot QC*

Description

This plot shows QC for skewing estimates

Usage

```
plotQC(xci_table, xcig = NULL, gene_names = "")
```

Arguments

xci_table	A data.table. Data to plot. Should be the results of betaBinomXI, getGenicDP or one of the annotation functions.
xcig	A character vector. The names of the genes in the inactivated training set.
gene_names	A character. If left blank, only genes that are further than 20 to "all", all genes will be named. Set to "none" to remove all annotations. Alternately, a character vector can be passed to annotate specific genes of interest.

Value

An invisible plot object.

Examples

```
library(data.table)
# Simulated data
dtf <- system.file("extdata/data2_vignette.tsv", package = "XCIR")
dt <- fread(dtf)
xcigf <- system.file("extdata/xcig_vignette.txt", package = "XCIR")
xcig <- readLines(xcigf)
# Run all models on the data
all <- betaBinomXI(dt, xciGenes = xcig)
# Simple BetaBinomial model and show histogram of skewing
bb <- betaBinomXI(dt, xciGenes = xcig, model = "BB", hist = TRUE)

# Plotting fits
stoshow <- paste0("sample", c(31, 33, 35, 40)) #interesting samples
plotQC(all[sample %in% stoshow], xcig = xcig)

# Summarizing results
# Sample information
samps <- sample_clean(all)
# Gene-level predictions
xcistates <- getXCIstate(all)
```

---

readRNASNPs *Read SNPs from RNA-Seq*

Description

Read SNPs from RNA-Seq that have not been phased.

Usage

```
readRNASNPs(vcf_file)
```

Arguments

vcf_file	A character. The path to a vcf file.

Details

For phased samples, use `readXVcf`.

Value

A `data.table` of allele specific read counts.

Examples

```
# Example workflow for documentation

vcff <- system.file("extdata/AD_example.vcf", package = "XCIR")
# Reading functions
vcf <- readRNASNPs(vcff)
vcf <- readVCF4(vcff)

# Annotation functions
# Using seqminer (requires additional annotation files)

anno <- addAnno(vcf)

# Using biomaRt
anno <- annotateX(vcf)
# Do not remove SNPs with 0 count on minor allele
anno0 <- annotateX(vcf, het_cutoff = 0)

# Summarise read counts per gene
# Assuming data is phased, reads can be summed across genes.
genic <- getGenicDP(anno, highest_expr = FALSE)
# Unphased data, select SNP with highest overall expression.
genic <- getGenicDP(anno, highest_expr = TRUE)
```

---

readVCF4 *Read VCF file*

Description

Read ASE from a VCF file

Usage

```
readVCF4(vcf_file)
```

Arguments

`vcf_file`	A `character`. The path to a vcf file. The file must have the REF, ALT and AD fields.

Value

A data.table of allele specific read counts.

Examples

```
# Example workflow for documentation

vcff <- system.file("extdata/AD_example.vcf", package = "XCIR")
# Reading functions
vcf <- readRNASNPs(vcff)
vcf <- readVCF4(vcff)

# Annotation functions
# Using seqminer (requires additional annotation files)

anno <- addAnno(vcf)

# Using biomaRt
anno <- annotateX(vcf)
# Do not remove SNPs with 0 count on minor allele
anno0 <- annotateX(vcf, het_cutoff = 0)

# Summarise read counts per gene
# Assuming data is phased, reads can be summed across genes.
genic <- getGenicDP(anno, highest_expr = FALSE)
# Unphased data, select SNP with highest overall expression.
genic <- getGenicDP(anno, highest_expr = TRUE)
```

---

readXCI *Read a list of known inactivated genes*

Description

Read a list of gene symbols of known inactivated genes to be used as training set in betaBinomXI.

Usage

```
readXCI(xciGenes = NULL)
```

Arguments

xciGenes	A character or codeNULL. By defaults, return a vector of 177 genes. Other available choices include "cotton" and "intersect". If a file path is given, the genes will be read from the file.

Details

Both gene lists are extracted from Cotton et al. Genome Biology (2013). doi:10.1186/gb-2013-14-11-r122. By default, the function returns a list that was used as training set in the paper. This training set was generated as the intersection of the silenced genes identified by both expression (Carrel & Willard, 2005) and DNA methylation analysis (Cotton et al, 2011). Setting it to "cotton" will instead return a list of 294 genes that were classified as inactivated by Cotton et al. "intersect" is the most stringent list which returns the intersection of training and predicted set.

Value

A `character` vector of gene names.

See Also

`betaBinomXI`

Examples

```
xcig <- readXCI()
xcig <- readXCI("cotton")
```

`sample_clean` *Sample estimates*

Description

Return sample specific information from XCIR results

Usage

```
sample_clean(bb_table)
```

Arguments

`bb_table`	A `data.table`. The table returned by `betaBinomXI`.

Value

A `data.table` with one entry per sample and information regarding skewing and model fitting.

Examples

```
library(data.table)
# Simulated data
dtf <- system.file("extdata/data2_vignette.tsv", package = "XCIR")
dt <- fread(dtf)
xcigf <- system.file("extdata/xcig_vignette.txt", package = "XCIR")
xcig <- readLines(xcigf)
# Run all models on the data
all <- betaBinomXI(dt, xciGenes = xcig)
# Simple BetaBinomial model and show histogram of skewing
bb <- betaBinomXI(dt, xciGenes = xcig, model = "BB", hist = TRUE)

# Plotting fits
stoshow <- paste0("sample", c(31, 33, 35, 40)) #interesting samples
plotQC(all[sample %in% stoshow], xcig = xcig)

# Summarizing results
# Sample information
samps <- sample_clean(all)
# Gene-level predictions
xcistates <- getXCIstate(all)
```

Index