Package: HACSim (via r-universe)

August 27, 2024

Type Package

Title Iterative Extrapolation of Species' Haplotype Accumulation Curves for Genetic Diversity Assessment

Version 1.0.6

Date 2022-05-23

Author Jarrett D. Phillips [aut, cre], Steven H. French [ctb], Navdeep Singh [ctb]

Maintainer Jarrett D. Phillips <phillipsjarrett1@gmail.com>

Description Performs iterative extrapolation of species' haplotype accumulation curves using a nonparametric stochastic (Monte Carlo) optimization method for assessment of specimen sampling completeness based on the approach of Phillips et al. (2015) <doi:10.1515/dna-2015-0008>, Phillips et al. (2019) <doi:10.1002/ece3.4757> and Phillips et al. (2020) <doi:10.7717/peerj-cs.243>. 'HACSim' outputs a number of useful summary statistics of sampling coverage (``Measures of Sampling Closeness''), including an estimate of the likely required sample size (along with desired level confidence intervals) necessary to recover a given number/proportion of observed unique species' haplotypes. Any genomic marker can be targeted to assess likely required specimen sample sizes for genetic diversity assessment. The method is particularly well-suited to assess sampling sufficiency for DNA barcoding initiatives. Users can also simulate their own DNA sequences according to various models of nucleotide substitution. A Shiny app is also available.

License GPL-3

URL <https://github.com/jphill01/HACSim.R> <https://github.com/jphill01/HACSim-RShiny-App> <https://jphill01.shinyapps.io/HACSim>

NeedsCompilation yes

Imports ape (>= 5.3), data.table (>= 1.12.8), graphics (>= 3.6.1), matrixStats (>= 0.56.0), pegas (>= 0.13), Rcpp (>= 1.0.3), shiny (>= 1.6.0), stats (>= 3.6.1), utils (>= 3.6.1)

LinkingTo Rcpp, RcppArmadillo
RoxygenNote 6.1.1
Repository https://jphill01.r-universe.dev
RemoteUrl https://github.com/jphill01/hacsim.r
RemoteRef HEAD
RemoteSha 2a525ad40c5cd198a652107eee63044bdb30b521

Contents

`HACSim-package` *Iterative Extrapolation of Species' Haplotype Accumulation Curves for Genetic Diversity Assessment*

Description

HACSim (**H**aplotype **A**ccumulation **C**urve **Sim**ulator) employs a novel nonparametric stochastic (Monte Carlo) optimization method of iteratively generating species' haplotype accumulation curves through extrapolation to assess sampling completeness based on the approach outlined in Phillips et al. (2015) <doi:10.1515/dna-2015-0008>, Phillips et al. (2019) <doi:10.1002/ece3.4757> and Phillips et al. (2020) <doi: 10.7717/peerj-cs.243>. HACSim outputs a number of useful summary statistics of sampling coverage ("Measures of Sampling Closeness"), including an estimate of the likely required sample size (along with desired level confidence intervals) necessary to recover a given number/proportion of observed unique species' haplotypes. Any genomic marker can be targeted to assess likely required specimen sample sizes for genetic diversity assessment. The method is particularly well-suited to assess sampling sufficiency for DNA barcoding initiatives. Users can also simulate their own DNA sequences according to various models of nucleotide substitution. A Shiny app is also available.

Details

The DESCRIPTION file: This package was not yet installed at build time.

Index: This package was not yet installed at build time.

Author(s)

Jarrett D. Phillips [aut, cre], Steven H. French [ctb], Navdeep Singh [ctb]

Maintainer: Jarrett D. Phillips <phillipsjarrett1@gmail.com>

References

Phillips, J.D., Gwiazdowski, R.A., Ashlock, D. and Hanner, R. (2015). An exploration of sufficient sampling effort to describe intraspecific DNA barcode haplotype diversity: examples from the ray-finned fishes (Chordata: Actinopterygii). DNA Barcodes, 3: 66-73.

Phillips, J.D., Gillis, D.J. and Hanner, R.H. (2019). Incomplete estimates of genetic diversity within species: Implications for DNA barcoding. Ecology and Evolution, 9(5): 2996-3010.

Phillips, J.D., Gillis, D.J. and Hanner, R.H. (2020). HACSim: An R package to estimate intraspecific sample sizes for genetic diversity assessment using haplotype accumulation curves. PeerJ Computer Science

Examples

```
## Simulate hypothetical species ##

N <- 100 # total number of sampled individuals
Hstar <- 10 # total number of haplotypes
probs <- rep(1/Hstar, Hstar) # equal haplotype frequency distribution

HACSObj <- HACHypothetical(N = N, Hstar = Hstar,
probs = probs, filename = "output") # outputs a CSV
# file called "output.csv"

## Simulate hypothetical species - subsampling ##
HACSObj <- HACHypothetical(N = N, Hstar = Hstar,
probs = probs, perms = 1000, p = 0.95,
subsample = TRUE, prop = 0.25, conf.level = 0.95,
filename = "output")

## Simulate hypothetical species and all paramaters changed - subsampling ##
HACSObj <- HACHypothetical(N = N, Hstar = Hstar, probs = probs,
perms = 10000, p = 0.90, subsample = TRUE, prop = 0.15,
conf.level = 0.95, filename = "output")

HAC.simrep(HACSObj) # runs a simulation


## Simulate real species ##

## Not run:
## Simulate real species ##
# outputs file called "output.csv"
HACSObj <- HACReal(filename = "output")

## Simulate real species - subsampling ##
HACSObj <- HACReal(subsample = TRUE, prop = 0.15,
```

```
conf.level = 0.95, filename = "output")

## Simulate real species and all parameters changed - subsampling ##
HACSObj <- HACReal(perms = 10000, p = 0.90, subsample = TRUE,
prop = 0.15, conf.level = 0.99, filename = "output")

# user prompted to select appropriate FASTA file
HAC.simrep(HACSObj)
## End(Not run)

## Not run:
## Simulate DNA sequences ##

num.seqs <- 100 # number of DNA sequences
num.haps <- 15 # number of haplotypes
length.seqs <- 658 # length of DNA sequences
count.haps <- c(60, rep(10, 2), rep(5, 2), rep(1, 5)) # haplotype frequency distribution
nucl.freqs <- rep(0.25, 4) # nucleotide frequency distribution
subst.model <- "JC69" # desired nucleotide substitution model
mu.rate <- 1e-3 # mutation rate
transi.rate <- NULL # transition rate
transv.rate <- NULL # transversion rate

sim.seqs(num.seqs = num.seqs, num.haps = num.haps, length.seqs = length.seqs,
nucl.freqs = nucl.freqs, count.haps = count.haps, subst.model = subst.model,
transi.rate = transi.rate, transv.rate = transv.rate)

# outputs file called "output.csv"
HACSObj <- HACReal(filename = "output")

## Simulate DNA sequences - subsampling ##
HACSObj <- HACReal(subsample = TRUE, prop = 0.15,
conf.level = 0.95, filename = "output")

## Simulate DNA sequences and all parameters changed - subsampling ##
HACSObj <- HACReal(perms = 10000, p = 0.90, subsample = TRUE,
prop = 0.15, conf.level = 0.99, filename = "output")

# user prompted to select appropriate FASTA file
HAC.simrep(HACSObj)
## End(Not run)
```

`accumulate` *Internal C++ code*

Description

`accumulate` comprises internal C++ code employed by `HAC.sim`. It is not directly called by the user.

envr *Simulation variable storage environment*

Description

envr is a new (initially empty) environment that is created when HACSim is loaded.

Value

When a simulation is run via HAC.simrep, envr will contain 26 elements as follows:

ci.type	Type of confidence interval to compute and plot. Default is conf.type = "quantile".
conf.level	The desired confidence level. Default is conf.level = 0.95.
d	A dataframe with Nstar - X rows and five columns: specimens (specs), accumulated haplotypes (means), standard deviations (sds) and quantiles (both lower and upper)
df.out	A dataframe with iters rows and six columns displaying "Measures of Sampling Closeness".
filename	The name of the file where results are to be saved. Default is NULL.
Hstar	Number of unique species' haplotypes
input.seqs	Should DNA sequences be inputted? Default is FALSE.
iters	The number of iterations required to reach convergence
N	The starting sample size used to initialize the algorithm
Nstar	The final (extrapolated) sample size
Nstar.high	The upper endpoint of the desired level confidence interval for the 'true' required sample size
Nstar.low	The lower endpoint of the desired level confidence interval for the 'true' required sample size
num.iters	Number of iterations to compute. num.iters = NULL by default (i.e., all iterations are computed; users can specify num.iters = 1 for the first iteration.)
p	The user-specified level of haplotype recovery. Default is p = 0.95.
perms	The user-specified number of permutations (replications). Default is perms = 10000.
probs	Haplotype frequency distribution vector
progress	Should iteration results be outputted to the console? Default is TRUE.
prop.haps	If subset.haps = TRUE, the user-specified proportion of haplotype labels to recover
prop.seqs	If subset.seqs = TRUE, the user-specified proportion of DNA sequences to recover
ptm	A timer to track progress of the algorithm in seconds
R	The proportion of haplotypes recovered by the algorithm

R.low	The lower endpoint of the desired level confidence interval for the 'true' fraction of haplotypes captured
R.up	The upper endpoint of the desired level confidence interval for the 'true' fraction of haplotypes captured
subset.haps	Should a subsample of haplotype labels be taken? Default is FALSE.
subset.seqs	Should a subsample of DNA sequences be taken? Default is FALSE.
X	Mean number of specimens not sampled

Examples

```
# Returns the frequencies of each haplotype in the extrapolated sample
max(envr$d$specs) * envr$probs

# Returns the extrapolated sample size corresponding to the dotted line
# in the last iteration plot
envr$d[which(envr$d$means >= envr$p * envr$Hstar), ][1, 1]
```

HAC.sim *Internal R code*

Description

HAC.sim comprises internal R code used by HAC.simrep and is not directly called by the user.

HAC.simrep *Run a simulation of haplotype accumulation curves for hypothetical or real species*

Description

Runs the HACSim algorithm by successively calling HAC.sim to iteratively extrapolate haplotype accumulation curves to determine likely specimen sample sizes for hypothetical or real species

The algorithm employs the following iterative methods when calculating the "Measures of Sampling Closeness":

- Mean number of haplotype sampled: H_i
- Mean number of haplotypes not sampled $H^* - H_i$
- Proportion of haplotypes sampled: $\frac{H_i}{H^*}$
- Proportion of haplotypes not sampled: $1 - \frac{H_i}{H^*}$
- Mean value of $N*$: $\frac{N_i H^*}{H_i}$
- Mean number of specimens not sampled: $\frac{N_i H^*}{H_i} - N_i$

where H_i is stochastically-determined through sampling from probs, the observed species' haplotype frequency distribution vector.

As the algorithm proceeds, H_i will approach H^* asymptotically (and hence, N_i will converge to N^*), but will likely fluctuate randomly from one iteration to the next. However, estimates of N^* found at each iteration will be monotonically-increasing.

Usage

```
HAC.simrep(HACSObject)
```

Arguments

HACSObject	object containing the desired simulation parameters

Value

Iteration results are outputted to the console and graphs displayed in the plot window. Plots depict haplotype accumulation (along with shaded confidence intervals for the mean number of haplotypes found). Dashed lines correspond to the endpoint of the curve and reflect haplotype recovery for a user-defined cutoff (default p = 0.95, 95% haplotype diversity). Output from the first iteration is useful for judging levels of haplotype diversity and recovery found in observed intraspecific sequence datasets, reflecting current sampling depth. The required sample size is displayed in the second- last iteration. All other information corresponding to the extrapolated sample size can be found in the last iteration. Iteration results can optionally be saved to a CSV file. Subsampled DNA sequences are automatically saved to a FASTA file.

Note

When simulating real species via HACReal(...), a pop-up window will appear prompting the user to select an intraspecific FASTA file of aligned/trimmed DNA sequences. The alignment must not contain missing or ambiguous nucleotides (i.e., it should only contain A, C, G or T); otherwise, haplotype diversity may be overestimated. Excluding sequences or alignment sites with missing/ambiguous data is an option.

Examples

```
## Simulate hypothetical species ##

N <- 100 # total number of sampled individuals
Hstar <- 10 # total number of haplotypes
probs <- rep(1/Hstar, Hstar) # equal haplotype frequency distribution

HACSObj <- HACHypothetical(N = N, Hstar = Hstar , probs = probs,
filename = "output") # outputs a CSV file called "output.csv"

## Simulate hypothetical species - subsampling ##
HACSObj <- HACHypothetical(N = N, Hstar = Hstar, probs = probs,
perms = 1000, p = 0.95, subsample = TRUE, prop = 0.25,
conf.level = 0.95, filename = "output")

## Simulate hypothetical species and all paramaters changed - subsampling ##
HACSObj <- HACHypothetical(N = N, Hstar = Hstar, probs = probs,
perms = 10000, p = 0.90, subsample = TRUE, prop = 0.15, conf.level = 0.95,
filename = "output")

try(HAC.simrep(HACSObj)) # runs a simulation

## Simulate real species ##
```

```
## Not run:
  ## Simulate real species ##
  # outputs file called "output.csv"
  HACSObj <- HACReal(filename = "output")

  ## Simulate real species - subsampling ##
  HACSObj <- HACReal(subsample = TRUE, prop = 0.15, conf.level = 0.95,
  filename = "output")

  ## Simulate real species and all parameters changed - subsampling ##
  HACSObj <- HACReal(perms = 10000, p = 0.90, subsample = TRUE,
  prop = 0.15, conf.level = 0.99, filename = "output")

  # user prompted to select appropriate FASTA file
  try(HAC.simrep(HACSObj))

## End(Not run)
```

`HACClass` *Internal R code*

Description

`HACClass` comprises internal R code used to generate an object used by `HAC.simrep`. It is not directly called by the user.

`HACHypothetical` *Set up an object to simulate haplotype accumulation curves for a hypothetical species*

Description

Helper function which creates an object containing necessary information to run a simulation of haplotype accumulation for a hypothetical species of interest

Usage

```
HACHypothetical(N, Hstar, probs, perms = 10000, p = 0.95,
conf.level = 0.95, ci.type = "quantile", subsample = FALSE, prop = NULL,
progress = TRUE, num.iters = NULL, filename = NULL)
```

Arguments

N	Number of individuals
Hstar	Number of unique species' haplotypes
probs	Haplotype frequency distribution vector
perms	Number of permutations (replications)
p	Proportion of haplotypes to recover
conf.level	Desired confidence level for graphical output and interval estimation
ci.type	Type of confidence interval for graphical output. Choose from "quantile" or "asymptotic"
subsample	Is a subsample of haplotype labels desired?
prop	If subsample = TRUE, the proportion of haplotype labels to subsample
num.iters	Number of iterations to compute
progress	Should iteration output be printed to the R console?
filename	Name of file where simulation results are to be saved

Value

A list object of class "HAC" with 13 elements that can be passed to `HAC.simrep` as follows:

input.seqs	Should a FASTA file of aligned/trimmed DNA sequences be inputted? Default is FALSE
subset.seqs	Should a subsample of DNA sequences be taken? Default is FALSE
prop.seqs	Proportion of DNA sequences to subsample. Default is NULL
prop.haps	Proportion of haplotype labels to subsample. Default is NULL (can be altered by user)
subset.haps	Should a subsample of haplotype labels be taken? Default is NULL (can be altered by user)
N	Number of individuals. NA by default (provided by user)
Hstar	Number of unique species' haplotypes. NA by default (provided by user)
probs	Haplotype frequency distribution vector. NA by default (provided by user)
p	Proportion of haplotypes to recover. p = 0.95 by default.
perms	Number of permutations (replications). perms = 10000 by default.
conf.level	Desired confidence level for graphical output and interval estimation. conf.level = 0.95 by default.
ci.type	Type of confidence interval for graphical output. ci.type = "quantile" by default
num.iters	Number of iterations to compute. num.iters = NULL by default (i.e., all iterations are computed; users can specify num.iters = 1 for the first iteration.)
progress	Should iteration output be printed to the R console? Default is TRUE
filename	Name of file where simulation results are to be saved.

Note

N must be greater than 1 and greater than or equal to Hstar.

Hstar must be greater than 1.

probs must have a length equal to Hstar and its elements must sum to 1.

Examples

```
## Simulate hypothetical species ##

N <- 100 # total number of sampled individuals
Hstar <- 10 # total number of haplotypes
probs <- rep(1/Hstar, Hstar) # equal haplotype frequency distribution

# outputs a CSV file called "output.csv"
HACSObj <- HACHypothetical(N = N, Hstar = Hstar, probs = probs,
filename = "output")

## Simulate hypothetical species - subsampling ##
# subsamples 25% of haplotype labels
HACSObj <- HACHypothetical(N = N, Hstar = Hstar, probs = probs,
perms = 1000, p = 0.95, subsample = TRUE, prop = 0.25,
conf.level = 0.95, filename = "output")

## Simulate hypothetical species and all paramaters changed - subsampling ##
HACSObj <- HACHypothetical(N = N, Hstar = Hstar, probs = probs,
perms = 10000, p = 0.90, subsample = TRUE, prop = 0.15, conf.level = 0.95,
num.iters = 1, filename = "output")
```

HACReal *Set up an object to simulate haplotype accumulation curves for a real species*

Description

Helper function which creates an object containing necessary information to run a simulation of haplotype accumulation for a real species of interest

Usage

```
HACReal(perms = 10000, p = 0.95, conf.level = 0.95,
ci.type = "quantile", subsample = FALSE, prop = NULL,
progress = TRUE, num.iters = NULL, filename = NULL)
```

Arguments

perms	Number of permutations (replications)
p	Proportion of haplotypes to recover
conf.level	Desired confidence level for graphical output and interval estimation
ci.type	Type of confidence interval for graphical output. Choose from "quantile" or "asymptotic"
subsample	Is a subsample of DNA sequences desired?
prop	If subsample = TRUE, the proportion of DNA sequences to subsample
num.iters	Number of iterations to compute
progress	Should iteration output be printed to the R console?
filename	Name of file where simulation results are to be saved

Value

A list object of class "HAC" with 13 elements that can be passed to HAC.simrep as follows:

input.seqs	Should a FASTA file of aligned/trimmed DNA sequences be inputted? Default is TRUE
subset.seqs	Should a subsample of DNA sequences be taken? Default is FALSE (can be altered by user)
prop.seqs	Proportion of DNA sequences to subsample. Default is NA (can be altered by user)
prop.haps	Proportion of haplotype labels to subsample. Default is NULL
subset.haps	Should a subsample of haplotype labels be taken? Default is NULL
N	Number of individuals. NA by default (computed automatically by algorithm)
Hstar	Number of unique species' haplotypes. NA by default (computed automatically by algorithm)
probs	Haplotype frequency distribution vector. NA by default (computed automatically by algorithm)
p	Proportion of haplotypes to recover. p = 0.95 by default.
perms	Number of permutations (replications). perms = 10000 by default.
conf.level	Desired confidence level for graphical output and interval estimation. conf.level = 0.95 by default.
ci.type	Type of confidence interval for graphical output. ci.type = "quantile" by default
num.iters	Number of iterations to compute. num.iters = NULL by default (i.e., all iterations are computed; users can specify num.iters = 1 for the first iteration.)
progress	Should iteration output be printed to the R console? Default is TRUE
filename	Name of file where simulation results are to be saved.

Examples

```
## Simulate real species ##
# outputs file called "output.csv"
HACSObj <- HACReal(filename = "output")

## Simulate real species - subsampling ##
# subsamples 25% of DNA sequences
HACSObj <- HACReal(subsample = TRUE, prop = 0.25, conf.level = 0.95,
filename = "output")

## Simulate real species and all parameters changed - subsampling ##
HACSObj <- HACReal(perms = 10000, p = 0.90, subsample = TRUE,
prop = 0.15, conf.level = 0.99, num.iters = 1, filename = "output")
```

---

launchApp *Launch HACSim R Shiny web app*

Description

Launch HACSim R Shiny web app locally on a user's R session

Usage

```
launchApp()
```

---

sim.seqs *Simulate DNA sequences according to DNA substitution models*

Description

Simulates DNA sequences according to various DNA substitution models:

- Jukes-Cantor (1969)
- Kimura (1980)
- Felsenstein (1981)
- Hasegawa-Kishino-Yano (1985)

Usage

```
sim.seqs(num.seqs, num.haps, length.seqs, count.haps, nucl.freqs,
codon.tbl = c("standard", "vertebrate mitochondrial",
"invertebrate mitochondrial"), subst.model = c("JC69", "K80", "F81", "HKY85"),
mu.rate, transi.rate, transv.rate)
```

Arguments

num.seqs	Number of simulated DNA sequences
num.haps	Number of simulated unique species' haplotypes
length.seqs	Basepair length of DNA sequences
count.haps	Haplotype frequency distribution vector
nucl.freqs	Nucleotide frequency distribution vector of A, C, G, and T respectively
codon.tbl	Codon table
subst.model	Model of DNA substitution
mu.rate	Overall nucleotide mutation rate/site/generation
transi.rate	Nucleotide transition rate/site/generation
transv.rate	Nucleotide transversion rate/site/generation

Value

A FASTA file of DNA sequences

Note

num.seqs must be greater than or equal to num.haps.

Both num.seqs and num.haps must be greater than 1.

nucl.freqs must have a length of four and its elements must sum to 1.

count.haps must have a length of num.haps and its elements must sum to num.seqs.

subst.model must be one of "JC69" (Jukes Cantor corrected p-distance), "K80" (Kimura-2-Parameter (K2P), "F81" (Felenstein) or "HKY85" (Hasegawa-Kishino-Yano)

mu.rate must be specified for both "JC69" and "F81" models

transi.rate and transv.rate must be specified for both "K80" and "HKY85" models

All elements nucl.freqs must be equal to 0.25 when subst.model is either "JC69" or "K80"

All elements nucl.freqs must differ from 0.25 when subst.model is either "F81" or "HKY85"

Examples

```
## Not run:

# Simulate DNA sequences from the 5'-COI DNA barcode region under a Jukes
# Cantor nucleotide substitution model

num.seqs <- 100 # number of DNA sequences
num.haps <- 10 # number of haplotypes
length.seqs <- 658 # length of DNA sequences
count.haps <- c(60, rep(10, 2), rep(5, 2), rep(1, 5)) # haplotype frequency distribution
nucl.freqs <- rep(0.25, 4) # nucleotide frequency distribution
codon.tbl <- "vertebrate mitochondrial"
subst.model <- "JC69" # desired nucleotide substitution model
mu.rate <- 1e-3 # mutation rate
```

```
transi.rate <- NULL # transition rate
transv.rate <- NULL # transversion rate

sim.seqs(num.seqs = num.seqs, num.haps = num.haps, length.seqs = length.seqs,
count.haps = count.haps, nucl.freqs = nucl.freqs, subst.model = subst.model,
codon.tbl = codon.tbl, transi.rate = transi.rate, transv.rate = transv.rate)


## End(Not run)
```

Index