Package ‘poolfstat’

October 22, 2019

Maintainer Mathieu Gautier <mathieu.gautier@inra.fr>

Author Mathieu Gautier, Valentin Hivert and Renaud Vitalis

Version 1.1.1

License GPL (>= 2)

Title Computing F-Statistics from Pool-Seq Data

Description

Functions for the computation of F-statistics from Pool-Seq data in population genomics studies. The package also includes several utilities to manipulate Pool-Seq data stored in standard format ('vcf' or 'rsync' files generated by the the 'PoPoolation' software) and perform conversion to alternative format (as used in the 'BayPass' and 'SelEstim' software).

Depends R (>= 3.0), methods, utils, foreach, doParallel, parallel

RoxygenNote 6.1.1

NeedsCompilation no

Repository CRAN

Date/Publication 2019-10-22 09:30:06 UTC

R topics documented:

computeFST *Compute FST from Pool-Seq data*

Description

Compute FST from Pool-Seq data

Usage

```
computeFST(pooldata, method = "Anova", snp.index = NA)
```

Arguments

`pooldata` A pooldata object containing Pool-Seq information

`method` Either "Anova" (default method as described in the manuscript) or "Identity" (relies on an alternative modeling consisting in estimating unbiased Probability of Identity within and across pairs of pools)

`snp.index` A list of SNP to be considered in the computation (by default all the SNP are considered)

Value

A list with the four following elements:

1. "FST": a scalar corresponding to the estimate the global FST
2. "snp.FST": a vector containing estimates of SNP-specific FST
3. "snp.Q1": a vector containing estimates of the overall within pop. SNP-specific probability of identity
4. "snp.Q2": a vector containing estimates of the overall between pop. SNP-specific probability of identity

See Also

To generate pooldata object, see `vcf2pooldata`, `popsync2pooldata`

Examples

```
 make.example.files(writing.dir=tempdir())
pooldata=popsync2pooldata(sync.file=paste0(tempdir(),"/ex.sync.gz"),poolsizes=rep(50,15))
 res.fst=computeFST(pooldata)
```

computePairwiseFSTmatrix

Compute pairwise population population FST matrix (and possibly all pairwise SNP-specific FST)

Description

Compute pairwise population population FST matrix (and possibly all pairwise SNP-specific FST)

Usage

```
computePairwiseFSTmatrix(pooldata, method = "Anova",
  min.cov.per.pool = -1, max.cov.per.pool = 1e+06, min.maf = -1,
  output.snp.values = FALSE)
```

Arguments

`pooldata` A pooldata object containing Pool-Seq information

`method` Either "Anova" (default method as described in the manuscript) or "Identity" (relies on an alternative modeling consisting in estimating unbiased Probability of Identity within and across pairs of pools)

`min.cov.per.pool` Minimal allowed read count (per pool). If at least one pool is not covered by at least min.cov.perpool reads, the position is discarded in the corresponding pairwise comparisons.

`max.cov.per.pool` Maximal allowed read count (per pool). If at least one pool is covered by more than min.cov.perpool reads, the position is discarded in the corresponding pairwise comparisons.

`min.maf` Minimal allowed Minor Allele Frequency (computed from the ratio overal read counts for the reference allele over the read coverage) in the pairwise comparisons.

`output.snp.values` If TRUE, provide SNP-specific pairwise FST for each comparisons (may lead to a huge result object if the number of pools and/or SNPs is large)

Value

A list with 2 (or 5 if output.snp.values=TRUE) elements:

1. "PairwiseFSTmatrix": a matrix with npools rows and npools columns containing the pairwise pool FST estimates
2. "NbOfSNPs": a matrix with npools rows and npools columns containing the number of SNPs satisfying the filtering criteria in pairs of pools (and within each pool in the diagonal)
3. "PairwiseSnpFST" (if output.snp.values=TRUE): a matrix with nsnp rows and (npools*(npools-1))/2 columns containing the SNP-specific FST estimates for each pair of pools #’

4. "PairwiseSnpQ1" (if output.snp.values=TRUE): a matrix with nsnp rows and (npools*(npools-1))/2 columns containing the SNP-specific Q1 estimates for each pair of pools #'
5. "PairwiseSnpQ2" (if output.snp.values=TRUE): a matrix with nsnp rows and (npools*(npools-1))/2 columns containing the SNP-specific Q2 estimates for each pair of pools

See Also

To generate subset of pooldata object, see `pooldata.subset`

Examples

```
make.example.files(writing.dir=tempdir())
pooldata=popsync2pooldata(sync.file=paste0(tempdir(),"/ex.sync.gz"),poolsizes=rep(50,15))
PairwiseFST=computePairwiseFSTmatrix(pooldata)
```

`genobaypass2pooldata` *Convert BayPass read count and haploid pool size input files into a pooldata object*

Description

Convert BayPass read count and haploid pool size input files into a pooldata object

Usage

```
genobaypass2pooldata(genobaypass.file = "", poolsize.file = "",
  poolnames = NA, min.cov.per.pool = -1, max.cov.per.pool = 1e+06,
  min.maf = -1, nlines.per.readblock = 1e+06)
```

Arguments

`genobaypass.file`
The name (or a path) of the BayPass read count file (see the BayPass manual http://www1.montpellier.inra.fr/CBGP/software/baypass/)

`poolsize.file`
The name (or a path) of the BayPass (haploid) pool size file (see the BayPass manual http://www1.montpellier.inra.fr/CBGP/software/baypass/)

`poolnames`
A character vector with the names of pool

`min.cov.per.pool`
Minimal allowed read count (per pool). If at least one pool is not covered by at least min.cov.perpool reads, the position is discarded

`max.cov.per.pool`
Maximal allowed read count (per pool). If at least one pool is covered by more than min.cov.perpool reads, the position is discarded

`min.maf`
Minimal allowed Minor Allele Frequency (computed from the ratio overal read counts for the reference allele over the read coverage)

`nlines.per.readblock`
Number of Lines read simultaneously. Should be adapted to the available RAM.

Value

A pooldata object containing 7 elements:

1. "refallele.readcount": a matrix with nsnp rows and npools columns containing read counts for the reference allele (chosen arbitrarily) in each pool
2. "readcoverage": a matrix with nsnp rows and npools columns containing read coverage in each pool
3. "snp.info": a matrix with nsnp rows and four columns containing respectively the contig (or chromosome) name (1st column) and position (2nd column) of the SNP; the allele in the reference assembly (3rd column); the allele taken as reference in the refallele matrix.readcount matrix (4th column); and the alternative allele (5th column)
4. "poolsizes": a vector of length npools containing the haploid pool sizes
5. "poolnames": a vector of length npools containing the names of the pools
6. "nsnp": a scalar corresponding to the number of SNPs
7. "npools": a scalar corresponding to the number of pools

Examples

```
 make.example.files(writing.dir=tempdir())
pooldata=popsync2pooldata(sync.file=paste0(tempdir(),"/ex.sync.gz"),poolsizes=rep(50,15))
 pooldata2genobaypass(pooldata=pooldata,writing.dir=tempdir())
 pooldata=genobaypass2pooldata(genobaypass.file=paste0(tempdir(),"/genobaypass"),
                               poolsize.file=paste0(tempdir(),"/poolsize"))
```

genoselestim2pooldata *Convert SelEstim read count input files into a pooldata object*

Description

Convert SelEstim read count input files into a pooldata object

Usage

```
genoselestim2pooldata(genoselestim.file = "", poolnames = NA,
  min.cov.per.pool = -1, max.cov.per.pool = 1e+06, min.maf = -1,
  nlines.per.readblock = 1e+06)
```

Arguments

`genoselestim.file`
: The name (or a path) of the SelEstim read count file (see the SelEstim manual http://www1.montpellier.inra.fr/CBGP/software/selestim/)

`poolnames`
: A character vector with the names of pool

`min.cov.per.pool`
: Minimal allowed read count (per pool). If at least one pool is not covered by at least min.cov.perpool reads, the position is discarded

`max.cov.per.pool`
Maximal allowed read count (per pool). If at least one pool is covered by more than min.cov.perpool reads, the position is discarded

`min.maf` Minimal allowed Minor Allele Frequency (computed from the ratio overal read counts for the reference allele over the read coverage)

`nlines.per.readblock`
Number of Lines read simultaneously. Should be adapted to the available RAM.

Value

A pooldata object containing 7 elements:

1. "refallele.readcount": a matrix with nsnp rows and npools columns containing read counts for the reference allele (chosen arbitrarily) in each pool
2. "readcoverage": a matrix with nsnp rows and npools columns containing read coverage in each pool
3. "snp.info": a matrix with nsnp rows and four columns containing respectively the contig (or chromosome) name (1st column) and position (2nd column) of the SNP; the allele in the reference assembly (3rd column); the allele taken as reference in the refallele matrix.readcount matrix (4th column); and the alternative allele (5th column)
4. "poolsizes": a vector of length npools containing the haploid pool sizes
5. "poolnames": a vector of length npools containing the names of the pools
6. "nsnp": a scalar corresponding to the number of SNPs
7. "npools": a scalar corresponding to the number of pools

Examples

```
 make.example.files(writing.dir=tempdir())
pooldata=popsync2pooldata(sync.file=paste0(tempdir(),"/ex.sync.gz"),poolsizes=rep(50,15))
 pooldata2genoselestim(pooldata=pooldata,writing.dir=tempdir())
 pooldata=genoselestim2pooldata(genoselestim.file=paste0(tempdir(),"/genoselestim"))
```

`is.pooldata` *Check pooldata objects*

Description

Check pooldata objects

Usage

```
is.pooldata(x)
```

Arguments

`x` The name (or a path) of the Popoolation sync file (might be in compressed format)

make.example.files *Create example files*

Description

Write in the current directory example files corresponding to a sync (as obtained when parsing mpileup files with PoPoolation) and vcf (as obtained when parsing mpileup files with VarScan) gzipped files

Usage

```
make.example.files(writing.dir = "")
```

Arguments

`writing.dir` Directory where to copy example files (e.g., set writing.dir=getwd() to copy in the current working directory)

Examples

```
make.example.files(writing.dir=tempdir())
```

pooldata-class *An S4 class to represent a Pool-Seq data set.*

Description

An S4 class to represent a Pool-Seq data set.

Slots

`npools` The number of pools

`nsnp` The number of SNPs

`refallele.readcount` A matrix (nsnp rows and npools columns) with read count data for the reference allele

`readcoverage` A matrix (nsnp rows and Npools columns) with overall read coverage

`snp.info` A matrix (nsnp rows and 4 columns) detailing for each SNP, the chromosome (or scaffold), the position, allele 1 and allele 2

`poolsizes` A vector of length npools with the corresponding haploid pool sizes

`poolnames` A vector of length npools with the corresponding haploid pool names

See Also

To generate pooldata object, see `vcf2pooldata`, `popsync2pooldata`, `genobaypass2pooldata` and `genoselestim2pooldata`

pooldata.subset *Create a subset of the pooldata object that contains Pool-Seq data*

Description

Create a subset of the pooldata object that contains Pool-Seq data

Usage

```
pooldata.subset(pooldata, pool.index = c(1, 2), min.cov.per.pool = -1,
  max.cov.per.pool = 1e+06, min.maf = -1)
```

Arguments

`pooldata` A pooldata object containing Pool-Seq information

`pool.index` Indexes of the pools (at least two), that should be selected to create the new pooldata object

`min.cov.per.pool` Minimal allowed read count (per pool). If at least one pool is not covered by at least min.cov.perpool reads, the position is discarded

`max.cov.per.pool` Maximal allowed read count (per pool). If at least one pool is covered by more than min.cov.perpool reads, the position is discarded

`min.maf` Minimal allowed Minor Allele Frequency (computed from the ratio overal read counts for the reference allele over the read coverage)

Value

A pooldata object with 7 elements:

1. "refallele.readcount": a matrix with nsnp rows and npools columns containing read counts for the reference allele (chosen arbitrarily) in each pool
2. "readcoverage": a matrix with nsnp rows and npools columns containing read coverage in each pool
3. "snp.info": a matrix with nsnp rows and four columns containing respectively the contig (or chromosome) name (1st column) and position (2nd column) of the SNP; the allele in the reference assembly (3rd column); the allele taken as reference in the refallele matrix.readcount matrix (4th column); and the alternative allele (5th column)
4. "poolsizes": a vector of length npools containing the haploid pool sizes
5. "poolnames": a vector of length npools containing the names of the pools
6. "nsnp": a scalar corresponding to the number of SNPs
7. "npools": a scalar corresponding to the number of pools

See Also

To generate pooldata object, see `vcf2pooldata`, `popsync2pooldata`

Examples

```
 make.example.files(writing.dir=tempdir())
pooldata=popsync2pooldata(sync.file=paste0(tempdir(),"/ex.sync.gz"),poolsizes=rep(50,15))
 pooldata.subset=pooldata.subset(pooldata,pool.index=c(1,2))
```

---

pooldata2genobaypass *Convert a pooldata object into BayPass input files.*

Description

Convert a pooldata object into BayPass allele read count and haploid pool size files. A file containing SNP details is also printed out. Options to generate sub-samples (e.g., for large number of SNPs) are also available.

Usage

```
pooldata2genobaypass(pooldata, writing.dir = getwd(), prefix = "",
  subsamplesize = -1, subsamplingmethod = "thinning")
```

Arguments

`pooldata` A pooldata object containing Pool-Seq information (see `vcf2pooldata` and `popsync2pooldata`)

`writing.dir` Directory where to create the files (e.g., set writing.dir=getwd() to copy in the current working directory)

`prefix` Prefix used for output file names

`subsamplesize` Size of the sub-samples. If <=1 (default), all the SNPs are considered in the output

`subsamplingmethod`
If sub-sampling is activated (argument subsamplesize), define the method used for subsampling that might be either i) "random" (A single data set consisting of randmly chosen SNPs is generated) or ii) "thinning", sub-samples are generated by taking SNPs one every nsub=floor(nsnp/subsamplesize) in the order of the map (a suffix ".subn" is added to each sub-sample files where n varies from 1 to nsub).

Value

Files containing allele count (in BayPass format), haploid pool size (in BayPass format), and SNP details (as in the snp.info matrix from the pooldata object)

See Also

To generate pooldata object, see `vcf2pooldata`, `popsync2pooldata`

Examples

```
make.example.files(writing.dir=tempdir())
pooldata=popsync2pooldata(sync.file=paste0(tempdir(),"/ex.sync.gz"),poolsizes=rep(50,15))
pooldata2genobaypass(pooldata=pooldata,writing.dir=tempdir())
```

---

pooldata2genoselestim *Convert a pooldata object into SelEstim input files.*

Description

Convert a pooldata object into SelEstim allele read count. A file containing SNP details is also printed out. Options to generate sub-samples (e.g., for large number of SNPs) are also available.

Usage

```
pooldata2genoselestim(pooldata, writing.dir = getwd(), prefix = "",
  subsamplesize = -1, subsamplingmethod = "thinning")
```

Arguments

pooldata	A pooldata object containing Pool-Seq information (see vcf2pooldata and popsync2pooldata)
writing.dir	Directory where to create the files (e.g., set writing.dir=getwd() to copy in the current working directory)
prefix	Prefix used for output file names
subsamplesize	Size of the sub-samples. If <=1 (default), all the SNPs are considered in the output
subsamplingmethod	If sub-sampling is activated (argument subsamplesize), define the method used for subsampling that might be either i) "random" (A single data set consisting of randmly chosen SNPs is generated) or ii) "thinning", sub-samples are generated by taking SNPs one every nsub=floor(nsnp/subsamplesize) in the order of the map (a suffix ".subn" is added to each sub-sample files where n varies from 1 to nsub).

Value

Files containing allele count (in SelEstim Pool-Seq format) and SNP details (as in the snp.info matrix from the pooldata object)

See Also

To generate pooldata object, see vcf2pooldata, popsync2pooldata

Examples

```
make.example.files(writing.dir=tempdir())
pooldata=popsync2pooldata(sync.file=paste0(tempdir(),"/ex.sync.gz"),poolsizes=rep(50,15))
pooldata2genoselestim(pooldata=pooldata,writing.dir=tempdir())
```

`poolfstat` *PoolFstat*

Description

Functions for the computation of F-statistics from Pool-Seq data in population genomics studies. The package also includes several utilities to manipulate Pool-Seq data stored in standard format ('vcf' or 'rsync' files generated by the the 'PoPoolation' software) and perform conversion to alternative format (as used in the 'BayPass' and 'SelEstim' software).

Details

Computing F-Statistics from Pool-Seq Data

`popsync2pooldata` *Convert Popoolation Sync files into a pooldata object*

Description

Convert Popoolation Sync files into a pooldata object

Usage

```
popsync2pooldata(sync.file = "", poolsizes = NA, poolnames = NA,
  min.rc = 1, min.cov.per.pool = -1, max.cov.per.pool = 1e+06,
  min.maf = 0.01, noindel = TRUE, nlines.per.readblock = 1e+06,
  nthreads = 1)
```

Arguments

`sync.file` The name (or a path) of the Popoolation sync file (might be in compressed format)

`poolsizes` A numeric vector with haploid pool sizes

`poolnames` A character vector with the names of pool

`min.rc` Minimal allowed read count per base. Bases covered by less than min.rc reads are discarded and considered as sequencing error. For instance, if nucleotides A, C, G and T are covered by respectively 100, 15, 0 and 1 over all the pools, setting min.rc to 0 will lead to discard the position (the polymorphism being considered as tri-allelic), while setting min.rc to 1 (or 2, 3..14) will make the position be considered as a SNP with two alleles A and C (the only read for allele T being disregarded).

`min.cov.per.pool` Minimal allowed read count (per pool). If at least one pool is not covered by at least min.cov.perpool reads, the position is discarded

`max.cov.per.pool`
: Maximal allowed read count (per pool). If at least one pool is covered by more than min.cov.perpool reads, the position is discarded

`min.maf` Minimal allowed Minor Allele Frequency (computed from the ratio overal read counts for the reference allele over the read coverage)

`noindel` If TRUE, positions with at least one indel count are discarded

`nlines.per.readblock`
: Number of Lines read simultaneously. Should be adapted to the available RAM.

`nthreads` Number of available threads for parallelization of some part of the parsing (default=1, i.e., no parallelization)

Value

A pooldata object containing 7 elements:

1. "refallele.readcount": a matrix with nsnp rows and npools columns containing read counts for the reference allele (chosen arbitrarily) in each pool
2. "readcoverage": a matrix with nsnp rows and npools columns containing read coverage in each pool
3. "snp.info": a matrix with nsnp rows and four columns containing respectively the contig (or chromosome) name (1st column) and position (2nd column) of the SNP; the allele in the reference assembly (3rd column); the allele taken as reference in the refallele matrix.readcount matrix (4th column); and the alternative allele (5th column)
4. "poolsizes": a vector of length npools containing the haploid pool sizes
5. "poolnames": a vector of length npools containing the names of the pools
6. "nsnp": a scalar corresponding to the number of SNPs
7. "npools": a scalar corresponding to the number of pools

Examples

```
 make.example.files(writing.dir=tempdir())
pooldata=popsync2pooldata(sync.file=paste0(tempdir(),"/ex.sync.gz"),poolsizes=rep(50,15))
```

`vcf2pooldata` *Convert a VCF file into a pooldata object.*

Description

Convert VCF files into a pooldata object.

Usage

```
vcf2pooldata(vcf.file = "", poolsizes = NA, poolnames = NA,
  min.cov.per.pool = -1, min.rc = 1, max.cov.per.pool = 1e+06,
  min.maf = 0.01, nlines.per.readblock = 1e+06, nthreads = 1)
```

Arguments

`vcf.file` The name (or a path) of the Popoolation sync file (might be in compressed format)

`poolsizes` A numeric vector with haploid pool sizes

`poolnames` A character vector with the names of pool

`min.cov.per.pool`
Minimal allowed read count (per pool). If at least one pool is not covered by at least min.cov.perpool reads, the position is discarded

`min.rc` Minimal allowed read count per base (options silenced for VarScan vcf). Bases covered by less than min.rc reads are discarded and considered as sequencing error. For instance, if nucleotides A, C, G and T are covered by respectively 100, 15, 0 and 1 over all the pools, setting min.rc to 0 will lead to discard the position (the polymorphism being considered as tri-allelic), while setting min.rc to 1 (or 2, 3..14) will make the position be considered as a SNP with two alleles A and C (the only read for allele T being disregarded). For VarScan vcf, markers with more than one alternative allele are discarded because the VarScan AD field only contains one alternate read count.

`max.cov.per.pool`
Maximal allowed read count (per pool). If at least one pool is covered by more than min.cov.perpool reads, the position is discarded

`min.maf` Minimal allowed Minor Allele Frequency (computed from the ratio overal read counts for the reference allele over the read coverage)

`nlines.per.readblock`
Number of Lines read simultaneously. Should be adapted to the available RAM.

`nthreads` Number of available threads for parallelization of some part of the parsing (default=1, i.e., no parallelization)

Details

Genotype format in the vcf file for each pool is assumed to contain either i) an AD field containing allele counts separated by a comma (as produced by popular software such as GATK or samtools/bcftools) or ii) both a RD (reference allele count) and a AD (alternate allele count) as obtained with the VarScan mpileup2snp program (when run with the –output-vcf option). The underlying format is automatically detected by the function. For VarScan generated vcf, it should be noticed that SNPs with more than one alternate allele are discarded (because only a single count is then reported in the AD fields) making the min.rc unavailable. The VarScan –min-reads2 option might replace to some extent this functionalities although SNP where the two major alleles in the Pool-Seq data are different from the reference allele (e.g., expected to be more frequent when using a distantly related reference genome for mapping) will be disregarded.

Value

A pooldata object containing 7 elements:

1. "refallele.readcount": a matrix with nsnp rows and npools columns containing read counts for the reference allele (chosen arbitrarily) in each pool

2. "readcoverage": a matrix with nsnp rows and npools columns containing read coverage in each pool
3. "snp.info": a matrix with nsnp rows and four columns containing respectively the contig (or chromosome) name (1st column) and position (2nd column) of the SNP; the allele in the reference assembly (3rd column); the allele taken as reference in the refallele matrix.readcount matrix (4th column); and the alternative allele (5th column)
4. "poolsizes": a vector of length npools containing the haploid pool sizes
5. "poolnames": a vector of length npools containing the names of the pools
6. "nsnp": a scalar corresponding to the number of SNPs
7. "npools": a scalar corresponding to the number of pools

Examples

```
make.example.files(writing.dir=tempdir())
pooldata=vcf2pooldata(vcf.file=paste0(tempdir(),"/ex.vcf.gz"),poolsizes=rep(50,15))
```

Index