Package ‘skeleSim’

November 27, 2017

Type     Package
Title    Genetic Simulation Engine
Version  0.9.8
URL      https://github.com/christianparobek/skeleSim
BugReports https://github.com/christianparobek/skeleSim/issues
Description  A shiny interface and supporting tools to guide users in choosing appropriate simulations, setting parameters, calculating summary genetic statistics, and organizing data output, all within the R environment. In addition to supporting existing forward and reverse-time simulators, new simulators can be integrated into the environment relatively easily.
License   GPL-2
LazyLoad  Yes
LazyData  true
Depends  R (>= 3.4.0), igraph, shiny (>= 0.13.0), shinyFiles
Imports  adegenet, ape, hierfstat, pegas, markdown, methods, reshape2, rmetasim (>= 3.0.0), strataG (>= 1.0), swfscMisc (>= 1.1)
Suggests knitr, rmarkdown, RJSONIO
Collate  skeleSim-package.R skeleSim.classes.R fsc.classes.R
         rms.classes.R skeleSim.internals.R skeleSimGUI.R
         rms.scenarioCheck.R runSim.R skeleSim.checks.R
         getGammaMutRates.R
VignetteBuilder knitr
RoxygenNote  6.0.1
NeedsCompilation no
Author  Eric Archer [aut],
        Michelle DePrenger-Levin [aut],
        Sean Hoban [aut],
Libby Liggins [aut],
Christian Parobek [aut],
Allan Strand [aut, cre]

Maintainer Allan Strand <stranda@gmail.com>
Repository CRAN
Date/Publication 2017-11-27 22:37:36 UTC

R topics documented:

skeleSim-package ............................................. 2
analysis.funcs ................................................. 3
fastsimcoal.params-class ................................. 4
fsc.loadScenario ............................................ 4
fsc.run .......................................................... 6
fsc.scenarioCheck ........................................... 6
fsc.write ....................................................... 7
getGammaMutRates ............................................. 7
rmetasim.params-class ................................. 8
rms.convert ..................................................... 8
rms.init.landscape ........................................ 9
rms.init.landscape.func .................................. 10
rms.run .......................................................... 11
rms.scenarioCheck ......................................... 11
rms.write ....................................................... 12
runSim ........................................................... 12
skeleSim.checks ............................................. 13
skeleSim.classes ........................................... 13
skeleSim.internals ....................................... 15
skeleSimGUI .................................................. 15

Index 16

skeleSim-package Genetic simulation engine

Description

skeleSim
**Description**

Run Global, Locus, and Pairwise analyses on results from a single simulation replicate stored in `params@rep.sample#`.

**Usage**

```r
analysisFunc(params)
loadResultsMatrix(params, mat, label)
formatOverallStats(g, num.perm.reps)
globalAnalysis(g, num.perm.reps)
locusAnalysisGenotypes(g)
hapSmryFunc(g)
locusAnalysisHaplotypes(g)
calcChordDist(dat)
pairwiseAnalysis(g, num.perm.reps)
```

**Arguments**

- `params` a `skeleSim.params` object.
- `mat` results matrix to be loaded into params object.
- `label` analysis type label ("Global", "Locus", or "Pairwise").
- `g` a `gtypes` object.
- `num.perm.reps` number of permutation replicates.
- `dat` data.frame in hierfstat format (see `genet.dist`).
fastsimcoal.params-class

**fastsimcoal Parameters Class**

**Description**

An S4 class storing parameters specific to fastsimcoal

**Slots**

- `fastsimcoal.exec` character string for the fastsimcoal command line executable.
- `pop.info` matrix of population sampling information created by the `fscPopInfo` function.
- `hist.ev` matrix of historical events created by the `fschistEv` function.
- `locus.params` data.frame specifying loci to simulate created by the `fscLocusParams` function.
- `growth.rate` a vector `num.pop` long giving the growth rate of each population.
- `sample.times` a vector `num.pop` long giving the time (in generations) that samples are drawn from in population.

**fsc.loadScenario**

*Load skeleSim scenario parameters for fastsimcoal*

**Description**

Load skeleSim scenario parameters for fastsimcoal

**Usage**

```r
fsc.loadScenario(num.pops, pop.size, sample.size, mut.rate, migration = NULL, sample.times = NULL, growth.rate = NULL, locus.type = c("dna", "msat", "snp"), sequence.length = NULL, num.loci = NULL, transition.rate = NULL, gsm.param = NULL, range.constraint = NULL, recomb.rate = NULL, chromosome = NULL, num.chrom = NULL, num.gen = NULL, source.deme = NULL, sink.deme = NULL, prop.migrants = NULL, new.sink.size = NULL, new.sink.growth = NULL, new.mig.mat = NULL, fastsimcoal.exec = "fsc252")
```

**Arguments**

- `num.pops` number of populations.
- `pop.size` a vector giving size of each population.
- `sample.size` a vector giving the number of samples to take from each population.
- `mut.rate` dna, msat: per base pair or locus mutation rate.
migration a num.pop x num.pop matrix or list of matrices giving the migration rates between each population.
sample.times a vector giving the number of generations in the past at which samples are taken.
growth.rate a vector giving the growth rate of each population.
locus.type a character representation of what type of marker to simulate. Can be "dna", "msat", or 'snp'.
sequence.length
  dna: number of DNA base pairs to use.
  msat, snp: number of loci to simulate.
transition.rate
dna: fraction of substitutions that are transitions.
gsm.param msat: Value of the geometric parameter for a Generalized Stepwise Mutation (GSM) model. This value represents the proportion of mutations that will change the allele size by more than one step. Values between 0 and 1 are required. A value of 0 is for a strict Stepwise Mutation Model (SMM).
range.constraint
  msat: Range constraint (number of different alleles allowed). A value of 0 means no range constraint.
recomb.rate recombination rate between adjacent markers.
chromosome number or character identifying which chromosome the marker is on.
num.chrom a value giving the number of chromosomes that the locus.params marker specifications should be copied for. If NULL, then chromosome assignment is taken from the chromosome column in locus.params. Any non-NULL integer will cause the chromosome column to be ignored.
num.gen Number of generations, $t$, before present at which the historical event happened.
source.deme Source deme (the first listed deme has index 0)
sink.deme Sink deme
prop.migrants Expected proportion of migrants to move from source to sink.
new.sink.size New size for the sink deme, relative to its size at generation $t$.
new.sink.growth New growth rate for the sink deme.
new.mig.mat New migration matrix to be used further back in time.
fastsimcoal.exec executable

Value
a scenario.params object to be loaded into a list in the scenarios slot of a skeleSim.params object.

Note
Vectors for pop.size, sample.size, sample.times, and growth.rate will be expanded/recycled to ensure they are as long as num.pops.
Depending on the choice of locus.type, values for some arguments may be ignored. See argument list above for which arguments are applicable to which locus.type.
**fsc.run**

**Run fastsimcoal**

**Description**

Run fastsimcoal

**Usage**

fsc.run(params)

**Arguments**

- **params**
  
a skeleSim.params object.

**Value**

a modified skeleSim.params object with the results of a fastsimcoal run.

---

**fsc.scenarioCheck**

**Check parameters for fastsimcoal**

**Description**

Check parameters for fastsimcoal

**Usage**

fsc.scenarioCheck(params)

fsc.histEvConverges(hist.ev, pop.size, growth.rate, num.mig.mats = NULL)

fsc.histEvCheck(hist.ev, pop.size, growth.rate, num.mig.mats = NULL)

**Arguments**

- **params**
  
a skeleSim.params object.
- **hist.ev**
  
a matrix describing historical events, with one row per event.
- **pop.size**
  
umerical vector giving size of each population.
- **growth.rate**
  
umerical vector giving growth rate of each population.
- **num.mig.mats**
  
number of migration matrices.
**fsc.write**

**Write fastsimcoal files**

**Description**

Run fastsimcoal

**Usage**

fsc.write(params)

**Arguments**

params a skeleSim.params object.

**Value**

a modified skeleSim.params object with the results of a fastsimcoal run.

---

**getGammaMutRates**

*Get a Vector of Randomly Chosen Mutation Rates*

**Description**

Get a vector of mutation rates from a gamma distribution by specifying the mean and std. dev. of the distribution

**Usage**

gammaMutRates(n, gmean = 1e-04, gstd = 1e-05)

**Arguments**

n Number of loci to select rates for

gmean Mean of the gamma distribution

gstd Std Dev of the gamma distribution

**Value**

a vector of mutation rates (type numeric)

**Examples**

```r
rates = getGammaMutRates(1000, gmean=0.0001, gstd=0.0001)
hist(rates)
rates = getGammaMutRates(1000, gmean=0.0001, gstd=0.00001)
hist(rates)
```
**rmetasim.params-class  rmetasim Parameters Class**

**Description**
An S4 class storing parameters specific to rmetasim

**Slots**
- `num.stg` a single integer for number of stages
- `self.rate` a single numeric for selfing rate
- `surv.matr` a matrix of numerics for survival transitions between stages
- `repr.matr` a matrix of integers for reproductive (females)
- `male.matr` a matrix of integers for male contribution (males)
- `carrying` a single integer for carrying capacity
- `init.pop.sizes` a vector of integers for initial starting population census sizes (length number of pop'ns)
- `num.alleles` a vector of integers for number of alleles per locus (length number of loci)
- `allele.freq` a vector of numerics for initial allele frequencies ???
- `num.gen` a single integer of number of generations to run

---

**rms.convert  Extract genetic data from landscape object**

**Description**
Extract genetic data from landscape object

**Usage**
rms.convert(Rland, locus.type)

**Arguments**
- `Rland` a `rmetasim` landscape object.
- `locus.type` character giving type of locus (microsat or sequence)

**Value**
A gtypes object containing genotypes (or sequences)
Initialize a landscape object

**Usage**

```r
rms.init.landscape(num.pops = NULL, carrying = NULL, sample.size = NULL,
    mig.rates = NULL, num.loc = NULL, loc.type = NULL, mut.rate = NULL,
    seq.length = NULL, num.stgs = NULL, selfing = NULL, surv.matr = NULL,
    repr.matr = NULL, male.matr = NULL, init.pop.sizes = NULL,
    num.gen = NULL, num.alleles = NULL, allele.freqs = NULL)
```

**Arguments**

- `num.pops`: number of populations to simulate
- `carrying`: maximum population sizes for each population
- `sample.size`: size of sample to be pulled from each population
- `mig.rates`: a list of among-population migration matrices
- `num.loc`: number of independently segregating loci
- `loc.type`: sequence or microsatellite
- `mut.rate`: per gene mutation rate
- `seq.length`: if simulating a sequence, the length of the molecule
- `num.stgs`: number of demographic stages in a population
- `selfing`: selfing rate (must range from 0 [random mating] to 1 [complete selfing])
- `surv.matr`: within pop survival matrices
- `repr.matr`: within pop reproduction matrices
- `male.matr`: with pop male repro matrices
- `init.pop.sizes`: starting population sizes
- `num.gen`: number of generations to simulate
- `num.alleles`: vector of the number of alleles per locus
- `allele.freqs`: list of allele freqs for each locus (range 0-1)
rms.init.landscape.func

Initialize a landscape object

Description

Initialize a landscape object

Usage

rms.init.landscape.func(num.pops = NULL, carrying = NULL,
sample.size = NULL, mig.rates = NULL, num.loc = NULL, loc.type = NULL,
mut.rate = NULL, seq.length = NULL, num.stgs = NULL, selfing = NULL,
surv.matr = NULL, repr.matr = NULL, male.matr = NULL,
init.pop.sizes = NULL, num.gen = NULL, num.alleles = NULL,
allele.freqs = NULL)

Arguments

num.pops number of populations to simulate
carrying maximum population sizes for each population
sample.size size of sample to be pulled from each population
mig.rates a list of among-population migration matrices
num.loc number of independently segregating loci
loc.type sequence or microsatellite
mut.rate per gene mutation rate
seq.length if simulating a sequence, the length of the molecule
num.stgs number of demographic stages in a population
selfing selfing rate (must range from 0 [random mating] to 1 [complete selfing]
surv.matr within pop survival matrices
repr.matr within pop reproduction matrices
male.matr with pop male repro matrices
init.pop.sizes starting population sizes
num.gen number of generations to simulate
num.alleles vector of the number of alleles per locus
allele.freqs list of allele freqs for each locus (range 0-1)
rms.run  

**Run rmetasim**

---

### Description

Run rmetasim

### Usage

```r
rms.run(params)
```

### Arguments

- `params`  
  a `skeleSim.params` object.

### Value

a modified `skeleSim.params` object with the results of a rmetasim run.

---

rms.scenarioCheck  

**Check parameters for rmetasim**

---

### Description

Check parameters for rmetasim

### Usage

```r
rms.scenarioCheck(params)
```

### Arguments

- `params`  
  a `skeleSim.params` object.
runSim

rms.write

Write metasim file

Description
Write metasim landscape script to disk (one per scenario)

Usage
rms.write(params)

Arguments
params a skeleSim.params object.

Value
Nothing

runSim

Run simulation

Description
Run simulation

Usage
runSim(params, num.secs = NULL)

Arguments
params a skeleSim.params object.
num.secs number of seconds to run timing checks
skeleSim.checks

Check all simulation parameters

Description
Check all simulation parameters

Usage
overall.check(params)
non.scenario.check(params)
gen.scenario.check(params)
analyses.check(analyses.requested)

Arguments
params a skeleSim.params object.
analyses.requested A named logical vector with elements named Global, Locus, and Pairwise

skeleSim.classes skeleSim Parameters Class

Description
An S4 class storing generic parameters used throughout the workflow

Slots
title a title for the simulation. Used in labelling of output files.
date datestamp for the simulation.
quiet logical determining whether to limit progress reports.
question a single character representing type of analytical question being addressed. Can be one of: (n)ull, p(o)wer, (p)erformance.
simulator.type a single character representing which type of simulator to use. Can be one of: (c)oalescent or (f)orward-time.
simulator a three character code representing which simulator is being run. Currently codes for fastsimcoal(fsc) and rmetasim(rms) exist.
scenarios a list of scenario.params objects.
num.sim.reps number of replicates to run.
sim.func  a function that runs one replicate of the simulator. Must take and return only a skeleSim.params object.

current.scenario  number of current scenario being run.

current.replicate  number of current replicate within current scenario being run.

rep.sample  result of last call to sim.func.

rep.analysis.func  a function that analyzes the results of one simulation replicate.

num.perm.reps  number of permutation replicates to run for population structure statistics.

rep.result  result from last call to rep.analysis.func.

analysis.results  a matrix containing result of all replicate analyses.

sim.summary.func  a function to summarize rep.analysis.

summary.results  a list containing result from call to sim.summary.func.

sim.check.func  a function to check the parameters object prior to running the simulations

sim.scen.checks  a matrix containing results of 'checks' on scenario elements (T/F)

timing  list containing elapsed time for a simulation

other.checks  a vector containing results of 'checks' on other param object elements

scenario.reps  a two column matrix describing which iteration matches which scenario/replicate

analyses.requested  vector of logicals specifying "Global", "Locus", or "Pairwise" analyses have been requested.

num.pops  number of populations.

pop.size  a vector num.pop long giving size of each population.

sample.size  a vector num.pop long giving the number of samples to take from each population.

migration  a list of one or more num.pop x num.pop matrices giving the migration rates between each population.

locus.type  a character representation of what type of marker to simulate. Can be "dna", "msat", or "snp".

mig.helper  a list of flags and values that are needed for the shiny interface but are not needed for the simulation itself. Makes it easier to keep track of different ways to specify migration matrices for different scenarios. List elements will include migration model, rows and columns of landscape and distance function.

num.loci  number of msat or snp loci to simulate.

sequence.length  number of DNA base pairs to use.

mut.rate  mutation rate for DNA or msat.

simulator.params  an object storing simulator-specific parameters. Can be a list or a simulator-specific class.
skeleSimGUI

---

skeleSim.internals  *SkeleSim internal functions*

---

**Description**

SkeleSim internal functions

**Usage**

currentScenario(params)
currentLabel(params)

**Arguments**

params  a *skeleSim.params* object.

**Value**

currentScenario  the parameters for the current scenario.
currentLabel   a character label representing current scenario and replicate.

---

skeleSimGUI  *GUI for skeleSim system*

---

**Description**

This function starts the shiny simulation control panel

**Usage**

skeleSimGUI(launch.browser = TRUE)

**Arguments**

launch.browser  If true, the system’s default web browser will be launched automatically after the app is started.

**Value**

NULL

# markdown and shinyFiles import added for shiny app
Index

*Topic internals
  skeleSim.internals, 15
*Topic package
  skeleSim-package, 2

analyses.check (skeleSim.checks), 13
analysis.funcs, 3
analysisFunc (analysis.funcs), 3

calcChordDist (analysis.funcs), 3
currentLabel (skeleSim.internals), 15
currentScenario (skeleSim.internals), 15

fastsimcoal.params
  (fastsimcoal.params-class), 4
fastsimcoal.params-class, 4
formatOverallStats (analysis.funcs), 3
fsc.histEvCheck (fsc.scenarioCheck), 6
fsc.histEvConverges
  (fsc.scenarioCheck), 6
fsc.loadScenario, 4
fsc.run, 6
fsc.scenarioCheck, 6
fsc.write, 7
fscHistEv, 4
fscLocusParams, 4
fscPopInfo, 4

gene.scenario.check (skeleSim.checks), 13
gen.dist, 3
gammaMutRates, 7
globalAnalysis (analysis.funcs), 3
gtypes, 3

hapSmryFunc (analysis.funcs), 3
loadResultsMatrix (analysis.funcs), 3
locusAnalysisGenotypes
  (analysis.funcs), 3
locusAnalysisHaplotypes
  (analysis.funcs), 3

non.scenario.check (skeleSim.checks), 13
overall.check (skeleSim.checks), 13
pairwiseAnalysis (analysis.funcs), 3
rmetasim.params-class, 8
rms.convert, 8
rms.init.landscape, 9
rms.init.landscape.func, 10
rms.run, 11
rms.scenarioCheck, 11
rms.write, 12
runSim, 12
scenario.params, 5
scenario.params (skeleSim.classes), 13
scenario.params-class
  (skeleSim.classes), 13
skeleSim (skeleSim-package), 2
skeleSim-package, 2
skeleSim.checks, 13
skeleSim.classes, 13
skeleSim.internals, 15
skeleSim.params, 3, 5–7, 11–13, 15
skeleSim.params (skeleSim.classes), 13
skeleSim.params-class
  (skeleSim.classes), 13
skeleSimGUI, 15