Package: tripsAndDipR (via r-universe)

September 10, 2024

Type Package

Title Inference of Ploidy from Sequencing Data

Version 0.2.0

Description Uses read counts for biallelic single nucleotide polymorphisms (SNPs) to infer ploidy. It allows parameters to be specified to account for sequencing error rates and allelic bias. For details of the algorithms, please see Delomas (2019) <doi:10.1111/1755-0998.13073> and Delomas et al. (2021) <doi:10.1111/1755-0998.13431>.

Imports stats, Rcpp (>= 1.0.2)

SystemRequirements C++11

LinkingTo Rcpp

URL https://github.com/delomast/tripsAndDipR

BugReports https://github.com/delomast/tripsAndDipR/issues

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.1.1

Suggests knitr, rmarkdown

VignetteBuilder knitr

Repository https://delomast.r-universe.dev

RemoteUrl https://github.com/delomast/tripsanddipr

RemoteRef HEAD

RemoteSha 291d186426b6fbd1adf6623811241bf3c63c664d

Contents

`funkyPloid` *calculate LLR's for a group of samples and a given set of ploidy values*

Description

calculate LLR's for a group of samples and a given set of ploidy values

Usage

```
funkyPloid(
  counts,
  counts_alt = NULL,
  ploidy,
  h = NULL,
  eps = NULL,
  maxIter = 10000,
  maxDiff = 1e-04,
  model = c("BB_noise", "BB", "Bin"),
  maxSubIter = 500,
  IC = FALSE
)
```

Arguments

`counts`	A numeric matrix with each row corresponding to a different sample. There are two options for formatting the input. Either the columns correspond to the read counts for each locus, in a two column per locus format: column 1 is the read counts for locus1ReferenceAllele, column two is the read counts for locus1AlternateAllele2, locus2Reference, locus2Alternate, ... OR this contains read counts for the reference allele, and `counts_alt` contains read counts for the alternate allele The rownames should be the sample names.
`counts_alt`	Either `NULL` or a numeric matrix with each row corresponding to a different sample. The matrix contains counts for the alternate allele, with samples and loci having the same order as in `counts` If this parameter is `NULL`, `counts` is assumed to have both the reference and alternate allele counts.
`ploidy`	A numeric vector containing the ploidies (as integers) to test.
`h`	Either `NULL`, or a numeric vector of h values for each locus in the same order that the loci are ordered in counts. These h values are as defined by Gerard et al. (2018) "Genotyping polyploids from messy sequencing data" Genetics 210:789-807. with h expressed as alternate / reference. These values can be estimated using the R package "updog". If `NULL`, h values of 1 (unbiased) are used for all loci.
`eps`	Either `NULL`, or a numeric vector of values for the error rate per read for each locus in the same order that the loci are ordered in counts. These are expressed as proportions, so a rate of 1% should be given as 0.01. These values can be

	estimated using the R package "updog". If NULL, error rates of .01 are assumed for all loci.
maxIter	The maximum number of iterations of the EM algorithm to run for a given sample
maxDiff	This is the maximum change in log-likelihood from the previous iteration to accept as convergence and stop the EM algorithm.
model	the model to fit: BB_noise - mixture of beta-binomials WITH uniform noise, BB - mixture of beta-binomials WITHOUT uniform noise, Bin - mixture of binomials
maxSubIter	If model is BB_noise or BB, this is the maximum number of iterations of to perform during the M-step of the EM algorithm.
IC	TRUE to calculate AIC and BIC.

---

genoProps *Fit the model for one ploidy*

Description

For each sample, this returns the log-likelihood and proportions of each genotype category for a given ploidy value

Usage

```
genoProps(
  counts,
  counts_alt = NULL,
  ploidy,
  h = NULL,
  eps = NULL,
  maxIter = 10000,
  maxDiff = 1e-04,
  model = c("BB_noise", "BB", "Bin"),
  maxSubIter = 500,
  IC = FALSE
)
```

Arguments

counts	A numeric matrix with each row corresponding to a different sample. There are two options for formatting the input. Either the columns correspond to the read counts for each locus, in a two column per locus format: column 1 is the read counts for locus1ReferenceAllele, column two is the read counts for locus1AlternateAllele2, locus2Reference, locus2Alternate, ... OR this contains read counts for the reference allele, and counts_alt contains read counts for the alternate allele The rownames should be the sample names.

counts_alt — Either NULL or a numeric matrix with each row corresponding to a different sample. The matrix contains counts for the alternate allele, with samples and loci having the same order as in counts If this parameter is NULL, counts is assumed to have both the reference and alternate allele counts.

ploidy — The ploidy (as an integer) to fit.

h — Either NULL, or a numeric vector of h values for each locus in the same order that the loci are ordered in counts. These h values are as defined by Gerard et al. (2018) "Genotyping polyploids from messy sequencing data" Genetics 210:789-807. with h expressed as alternate / reference. These values can be estimated using the R package "updog". If NULL, h values of 1 (unbiased) are used for all loci.

eps — Either NULL, or a numeric vector of values for the error rate per read for each locus in the same order that the loci are ordered in counts. These are expressed as proportions, so a rate of 1% should be given as 0.01. These values can be estimated using the R package "updog". If NULL, error rates of .01 are assumed for all loci.

maxIter — The maximum number of iterations of the EM algorithm to run for a given sample

maxDiff — This is the maximum change in log-likelihood from the previous iteration to accept as convergence and stop the EM algorithm.

model — the model to fit: BB_noise - mixture of beta-binomials WITH uniform noise, BB - mixture of beta-binomials WITHOUT uniform noise, Bin - mixture of binomials

maxSubIter — If model is BB_noise or BB, this is the maximum number of iterations of to perform during the M-step of the EM algorithm.

IC — TRUE to calculate AIC and BIC.

---

rPloidySamples *simulate read counts for random samples*

Description

for nSamps samples, read counts are simulated by: 1. If genotypePropsAreKnown is FALSE, draw genotype proportions from a Dirichlet posterior calculated using genotypeCounts (treated as a multinomial) and a uniform Dirichlet prior. If genotypePropsAreKnown is TRUE, treat genotypeCounts as the true genotype proportions. 2. Draw genotypes from categorical distributions using the genotype proportions from step 1. 3. Draw read counts per locus from a Dirichlet-multinomial with alpha describing the Dirichlet and reads total reads 4. Draw reads of reference and alternative alleles as a binomial using the probability calculated from the allele dosage of the genotypes drawn in step 2, eps, and h

Usage

```
rPloidySamples(
  nSamps,
  reads,
  truePloidy,
  alpha,
  eps = NULL,
  h = NULL,
  genotypeCounts,
  genotypePropsAreKnown = FALSE
)
```

Arguments

nSamps	number of samples to simulate
reads	number of reads per sample
truePloidy	the true ploidy, as an integer, to simulate
alpha	the alpha values of a Dirichlet distribution describing the proportion of reads for each locus
eps	Either NULL, or a numeric vector of values for the error rate per read for each locus in the same order that the loci are ordered in alpha
h	Either NULL, or a numeric vector of h values for each locus in the same order that the loci are ordered in alpha
genotypeCounts	a matrix with each row representing a locus (in the same order as alpha), and each column representing a genotype with column 1 being 0 copies of the reference allele, column 2 being 1 copy of the reference, ..., column truePloidy + 1 being truePloidy copies of the reference
genotypePropsAreKnown	boolean indicating whether to threat genotypeCounts as true proportions, or the observed number of genotypes in a sample

---

tripsAndDip *Uses read counts for biallelic SNPs to determine if a sample is diploid or triploid*

Description

tripsAndDip calculates log-likelihood ratios comparing whether a sample is likely diploid or triploid based on the read counts for biallelic SNPs.

Usage

```
tripsAndDip(
  counts,
  counts_alt = NA,
  h,
  eps,
  min_reads = 30,
  min_loci = 15,
  binom_p_value = 0.05
)
```

Arguments

counts	Either a numeric matrix or a dataframe with each row corresponding to a different sample. There are two options for formatting the input. Either the columns correspond to the read counts for each locus, in a two column per locus format: column 1 is the read counts for locus1ReferenceAllele, column two is the read counts for locus1AlternateAllele2, locus2Reference, locus2Alternate, ... OR this contains read counts for the reference allele, and counts_alt contains read counts for the alternate allele The rownames should be the sample names.
counts_alt	This is a numeric matrix or a dataframe with each row corresponding to a different sample. The matrix contains counts for the alternate allele, with samples and loci having the same order as in counts If this parameter is NA or NULL, counts is assumed to have both the reference and alternate allele counts.
h	A numeric vector of h values for each locus in the same order that the loci are ordered in counts. These h values are as defined by Gerard et al. (2018) "Genotyping polyploids from messy sequencing data" Genetics 210:789-807. with h expressed as alternate / reference. These values can be estimated using the R package "updog".
eps	A numeric vector of values for the error rate per read for each locus in the same order that the loci are ordered in counts. These are expressed as proportions, so a rate of 1% should be given as 0.01. These values can be estimated using the R package "updog".
min_reads	The minimum number of reads to consider a locus.
min_loci	The minimum number of usable loci in a sample to calculate a log-likelihood ratio.
binom_p_value	The alpha value to use when applying a binomial test to determine whether to include a locus in the calculation.

Details

tripsAndDip calculates log-likelihood ratios comparing the likelihoods of the read counts under diploidy or triploidy for a sample using biallelic SNPs.This function was designed with amplicon sequencing data in mind, but may be useful for other genotyping techniques that also yield read counts for each allele in a given locus. Full details of the calculations can be found in Delomas (2019) Differentiating diploid and triploid individuals using single nucleotide polymorphisms genotyped by amplicon-sequencing. Molecular Ecology Resources.

Value

a dataframe with column 1 containing sample names, column 2 containing calculated LLRs (larger means more likely given triploidy) and column 3 containing the number of loci used to calculate the LLR

Examples

```
# make up some data
triploid_allele1 <- rbinom(60, 75, 2/3)
triploid_allele2 <- 75 - triploid_allele1
diploid_allele1 <- rbinom(60, 75, 1/2)
diploid_allele2 <- 75 - diploid_allele1
# interleave allele counts
triploid <- c(rbind(triploid_allele1, triploid_allele2))
diploid <- c(rbind(diploid_allele1, diploid_allele2))

# create counts matrix
allele_counts <- matrix(data = c(triploid, diploid), byrow = TRUE, nrow = 2, ncol = 120)
rownames(allele_counts) <- c("triploid", "diploid")

#create h and eps vectors
h_constant <- rep(1, 60)
eps_constant <- rep(.01, 60)

#run function
ploidy <- tripsAndDip(allele_counts, h = h_constant, eps = eps_constant)
```

Index