Package 'deepSNV'

March 25, 2024

Maintainer Moritz Gerstung <moritz.gerstung@ebi.ac.uk>

License GPL-3

Title Detection of subclonal SNVs in deep sequencing data.

biocViews GeneticVariability, SNP, Sequencing, Genetics, DataImport

LinkingTo Rhtslib (>= 1.13.1)

Type Package

LazyLoad yes

Description This package provides provides quantitative variant callers for detecting subclonal mutations in ultra-deep (>=100x coverage) sequencing experiments. The deepSNV algorithm is used for a comparative setup with a control experiment of the same loci and uses a beta-binomial model and a likelihood ratio test to discriminate sequencing errors and subclonal SNVs. The shearwater algorithm computes a Bayes classifier based on a beta-binomial model for variant calling with multiple samples for precisely estimating model parameters - such as local error rates and dispersion - and prior knowledge, e.g. from variation data bases such as COSMIC.

Version 1.48.0

Depends R (>= 2.13.0), methods, graphics, parallel, IRanges, GenomicRanges, SummarizedExperiment, Biostrings, VGAM, VariantAnnotation (>= 1.27.6),

Imports Rhtslib

Suggests RColorBrewer, knitr, rmarkdown

VignetteBuilder knitr

SystemRequirements GNU make

RoxygenNote 7.0.2

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

git_branch RELEASE_3_18

git_last_commit 38615a9

git_last_commit_date 2023-10-24

Repository Bioconductor 3.18

Date/Publication 2024-03-25

Author Niko Beerenwinkel [ths],
Raul Alcantara [ctb],
David Jones [ctb],
John Marshall [ctb],
Inigo Martincorena [ctb],
Moritz Gerstung [aut, cre]

R topics documented:

---

deepSNV-package *Detection of subclonal SNVs in deep sequencing experiments*

Description

Detection of subclonal SNVs in deep sequencing experiments

Details

This packages provides algorithms for detecting subclonal single nucleotide variants (SNVs) and their frequencies from ultra-deep sequencing data. It retrieves the nucleotide counts at each position and each strand from two .bam files and tests for differences between the two experiments with a likelihood ratio test using either a binomial or and overdispersed beta-binomial model. The statistic can be tuned across genomic sites by a shared Dirichlet prior and there package provides procedures for normalizing sequencing data from different runs.

Author(s)

Moritz Gerstung, Wellcome Trust Sanger Institute, <moritz.gerstung@sanger.ac.uk>

References

Gerstung M, Beisel C, Rechsteiner M, Wild P, Schraml P, Moch H, and Beerenwinkel N. Reliable detection of subclonal single-nucleotide variants in tumour cell populations. Nat Commun 3:811 (2012). DOI:10.1038/ncomms1814.

See Also

deepSNV

Examples

```
## Short example with 2 SNVs at frequency ~10%
regions <- data.frame(chr="B.FR.83.HXB2_LAI_IIIB_BRU_K034", start = 3120, stop=3140)
ex <- deepSNV(test = system.file("extdata", "test.bam", package="deepSNV"), control = system.file("extdata", "cont
show(ex)   # show method
plot(ex)   # scatter plot
summary(ex)   # summary with significant SNVs
ex[1:3,]   # subsetting the first three genomic positions
tail(test(ex, total=TRUE))   # retrieve the test counts on both strands
tail(control(ex, total=TRUE))

## Not run: Full example with ~ 100 SNVs. Requires an internet connection, but try yourself.
# regions <- data.frame(chr="B.FR.83.HXB2_LAI_IIIB_BRU_K034", start = 2074, stop=3585)
# HIVmix <- deepSNV(test = "http://www.bsse.ethz.ch/cbg/software/deepSNV/data/test.bam", control = "http://www.bs
data(HIVmix) # attach data instead..
show(HIVmix)
plot(HIVmix)
```

```
head(summary(HIVmix))
```

---

bam2R *Read nucleotide counts from a .bam file*

Description

This function uses a C interface to read the nucleotide counts on each position of a .bam alignment. The counts of both strands are reported separately and nucleotides below a quality cutoff are masked. It is called by deepSNV to parse the alignments of the test and control experiments, respectively.

Usage

```
bam2R(
  file,
  chr,
  start,
  stop,
  q = 25,
  mq = 0,
  s = 2,
  head.clip = 0,
  max.depth = 1e+06,
  verbose = FALSE,
  mask = 0,
  keepflag = 0,
  max.mismatches = NULL
)
```

Arguments

file	The name of the .bam file as a string.
chr	The chromosome as a string.
start	The start position (1-indexed).
stop	The end position (1-indexed).
q	An optional cutoff for the nucleotide Phred quality. Default q = 25. Nucleotides with Q < q will be masked by 'N'.
mq	An optional cutoff for the read mapping quality. Default mq = 0 (no filter). reads with MQ < mq will be discarded.
s	Optional choice of the strand. Defaults to s = 2 (both).
head.clip	Should n nucleotides from the head of reads be clipped? Default 0.
max.depth	The maximal depth for the pileup command. Default 1,000,000.

verbose	Boolean. Set to TRUE if you want to get additional output.
mask	Integer indicating which flags to filter. Default 0 (no mask). Try 3844 (UNMAP|SECONDARY|QCFAIL|DUP|SUPPLEMENTARY).
keepflag	Integer indicating which flags to keep. Default 0 (no mask). Try 3 (PAIRED|PROPERLY_PAIRED).
max.mismatches	Integer indicating maximum NM value to allow in a read. Default NULL (no filter).

Value

A named matrix with rows corresponding to genomic positions and columns for the nucleotide counts (A, T, C, G, -), masked nucleotides (N), (INS)ertions, (DEL)etions, (HEAD)s and (TAIL)s that count how often a read begins and ends at the given position, respectively, and the sum of alignment (QUAL)ities, which can be indicative of alignment problems. Counts from matches on the reference strand (s=0) are uppercase, counts on the complement (s=1) are lowercase. The returned matrix has 11 * 2 (strands) = 22 columns and (stop - start + 1) rows.

Author(s)

Moritz Gerstung

Examples

```
## Simple example:
counts <- bam2R(file = system.file("extdata", "test.bam", package="deepSNV"), chr="B.FR.83.HXB2_LAI_IIIB_BRU_K034
show(counts)
## Not run: Requires an internet connection, but try yourself.
# bam <- bam2R(file = "http://www.bsse.ethz.ch/cbg/software/deepSNV/data/test.bam", chr="B.FR.83.HXB2_LAI_IIIB_B
# head(bam)
```

---

bbb *Bayesian beta-binomal test, codename shearwater*

Description

This is the workhorse of the shearwater test. It computes the Bayes factor for each sample, nucleotide and position of the null-model vs. the alternative of a real variant.

Usage

```
bbb(
  counts,
  rho = NULL,
  alternative = "greater",
  truncate = 0.1,
  rho.min = 1e-04,
  rho.max = 0.1,
  pseudo = .Machine$double.eps,
```

```
  return.value = c("BF", "P0", "err"),
  model = c("OR", "AND", "adaptive"),
  min.cov = NULL,
  max.odds = 10,
  mu.min = 1e-06,
  mu.max = 1 - mu.min
)
```

Arguments

counts	An array of nucleotide counts (samples x positions x 10 nucleotides in forward and reverse orientation), typically from loadAllData
rho	Disperision factor. If NULL, estimated from the data.
alternative	The alternative. Currently only "greater" is implemented.
truncate	The model uses a compound control sample which is the sum of all samples with a relative nucleotide frequency below truncate at this locus. Default = 0.1.
rho.min	Lower bound for the method of moment estimate of the dispersion factor rho.
rho.max	Upper bound for the method of moment estimate of the dispersion factor rho.
pseudo	A pseudo count to be added to the counts to avoid problems with zeros.
return.value	Return value. Either "BF" for Bayes Factor of "P0" for the posterior probability (assuming a prior of 0.5).
model	The null model to use. For "OR" it requires the alternative model to be violated on either of the strands, for "AND" the null is specified such that the error rates of the sample of interest and the compound control sample are identical on both strands. "AND" typically yield many more calls. The most recent addition is "adaptive", which switches from "OR" to "AND", if the coverage is less than min.cov, or if the odds of forward and reverse coverage is greater than max.odds. Default = "OR".
min.cov	Minimal coverage to swith from OR to AND, if model is "adaptive"
max.odds	Maximal odds before switching from OR to AND if model is "adaptive" and min.cov=NULL.
mu.min	Minimum of the error rate mu.
mu.max	Maximal error rate mu.

Value

An array of Bayes factors

Note

Experimental code, subject to changes

Author(s)

mg14

Examples

```
## Load data from deepSNV example
regions <- GRanges("B.FR.83.HXB2_LAI_IIIB_BRU_K034", IRanges(start = 3120, end=3140))
files <- c(system.file("extdata", "test.bam", package="deepSNV"), system.file("extdata", "control.bam", package='
counts <- loadAllData(files, regions, q=10)

## Run (bbb) computes the Bayes factor
bf <- bbb(counts, model = "OR", rho=1e-4)
vcf <- bf2Vcf(bf, counts, regions, samples = files, prior = 0.5, mvcf = TRUE)

## Compare to deepSNV
bf <- bbb(counts, model = "AND", rho=1e-4)
dpSNV <- deepSNV(test = files[1], control = files[2], regions=regions, q=10)
plot(p.val(dpSNV), bf[1,,]/(1+bf[1,,]), log="xy")
```

---

betabinLRT *ShearwaterML*

Description

Maximum likelihood version of Shearwater producing p-values instead of Bayes factors.

Usage

```
betabinLRT(
  counts,
  rho = NULL,
  truncate = 0.05,
  rho.min = 1e-04,
  rho.max = 0.8,
  maxvaf = 0.3,
  mindepth = 10,
  maxtruncate = 0.5
)
```

Arguments

counts	The array of counts typically generated by loadAllData.
rho	Use this variable to fix the dispersion parameter to a value of interest. Default: NULL, rho will be estimated from the data.
truncate	Samples with variant allele frequencies higher than "truncate" will be excluded from the background error model.
rho.min	If rho=NULL, rho will be estimated from the data in the interval [rho.min,rho.max].
rho.max	If rho=NULL, rho will be estimated from the data in the interval [rho.min,rho.max].
maxvaf	Sites with an average rate of mimatches higher than maxvaf will not be considered (e.g. SNPs or reference sites).

`mindepth`	Minimum coverage required to test a site.
`maxtruncate`	Maximum number of samples that can be excluded from the background error model by truncate for a site to be tested.

Value

A list with two arrays for P- and Q-values.

Author(s)

Inigo Martincorena and Moritz Gerstung

References

Martincorena I, Roshan A, Gerstung M, et al. (2015). High burden and pervasive positive selection of somatic mutations in normal human skin. _Science_ (Under consideration).

Examples

```
# code to be added
```

`bf2Vcf` *Function to create a* `VCF` *object with variant calls from an array of Bayes factors.*

Description

This function thresholds the Bayes factors computed by the shearwater algorithm and creates a `VCF` object as output.

Usage

```
bf2Vcf(
  BF,
  counts,
  regions,
  samples = 1:nrow(counts),
  err = NULL,
  mu = NULL,
  cutoff = 0.05,
  prior = 0.5,
  mvcf = TRUE
)
```

Arguments

BF	array of Bayes factors from bbb.
counts	array of counts from loadAllData.
regions	GRanges with the regions corresponding to counts and BF.
samples	vector of samples names.
err	Optional matrix of error rates, otherwise recomputed from counts.
mu	Optional matrix of relative frequencies, otherwise recomputed from counts.
cutoff	Cutoff for the posterior artifact probability below which a variant is considered to be true (default = 0.05)
prior	matrix of prior probabilities for finding a true call, typically from makePrior. Alternatively a single fixed number.
mvcf	boolean flag, if TRUE compute a large VCF with as many genotype columns as samples. Default TRUE. Otherwise use duplicate rows and only one genotype column. The sample is then provided by the info:PD field. Can be inefficient for large sample sizes.

Value

A VCF object

Note

Experimental code, subject to changes

Author(s)

mg14

---

consensusSequence *Calculate the consensus sequence.*

Description

This function computes the consensus sequence from a matrix of nucleotide counts, or the control slot of a deepSNV object.

Usage

```
consensusSequence(x, ...)

## S4 method for signature 'matrix'
consensusSequence(x, vector=FALSE, haploid=TRUE, het.cut = .333)

## S4 method for signature 'deepSNV'
consensusSequence(x, vector=FALSE, haploid=TRUE, het.cut = .333)
```

Arguments

x	An object. Either an deepSNV-class object, or a named matrix with nucleotide counts.
...	Additional arguments passed to methods.
vector	Boolean where TRUE indicates that a character vector should be returned.
haploid	Should the consensus be called for a haploid control? Otherwise, also all bases larger than het.cut are rerported. Default haploid = TRUE.
het.cut	Heterozygous cutoff. If haploid = FALSE, report all nucleotides with relative frequency larger than het.cut. Default = 0.333.

Value

A DNAString with the consensus sequence, or if vector = TRUE, a character vector.

Author(s)

Moritz Gerstung

Examples

```
data(HIVmix)
seq = consensusSequence(HIVmix)
consensusSequence(HIVmix, vector=TRUE)[1:10]
```

---

control *Get control counts*

Description

Convenience function to obtain the control counts from a deepSNV object.

Usage

```
control(deepSNV, ...)

## S4 method for signature 'deepSNV'
control(deepSNV, total = FALSE)
```

Arguments

deepSNV	a deepSNV-class object
...	Additional param passed to specific methods
total	Logical. If true the sum of both strands is returned

Value

A matrix with the absolute frequencies summed over both strands.

Examples

```
data(HIVmix)
control(HIVmix)[1:10,]
control(HIVmix, total=TRUE)[1:10,]
```

---

coordinates *Get coordinates*

Description

Convenience function to get the coordinates from a deepSNV object.

Usage

```
coordinates(deepSNV, ...)

## S4 method for signature 'deepSNV'
coordinates(deepSNV)
```

Arguments

deepSNV	a deepSNV-class object
...	Additional param passed to specific methods

Value

A data.frame with columns "chrom(osome)" and "pos(ition)".

Examples

```
data(HIVmix)
coordinates(HIVmix)[1:10,]
```

---

counts *Example count table*

Description

A table with counts of the HIVmix data set. Used for minimal unit testing.

Examples

```
data("counts", package="deepSNV")
countsFromBam <- bam2R(file = system.file("extdata", "test.bam", package="deepSNV"), chr="B.FR.83.HXB2_LAI_IIIB_E
all(counts == countsFromBam)
```

dbetabinom *Beta-binomial probability distribution*

Description

Beta-binomial probability distribution

Usage

```
dbetabinom(x, n, mu, rho, log = FALSE)
```

Arguments

x	Counts
n	Size
mu	Probability
rho	Dispersion. rho in (0,1)
log	Return logarithmic values

Value

d

Author(s)

mg14

deepSNV *Test two matched deep sequencing experiments for low-frequency SNVs.*

Description

This generic function can handle different types of inputs for the test and control experiments. It either reads from two .bam files, uses two matrices of nucleotide counts, or re-evaluates the test results from a deepSNV-class object. The actual test is a likelihood ratio test of a (beta-)binomial model for the individual nucleotide counts on each position under the hypothesis that both experiments share the same parameter, and the alternative that the parameters differ. Because the difference in degrees of freedom is 1, the test statistic $D = -2 \log \max L_0 / \max L_1$ is asymptotically distributed as χ^2_1. The statistic may be tuned by a nucleotide specific Dirichlet prior that is learned across all genomic sites, see estimateDirichlet. If the model is beta-binomial, a global dispersion parameter is used for all sites. It can be learned with estimateDispersion.

Usage

```
deepSNV(test, control, ...)

## S4 method for signature 'matrix,matrix'
deepSNV(test,control, alternative = c('greater', 'less', 'two.sided'), dirichlet.prior = NULL, pseudo.

## S4 method for signature 'deepSNV,missing'
deepSNV(test, control, ...)

## S4 method for signature 'character,character'
deepSNV(test, control, regions, q=25, s=2, head.clip=0, ...)

## S4 method for signature 'matrix,character'
deepSNV(test, control, regions, q=25, s=2, ...)

## S4 method for signature 'character,matrix'
deepSNV(test, control, regions, q=25, s=2, ...)
```

Arguments

test	The test experiment. Either a .bam file, or a matrix with nucleotide counts, or a deepSNV-class object.
control	The control experiment. Must be of the same type as test, or missing if test is a deepSNV-class object.
...	Additional arguments.
alternative	The alternative to be tested. One of greater, less, or two.sided.
dirichlet.prior	A base-sepecific Dirichlet prior specified as a matrix. Default NULL.
pseudo.count	If dirichlet.prior=NULL, a pseudocount can be used to define a flat prior.
combine.method	The method to combine p-values. One of "fisher" (default), "max", or "average". See p.combine for details.
over.dispersion	A numeric factor for the over.dispersion, if the model is beta-binomial. Default 100.
model	Which model to use. Either "bin", or "betabin". Default "bin".
regions	The regions to be parsed if test and control are .bam files. Either a data.frame with columns "chr" (chromosome), "start", "stop", or a GRanges object. If multiple regions are specified, the appropriate slots of the returned object are concatenated by row.
q	The quality arguement passed to bam2R if the experiments are .bam files.
s	The strand argument passed to bam2R if the experiments are .bam files.
head.clip	The head.clip argument passed to bam2R if the experiments are .bam files.

Value

A deepSNV object

Author(s)

Moritz Gerstung

Examples

```
## Short example with 2 SNVs at frequency ~10%
regions <- data.frame(chr="B.FR.83.HXB2_LAI_IIIB_BRU_K034", start = 3120, stop=3140)
ex <- deepSNV(test = system.file("extdata", "test.bam", package="deepSNV"), control = system.file("extdata", "cont
show(ex)   # show method
plot(ex)   # scatter plot
summary(ex)   # summary with significant SNVs
ex[1:3,]   # subsetting the first three genomic positions
tail(test(ex, total=TRUE))   # retrieve the test counts on both strands
tail(control(ex, total=TRUE))

## Not run: Full example with ~ 100 SNVs. Requires an internet connection, but try yourself.
# regions <- data.frame(chr="B.FR.83.HXB2_LAI_IIIB_BRU_K034", start = 2074, stop=3585)
# HIVmix <- deepSNV(test = "http://www.bsse.ethz.ch/cbg/software/deepSNV/data/test.bam", control = "http://www.bs
data(HIVmix) # attach data instead..
show(HIVmix)
plot(HIVmix)
head(summary(HIVmix))
```

---

deepSNV-class *deepSNV class.*

Description

This class stores the contents of the deepSNV test. It is typically initialized with deepSNV. This class has the following slots:

p.val The P-values of the test.

test A matrix with the nucleotide counts in the test experiment. The column names of the nucleotide counts are A, T, C, G, - for the positivie strand and a, t, c, g, _ for the reverse.

control A matrix with the nucleotide counts in the control experiment. The column names must be the same as for the test.

coordinates A data.frame with the genomic coordinates chr and pos, and other columns, if desired.

dirichlet.prior A matrix with the nucleotide-specific Dirichlet prior

pseudo.count The pseudo count if used)

alternative A string with the alternative used in the test.

nucleotides A character vector with the nucleotides tested.

regions A data.frame with columns chr, start, and stop.

files A list with two entries test and control storing the filenames (if the object was initialized from two bam-files).

combine.method The method for combining p-values as a character string.

model The statistical model, either bin for binomial, or betabin for beta-binomial

over.dispersion If the model is beta-binomial, the first parameter for the beta-binomial model, which is shared across sites.

call The last function call to deepSNV.

log.lik The log likelihood of the data under the null hypothesis. (Excluding zeros on the opposite site under a one-sided test.)

Author(s)

Moritz Gerstung

See Also

deepSNV

Examples

```
## Short example with 2 SNVs at frequency ~10%
regions <- data.frame(chr="B.FR.83.HXB2_LAI_IIIB_BRU_K034", start = 3120, stop=3140)
ex <- deepSNV(test = system.file("extdata", "test.bam", package="deepSNV"), control = system.file("extdata", "cont
show(ex)   # show method
plot(ex)   # scatter plot
summary(ex)   # summary with significant SNVs
ex[1:3,]   # subsetting the first three genomic positions
tail(test(ex, total=TRUE))   # retrieve the test counts on both strands
tail(control(ex, total=TRUE))

## Not run: Full example with ~ 100 SNVs. Requires an internet connection, but try yourself.
# regions <- data.frame(chr="B.FR.83.HXB2_LAI_IIIB_BRU_K034", start = 2074, stop=3585)
# HIVmix <- deepSNV(test = "http://www.bsse.ethz.ch/cbg/software/deepSNV/data/test.bam", control = "http://www.bs
data(HIVmix) # attach data instead..
show(HIVmix)
plot(HIVmix)
head(summary(HIVmix))
```

---

estimateDirichlet *Learn a base-specific Dirichlet prior.*

Description

The prior learns the parameters of a Dirichlet distribution seperately for each consensus base. The expected value of the Dirichlet distributions is the base-substitution matrix, where rows correspond to the initial nucleotide and columns to the substituted nucleotide. The absolute values determine the higher moments of the Dirichlet distributions. After having learned the prior the deepSNV-class test is recomputed.

Usage

```
estimateDirichlet(control)

## S4 method for signature 'matrix'
estimateDirichlet(control)

## S4 method for signature 'deepSNV'
estimateDirichlet(control)
```

Arguments

control	Either a matrix with nucleotide counts or a deepSNV-class object.

Value

An deepSNV-class object.

Author(s)

Moritz Gerstung

Examples

```
data(phiX)
estimateDirichlet(phiX)
```

---

estimateDispersion *Estimate the Dispersion factor in a beta-binomial model.*

Description

This function estimates the dispersion factor in a beta-binomial model of the nucleotide counts. This model assumes that the count for nucleotide j at position i is distributed after a beta-binomial $X_{i,j} \sim \mathrm{BB}(n_i; \alpha, \beta_{ij})$, where n_i is the coverage. The base and nucleotide specific parameter β_{ij} is estimated from the local mean by the method-of-moments estimate, α is a shared overdispersion parameter. It is estimated via a numerical optimization of the likelihood under the null-hypothesis.

Usage

```
estimateDispersion(test, control, ...)

## S4 method for signature 'deepSNV,missing'
estimateDispersion(test, control, alternative = NULL, interval = c(0,1000))

## S4 method for signature 'matrix,matrix'
estimateDispersion(test, control, alternative = NULL, interval = c(0,1000))
```

Arguments

test	Either a deepSNV object, or a matrix with the test counts.
control	Missing if test is a deepSNV object, otherwise missing.
...	Additional param passed to specific methods
alternative	The alternative to be tested. One of "greater", "less", "two-sided" (default). If test is a deepSNV object, automatically taken from the corresponding slot if unspecified.
interval	The interval to be screened for the overdispersion factor. Default (0,1000).

Value

A deepSNV-class object if the input was a deepSNV object. Otherwise the loglikelihood and the estimated parameter.

Author(s)

Moritz Gerstung

Examples

```
data("RCC", package="deepSNV")
plot(RCC)
summary(RCC)[,1:6]
RCC.bb = estimateDispersion(RCC, alternative = "two.sided")
summary(RCC.bb)
```

---

estimateRho *Helper function for estimating the dispersion factor rho*

Description

It uses a method of moments approximation to estimate rho from the variances of the relative frequencies nu across samples

Usage

```
estimateRho(x, mu, ix, pseudo.rho = .Machine$double.eps)
```

Arguments

x	counts
mu	relative frequency across all samples
ix	index indicating the set of samples to use (typically indicating those with relative frequency smaller than 0.1).
pseudo.rho	a pseudo count added to each sample to avoid problems with zeros. Default = .Machine$double.eps

Value

rho

Note

Experimental code, subject to changes

Author(s)

mg14

Extract *Subsetting for deepSNV objects.*

Description

Subsetting for deepSNV objects.

Usage

```
## S4 method for signature 'deepSNV,ANY,ANY,ANY'
x[i, j]
```

Arguments

x	A deepSNV-class object.
i	Row indeces.
j	Column (nucleotide) indeces.

Value

A deepSNV-class object.

Author(s)

Moritz Gerstung

Examples

```
data(HIVmix)
HIVmix[1:10,]
```

loadAllData — *Function to load all data from a list of bam files*

Description

This function uses the parallel package and the bam2R interface to load all nucleotide counts from a list of bam files and a set of regions into a large array.

Usage

```
loadAllData(files, regions, ..., mc.cores = 1)
```

Arguments

files	A character vector with the paths to all bam files
regions	Either a GRanges or data.frame with the coordinates of interest
...	Arguments passed to bam2R
mc.cores	Number of cores used for loading, default = 1

Value

counts

Note

Experimental code, subject to changes

Author(s)

mg14

makePrior — *Compute a prior from a COSMIC VCF object*

Description

This function computes the prior probability of detecting a true variant from a variation data base. It assumes a VCF file with a CNT slot for the count of a given base substitution. Such a VCF file can be downloaded at ftp://ngs.sanger.ac.uk/production/cosmic/. The prior probability is simply defined as pi.mut * CNT[i]/sum(CNT). On sites with no count, a background probability of pi0 is used.

Usage

```
makePrior(COSMIC, regions, pi.gene = 0.1, pi.backgr = 1e-04)
```

Arguments

COSMIC	A VCF object from COSMIC VCF export.
regions	A GRanges object with the regions (gene) of interest.
pi.gene	Probability that a gene is mutated
pi.backgr	Background probability of a locus being mutated. Default 1e-4, corresponding to an expected value of 1 SNV per 1e4 bases.

Value

A vector of prior values with length given by the length of the regions GRanges object.

Note

Experimental code, subject to changes

Author(s)

mg14

Examples

```
## Make prior (not run)
#COSMIC <- readVcf("PATHTO/CosmicCodingMuts_v64_02042013_noLimit.vcf.gz", genome="GChr37")
#prior <- makePrior(COSMIC[info(COSMIC)$GENE=="TP53"], regions=GRanges(17, IRanges(7571720,7578811)))
#plot(prior[,1], type="h")
```

---

manhattanPlot *Manhattan plot.*

Description

This functions performs a Manhattan plot of the p-values of a deepSNV test against the position

Usage

```
manhattanPlot(x, col = nt.col)
```

Arguments

x	An deepSNV object.
col	An optional vector of colors for the nucleotides.

Value

NULL.

Author(s)

Moritz Gerstung

Examples

```
data(HIVmix)
manhattanPlot(HIVmix)
```

---

mcChunk *Little helper function to split the count objects into a smaller digestible chunks and run function FUN on each subset*

Description

Little helper function to split the count objects into a smaller digestible chunks and run function FUN on each subset

Usage

```
mcChunk(FUN, X, split = 250, mc.cores = 1, ...)
```

Arguments

FUN	The function to call on each chunk
X	The object to be subsetted using [,i,]
split	The size of each chunk
mc.cores	The number of cores to use
...	Additional arguments passed to FUN

Value

The value of FUN

Note

Experimental code, subject to changes

Author(s)

mg14

normalize *Normalize nucleotide counts.*

Description

This functions performs a loess normalization of the nucleotide. This experimental feature can be used to compare experiments from different libraries or sequencing runs that may have differing noise characteristics.

Usage

```
normalize(test, control, ...)

## S4 method for signature 'matrix,matrix'
normalize(test, control, round=TRUE, ...)

## S4 method for signature 'deepSNV,missing'
normalize(test, control,  ...)
```

Arguments

test	Either an deepSNV-class object or a named matrix with nucleotide counts.
control	Missing if test is an link{deepSNV-class} object, otherwise a matrix with nucleotide counts.
...	Parameters passed to loess.
round	Logical. Should normalized counts be rounded to integers? Default=TRUE

Value

A deepSNV-class object.

Note

This feature is somewhat experimental and the results should be treated with care. Sometimes it can be better to leave the data unnormalized and use a model with greater dispersion instead.

Author(s)

Moritz Gerstung

Examples

```
data(phiX, package = "deepSNV")
plot(phiX)
phiN <- normalize(phiX, round = TRUE)
plot(phiN)
```

p.combine *Combine two p-values*

Description

This function combines two P-values into a single one using a statistic defined by method. "fisher" uses the product of the two, in this case the logarithm of the product is χ^2_4 distributed. If the method = "max", the resulting P-value is $\max\{P_1, P_2\}^2$. For method = "average" the mean is used, yielding a P-value of $2x^2$ if $x = (P_1 + P_2)/2 < .5$ and $1 - 2x^2$ otherwise. "negfisher" is the negative of Fisher's method using $1-F(1-P_1, 1-P_2)$, where F is the combination function of Fisher's method; for small P_1,P_2, the result is very similar to method="average". Fisher's method behaves a bit like a logical AND of the joint null-hypothesis, whereas negative Fisher is like an OR.

Usage

```
p.combine(p1, p2, method = c("fisher", "max", "average", "prod", "negfisher"))
```

Arguments

p1	P-value 1
p2	P-value 2
method	One of "fisher" (default), "max" or "average"

Value

p-values

Author(s)

Moritz Gerstung

Examples

```
p1 <- runif(1000)
p2 <- runif(1000)
hist(p1)
p.avg = p.combine(p1,p2, method="average")
hist(p.avg)
p.fish = p.combine(p1,p2, method="fisher")
hist(p.fish)
p.max = p.combine(p1,p2, method="max")
hist(p.max)
pairs(data.frame(p1,p2,p.fish,p.max,p.avg))
```

p.val *Get p-values*

Description

Convenience function to get the p-values from a deepSNV object.

Usage

```
p.val(deepSNV, ...)

## S4 method for signature 'deepSNV'
p.val(deepSNV)
```

Arguments

deepSNV	a deepSNV-class object
...	Additional param passed to specific methods

Value

A matrix with the p-values.

Examples

```
data(HIVmix)
p.val(HIVmix)[1:10,]
```

pbetabinom *Cumulative beta-binomial probability distribution*

Description

Cumulative beta-binomial probability distribution

Usage

```
pbetabinom(x, n, mu, rho, log = FALSE)
```

Arguments

x	Counts
n	Sample size
mu	Probability
rho	Dispersion. rho in (0,1)
log	Return logarithmic values

Value

Probability

Author(s)

mg14

---

phiX *Example phiX data*

Description

Data from two phiX experiments sequenced on a GAIIx.

Examples

```
data(phiX, package="deepSNV")
plot(phiX)
phiN <- normalize(phiX, round=TRUE)
plot(phiN)
```

---

pi *Example prior*

Description

Prior from COSMIC v63 for the TP53 gene

Examples

```
data("pi", package="deepSNV")
plot(pi[,1], type="h")
```

`plot.deepSNV` *Scatter plot of relative nucleotide frequencies.*

Description

This function plots the relative nucleotide frequencies of the test against the control experiment on a logarithmit scale. The color of the symbols denotes the nucleotide, and the area of the circle is proportional to the $-\log$ of the p-value.

Usage

```
## S3 method for class 'deepSNV'
plot(
  x,
  sig.level = NULL,
  col = NULL,
  col.null = "grey",
  cex.min = 0.2,
  ylab = "Relative Frequency in Test",
  xlab = "Relative Frequency in Control",
  pch = 16,
  ...
)
```

Arguments

`x`	A deep SNV object.
`sig.level`	By default, p-values below sig.level are drawn as filled circles.
`col`	Color of the nucleotides.
`col.null`	Color of insignificant nucleotides.
`cex.min`	The minimal size of the points.
`ylab`	The y-axis label.
`xlab`	The x-axis label.
`pch`	The plotting symbol. Default = 16 (filled circle)
`...`	Additional arguments passed to plot.

Author(s)

Moritz Gerstung

Examples

```
## Short example with 2 SNVs at frequency ~10%
regions <- data.frame(chr="B.FR.83.HXB2_LAI_IIIB_BRU_K034", start = 3120, stop=3140)
ex <- deepSNV(test = system.file("extdata", "test.bam", package="deepSNV"), control = system.file("extdata", "cont
show(ex)   # show method
plot(ex)   # scatter plot
summary(ex)   # summary with significant SNVs
ex[1:3,]   # subsetting the first three genomic positions
tail(test(ex, total=TRUE))   # retrieve the test counts on both strands
tail(control(ex, total=TRUE))

## Not run: Full example with ~ 100 SNVs. Requires an internet connection, but try yourself.
# regions <- data.frame(chr="B.FR.83.HXB2_LAI_IIIB_BRU_K034", start = 2074, stop=3585)
# HIVmix <- deepSNV(test = "http://www.bsse.ethz.ch/cbg/software/deepSNV/data/test.bam", control = "http://www.bs
data(HIVmix) # attach data instead..
show(HIVmix)
plot(HIVmix)
head(summary(HIVmix))
```

`qvals2Vcf` *Function to create a VCF object with variant calls from an array of q-values.*

Description

This function thresholds the q-values computed by the shearwater algorithm and creates a VCF object as output.

Usage

```
qvals2Vcf(
  qvals,
  counts,
  regions,
  samples = 1:nrow(counts),
  err = NULL,
  mu = NULL,
  cutoff = 0.05,
  mvcf = TRUE
)
```

Arguments

qvals	array of q-values from betabinLRT.
counts	array of counts from loadAllData.
regions	GRanges with the regions corresponding to counts and qvals.

samples	vector of samples names.
err	Optional matrix of error rates, otherwise recomputed from counts.
mu	Optional matrix of relative frequencies, otherwise recomputed from counts.
cutoff	Cutoff for the q-values below which a variant is considered to be true (default = 0.05)
mvcf	boolean flag, if TRUE compute a large VCF with as many genotype columns as samples. Default TRUE. Otherwise use duplicate rows and only one genotype column. The sample is then provided by the info:PD field. Can be inefficient for large sample sizes.

Value

A VCF object

Note

Experimental code, subject to changes

Author(s)

mg14

RCC *Example RCC data*

Description

Deep sequencing experiments of a renal cell carcinoma and healthy control tissue.

Examples

```
data("RCC", package="deepSNV")
summary(RCC, adjust.method="bonferroni")[,1:6]
plot(RCC)
RCC.bb <- estimateDispersion(RCC, alternative="two.sided")
summary(RCC.bb, adjust.method="bonferroni")[,1:6]
plot(RCC.bb)
```

repeatMask *Mask homopolymeric repeats.*

Description

This function masks homopolymeric repeats longer than a given width. These are hot-spots of sequencing error and can confound the analysis.

Usage

```
repeatMask(x, ...)

## S4 method for signature 'DNAString'
repeatMask(x, w=5, flank=TRUE)

## S4 method for signature 'deepSNV'
repeatMask(x, w=5, flank=TRUE)
```

Arguments

x	An object. Either a deepSNV-class object or a DNAString with the nucleotide sequence.
...	Additional param passed to specific methods
w	Integer. The minimal length at which repeats should be masked. Default w=0.
flank	Boolean. Indicates whether the sites adjacent to the repeat should also be masked.

Value

A boolean vector where TRUE indicates a non-homopolymeric region.

Author(s)

Moritz Gerstung

Examples

```
data(HIVmix)
which(repeatMask(HIVmix))
```

RF *Relative frequencies.*

Description

Convenience function to compute the relative frequencies from a matrix with absolute counts.

Usage

```
RF(freq, total = FALSE)
```

Arguments

freq	A matrix with nucleotide counts.
total	If the nucleotide counts have columns for forward and reverse direction, return each strand sepratatelu (FALSE), or add the two (TRUE).

Value

A matrix with the relative frequencies.

Author(s)

Moritz Gerstung

Examples

```
data(HIVmix)
RF(test(HIVmix))[1:10,]
RF(test(HIVmix), total=TRUE)[1:10,]
```

show,deepSNV-method *Show method for deepSNV objects*

Description

Show method for deepSNV objects

Usage

```
## S4 method for signature 'deepSNV'
show(object)
```

Arguments

object	A deepSNV-class object.

Author(s)

Moritz Gerstung

Examples

```
## Short example with 2 SNVs at frequency ~10%
regions <- data.frame(chr="B.FR.83.HXB2_LAI_IIIB_BRU_K034", start = 3120, stop=3140)
ex <- deepSNV(test = system.file("extdata", "test.bam", package="deepSNV"), control = system.file("extdata", "cont
show(ex)   # show method
plot(ex)   # scatter plot
summary(ex)   # summary with significant SNVs
ex[1:3,]   # subsetting the first three genomic positions
tail(test(ex, total=TRUE))   # retrieve the test counts on both strands
tail(control(ex, total=TRUE))

## Not run: Full example with ~ 100 SNVs. Requires an internet connection, but try yourself.
# regions <- data.frame(chr="B.FR.83.HXB2_LAI_IIIB_BRU_K034", start = 2074, stop=3585)
# HIVmix <- deepSNV(test = "http://www.bsse.ethz.ch/cbg/software/deepSNV/data/test.bam", control = "http://www.bs
data(HIVmix) # attach data instead..
show(HIVmix)
plot(HIVmix)
head(summary(HIVmix))
```

---

summary *Summary of a deepSNV object*

Description

Tabularize significant SNVs by evalutating the p-values of the deepSNV test.

Usage

```
## S4 method for signature 'deepSNV'
summary(
  object,
  sig.level = 0.05,
  adjust.method = "bonferroni",
  fold.change = 1,
  value = c("data.frame", "VCF")
)
```

Arguments

object	A deepSNV-class object.
sig.level	The desired significance level.
adjust.method	The adjustment method for multiple testing corrections. See p.adjust for details. Set to NULL, for no adjustment. Default "bonferroni".

fold.change	The minimal fold change required of the relative frequency. Default 1.
value	String. The type of the returned object. Either "data.frame" for a data.frame (default) or "VCF" for an ExtendedVCF-class object.

Value

If value="data.frame", a data.frame with the following columns:

chr	The chromosome
pos	The position (1-based)
ref	The reference (consensus) nucleotide
var	The variant nucleotide
p.val	The (corrected) p-value
freq.var	The relative frequency of the SNV
sigma2.freq.var	The estimated variance of the frequency
n.tst.fw	The variant counts in the test experiment, forward strand
cov.tst.fw	The coverage in the test experiment, forward strand
n.tst.bw	The variant counts in the test experiment, backward strand
cov.tst.bw	The coverage in the test experiment, backward strand
n.ctrl.fw	The variant counts in the control experiment, forward strand
cov.ctrl.fw	The coverage in the control experiment, forward strand
n.ctrl.bw	The variant counts in the control experiment, backward strand
cov.ctrl.bw	The coverage in the control experiment, backward strand
raw.p.val	The raw p-value

If value = "VCF", this functions returns a VCF-class object with the following entries: FIXED:

REF	Reference allele in control sample. Note that deletions in the control sample will be reported like insertions, e.g. if the consensus of the control is A,- at positions 1 and 2 (relative to the reference) and the test was A,A, then this would be denoted as REF="A" and VAR="AA" with coordinate IRanges(1,2). This may cause ambiguities when the VCF object is written to text with writeVcf(), which discards the width of the coordinate, and this variant remains indistinguishable from an insertion to the _reference_ genome.
VAR	Variant allele in test sample
QUAL	-10*log10(raw.p.val)

INFO:

VF	Variant frequency. Variant allele frequency in the test minus variant allele frequency in the control.
VFV	Variant frequency variance. Variance of the variant frequency; can be thought of as confidence interval.

GENO (one column for test and one column for control):

FW	Forward allele count
BW	Backward allele count
DFW	Forward read depth
DBW	Backward read depth

Author(s)

Moritz Gerstung

Examples

```
## Short example with 2 SNVs at frequency ~10%
regions <- data.frame(chr="B.FR.83.HXB2_LAI_IIIB_BRU_K034", start = 3120, stop=3140)
ex <- deepSNV(test = system.file("extdata", "test.bam", package="deepSNV"), control = system.file("extdata", "cont
show(ex)   # show method
plot(ex)   # scatter plot
summary(ex)   # summary with significant SNVs
ex[1:3,]   # subsetting the first three genomic positions
tail(test(ex, total=TRUE))   # retrieve the test counts on both strands
tail(control(ex, total=TRUE))

## Not run: Full example with ~ 100 SNVs. Requires an internet connection, but try yourself.
# regions <- data.frame(chr="B.FR.83.HXB2_LAI_IIIB_BRU_K034", start = 2074, stop=3585)
# HIVmix <- deepSNV(test = "http://www.bsse.ethz.ch/cbg/software/deepSNV/data/test.bam", control = "http://www.bs
data(HIVmix) # attach data instead..
show(HIVmix)
plot(HIVmix)
head(summary(HIVmix))
```

---

test *Get test counts*

Description

Convenience function to obtain the test counts from a deepSNV object.

Usage

```
test(deepSNV, ...)

## S4 method for signature 'deepSNV'
test(deepSNV, total = FALSE)
```

Arguments

deepSNV	a deepSNV-class object
...	Additional param passed to specific methods
total	Logical. If true the sum of both strands is returned

Value

A matrix with the absolute frequencies summed over both strands.

Examples

```
data(HIVmix)
test(HIVmix)[1:10,]
test(HIVmix, total=TRUE)[1:10,]
```

---

trueSNVs *Example .bam data and true SNVs.*

Description

Two .bam alignments as example data sets are downloaded remotely via http. Sequenced were a 1,512 nt fragment of the HIV genome and a mixture (90% + 10%) with another variants. The two sequences were confirmed by Sanger sequencing and stored in the table trueSNVs.

Examples

```
data(HIVmix)
data(trueSNVs)
table(p.adjust(p.val(HIVmix), method="BH") < 0.05, trueSNVs)
```

Index