Constructing hyperprior for hierarchical co-demographic model

Description

`build.dice` builds a hyperprior across Ψ/ψ and ζ/ζ_s according to: 1) uniform distribution for Ψ/ψ, for ζ_T within each discrete Ψ/ψ value, and across all combinations of the vector ζ/ζ_s within each discrete ζ_T value; 2) Dirichlet-process that weighs all allowable combinations of Ψ/ψ and ζ/ζ_s according to possible combinations of taxa assignment; 3) customized distribution(s).

Usage

```
build.dice(num.taxa, num.partitions=1, tau.psi.prior=NULL,
epsilon.psi.prior=NULL, NE.psi.prior=NULL, tau.zeta.prior=NULL,
tau2.zeta.prior=NULL, epsilon.zeta.prior=NULL, epsilon2.zeta.prior=NULL,
NE.zeta.prior=NULL, tau.zeta.total.prior=NULL, tau2.zeta.total.prior=NULL,
epsilon.zeta.total.prior=NULL, epsilon2.zeta.total.prior=NULL,
NE.zeta.total.prior=NULL, dirichlet.process=F, idiosyncratic=T,
min.net.tau.zeta.total=NULL, min.net.tau2.zeta.total=NULL,
min.net.epsilon.zeta.total=NULL, min.net.epsilon2.zeta.total=NULL,
min.net.NE.zeta.total=NULL, max.net.tau.zeta.total=NULL,
max.net.tau2.zeta.total=NULL, max.net.epsilon.zeta.total=NULL,
max.net.epsilon2.zeta.total=NULL, max.net.NE.zeta.total=NULL,
min.net.tau.zeta.per.pulse=NULL, min.net.tau2.zeta.per.pulse=NULL,
min.net.epsilon.zeta.per.pulse=NULL, min.net.epsilon2.zeta.per.pulse=NULL,
min.net.NE.zeta.per.pulse=NULL, max.net.tau.zeta.per.pulse=NULL,
max.net.tau2.zeta.per.pulse=NULL, max.net.epsilon.zeta.per.pulse=NULL,
max.net.epsilon2.zeta.per.pulse=NULL, max.net.NE.zeta.per.pulse=NULL)
```

Arguments

DATA

`num.taxa`	List, vector of length = `num.partitions`, or positive integer. Number of taxa per partition. Total sum across partitions equals the total number of taxa *n* in dataset. See also `Details`. Required.
`num.partitions`	Positive integer. Number of partitions for taxa in dataset. Allows differential data and model specifications across user-specified taxa groupings, including sampling size of individuals, generation times, demographic syndrome, and taxon-specific nuisance parameter prior distributions. See also `Details`.

PRIORS

`tau.psi.prior`, `epsilon.psi.prior`, `NE.psi.prior`

List, vector, or non-negative integer. Hyperprior distribution for Ψ/ψ of τ, ε, and N_E, respectively. For τ and ε, if list of length = 2, then the first list element applies to the first more recent size change event (*e.g.* $\tau 1$, $\varepsilon 1$) and the second list element applies to the second more ancient size change event (*e.g.* $\tau 2$, $\varepsilon 2$), per taxon. The arguments(s) specified here and their according list lengths activate which taxon-specific demographic parameters are to be hyperparameterized via Ψ/ψ as well as $\zeta/\zeta_s/\zeta_T$ downstream. See also `Details`. At least one is required.

`tau.zeta.prior`, `tau2.zeta.prior`, `epsilon.zeta.prior`, `epsilon2.zeta.prior`, `NE.zeta.prior`

List of length = `num.partitions` or 1, vector, or non-negative proportion (*i.e.* ≤ 1 and ≥ 0). Hyperprior distribution for ζ_j of $\tau 1$, $\tau 2$, $\varepsilon 1$, $\varepsilon 2$, and N_E, respectively, for each *j*th pulse from 1 to Ψ/ψ, as specified by the corresponding `psi.prior`, and per partition. See also `Details`. Required for each corresponding `psi.prior` specified, unless the maximum value in the corresponding `psi.prior` = 0.

`tau.zeta.total.prior`, `tau2.zeta.total.prior`, `epsilon.zeta.total.prior`, `epsilon2.zeta.total.prior`, `NE.zeta.total.prior`

List of length = 1, vector, or non-negative proportion (*i.e.* ≤ 1 and ≥ 0). Hyperprior distribution for ζ_T of $\tau 1$, $\tau 2$, $\varepsilon 1$, $\varepsilon 2$, and N_E, respectively. Activates a uniform hyperprior such that each discrete Ψ/ψ value, as specified by the corresponding `psi.prior`, is first weighted with equal hyperprior probability, then all discrete ζ_T values are weighted equally per Ψ/ψ value, and finally every possible associated vector ζ/ζ_s is weighted equally per ζ_T value. See also `Details`.

MODEL SPECIFICATIONS

`dirichlet.process`

Logical value. Activates a Dirichlet-process hyperprior that weighs all allowable combinations of Ψ/ψ and ζ/ζ_s according to possible combinations of taxa assignment. See also `Details`.

`idiosyncratic`

Logical value. Allows idiosyncratic taxa that freely vary *i.e.* are ungrouped from any of the pulses, as specified by the `psi.prior` arguments. See also `Details`.

`min.net.tau.zeta.total`, `min.net.tau2.zeta.total`, `min.net.epsilon.zeta.total`, `min.net.epsilon2.zeta.total`, `min.net.NE.zeta.total`, `max.net.tau.zeta.total`, `max.net.tau2.zeta.total`,

Non-negative proportion (*i.e.* ≤ 1 and ≥ 0). Rule for the minimum/maximum ζ_T value, across all pulses (as specified by the corresponding `psi.prior`) and partitions, for $\tau 1$, $\tau 2$, $\varepsilon 1$, $\varepsilon 2$, and N_E, respectively. See also `Details`.

`max.net.epsilon.zeta.total`,
`max.net.epsilon2.zeta.total`,
`max.net.NE.zeta.total`

`min.net.tau.zeta.per.pulse`,
`min.net.tau2.zeta.per.pulse`,
`min.net.epsilon.zeta.per.pulse`,
`min.net.epsilon2.zeta.per.pulse`,
`min.net.NE.zeta.per.pulse`,
`max.net.tau.zeta.per.pulse`,
`max.net.tau2.zeta.per.pulse`,
`max.net.epsilon.zeta.per.pulse`,
`max.net.epsilon2.zeta.per.pulse`,
`max.net.NE.zeta.per.pulse`

List, vector of length = maximum value in corresponding `psi.prior`, or non-negative proportion (*i.e.* ≤ 1 and ≥ 0). Rule for the minimum/maximum ζ_j value of $\tau 1$, $\tau 2$, $\varepsilon 1$, $\varepsilon 2$, and N_E, respectively, for each *j*th pulse from 1 to Ψ/ψ (as specified by the corresponding `psi.prior`) across all partitions. See also `Details`.

Arguments from other `Multi-DICE` functions may be included here and are ignored if not applicable.

Details

For more information, see `Multi-DICE` Manual.

`Multi-DICE` cannot currently accommodate models with more than one population per taxon, events aside from population size change, and more than two size change events.

For $\tau 1$ and $\tau 2$, units are in numbers of generations, and thus may only be positive integers. For $\varepsilon 1$ and $\varepsilon 2$, units are in ratio of size change from the ancestral effective population size to current effective population size, such that expansions are < 1 and contractions are > 1, and thus may only be positive values. For N_E, unit is in number of effective haploid individuals, and thus may only be positive integers.

For `num.taxa`, `min.net.tau.zeta.per.pulse`, `min.net.tau2.zeta.per.pulse`, `min.net.epsilon.zeta.per.pulse`, `min.net.epsilon2.zeta.per.pulse`, `min.net.NE.zeta.per.pulse`, `max.net.tau.zeta.per.pulse`, `max.net.tau2.zeta.per.pulse`, `max.net.epsilon.zeta.per.pulse`, `max.net.epsilon2.zeta.per.pulse`, and `max.net.NE.zeta.per.pulse`, if list, then all list elements are concatenated to form a single vector, with the ordering within list elements and then between list elements preserved (*e.g.* for list of length = 2, with first list element of length = 2 and second list element of length = 1, the order from first to last is: 1) first vector element in first list element; 2) second vector element in first list element; 3) sole vector element in second list element).

For `tau.psi.prior`, `epsilon.psi.prior`, `NE.psi.prior`, `tau.zeta.prior`, `tau2.zeta.prior`, `epsilon.zeta.prior`, `epsilon2.zeta.prior`, `NE.zeta.prior`, `tau.zeta.total.prior`, `tau2.zeta.total.prior`, `epsilon.zeta.total.prior`, `epsilon2.zeta.total.prior`, and `NE.zeta.total.prior`, each list element contains an entire individual discrete distribution; if vector, then converted to list of length = 1 with all vector elements comprising the entirety of a single discrete distribution. Per list element, vector elements within (*i.e.* the discrete distribution) do not need to be in any particular order. Relatedly, each vector

element is treated as an independent value, thus weighted distributions (*i.e.* not uniform) may be employed by duplicating values (*e.g.* a distribution of `c(0,1,1,1)` signifies 75% probability of drawing “1” and 25% probability of drawing “0”), allowing the specification of any discretized distribution (*e.g.* gamma, beta, log-uniform). Accordingly, a uniform distribution with no gaps for integer values would be of length = range of distribution.

For `num.taxa`, the order of vector elements corresponds to the order of partitions, and for `tau.zeta.prior`, `tau2.zeta.prior`, `epsilon.zeta.prior`, `epsilon2.zeta.prior`, and `NE.zeta.prior`, the order of list elements corresponds to the order of partitions. Additionally, if length = 1 and `num.partitions` > 1, then the sole element is used for all partitions. Similarly, if length < `num.partitions`, then the first element is used for all partitions while ignoring any remaining elements, and if length > `num.partitions`, then the remaining elements beyond length = `num.partitions` are ignored; a caution is provided when the length does not equal 1 or `num.partitions`.

For `min.net.tau.zeta.per.pulse`, `min.net.tau2.zeta.per.pulse`, `min.net.epsilon.zeta.per.pulse`, `min.net.epsilon2.zeta.per.pulse`, `min.net.NE.zeta.per.pulse`, `max.net.tau.zeta.per.pulse`, `max.net.tau2.zeta.per.pulse`, `max.net.epsilon.zeta.per.pulse`, `max.net.epsilon2.zeta.per.pulse`, and `max.net.NE.zeta.per.pulse`, the order of vector elements corresponds to the temporal order, from most recent to most ancient, of pulses (as specified by the corresponding `psi.prior`). Additionally, if length = 1 and maximum value in corresponding `psi.prior` > 1, then the sole element is used for all pulses. Similarly, if length < maximum value in corresponding `psi.prior`, then the first element is used for all pulses while ignoring any remaining elements, and if length > maximum value in corresponding `psi.prior`, then the remaining elements beyond length = maximum value in corresponding `psi.prior` are ignored; a caution is provided when the length does not equal 1 or maximum value in corresponding `psi.prior`.

If `num.partitions`=*n*, then rearrangement of bins across taxa within allele frequency classes based on descending order of the relative SNP proportions is not performed to construct the aSFS and taxon-specific inference of demographic parameters is possible. However, in general, more partitions results in more parameter space with respect to taxa samples that must be explored due to a decrease in order-independence and assumed exchangeability, thus multiple-fold more simulations must be conducted to achieve comparable accuracy in hyperparameter estimation as without partitioning.

For `tau.psi.prior`, `epsilon.psi.prior`, and `NE.psi.prior`, distinguishing between Ψ and ψ is accomplished via the corresponding `zeta.prior`, `zeta.total.prior`, `idiosyncratic` setting, and/or `min`/`max.net zeta.total`/`per.pulse`, except for values of 0, which are explicitly for $\psi = 0$ and thus indicate full idiosyncrasy. If list length > 2 for τ and ε or list length > 1 for N_E, then a caution is provided and the remaining elements beyond length = 2 for τ and ε and length = 1 for N_E are ignored. Applies across all partitions, such that it is regardless of partitioning.

For tau.zeta.prior, tau2.zeta.prior, epsilon.zeta.prior, epsilon2.zeta.prior, and NE.zeta.prior, if num.partitions > 1, may be necessary to include “0.0” as a value, but can control ζ_s and ζ_T across partitions via corresponding zeta.total.prior and/or min/max.net zeta.total/per.pulse. Attributes to each partition individually, but proportion values are out of the entirety of taxa dataset, thus if num.partitions > 1, then the upper bound for each partition should be the number of taxa within that partition divided by *n*. When $\psi = \{0, 1\}$, equivalent to hyperprior distribution for ζ_T. If identical across all partitions, it is more computationally efficient to specify only one *i.e.* list of length = 1, or a vector.

For tau.zeta.total.prior, tau2.zeta.total.prior, epsilon.zeta.total.prior, epsilon2.zeta.total.prior, and NE.zeta.total.prior, if length > 1, then remaining elements beyond the first are ignored and a caution is provided.

To build a hyperprior, Multi-DICE first looks if the corresponding zeta.total.prior is specified, then if dirichlet.process=T, and if neither is such case, then all possible combinations of corresponding psi.prior and zeta.prior draws are equally weighted. Therefore, if num.partitions=1, corresponding psi.prior=1, and dirichlet.process=F, then tau.zeta.total.prior, tau2.zeta.total.prior, epsilon.zeta.total.prior, epsilon2.zeta.total.prior, and NE.zeta.total.prior equivalent to corresponding zeta.prior and thus unnecessary to specify.

When $\psi = 0$ *i.e.* full idiosyncrasy, the arguments idiosyncratic, min.net.tau.zeta.total, min.net.tau2.zeta.total, min.net.epsilon.zeta.total, min.net.epsilon2.zeta.total, min.net.NE.zeta.total, min.net.tau.zeta.per.pulse, min.net.tau2.zeta.per.pulse, min.net.epsilon.zeta.per.pulse, min.net.epsilon2.zeta.per.pulse, and min.net.NE.zeta.per.pulse are ignored.

Multi-DICE proceeds only if a valid draw is allowed given corresponding num.taxa, psi.prior, zeta.prior (*e.g.* minimum value in psi.prior * minimum value in zeta.prior $\leq$ minimum value in num.taxa), zeta.total.prior, idiosyncratic setting, and min/max.net zeta.total/per.pulse. Importantly, there is no check on if all values in psi.prior have a valid draw, only if there is a valid draw among any of the values.

Value

Returned value is a list object with each element attributed to a hyperparameterized demographic parameter and accordingly named (*i.e.* tau, tau2, epsilon, epsilon2, NE). Each of these list elements contain the following components:

draws — List of matrices. The order of list elements/matrices corresponds to the order of partitions. Per matrix, each row represents a valid draw, with rows across matrices corresponding to each other (*i.e.* the same row number across matrices refers to the same individual draw). The first column is the Ψ/ψ drawn value as specified by the

corresponding `psi.prior`. The second column is a row index number corresponding to the matrices in the `combos` component, which contains information about the ζ/ζ_s drawn value(s).

`combos` List of list of matrices. The order of list elements corresponds to the order of partitions. Each of these list elements contain another list of matrices, with these list elements/matrices corresponding to the different unique Ψ/ψ values in the corresponding `psi.prior`, in ascending order of Ψ/ψ values, and accordingly named with the prefix "pulse" concatenated with Ψ/ψ value as a suffix (*e.g.* "pulse1", "pulse2"). Per matrix, each row represents a possible per-partition draw, each cell is a *S* drawn value, and the order of columns corresponds to the temporal order, from most recent to most ancient, of pulses (as specified by the corresponding `psi.prior`).

`hyperprior` Vector. Contains constructed hyperprior distribution, with each element an indexed draw referring to the row numbers in `draws`.

Author(s)

Alexander T. Xue

References

Chan YL, Schanzenbach D, Hickerson MJ (2014) Detecting concerted demographic response across community assemblages using hierarchical approximate Bayesian computation. *Molecular Biology and Evolution*, **31**, 2501–2515.

Hickerson MJ, Stahl E, Takebayashi N (2007) msBayes: Pipeline for testing comparative phylogeographic histories using hierarchical approximate Bayesian computation. *BMC bioinformatics*, **8**, 268.

Huang W, Takebayashi N, Qi Y, Hickerson MJ (2011) MTML-msBayes: approximate Bayesian comparative phylogeographic inference from multiple taxa and multiple loci with rate heterogeneity. *BMC bioinformatics*, **12**, 1.

Xue AT (2017) Multi-DICE Manual.

Xue AT, Hickerson MJ (2015) The aggregate site frequency spectrum for comparative population genomic inference. *Molecular Ecology*, **24**, 6223–6240.

Xue AT, Hickerson MJ (*submitted*) Multi-DICE: R package for comparative population genomic inference under multi-taxa hierarchical co-demographic models.

See Also

`roll.dice`, `play.dice`, `dice.sims`, `dice.aSFS`, `dice.sumstats`

Examples

```
#simplest execution akin to approach in Xue and Hickerson (2015)
```

```
build.dice(num.taxa=10, tau.psi.prior=c(1), tau.zeta.prior=c(1:10)/10)

#simplest execution akin to approach in software package msBayes
build.dice(num.taxa=10, tau.psi.prior=c(1:10), tau.zeta.prior=c(1:10)/10,
idiosyncratic=F)
```

Hyperprior and parameter summary prior draws for hierarchical co-demographic model

Description

`roll.dice` conducts random draws from the hyperprior distribution constructed in `build.dice`, as well as from user-specified prior distributions for shared pulse values. These shared pulse values (*e.g.* $\tau 1_s$, $\tau 2_s$, $\varepsilon 1_s$, $\varepsilon 2_s$, N_s) are parameter summaries of the taxon-specific demographic parameter values for the shared pulses, as specified by the `psi.prior` arguments. `build.dice` is embedded here and is automatically deployed if `build.object` is not specified.

Usage

```
roll.dice(num.sims, num.taxa, num.partitions=1, tau.psi.prior=NULL,
epsilon.psi.prior=NULL, NE.psi.prior=NULL, tau.zeta.prior=NULL,
tau2.zeta.prior=NULL, epsilon.zeta.prior=NULL, epsilon2.zeta.prior=NULL,
NE.zeta.prior=NULL, tau.zeta.total.prior=NULL, tau2.zeta.total.prior=NULL,
epsilon.zeta.total.prior=NULL, epsilon2.zeta.total.prior=NULL,
NE.zeta.total.prior=NULL, tau.shared.prior=NULL, tau2.shared.prior=NULL,
epsilon.shared.prior=NULL, epsilon2.shared.prior=NULL,
NE.shared.prior=NULL, dirichlet.process=F, idiosyncratic=T,
min.net.tau.zeta.total=NULL, min.net.tau2.zeta.total=NULL,
min.net.epsilon.zeta.total=NULL, min.net.epsilon2.zeta.total=NULL,
min.net.NE.zeta.total=NULL, max.net.tau.zeta.total=NULL,
max.net.tau2.zeta.total=NULL, max.net.epsilon.zeta.total=NULL,
max.net.epsilon2.zeta.total=NULL, max.net.NE.zeta.total=NULL,
min.net.tau.zeta.per.pulse=NULL, min.net.tau2.zeta.per.pulse=NULL,
min.net.epsilon.zeta.per.pulse=NULL, min.net.epsilon2.zeta.per.pulse=NULL,
min.net.NE.zeta.per.pulse=NULL, max.net.tau.zeta.per.pulse=NULL,
max.net.tau2.zeta.per.pulse=NULL, max.net.epsilon.zeta.per.pulse=NULL,
max.net.epsilon2.zeta.per.pulse=NULL, max.net.NE.zeta.per.pulse=NULL,
tau.buffer=0, tau2.buffer=0, epsilon.buffer=0, epsilon2.buffer=0,
NE.buffer=0, build.object=NULL)
```

Arguments

`num.sims` — Positive integer. Number of simulations. Required.

DATA

`num.taxa` — List, vector of length = `num.partitions`, or positive integer. Number of taxa per partition. Total sum across partitions

equals the total number of taxa *n* in dataset. See also `Details`. Required.

`num.partitions`

Positive integer. Number of partitions for taxa in dataset. Allows differential data and model specifications across user-specified taxa groupings, including sampling size of individuals, generation times, demographic syndrome, and taxon-specific nuisance parameter prior distributions. See also `Details`.

PRIORS

`tau.psi.prior, epsilon.psi.prior, NE.psi.prior`

List, vector, or non-negative integer. Hyperprior distribution for Ψ/ψ of τ, ε, and N_E, respectively. For τ and ε, if list of length = 2, then the first list element applies to the first more recent size change event (*e.g.* $\tau 1$, $\varepsilon 1$) and the second list element applies to the second more ancient size change event (*e.g.* $\tau 2$, $\varepsilon 2$), per taxon. The arguments(s) specified here and their according list lengths activate which taxon-specific demographic parameters are to be hyperparameterized via Ψ/ψ as well as $\zeta/\zeta_s/\zeta_T$ downstream. See also `Details`. At least one is required.

`tau.zeta.prior, tau2.zeta.prior, epsilon.zeta.prior, epsilon2.zeta.prior, NE.zeta.prior`

List of length = `num.partitions` or 1, vector, or non-negative proportion (*i.e.* ≤ 1 and ≥ 0). Hyperprior distribution for ζ_j of $\tau 1$, $\tau 2$, $\varepsilon 1$, $\varepsilon 2$, and N_E, respectively, for each *j*th pulse from 1 to Ψ/ψ, as specified by the corresponding `psi.prior`, and per partition. See also `Details`. Required for each corresponding `psi.prior` specified, unless the maximum value in the corresponding `psi.prior` = 0.

`tau.zeta.total.prior, tau2.zeta.total.prior, epsilon.zeta.total.prior, epsilon2.zeta.total.prior, NE.zeta.total.prior`

List of length = 1, vector, or non-negative proportion (*i.e.* ≤ 1 and ≥ 0). Hyperprior distribution for ζ_T of $\tau 1$, $\tau 2$, $\varepsilon 1$, $\varepsilon 2$, and N_E, respectively. Activates a uniform hyperprior such that each discrete Ψ/ψ value, as specified by the corresponding `psi.prior`, is first weighted with equal hyperprior probability, then all discrete ζ_T values are weighted equally per Ψ/ψ value, and finally every possible associated vector ζ/ζ_s is weighted equally per ζ_T value. See also `Details`.

`tau.shared.prior, tau2.shared.prior, epsilon.shared.prior, epsilon2.shared.prior, NE.shared.prior`

List, vector, or positive value. Prior distribution for the demographic parameter summaries $\tau 1_s$, $\tau 2_s$, $\varepsilon 1_s$, $\varepsilon 2_s$, and N_s, respectively (or $\tau 1$, $\tau 2$, $\varepsilon 1$, $\varepsilon 2$, and N, respectively, if corresponding `psi.prior` specifies Ψ). See also `Details`. Required for each corresponding `psi.prior` specified, unless the maximum value in the corresponding `psi.prior` = 0.

MODEL SPECIFICATIONS

`dirichlet.process` — Logical value. Activates a Dirichlet-process hyperprior that weighs all allowable combinations of Ψ/ψ and ζ/ζ_s according to possible combinations of taxa assignment. See also `Details`.

`idiosyncratic` — Logical value. Allows idiosyncratic taxa that freely vary *i.e.* are ungrouped from any of the pulses, as specified by the `psi.prior` arguments. See also `Details`.

`min.net.tau.zeta.total`, `min.net.tau2.zeta.total`, `min.net.epsilon.zeta.total`, `min.net.epsilon2.zeta.total`, `min.net.NE.zeta.total`, `max.net.tau.zeta.total`, `max.net.tau2.zeta.total`, `max.net.epsilon.zeta.total`, `max.net.epsilon2.zeta.total`, `max.net.NE.zeta.total` — Non-negative proportion (*i.e.* ≤ 1 and ≥ 0). Rule for the minimum/maximum ζ_T value, across all pulses (as specified by the corresponding `psi.prior`) and partitions, for $\tau 1$, $\tau 2$, $\varepsilon 1$, $\varepsilon 2$, and N_E, respectively. See also `Details`.

`min.net.tau.zeta.per.pulse`, `min.net.tau2.zeta.per.pulse`, `min.net.epsilon.zeta.per.pulse`, `min.net.epsilon2.zeta.per.pulse`, `min.net.NE.zeta.per.pulse`, `max.net.tau.zeta.per.pulse`, `max.net.tau2.zeta.per.pulse`, `max.net.epsilon.zeta.per.pulse`, `max.net.epsilon2.zeta.per.pulse`, `max.net.NE.zeta.per.pulse` — List, vector of length = maximum value in corresponding `psi.prior`, or non-negative proportion (*i.e.* ≤ 1 and ≥ 0). Rule for the minimum/maximum ζ_j value of $\tau 1$, $\tau 2$, $\varepsilon 1$, $\varepsilon 2$, and N_E, respectively, for each *j*th pulse from 1 to Ψ/ψ (as specified by the corresponding `psi.prior`) across all partitions. See also `Details`.

`tau.buffer`, `tau2.buffer`, `epsilon.buffer`, `epsilon2.buffer`, `NE.buffer` — Non-negative value or function. Pulse buffer β of the demographic parameter summaries $\tau 1_s$, $\tau 2_s$, $\varepsilon 1_s$, $\varepsilon 2_s$, and N_s, respectively. See also `Details`.

OBJECTS FROM PRECEDING FUNCTIONS

`build.object` — Output from function `build.dice`. See also `Details`.

Arguments from other `Multi-DICE` functions may be included here and are ignored if not applicable.

Details

For more information, see `Multi-DICE` Manual.

`Multi-DICE` cannot currently accommodate models with more than one population per taxon, events aside from population size change, and more than two size change events.

For τ1 and τ2, units are in numbers of generations, and thus may only be positive integers. For ε1 and ε2, units are in ratio of size change from the ancestral effective population size to current effective population size, such that expansions are < 1 and contractions are > 1, and thus may only be positive values. For N_E, unit is in number of effective haploid individuals, and thus may only be positive integers.

For `num.taxa`, `min.net.tau.zeta.per.pulse`, `min.net.tau2.zeta.per.pulse`, `min.net.epsilon.zeta.per.pulse`, `min.net.epsilon2.zeta.per.pulse`, `min.net.NE.zeta.per.pulse`, `max.net.tau.zeta.per.pulse`, `max.net.tau2.zeta.per.pulse`, `max.net.epsilon.zeta.per.pulse`, `max.net.epsilon2.zeta.per.pulse`, and `max.net.NE.zeta.per.pulse`, if list, then all list elements are concatenated to form a single vector, with the ordering within list elements and then between list elements preserved (*e.g.* for list of length = 2, with first list element of length = 2 and second list element of length = 1, the order from first to last is: 1) first vector element in first list element; 2) second vector element in first list element; 3) sole vector element in second list element).

For `tau.psi.prior`, `epsilon.psi.prior`, `NE.psi.prior`, `tau.zeta.prior`, `tau2.zeta.prior`, `epsilon.zeta.prior`, `epsilon2.zeta.prior`, `NE.zeta.prior`, `tau.zeta.total.prior`, `tau2.zeta.total.prior`, `epsilon.zeta.total.prior`, `epsilon2.zeta.total.prior`, `NE.zeta.total.prior`, `tau.shared.prior`, `tau2.shared.prior`, `epsilon.shared.prior`, `epsilon2.shared.prior`, and `NE.shared.prior`, each list element contains an entire individual discrete distribution; if vector, then converted to list of length = 1 with all vector elements comprising the entirety of a single discrete distribution. Per list element, vector elements within (*i.e.* the discrete distribution) do not need to be in any particular order. Relatedly, each vector element is treated as an independent value, thus weighted distributions (*i.e.* not uniform) may be employed by duplicating values (*e.g.* a distribution of `c(0,1,1,1)` signifies 75% probability of drawing “1” and 25% probability of drawing “0”), allowing the specification of any discretized distribution (*e.g.* gamma, beta, log-uniform). Accordingly, a uniform distribution with no gaps for integer values would be of length = range of distribution.

For `num.taxa`, the order of vector elements corresponds to the order of partitions, and for `tau.zeta.prior`, `tau2.zeta.prior`, `epsilon.zeta.prior`, `epsilon2.zeta.prior`, and `NE.zeta.prior`, the order of list elements corresponds to the order of partitions. Additionally, if length = 1 and `num.partitions` > 1, then the sole element is used for all partitions. Similarly, if length < `num.partitions`, then the first element is used for all partitions while ignoring any remaining elements, and if length > `num.partitions`, then the remaining elements beyond length = `num.partitions` are ignored; a caution is provided when the length does not equal 1 or `num.partitions`.

For `min.net.tau.zeta.per.pulse`, `min.net.tau2.zeta.per.pulse`, `min.net.epsilon.zeta.per.pulse`, `min.net.epsilon2.zeta.per.pulse`, `min.net.NE.zeta.per.pulse`, `max.net.tau.zeta.per.pulse`, `max.net.tau2.zeta.per.pulse`, `max.net.epsilon.zeta.per.pulse`,

max.net.epsilon2.zeta.per.pulse, and max.net.NE.zeta.per.pulse, the order of vector elements corresponds to the temporal order, from most recent to most ancient, of pulses (as specified by the corresponding psi.prior), and for tau.shared.prior, tau2.shared.prior, epsilon.shared.prior, epsilon2.shared.prior, and NE.shared.prior, the order of list elements corresponds to the temporal order, from most recent to most ancient, of pulses (as specified by the corresponding psi.prior). Additionally, if length = 1 and maximum value in corresponding psi.prior > 1, then the sole element is used for all pulses. Similarly, if length < maximum value in corresponding psi.prior, then the first element is used for all pulses while ignoring any remaining elements, and if length > maximum value in corresponding psi.prior, then the remaining elements beyond length = maximum value in corresponding psi.prior are ignored. A caution is provided when the length does not equal 1 or maximum value in corresponding psi.prior, except for the shared.prior arguments, which provide a caution if length > 1 and length < maximum value in corresponding psi.prior, but may have additional list elements for idiosyncratic distributions (not utilized here; see play.dice).

If num.partitions=*n*, then rearrangement of bins across taxa within allele frequency classes based on descending order of the relative SNP proportions is not performed to construct the aSFS and taxon-specific inference of demographic parameters is possible. However, in general, more partitions results in more parameter space with respect to taxa samples that must be explored due to a decrease in order-independence and assumed exchangeability, thus multiple-fold more simulations must be conducted to achieve comparable accuracy in hyperparameter estimation as without partitioning.

For tau.psi.prior, epsilon.psi.prior, and NE.psi.prior, distinguishing between Ψ and ψ is accomplished via the corresponding zeta.prior, zeta.total.prior, idiosyncratic setting, and/or min/max.net zeta.total/per.pulse, except for values of 0, which are explicitly for $\psi = 0$ and thus indicate full idiosyncrasy. If list length > 2 for τ and ε or list length > 1 for N_E, then a caution is provided and the remaining elements beyond length = 2 for τ and ε and length = 1 for N_E are ignored. Applies across all partitions, such that it is regardless of partitioning.

For tau.zeta.prior, tau2.zeta.prior, epsilon.zeta.prior, epsilon2.zeta.prior, and NE.zeta.prior, if num.partitions > 1, may be necessary to include “0.0” as a value, but can control ζ_S and ζ_T across partitions via corresponding zeta.total.prior and/or min/max.net zeta.total/per.pulse. Attributes to each partition individually, but proportion values are out of the entirety of taxa dataset, thus if num.partitions > 1, then the upper bound for each partition should be the number of taxa within that partition divided by *n*. When $\psi = \{0, 1\}$, equivalent to hyperprior distribution for ζ_T. If identical across all partitions, it is more computationally efficient to specify only one *i.e.* list of length = 1, or a vector.

For tau.zeta.total.prior, tau2.zeta.total.prior, epsilon.zeta.total.prior, epsilon2.zeta.total.prior, and NE.zeta.total.prior, if length > 1, then remaining elements beyond the first are ignored and a caution is provided.

To build a hyperprior, Multi-DICE first looks if the corresponding zeta.total.prior is specified, then if dirichlet.process=T, and if neither is such case, then all possible combinations of corresponding psi.prior and zeta.prior draws are equally weighted. Therefore, if num.partitions=1, corresponding psi.prior=1, and dirichlet.process=F, then tau.zeta.total.prior, tau2.zeta.total.prior, epsilon.zeta.total.prior, epsilon2.zeta.total.prior, and NE.zeta.total.prior equivalent to corresponding zeta.prior and thus unnecessary to specify.

For tau.shared.prior, tau2.shared.prior, epsilon.shared.prior, epsilon2.shared.prior, and NE.shared.prior, each successive list element/distribution must have a greater minimum and maximum value than its preceding list elements/distributions. If multiple distributions are utilized for every potential pulse and these distributions overlap in their bounds, running time may slow since, in this case, draws are made from all the distributions independently and then checked if abiding by ordering and buffering, with re-draws if not.

When $\psi = 0$ *i.e.* full idiosyncrasy, the arguments idiosyncratic, min.net.tau.zeta.total, min.net.tau2.zeta.total, min.net.epsilon.zeta.total, min.net.epsilon2.zeta.total, min.net.NE.zeta.total, min.net.tau.zeta.per.pulse, min.net.tau2.zeta.per.pulse, min.net.epsilon.zeta.per.pulse, min.net.epsilon2.zeta.per.pulse, and min.net.NE.zeta.per.pulse are ignored.

Multi-DICE proceeds only if a valid draw is allowed given corresponding num.taxa, psi.prior, zeta.prior (*e.g.* minimum value in psi.prior * minimum value in zeta.prior $\leq$ minimum value in num.taxa), zeta.total.prior, shared.prior, idiosyncratic setting, min/max.net zeta.total/per.pulse, and buffer. Importantly, there is no check on if all values in psi.prior have a valid draw, only if there is a valid draw among any of the values. Additionally, the range of a shared.prior argument must be greater than its corresponding $(\Psi/\psi - 1)(2\beta + 1)$. For tau2.shared.prior, consideration must also be given to any overlap with tau.shared.prior.

Value

Returned value is a list object with the following components:

draws.psi — List of matrices. Each list element/matrix is attributed to a hyperparameterized demographic parameter and accordingly named (*i.e.* tau, tau2, epsilon, epsilon2, NE). Per matrix, there is a single column and each cell is the Ψ/ψ value drawn from the corresponding psi.prior for that simulation.

draws.zeta — List of list of matrices. Each list element is attributed to a hyperparameterized demographic parameter and accordingly named (*i.e.* tau, tau2, epsilon, epsilon2, NE). Each of these list elements contain another list of matrices, with the order of these list elements/matrices corresponding to the order of partitions. Per matrix, each cell is the *S* value drawn from the corresponding

zeta.prior for that simulation.

draws.pulse.values — List of matrices. Each list element/matrix is attributed to a hyperparameterized demographic parameter and accordingly named (*i.e.* tau, tau2, epsilon, epsilon2, NE). Per matrix, each cell is the shared pulse value drawn from the corresponding shared.prior for that simulation.

Each row across matrices in draws.psi, draws.zeta, and draws.pulse.values represents an individual simulation and these rows correspond to each other *i.e.* the same row number across matrices refers to the same simulation. For matrices in draws.zeta and draws.pulse.values, the number of columns is the maximum value in the corresponding psi.prior that allows a valid draw, the order of columns corresponds to the temporal order, from most recent to most ancient, of pulses (as specified by the corresponding psi.prior), and non-applicable cells (*i.e.* columns beyond the corresponding draws.psi value) contain the value 0.

Author(s)

Alexander T. Xue

References

Chan YL, Schanzenbach D, Hickerson MJ (2014) Detecting concerted demographic response across community assemblages using hierarchical approximate Bayesian computation. *Molecular Biology and Evolution*, **31**, 2501–2515.

Hickerson MJ, Stahl E, Takebayashi N (2007) msBayes: Pipeline for testing comparative phylogeographic histories using hierarchical approximate Bayesian computation. *BMC bioinformatics*, **8**, 268.

Huang W, Takebayashi N, Qi Y, Hickerson MJ (2011) MTML-msBayes: approximate Bayesian comparative phylogeographic inference from multiple taxa and multiple loci with rate heterogeneity. *BMC bioinformatics*, **12**, 1.

Xue AT (2017) Multi-DICE Manual.

Xue AT, Hickerson MJ (2015) The aggregate site frequency spectrum for comparative population genomic inference. *Molecular Ecology*, **24**, 6223–6240.

Xue AT, Hickerson MJ (*submitted*) Multi-DICE: R package for comparative population genomic inference under multi-taxa hierarchical co-demographic models.

See Also

build.dice, play.dice, dice.sims, dice.aSFS, dice.sumstats

Examples

```
#simplest execution akin to approach in Xue and Hickerson (2015)
roll.dice(num.sims=5, num.taxa=10, tau.psi.prior=c(1),
tau.zeta.prior=c(1:10)/10, tau.shared.prior=c(1000:1000000))
```

```
#simplest execution akin to approach in Xue and Hickerson (2015); ln U
distribution applied on tau.shared.prior, with 100,000 intervals
discretized uniformly across ln(tau.shared.prior)
roll.dice(num.sims=5, num.taxa=10, tau.psi.prior=c(1),
tau.zeta.prior=c(1:10)/10,
tau.shared.prior=exp(c((log(1000)*100000):(log(1000000)*100000))/100000))

#simplest execution akin to approach in Xue and Hickerson (2015); assuming
build.dice was previously performed and the output was directed to object
build.object
roll.dice(num.sims=5, tau.psi.prior=c(1),
tau.shared.prior=c(1000:1000000), build.object=build.object)

#simplest execution akin to approach in software package msBayes
roll.dice(num.sims=5, num.taxa=10, tau.psi.prior=c(1:10),
tau.zeta.prior=c(1:10)/10, tau.shared.prior=c(1000:1000000),
idiosyncratic=F)
```

Idiosyncratic/nuisance prior draws for hierarchical co-demographic model

Description

`play.dice` conducts random draws from user-specified prior distributions for idiosyncratic and nuisance values, as well as determine parameter summary values. Nuisance values are for taxon-specific demographic parameters that are not hyperparameterized (*i.e.* not specified in the corresponding `psi.prior` argument and therefore values are not grouped into shared pulses) and are therefore drawn independently across taxa, similar to idiosyncratic draws. `build.dice` and `roll.dice` are embedded here and are automatically deployed if `build.object`/`roll.object` and `roll.object`, respectively, are not specified.

Usage

```
play.dice(num.sims, num.taxa, num.partitions=1, tau.psi.prior=NULL,
epsilon.psi.prior=NULL, NE.psi.prior=NULL, tau.zeta.prior=NULL,
tau2.zeta.prior=NULL, epsilon.zeta.prior=NULL, epsilon2.zeta.prior=NULL,
NE.zeta.prior=NULL, tau.zeta.total.prior=NULL, tau2.zeta.total.prior=NULL,
epsilon.zeta.total.prior=NULL, epsilon2.zeta.total.prior=NULL,
NE.zeta.total.prior=NULL, tau.shared.prior=NULL, tau2.shared.prior=NULL,
epsilon.shared.prior=NULL, epsilon2.shared.prior=NULL,
NE.shared.prior=NULL, tau.idio.prior=NULL, tau2.idio.prior=NULL,
epsilon.idio.prior=NULL, epsilon2.idio.prior=NULL, NE.idio.prior=NULL,
linked.param=NULL, attached.hyper=NULL, linked.param.partition=NULL,
attached.hyper.pulse=NULL, linked.param.prior=NULL,
linked.param.fixed=NULL, anchor.prior=NULL, change.prior=NULL,
exponential.growth.rate.prior=NULL, exponential.growth.rate.prior2=NULL,
dirichlet.process=F, idiosyncratic=T, min.net.tau.zeta.total=NULL,
min.net.tau2.zeta.total=NULL, min.net.epsilon.zeta.total=NULL,
min.net.epsilon2.zeta.total=NULL, min.net.NE.zeta.total=NULL,
max.net.tau.zeta.total=NULL, max.net.tau2.zeta.total=NULL,
max.net.epsilon.zeta.total=NULL, max.net.epsilon2.zeta.total=NULL,
max.net.NE.zeta.total=NULL, min.net.tau.zeta.per.pulse=NULL,
min.net.tau2.zeta.per.pulse=NULL, min.net.epsilon.zeta.per.pulse=NULL,
min.net.epsilon2.zeta.per.pulse=NULL, min.net.NE.zeta.per.pulse=NULL,
max.net.tau.zeta.per.pulse=NULL, max.net.tau2.zeta.per.pulse=NULL,
max.net.epsilon.zeta.per.pulse=NULL, max.net.epsilon2.zeta.per.pulse=NULL,
max.net.NE.zeta.per.pulse=NULL, tau.buffer=0, tau2.buffer=0,
epsilon.buffer=0, epsilon2.buffer=0, NE.buffer=0,
tau.idiosyncratic.buffer=NULL, tau2.idiosyncratic.buffer=NULL,
epsilon.idiosyncratic.buffer=NULL, epsilon2.idiosyncratic.buffer=NULL,
NE.idiosyncratic.buffer=NULL, idiosyncratic.rule='none', num.changes=1,
flip=F, net.zeta.total=F, net.zeta.per.pulse=F, mean.tau.shared=F,
```

```
mean.tau2.shared=F, mean.epsilon.shared=F, mean.epsilon2.shared=F,
mean.NE.shared=F, mean.tau=F, mean.tau2=F, mean.epsilon=F,
mean.epsilon2=F, mean.NE=F, disp.index.tau.shared=F,
disp.index.tau2.shared=F, disp.index.epsilon.shared=F,
disp.index.epsilon2.shared=F, disp.index.NE.shared=F, disp.index.tau=F,
disp.index.tau2=F, disp.index.epsilon=F, disp.index.epsilon2=F,
disp.index.NE=F, build.object=NULL, roll.object=NULL)
```

Arguments

`num.sims` — Positive integer. Number of simulations. Required.

DATA

`num.taxa` — List, vector of length = `num.partitions`, or positive integer. Number of taxa per partition. Total sum across partitions equals the total number of taxa *n* in dataset. See also `Details`. Required.

`num.partitions` — Positive integer. Number of partitions for taxa in dataset. Allows differential data and model specifications across user-specified taxa groupings, including sampling size of individuals, generation times, demographic syndrome, and taxon-specific nuisance parameter prior distributions. See also `Details`.

PRIORS

`tau.psi.prior, epsilon.psi.prior, NE.psi.prior` — List, vector, or non-negative integer. Hyperprior distribution for Ψ/ψ of τ, ε, and N_E, respectively. For τ and ε, if list of length = 2, then the first list element applies to the first more recent size change event (*e.g.* $\tau 1$, $\varepsilon 1$) and the second list element applies to the second more ancient size change event (*e.g.* $\tau 2$, $\varepsilon 2$), per taxon. The arguments(s) specified here and their according list lengths activate which taxon-specific demographic parameters are to be hyperparameterized via Ψ/ψ as well as $\zeta/\zeta_s/\zeta_T$ downstream. See also `Details`. At least one is required.

`tau.zeta.prior, tau2.zeta.prior, epsilon.zeta.prior, epsilon2.zeta.prior, NE.zeta.prior` — List of length = `num.partitions` or 1, vector, or non-negative proportion (*i.e.* ≤ 1 and ≥ 0). Hyperprior distribution for ζ_j of $\tau 1$, $\tau 2$, $\varepsilon 1$, $\varepsilon 2$, and N_E, respectively, for each *j*th pulse from 1 to Ψ/ψ, as specified by the corresponding `psi.prior`, and per partition. See also `Details`. Required for each corresponding `psi.prior` specified, unless the maximum value in the corresponding `psi.prior` = 0.

`tau.zeta.total.prior`, `tau2.zeta.total.prior`, `epsilon.zeta.total.prior`, `epsilon2.zeta.total.prior`, `NE.zeta.total.prior`

List of length = 1, vector, or non-negative proportion (*i.e.* ≤ 1 and ≥ 0). Hyperprior distribution for ζ_T of $\tau 1$, $\tau 2$, $\varepsilon 1$, $\varepsilon 2$, and N_E, respectively. Activates a uniform hyperprior such that each discrete Ψ/ψ value, as specified by the corresponding `psi.prior`, is first weighted with equal hyperprior probability, then all discrete ζ_T values are weighted equally per Ψ/ψ value, and finally every possible associated vector ζ/ζ_s is weighted equally per ζ_T value. See also `Details`.

`tau.shared.prior`, `tau2.shared.prior`, `epsilon.shared.prior`, `epsilon2.shared.prior`, `NE.shared.prior`

List, vector, or positive value. Prior distribution for the demographic parameter summaries $\tau 1_s$, $\tau 2_s$, $\varepsilon 1_s$, $\varepsilon 2_s$, and N_s, respectively (or $\tau 1$, $\tau 2$, $\varepsilon 1$, $\varepsilon 2$, and N, respectively, if corresponding `psi.prior` specifies Ψ). See also `Details`. Required for each corresponding `psi.prior` specified, unless the maximum value in the corresponding `psi.prior` = 0.

`tau.idio.prior`, `tau2.idio.prior`, `epsilon.idio.prior`, `epsilon2.idio.prior`, `NE.idio.prior`

List of length = `num.partitions` or 1, vector, or positive value. Prior distribution for the taxon-specific demographic parameters $\tau 1_i$, $\tau 2_i$, $\varepsilon 1_i$, $\varepsilon 2_i$, and N_i, respectively for idiosyncratic values, and $\tau 1$, $\tau 2$, $\varepsilon 1$, $\varepsilon 2$, and N, respectively for nuisance values. See also `Details`.

`linked.param`, `attached.hyper`

List, vector, or character string, with possible values being the names of the demographic parameters (*i.e.* `"tau"`, `"tau2"`, `"epsilon"`, `"epsilon2"`, `"NE"`). Activates nuisance parameters in `linked.param` to have prior distributions be linked to hyperparameterized demographic parameters in `attached.hyper`, such that prior distributions may differentiate across pulses and idiosyncratic taxa with respect to the hyperparameterized demographic parameter in `attached.hyper`. See also `Details`.

`linked.param.partition`, `attached.hyper.pulse`

List of length = length of `linked.param` or 1, vector, or positive integer. The partitions in the linked nuisance parameter, and the pulses in the attached hyperparameterized demographic parameter, for which each element in `linked.param` and `attached.hyper`, respectively, applies. Each list element may contain multiple partitions/pulses, respectively. See also `Details`.

`linked.param.prior`

List of length = length of `linked.param` or 1, vector, or positive value. Prior distribution for the nuisance demographic parameter in each element of `linked.param`. See also `Details`.

`linked.param.fixed`

List, vector of length = length of `linked.param`, or logical value. Activates a fixed nuisance demographic parameter value for all taxa to which the corresponding elements in `linked.param.partition` and `attached.hyper.pulse` apply. See also `Details`.

`anchor.prior, change.prior`

List, vector, or positive integer. Prior distribution for $\tau 2$ based on its difference δ with $\tau 1$. This difference value can be assigned to synchronous/shared pulses in $\tau 1_s$, as specified by `tau.psi.prior`, and accordingly inferred as a parameter summary vector δ_s (`anchor.prior`), or applied independently across taxa as an idiosyncratic or nuisance value (`change.prior`). See also `Details`.

`exponential.growth.rate.prior, exponential.growth.rate.prior2`

List of length = `num.partitions` or 1, vector, or double value. Prior distribution for the nuisance taxon-specific parameters *r*1 and *r*2, respectively. Activates exponential growth model $N_\tau = N_0 * e^{(r * \tau)}$ for the first and second event, respectively, instead of instantaneous growth. Negative values indicate expansion and positive values indicate contraction. See also `Details`.

MODEL SPECIFICATIONS

`dirichlet.process`

Logical value. Activates a Dirichlet-process hyperprior that weighs all allowable combinations of Ψ/ψ and ζ/ζ_s according to possible combinations of taxa assignment. See also `Details`.

`idiosyncratic`

Logical value. Allows idiosyncratic taxa that freely vary *i.e.* are ungrouped from any of the pulses, as specified by the `psi.prior` arguments. See also `Details`.

`min.net.tau.zeta.total, min.net.tau2.zeta.total, min.net.epsilon.zeta.total, min.net.epsilon2.zeta.total, min.net.NE.zeta.total, max.net.tau.zeta.total, max.net.tau2.zeta.total, max.net.epsilon.zeta.total, max.net.epsilon2.zeta.total, max.net.NE.zeta.total`

Non-negative proportion (*i.e.* ≤ 1 and ≥ 0). Rule for the minimum/maximum ζ_T value, across all pulses (as specified by the corresponding `psi.prior`) and partitions, for $\tau 1$, $\tau 2$, $\varepsilon 1$, $\varepsilon 2$, and N_E, respectively. See also `Details`.

`min.net.tau.zeta.per.pulse, min.net.tau2.zeta.per.pulse, min.net.epsilon.zeta.per.pulse,`

List, vector of length = maximum value in corresponding `psi.prior`, or non-negative proportion (*i.e.* ≤ 1 and ≥ 0). Rule for the minimum/maximum ζ_j value of $\tau 1$, $\tau 2$, $\varepsilon 1$, $\varepsilon 2$, and N_E,

`min.net.epsilon2.zeta.per.pulse`, `min.net.NE.zeta.per.pulse`, `max.net.tau.zeta.per.pulse`, `max.net.tau2.zeta.per.pulse`, `max.net.epsilon.zeta.per.pulse`, `max.net.epsilon2.zeta.per.pulse`, `max.net.NE.zeta.per.pulse`

respectively, for each *j*th pulse from 1 to Ψ/ψ (as specified by the corresponding `psi.prior`) across all partitions. See also `Details`.

`tau.buffer`, `tau2.buffer`, `epsilon.buffer`, `epsilon2.buffer`, `NE.buffer`

Non-negative value or function. Pulse buffer β of the demographic parameter summaries $\tau 1_s$, $\tau 2_s$, $\varepsilon 1_s$, $\varepsilon 2_s$, and N_s, respectively. See also `Details`.

`tau.idiosyncratic.buffer`, `tau2.idiosyncratic.buffer`, `epsilon.idiosyncratic.buffer`, `epsilon2.idiosyncratic.buffer`, `NE.idiosyncratic.buffer`

Non-negative value or function. Idiosyncratic buffer β_i of the idiosyncratic taxon-specific demographic parameters $\tau 1_i$, $\tau 2_i$, $\varepsilon 1_i$, $\varepsilon 2_i$, and N_i, respectively. See also `Details`.

`idiosyncratic.rule`

Character string with possible values `"recent"` and `"ancient"`. Activates rule forcing all idiosyncratic taxa to have values less than the first shared pulse, or values greater than the last shared pulse, respectively. Any other values results in no such rules being placed on idiosyncratic taxa. See also `Details`.

`num.changes`

List, vector of length = `num.partitions`, or value of 1 or 2. Number of demographic change events per taxon. See also `Details`.

`flip`

List, vector of length = `num.partitions`, or logical value. Activates $\tau 2$ to be more recent than $\tau 1$. See also `Details`.

PARAMETER SUMMARIES

`net.zeta.total`, `net.zeta.per.pulse`

Logical value. Activates output of ζ_T and the vector ζ/ζ_s across partitions, respectively, as a list element/matrix in the `roll.object` list element of the final output, for downstream estimation. Rows of the matrix correspond to individual simulations, which correspond to rows of other matrices in the `roll.object` list element of the final output, and each cell is the aforementioned value.

`mean.tau.shared`, `mean.tau2.shared`, `mean.epsilon.shared`, `mean.epsilon2.shared`, `mean.NE.shared`

Logical value. Activates output of $E(\tau 1_s)$, $E(\tau 2_s)$, $E(\varepsilon 1_s)$, $E(\varepsilon 2_s)$, and $E(N_s)$ weighted by the vector ζ/ζ_s, respectively, as a list element/matrix in the `roll.object` list element of the final output, for downstream estimation. Rows of the matrix correspond to individual simulations, which correspond to rows

	of other matrices in the `roll.object` list element of the final output, and each cell is the aforementioned value. See also `Details`.
`mean.tau, mean.tau2, mean.epsilon, mean.epsilon2, mean.NE`	Logical value. Activates output of E(τ1), E(τ2), E(ε1), E(ε2), and E(*N*), respectively, as a list element/matrix in the `roll.object` list element of the final output, for downstream estimation. Rows of the matrix correspond to individual simulations, which correspond to rows of other matrices in the `roll.object` list element of the final output, and each cell is the aforementioned value.
`disp.index.tau.shared, disp.index.tau2.shared, disp.index.epsilon.shared, disp.index.epsilon2.shared, disp.index.NE.shared`	Logical value. Activates output of $\Omega(\tau 1_s)$, $\Omega(\tau 2_s)$, $\Omega(\varepsilon 1_s)$, $\Omega(\varepsilon 2_s)$, and $\Omega(N_s)$ weighted by the vector ζ/ζ_s, respectively, as a list element/matrix in the `roll.object` list element of the final output, for downstream estimation. Rows of the matrix correspond to individual simulations, which correspond to rows of other matrices in the `roll.object` list element of the final output, and each cell is the aforementioned value. See also `Details`.
`disp.index.tau, disp.index.tau2, disp.index.epsilon, disp.index.epsilon2, disp.index.NE`	Logical value. Activates output of Ω(τ1), Ω(τ2), Ω(ε1), Ω(ε2), and Ω(*N*), respectively, as a list element/matrix in the `roll.object` list element of the final output, for downstream estimation. Rows of the matrix correspond to individual simulations, which correspond to rows of other matrices in the `roll.object` list element of the final output, and each cell is the aforementioned value. See also `Details`.

OBJECTS FROM PRECEDING FUNCTIONS

`build.object`	Output from function `build.dice`. See also `Details`.
`roll.object`	Output from function `roll.dice`. See also `Details`.

Arguments from other `Multi-DICE` functions may be included here and are ignored if not applicable.

Details

For more information, see `Multi-DICE` Manual.

`Multi-DICE` cannot currently accommodate models with more than one population per taxon, events aside from population size change, and more than two size change events.

For $\tau 1$ and $\tau 2$, units are in numbers of generations, and thus may only be positive integers. For $\varepsilon 1$ and $\varepsilon 2$, units are in ratio of size change from the ancestral effective population size to current effective population size, such that expansions are < 1 and contractions are > 1, and thus may only be positive values. For N_E, unit is in number of effective haploid individuals, and thus may only be positive integers.

For `num.taxa`, `linked.param`, `attached.hyper`, `linked.param.fixed`, `min.net.tau.zeta.per.pulse`, `min.net.tau2.zeta.per.pulse`, `min.net.epsilon.zeta.per.pulse`, `min.net.epsilon2.zeta.per.pulse`, `min.net.NE.zeta.per.pulse`, `max.net.tau.zeta.per.pulse`, `max.net.tau2.zeta.per.pulse`, `max.net.epsilon.zeta.per.pulse`, `max.net.epsilon2.zeta.per.pulse`, `max.net.NE.zeta.per.pulse`, `num.changes`, and `flip`, if list, then all list elements are concatenated to form a single vector, with the ordering within list elements and then between list elements preserved (*e.g.* for list of length = 2, with first list element of length = 2 and second list element of length = 1, the order from first to last is: 1) first vector element in first list element; 2) second vector element in first list element; 3) sole vector element in second list element).

For `tau.psi.prior`, `epsilon.psi.prior`, `NE.psi.prior`, `tau.zeta.prior`, `tau2.zeta.prior`, `epsilon.zeta.prior`, `epsilon2.zeta.prior`, `NE.zeta.prior`, `tau.zeta.total.prior`, `tau2.zeta.total.prior`, `epsilon.zeta.total.prior`, `epsilon2.zeta.total.prior`, `NE.zeta.total.prior`, `tau.shared.prior`, `tau2.shared.prior`, `epsilon.shared.prior`, `epsilon2.shared.prior`, `NE.shared.prior`, `tau.idio.prior`, `tau2.idio.prior`, `epsilon.idio.prior`, `epsilon2.idio.prior`, `NE.idio.prior`, `linked.param.prior`, `anchor.prior`, `change.prior`, `exponential.growth.rate.prior`, and `exponential.growth.rate.prior2`, each list element contains an entire individual discrete distribution; if vector, then converted to list of length = 1 with all vector elements comprising the entirety of a single discrete distribution. Per list element, vector elements within (*i.e.* the discrete distribution) do not need to be in any particular order. Relatedly, each vector element is treated as an independent value, thus weighted distributions (*i.e.* not uniform) may be employed by duplicating values (*e.g.* a distribution of `c(0,1,1,1)` signifies 75% probability of drawing “1” and 25% probability of drawing “0”), allowing the specification of any discretized distribution (*e.g.* gamma, beta, log-uniform). Accordingly, a uniform distribution with no gaps for integer values would be of length = range of distribution.

For `num.taxa`, `num.changes`, and `flip`, the order of vector elements corresponds to the order of partitions, and for `tau.zeta.prior`, `tau2.zeta.prior`, `epsilon.zeta.prior`, `epsilon2.zeta.prior`, `NE.zeta.prior`, `tau.idio.prior`, `tau2.idio.prior`, `epsilon.idio.prior`, `epsilon2.idio.prior`, `NE.idio.prior`, `change.prior`, `exponential.growth.rate.prior`, and `exponential.growth.rate.prior2`, the order of list elements corresponds to the order of partitions. Additionally, if length = 1 and `num.partitions` > 1, then the sole element is used for all partitions. Similarly, if length < `num.partitions`, then the first element is used for all partitions while ignoring any remaining elements, and if length >

num.partitions, then the remaining elements beyond length = num.partitions are ignored; a caution is provided when the length does not equal 1 or num.partitions.

For min.net.tau.zeta.per.pulse, min.net.tau2.zeta.per.pulse, min.net.epsilon.zeta.per.pulse, min.net.epsilon2.zeta.per.pulse, min.net.NE.zeta.per.pulse, max.net.tau.zeta.per.pulse, max.net.tau2.zeta.per.pulse, max.net.epsilon.zeta.per.pulse, max.net.epsilon2.zeta.per.pulse, and max.net.NE.zeta.per.pulse, the order of vector elements corresponds to the temporal order, from most recent to most ancient, of pulses (as specified by the corresponding psi.prior), and for tau.shared.prior, tau2.shared.prior, epsilon.shared.prior, epsilon2.shared.prior, NE.shared.prior, and anchor.prior, the order of list elements corresponds to the temporal order, from most recent to most ancient, of pulses (as specified by the corresponding psi.prior). Additionally, if length = 1 and maximum value in corresponding psi.prior > 1, then the sole element is used for all pulses. Similarly, if length < maximum value in corresponding psi.prior, then the first element is used for all pulses while ignoring any remaining elements. For the zeta.per.pulse arguments, if length > maximum value in corresponding psi.prior, then the remaining elements beyond length = maximum value in corresponding psi.prior are ignored; a caution is provided when the length does not equal 1 or maximum value in corresponding psi.prior. For the shared.prior arguments and anchor.prior, there may be additional list elements for idiosyncratic distributions, such that any total length = 1, maximum value in corresponding psi.prior, maximum value in corresponding psi.prior + 1, or maximum value in corresponding psi.prior + num.partitions, are allowed; for any other lengths, a caution is provided and excess elements beyond the highest acceptable length are ignored. See below for more information about adding idiosyncratic distributions to these arguments.

If num.partitions=*n*, then rearrangement of bins across taxa within allele frequency classes based on descending order of the relative SNP proportions is not performed to construct the aSFS and taxon-specific inference of demographic parameters is possible. However, in general, more partitions results in more parameter space with respect to taxa samples that must be explored due to a decrease in order-independence and assumed exchangeability, thus multiple-fold more simulations must be conducted to achieve comparable accuracy in hyperparameter estimation as without partitioning.

For tau.psi.prior, epsilon.psi.prior, and NE.psi.prior, distinguishing between Ψ and ψ is accomplished via the corresponding zeta.prior, zeta.total.prior, idiosyncratic setting, and/or min/max.net zeta.total/per.pulse, except for values of 0, which are explicitly for $\psi = 0$ and thus indicate full idiosyncrasy. If list length > 2 for τ and ε or list length > 1 for N_E, then a caution is provided and the remaining elements beyond length = 2 for τ and ε and length = 1 for N_E are ignored. Applies across all partitions, such that it is regardless of partitioning.

For tau.zeta.prior, tau2.zeta.prior, epsilon.zeta.prior, epsilon2.zeta.prior, and NE.zeta.prior, if num.partitions > 1, may be necessary to include “0.0” as a value, but can control ζ_S and ζ_T across partitions via corresponding zeta.total.prior and/or min/max.net

zeta.total/per.pulse. Attributes to each partition individually, but proportion values are out of the entirety of taxa dataset, thus if num.partitions > 1, then the upper bound for each partition should be the number of taxa within that partition divided by *n*. When $\psi = \{0, 1\}$, equivalent to hyperprior distribution for ζ_T. If identical across all partitions, it is more computationally efficient to specify only one *i.e.* list of length = 1, or a vector.

For tau.zeta.total.prior, tau2.zeta.total.prior, epsilon.zeta.total.prior, epsilon2.zeta.total.prior, and NE.zeta.total.prior, if length > 1, then remaining elements beyond the first are ignored and a caution is provided.

To build a hyperprior, Multi-DICE first looks if the corresponding zeta.total.prior is specified, then if dirichlet.process=T, and if neither is such case, then all possible combinations of corresponding psi.prior and zeta.prior draws are equally weighted. Therefore, if num.partitions=1, corresponding psi.prior=1, and dirichlet.process=F, then tau.zeta.total.prior, tau2.zeta.total.prior, epsilon.zeta.total.prior, epsilon2.zeta.total.prior, and NE.zeta.total.prior equivalent to corresponding zeta.prior and thus unnecessary to specify.

For tau.shared.prior, tau2.shared.prior, epsilon.shared.prior, epsilon2.shared.prior, and NE.shared.prior, each successive list element/distribution must have a greater minimum and maximum value than its preceding list elements/distributions. If multiple distributions are utilized for every potential pulse and these distributions overlap in their bounds, running time may slow since, in this case, draws are made from all the distributions independently and then checked if abiding by ordering and buffering, with re-draws if not.

For each of tau, tau2, epsilon, epsilon2, and NE, if hyperparameterized via its corresponding psi.prior while idiosyncratic=T or a 0 value is in said psi.prior, then a corresponding idiosyncratic prior is required. Multi-DICE first looks in change.prior (for $\tau 2$ only), then the corresponding idio.prior, and finally the corresponding shared.prior. If a shared.prior argument is utilized, the additional list elements beyond length = maximum value in corresponding psi.prior are considered; if the number of list elements is $\leq$ maximum value in corresponding psi.prior, then only the first list element is considered. These list elements undergo the same specifications as aforementioned for the corresponding idio.prior.

For each of tau, tau2, epsilon, epsilon2, and NE, if not hyperparameterized yet part of the specified model (*i.e.* tau, epsilon, and NE are always part of the model, and tau2 and epsilon2 are part of the model when num.changes=2), then a corresponding nuisance prior is required. A nuisance prior differs from an idiosyncratic prior in that variation in the respective nuisance parameter is being considered while not of interest with respect to hyperparameterization (*i.e.* variability in values across taxa governed hierarchically), whereas an idiosyncratic prior is still governed by hyperparameters in coordination with the corresponding shared prior. Multi-DICE first looks in anchor.prior (for $\tau 2$ only), next change.prior (for $\tau 2$ only), afterward linked.param.prior (if applicable; see below for more information), then the corresponding idio.prior, and finally the

corresponding `shared.prior`. If `linked.param.prior` is utilized for a particular nuisance parameter, either the corresponding `idio.prior` or `shared.prior` must still be specified even if all taxa are always covered by `linked.param.prior` across all simulations. If a `shared.prior` argument is utilized, it undergoes the same specifications as aforementioned for the corresponding `idio.prior`.

It is highly recommended that prior distribution bounds are as far apart as possible between $\tau 1$ and $\tau 2$, ideally mutually exclusive/non-overlapping. Otherwise, there could be a stop of operation due to an invalid/incompatible draw, computational lag, and/or statistical bias on prior distributions.

For `tau.buffer`, `tau2.buffer`, `epsilon.buffer`, `epsilon2.buffer`, and `NE.buffer`, β buffers are applied to draws of shared pulse values, thus affecting the corresponding shared prior; the corresponding idiosyncratic prior, from which draws are subsequently made, is accordingly affected by the shared pulse buffers. For `tau.idiosyncratic.buffer`, `tau2.idiosyncratic.buffer`, `epsilon.idiosyncratic.buffer`, `epsilon2.idiosyncratic.buffer`, and `NE.idiosyncratic.buffer`, β_i idiosyncratic buffers are then applied to idiosyncratic draws, which additionally affect the corresponding idiosyncratic prior that had already been initially transformed by the shared pulse buffers. In other words, β buffers idiosyncratic taxa from shared pulse values, and β_i buffers idiosyncratic taxa from each other. For example, if $n = 10$, $\psi_{\tau 1} = 2$, $\zeta_{\tau 1,T} = 0.8$, $\beta = 10{,}000$, $\beta_i = 1{,}000$, $\tau 1 \sim U(1{,}000, 100{,}000)$, $\tau 1_s = \{11{,}000, 12{,}001\}$, and $\tau 1_i = \{99{,}000\}$ after the first idiosyncratic draw, then given that the synchronous/shared pulse buffers result in invalid draws from $\sim U(1{,}000, 13{,}001)$ and the first idiosyncratic draw buffer results in invalid draws from $\sim U(98{,}000, 100{,}000)$, the remaining second idiosyncratic draw would be from the resulting transformed prior distribution $\tau 1_i \sim U(13{,}002, 97{,}999)$. A caution is provided if an `idiosyncratic.buffer` corresponding to a specified `psi.prior` is not specified. Buffers cannot be deployed for nuisance draws (though see below when $\psi = 0$).

For `num.changes`, if < 1, then there is a stop of operation, and if > 2, then converted to `num.changes=2` and a caution is provided. Notably, `num.changes` is only utilized under `num.changes=2` to activate nuisance demographic parameters for a second event; one event is the default, thus `num.changes=1` is unnecessary, and second list elements for `tau.psi.prior` and `epsilon.psi.prior` activate respective hyperparameterized second-event demographic parameters, thus if both are specified, then `num.changes=2` is unnecessary (though must be specified for `dice.sims`).

If `flip=T`, $\tau 2$ and $\varepsilon 2$ still refers to the second specified event, which in this case is the more recent one.

When $\psi = 0$ *i.e.* full idiosyncrasy, the arguments `idiosyncratic`, `min.net.tau.zeta.total`, `min.net.tau2.zeta.total`, `min.net.epsilon.zeta.total`, `min.net.epsilon2.zeta.total`, `min.net.NE.zeta.total`, `min.net.tau.zeta.per.pulse`, `min.net.tau2.zeta.per.pulse`, `min.net.epsilon.zeta.per.pulse`, `min.net.epsilon2.zeta.per.pulse`, `min.net.NE.zeta.per.pulse`, and `idiosyncratic.rule` are ignored, and the arguments

tau.idiosyncratic.buffer, tau2.idiosyncratic.buffer, epsilon.idiosyncratic.buffer, epsilon2.idiosyncratic.buffer, and NE.idiosyncratic.buffer are activated (even if idiosyncratic=F).

Multi-DICE proceeds only if a valid draw is allowed given corresponding num.taxa, psi.prior, zeta.prior (*e.g.* minimum value in psi.prior * minimum value in zeta.prior $\leq$ minimum value in num.taxa), zeta.total.prior, shared prior, idiosyncratic prior, idiosyncratic setting, min/max.net zeta.total/per.pulse, buffer, and idiosyncratic.buffer. Importantly, there is no check on if all values in psi.prior have a valid draw, only if there is a valid draw among any of the values. Additionally, the range of a shared prior must be greater than its corresponding $(\Psi/\psi - 1)(2\beta + 1)$, and the range of an idiosyncratic prior ought to be greater than its corresponding $(\Psi/\psi)(2\beta + 1) + (\sigma - 1)(2\beta_i + 1)$ across every possible combination of Ψ/ψ and σ and with consideration given to idiosyncratic.rule. For tau2.shared.prior and tau2.idio.prior, consideration must also be given to any overlap with tau.shared.prior and tau.idio.prior, respectively.

Value

Returned value is a list object with two elements, roll.object and sim.specs. The list element roll.object is similar to the output of the function roll.dice, except parameter summaries derived here are added (*e.g.* Ω, E(), δ_s; see above for more information); these represent values of interest for estimation. The list element sim.specs contains a list of matrices, with each list element/matrix attributed to a taxon-specific parameter and accordingly named (*i.e.* tau, tau2, epsilon, epsilon2, NE, exponential.growth.rate.prior, exponential.growth.rate.prior2). Per matrix, each row represents an individual simulation, each column represents an independent taxon to be simulated, and each cell is the according demographic parameter value to be used for simulation. Rows and columns across matrices correspond to each other *i.e.* across matrices, the same row number refers to the same simulation and the same column number refers to the same simulated independent taxon.

Author(s)

Alexander T. Xue

References

Chan YL, Schanzenbach D, Hickerson MJ (2014) Detecting concerted demographic response across community assemblages using hierarchical approximate Bayesian computation. *Molecular Biology and Evolution*, **31**, 2501–2515.

Hickerson MJ, Stahl E, Takebayashi N (2007) msBayes: Pipeline for testing comparative phylogeographic histories using hierarchical approximate Bayesian computation. *BMC bioinformatics*, **8**, 268.

Huang W, Takebayashi N, Qi Y, Hickerson MJ (2011) MTML-msBayes: approximate Bayesian comparative phylogeographic inference from multiple taxa and multiple loci with rate heterogeneity. *BMC bioinformatics*, **12**, 1.

Xue AT (2017) Multi-DICE Manual.
Xue AT, Hickerson MJ (2015) The aggregate site frequency spectrum for comparative population genomic inference. *Molecular Ecology*, **24**, 6223–6240.
Xue AT, Hickerson MJ (*submitted*) Multi-DICE: R package for comparative population genomic inference under multi-taxa hierarchical co-demographic models.

See Also

`build.dice`, `roll.dice`, `dice.sims`, `dice.aSFS`, `dice.sumstats`

Examples

```
#simplest execution akin to approach in Xue and Hickerson (2015)
play.dice(num.sims=5, num.taxa=10, tau.psi.prior=c(1),
tau.zeta.prior=c(1:10)/10, tau.shared.prior=c(1000:1000000),
epsilon.idio.prior=c(1000:10000)/100000, NE.idio.prior=c(1000:100000))

#simplest execution akin to approach in Xue and Hickerson (2015); ln U
distribution applied on tau.shared.prior, with 100,000 intervals
discretized uniformly across ln(tau.shared.prior)
play.dice(num.sims=5, num.taxa=10, tau.psi.prior=c(1),
tau.zeta.prior=c(1:10)/10,
tau.shared.prior=exp(c((log(1000)*100000):(log(1000000)*100000))/100000),
epsilon.idio.prior=c(1000:10000)/100000, NE.idio.prior=c(1000:100000))

#simplest execution akin to approach in Xue and Hickerson (2015); assuming
roll.dice was previously performed and the output was directed to object
roll.object
play.dice(num.sims=5, num.taxa=10, tau.psi.prior=c(1),
tau.idio.prior=c(1000:1000000), epsilon.idio.prior=c(1000:10000)/100000,
NE.idio.prior=c(1000:100000)), roll.object=roll.object)

#simplest execution akin to approach in software package msBayes
play.dice(num.sims=5, num.taxa=10, tau.psi.prior=c(1:10),
tau.zeta.prior=c(1:10)/10, tau.shared.prior=c(1000:1000000),
epsilon.idio.prior=c(1000:10000)/100000, NE.idio.prior=c(1000:100000),
idiosyncratic=F)
```

Simulating under hierarchical co-demographic model

Description

`dice.sims` is a wrapper function for the command-line program `fastsimcoal2` that performs multi-taxa coalescent simulation of per-taxon summary statistics under a unified hierarchical co-demographic model as specified by `build.dice`, `roll.dice`, and `play.dice`, which are also embedded here and are automatically deployed if `build.object`/`roll.object`/`play.object`, `roll.object`/`play.object`, and `play.object`, respectively, are not specified. `fastsimcoal2` must be separately user-installed. `bash` commands are called upon here, thus `dice.sims` can run only within a `bash` terminal environment (*e.g.* Mac, Linux).

Usage

```
dice.sims(num.sims, num.taxa, num.partitions=1, num.haploid.samples,
num.ind.sites=NULL, num.SNPs=NULL, length.seq=NULL, folded=T,
sampling.times=NULL, gen.times=NULL, tau.psi.prior=NULL,
epsilon.psi.prior=NULL, NE.psi.prior=NULL, tau.zeta.prior=NULL,
tau2.zeta.prior=NULL, epsilon.zeta.prior=NULL, epsilon2.zeta.prior=NULL,
NE.zeta.prior=NULL, tau.zeta.total.prior=NULL, tau2.zeta.total.prior=NULL,
epsilon.zeta.total.prior=NULL, epsilon2.zeta.total.prior=NULL,
NE.zeta.total.prior=NULL, tau.shared.prior=NULL, tau2.shared.prior=NULL,
epsilon.shared.prior=NULL, epsilon2.shared.prior=NULL,
NE.shared.prior=NULL, tau.idio.prior=NULL, tau2.idio.prior=NULL,
epsilon.idio.prior=NULL, epsilon2.idio.prior=NULL, NE.idio.prior=NULL,
linked.param=NULL, attached.hyper=NULL, linked.param.partition=NULL,
attached.hyper.pulse=NULL, linked.param.prior=NULL,
linked.param.fixed=NULL, anchor.prior=NULL, change.prior=NULL,
exponential.growth.rate.prior=NULL, exponential.growth.rate.prior2=NULL,
mut.rate.prior=NULL, dirichlet.process=F, idiosyncratic=T,
min.net.tau.zeta.total=NULL, min.net.tau2.zeta.total=NULL,
min.net.epsilon.zeta.total=NULL, min.net.epsilon2.zeta.total=NULL,
min.net.NE.zeta.total=NULL, max.net.tau.zeta.total=NULL,
max.net.tau2.zeta.total=NULL, max.net.epsilon.zeta.total=NULL,
max.net.epsilon2.zeta.total=NULL, max.net.NE.zeta.total=NULL,
min.net.tau.zeta.per.pulse=NULL, min.net.tau2.zeta.per.pulse=NULL,
min.net.epsilon.zeta.per.pulse=NULL, min.net.epsilon2.zeta.per.pulse=NULL,
min.net.NE.zeta.per.pulse=NULL, max.net.tau.zeta.per.pulse=NULL,
max.net.tau2.zeta.per.pulse=NULL, max.net.epsilon.zeta.per.pulse=NULL,
max.net.epsilon2.zeta.per.pulse=NULL, max.net.NE.zeta.per.pulse=NULL,
tau.buffer=0, tau2.buffer=0, epsilon.buffer=0, epsilon2.buffer=0,
NE.buffer=0, tau.idiosyncratic.buffer=NULL,
tau2.idiosyncratic.buffer=NULL, epsilon.idiosyncratic.buffer=NULL,
```

```
epsilon2.idiosyncratic.buffer=NULL, NE.idiosyncratic.buffer=NULL,
idiosyncratic.rule='none', num.changes=1, flip=F, net.zeta.total=F,
net.zeta.per.pulse=F, mean.tau.shared=F, mean.tau2.shared=F,
mean.epsilon.shared=F, mean.epsilon2.shared=F, mean.NE.shared=F,
mean.tau=F, mean.tau2=F, mean.epsilon=F, mean.epsilon2=F, mean.NE=F,
disp.index.tau.shared=F, disp.index.tau2.shared=F,
disp.index.epsilon.shared=F, disp.index.epsilon2.shared=F,
disp.index.NE.shared=F, disp.index.tau=F, disp.index.tau2=F,
disp.index.epsilon=F, disp.index.epsilon2=F, disp.index.NE=F, fsc2path,
messages.sims=NULL, output.directory, append.sims=F, keep.taxa.draws=F,
output.hyper.draws=T, output.taxa.draws=F, keep.fsc2.files=F,
build.object=NULL, roll.object=NULL, play.object=NULL)
```

Arguments

`num.sims` — Positive integer. Number of simulations. Required.

DATA

`num.taxa` — List, vector of length = `num.partitions`, or positive integer. Number of taxa per partition. Total sum across partitions equals the total number of taxa *n* in dataset. See also `Details`. Required.

`num.partitions` — Positive integer. Number of partitions for taxa in dataset. Allows differential data and model specifications across user-specified taxa groupings, including sampling size of individuals, generation times, demographic syndrome, and taxon-specific nuisance parameter prior distributions. See also `Details`.

`num.haploid.samples` — List, vector of length = `num.partitions`, or positive integer. Number of haploid samples per partition. See also `Details`. Required.

`num.ind.sites, num.SNPs, length.seq` — List, vector of length = `num.partitions`, or positive integer. Data sampling level per partition, in number of independent sites/SNPs using the `fastsimcoal2 FREQ` simulation model, in number of independent SNPs using the `fastsimcoal2 SNP` simulation model, and in sequence length using the `fastsimcoal2 SNP` simulation model, respectively. For each taxon, the former two simulate the SFS based on independent sites, while the latter one simulates single-sequence summary statistics. See also `Details`. At least one is required.

`folded` — List, vector of length = `num.partitions`, or logical value.

Activates folding of the SFS per partition; ignored if single-sequence summary statistics are simulated. See also `Details`.

`sampling.times`

List, vector of length = `num.partitions`, or non-negative integer. Sampling times per partition. Allows simulation of time-series or ancient data. Unnecessary if all data are collected simultaneously and in present-day, since this is assumed by default. See also `Details`.

`gen.times`

List, vector of length = `num.partitions` or *n*, or positive value. Generation times per partition or per taxon, in units of years per generation. If including multiple generation times within a partition (*i.e.* vector of length = *n*), generation times are randomly assigned to sets of parameter draws, though the order of output simulation files corresponds to the user-specified order (*i.e.* `dice.simulations1` corresponds to the first element, `dice.simulations2` to the second, etc.). See also `Details`.

PRIORS

`tau.psi.prior, epsilon.psi.prior, NE.psi.prior`

List, vector, or non-negative integer. Hyperprior distribution for Ψ/ψ of τ, ε, and N_E, respectively. For τ and ε, if list of length = 2, then the first list element applies to the first more recent size change event (*e.g.* $\tau 1$, $\varepsilon 1$) and the second list element applies to the second more ancient size change event (*e.g.* $\tau 2$, $\varepsilon 2$), per taxon. The arguments(s) specified here and their according list lengths activate which taxon-specific demographic parameters are to be hyperparameterized via Ψ/ψ as well as $\zeta/\zeta_s/\zeta_T$ downstream. See also `Details`. At least one is required.

`tau.zeta.prior, tau2.zeta.prior, epsilon.zeta.prior, epsilon2.zeta.prior, NE.zeta.prior`

List of length = `num.partitions` or 1, vector, or non-negative proportion (*i.e.* ≤ 1 and ≥ 0). Hyperprior distribution for ζ_j of $\tau 1$, $\tau 2$, $\varepsilon 1$, $\varepsilon 2$, and N_E, respectively, for each *j*th pulse from 1 to Ψ/ψ, as specified by the corresponding `psi.prior`, and per partition. See also `Details`. Required for each corresponding `psi.prior` specified, unless the maximum value in the corresponding `psi.prior` = 0.

`tau.zeta.total.prior, tau2.zeta.total.prior, epsilon.zeta.total.prior, epsilon2.zeta.total.prior, NE.zeta.total.prior`

List of length = 1, vector, or non-negative proportion (*i.e.* ≤ 1 and ≥ 0). Hyperprior distribution for ζ_T of $\tau 1$, $\tau 2$, $\varepsilon 1$, $\varepsilon 2$, and N_E, respectively. Activates a uniform hyperprior such that each discrete Ψ/ψ value, as specified by the corresponding

	`psi.prior`, is first weighted with equal hyperprior probability, then all discrete ζ_T values are weighted equally per Ψ/ψ value, and finally every possible associated vector ζ/ζ_s is weighted equally per ζ_T value. See also `Details`.
`tau.shared.prior`, `tau2.shared.prior`, `epsilon.shared.prior`, `epsilon2.shared.prior`, `NE.shared.prior`	List, vector, or positive value. Prior distribution for the demographic parameter summaries $\tau1_s$, $\tau2_s$, $\varepsilon1_s$, $\varepsilon2_s$, and N_s, respectively (or $\tau1$, $\tau2$, $\varepsilon1$, $\varepsilon2$, and N, respectively, if corresponding `psi.prior` specifies Ψ). See also `Details`. Required for each corresponding `psi.prior` specified, unless the maximum value in the corresponding `psi.prior` = 0.
`tau.idio.prior, tau2.idio.prior, epsilon.idio.prior, epsilon2.idio.prior, NE.idio.prior`	List of length = `num.partitions` or 1, vector, or positive value. Prior distribution for the taxon-specific demographic parameters $\tau1_i$, $\tau2_i$, $\varepsilon1_i$, $\varepsilon2_i$, and N_i, respectively for idiosyncratic values, and $\tau1$, $\tau2$, $\varepsilon1$, $\varepsilon2$, and N, respectively for nuisance values. See also `Details`.
`linked.param, attached.hyper`	List, vector, or character string, with possible values being the names of the demographic parameters (*i.e.* `"tau"`, `"tau2"`, `"epsilon"`, `"epsilon2"`, `"NE"`). Activates nuisance parameters in `linked.param` to have prior distributions be linked to hyperparameterized demographic parameters in `attached.hyper`, such that prior distributions may differentiate across pulses and idiosyncratic taxa with respect to the hyperparameterized demographic parameter in `attached.hyper`. See also `Details`.
`linked.param.partition`, `attached.hyper.pulse`	List of length = length of `linked.param` or 1, vector, or positive integer. The partitions in the linked nuisance parameter, and the pulses in the attached hyperparameterized demographic parameter, for which each element in `linked.param` and `attached.hyper`, respectively, applies. Each list element may contain multiple partitions/pulses, respectively. See also `Details`.
`linked.param.prior`	List of length = length of `linked.param` or 1, vector, or positive value. Prior distribution for the nuisance demographic parameter in each element of `linked.param`. See also `Details`.
`linked.param.fixed`	List, vector of length = length of `linked.param`, or logical value. Activates a fixed nuisance demographic parameter value for all taxa to which the corresponding elements in

`linked.param.partition` and `attached.hyper.pulse` apply. See also `Details`.

`anchor.prior, change.prior`

List, vector, or positive integer. Prior distribution for $\tau 2$ based on its difference δ with $\tau 1$. This difference value can be assigned to synchronous/shared pulses in $\tau 1_s$, as specified by `tau.psi.prior`, and accordingly inferred as a parameter summary vector δ_s (`anchor.prior`), or applied independently across taxa as an idiosyncratic or nuisance value (`change.prior`). See also `Details`.

`exponential.growth.rate.prior, exponential.growth.rate.prior2`

List of length = `num.partitions` or 1, vector, or double value. Prior distribution for the nuisance taxon-specific parameters $r1$ and $r2$, respectively. Activates exponential growth model $N_\tau = N_0 * e^{(r * \tau)}$ for the first and second event, respectively, instead of instantaneous growth. Negative values indicate expansion and positive values indicate contraction. See also `Details`.

`mut.rate.prior`

List of length = `num.partitions` or 1, vector, or positive value. Prior distribution for mutation rate μ. See also `Details`. Required if `length.seq` is specified.

MODEL SPECIFICATIONS

`dirichlet.process`

Logical value. Activates a Dirichlet-process hyperprior that weighs all allowable combinations of Ψ/ψ and ζ/ζ_s according to possible combinations of taxa assignment. See also `Details`.

`idiosyncratic`

Logical value. Allows idiosyncratic taxa that freely vary *i.e.* are ungrouped from any of the pulses, as specified by the `psi.prior` arguments. See also `Details`.

`min.net.tau.zeta.total, min.net.tau2.zeta.total, min.net.epsilon.zeta.total, min.net.epsilon2.zeta.total, min.net.NE.zeta.total, max.net.tau.zeta.total, max.net.tau2.zeta.total, max.net.epsilon.zeta.total, max.net.epsilon2.zeta.total, max.net.NE.zeta.total`

Non-negative proportion (*i.e.* ≤ 1 and ≥ 0). Rule for the minimum/maximum ζ_T value, across all pulses (as specified by the corresponding `psi.prior`) and partitions, for $\tau 1$, $\tau 2$, $\varepsilon 1$, $\varepsilon 2$, and N_E, respectively. See also `Details`.

`min.net.tau.zeta.per.pulse, min.net.tau2.zeta.per.pulse, min.net.epsilon.zeta.per.pulse,`

List, vector of length = maximum value in corresponding `psi.prior`, or non-negative proportion (*i.e.* ≤ 1 and ≥ 0). Rule

`min.net.epsilon2.zeta.per.pulse`, `min.net.NE.zeta.per.pulse`, `max.net.tau.zeta.per.pulse`, `max.net.tau2.zeta.per.pulse`, `max.net.epsilon.zeta.per.pulse`, `max.net.epsilon2.zeta.per.pulse`, `max.net.NE.zeta.per.pulse`

for the minimum/maximum ζ_j value of $\tau 1$, $\tau 2$, $\varepsilon 1$, $\varepsilon 2$, and N_E, respectively, for each *j*th pulse from 1 to Ψ/ψ (as specified by the corresponding `psi.prior`) across all partitions. See also `Details`.

`tau.buffer`, `tau2.buffer`, `epsilon.buffer`, `epsilon2.buffer`, `NE.buffer`

Non-negative value or function. Pulse buffer β of the demographic parameter summaries $\tau 1_s$, $\tau 2_s$, $\varepsilon 1_s$, $\varepsilon 2_s$, and N_s, respectively. See also `Details`.

`tau.idiosyncratic.buffer`, `tau2.idiosyncratic.buffer`, `epsilon.idiosyncratic.buffer`, `epsilon2.idiosyncratic.buffer`, `NE.idiosyncratic.buffer`

Non-negative value or function. Idiosyncratic buffer β_i of the idiosyncratic taxon-specific demographic parameters $\tau 1_i$, $\tau 2_i$, $\varepsilon 1_i$, $\varepsilon 2_i$, and N_i, respectively. See also `Details`.

`idiosyncratic.rule`

Character string with possible values `"recent"` and `"ancient"`. Activates rule forcing all idiosyncratic taxa to have values less than the first shared pulse, or values greater than the last shared pulse, respectively. Any other values results in no such rules being placed on idiosyncratic taxa. See also `Details`.

`num.changes`

List, vector of length = `num.partitions`, or value of 1 or 2. Number of demographic change events per taxon. See also `Details`.

`flip`

List, vector of length = `num.partitions`, or logical value. Activates $\tau 2$ to be more recent than $\tau 1$. See also `Details`.

PARAMETER SUMMARIES

`net.zeta.total`, `net.zeta.per.pulse`

Logical value. Activates output of ζ_T and the vector ζ/ζ_s across partitions, respectively, as a list element/matrix in the `roll.object` list element of the final output, for downstream estimation. Rows of the matrix correspond to individual simulations, which correspond to rows of other matrices in the `roll.object` list element of the final output, and each cell is the aforementioned value.

`mean.tau.shared`, `mean.tau2.shared`, `mean.epsilon.shared`, `mean.epsilon2.shared`, `mean.NE.shared`

Logical value. Activates output of $E(\tau 1_s)$, $E(\tau 2_s)$, $E(\varepsilon 1_s)$, $E(\varepsilon 2_s)$, and $E(N_s)$ weighted by the vector ζ/ζ_s, respectively, as a list element/matrix in the `roll.object` list element of the final output, for downstream estimation. Rows of the matrix correspond to individual simulations, which correspond to rows

of other matrices in the `roll.object` list element of the final output, and each cell is the aforementioned value. See also `Details`.

`mean.tau, mean.tau2, mean.epsilon, mean.epsilon2, mean.NE`

Logical value. Activates output of E(τ1), E(τ2), E(ε1), E(ε2), and E(*N*), respectively, as a list element/matrix in the `roll.object` list element of the final output, for downstream estimation. Rows of the matrix correspond to individual simulations, which correspond to rows of other matrices in the `roll.object` list element of the final output, and each cell is the aforementioned value.

`disp.index.tau.shared, disp.index.tau2.shared, disp.index.epsilon.shared, disp.index.epsilon2.shared, disp.index.NE.shared`

Logical value. Activates output of $\Omega(\tau 1_s)$, $\Omega(\tau 2_s)$, $\Omega(\varepsilon 1_s)$, $\Omega(\varepsilon 2_s)$, and $\Omega(N_s)$ weighted by the vector ζ/ζ_s, respectively, as a list element/matrix in the `roll.object` list element of the final output, for downstream estimation. Rows of the matrix correspond to individual simulations, which correspond to rows of other matrices in the `roll.object` list element of the final output, and each cell is the aforementioned value. See also `Details`.

`disp.index.tau, disp.index.tau2, disp.index.epsilon, disp.index.epsilon2, disp.index.NE`

Logical value. Activates output of Ω(τ1), Ω(τ2), Ω(ε1), Ω(ε2), and Ω(*N*), respectively, as a list element/matrix in the `roll.object` list element of the final output, for downstream estimation. Rows of the matrix correspond to individual simulations, which correspond to rows of other matrices in the `roll.object` list element of the final output, and each cell is the aforementioned value. See also `Details`.

SIMULATION SPECIFICATIONS

`fsc2path`

Character string of path for `fastsimcoal2` executable file. Must include name of `fastsimcoal2` executable file. May be absolute path (*i.e.* beginning with "/") or relative path from `output.directory`. Required.

`messages.sims`

Positive integer. Interval length in number of completed simulations for each cycle of messages (*e.g.* if = 10,000, a message is given when 10,000, 20,000, 30,000, etc. simulations are completed). Regardless of value or whether specified, a message is always given when the last simulation is completed.

`output.directory`

Character string of path for directory where output simulation

files are deposited, as well as from where `fastsimcoal2` is run (*i.e.* working directory changes to `output.directory` when running `fastsimcoal2`, but returns to current working directory). May or may not end with "/". May be absolute path (*i.e.* beginning with "/") or relative path from current working directory. See also `Details`. Required.

`append.sims` — Logical value. Allows output to be added to existing files with the same filename that are located within `output.directory`. See also `Details`.

`keep.taxa.draws`, `output.hyper.draws`, `output.taxa.draws`, `keep.fsc2.files` — Logical value. Activates whether taxon-specific parameter draws (*i.e.* `sim.specs` list elements) are outputted in `R` (not activating this may increase performance speed and decrease memory usage), whether hyperparameter and parameter summary values (*i.e.* `roll.object` list elements) are outputted to a simple text file within `output.directory`, whether taxon-specific parameter draws are outputted to a simple text file within `output.directory`, and whether the `fastsimcoal2` outputs `seed.txt` and `.lhood` are kept in simple text files per simulated independent taxon with respective filename suffixes `.seed` and `.fsc2` within `output.directory`, respectively.

OBJECTS FROM PRECEDING FUNCTIONS

`build.object` — Output from function `build.dice`.

`roll.object` — Output from function `roll.dice`.

`play.object` — Output from function `play.dice`. See also `Details`.

Arguments from other `Multi-DICE` functions may be included here and are ignored if not applicable.

Details

For more information, see `Multi-DICE` Manual.

`Multi-DICE` cannot currently accommodate models with more than one population per taxon, events aside from population size change, and more than two size change events.

For $\tau 1$ and $\tau 2$, units are in numbers of generations, and thus may only be positive integers. For $\varepsilon 1$ and $\varepsilon 2$, units are in ratio of size change from the ancestral effective population size to current effective population size, such that expansions are < 1 and contractions are > 1, and thus may only be positive

values. For N_E, unit is in number of effective haploid individuals, and thus may only be positive integers.

For `num.taxa`, `num.haploid.samples`, `num.ind.sites`, `num.SNPs`, `length.seq`, `folded`, `sampling.times`, `gen.times`, `linked.param`, `attached.hyper`, `linked.param.fixed`, `min.net.tau.zeta.per.pulse`, `min.net.tau2.zeta.per.pulse`, `min.net.epsilon.zeta.per.pulse`, `min.net.epsilon2.zeta.per.pulse`, `min.net.NE.zeta.per.pulse`, `max.net.tau.zeta.per.pulse`, `max.net.tau2.zeta.per.pulse`, `max.net.epsilon.zeta.per.pulse`, `max.net.epsilon2.zeta.per.pulse`, `max.net.NE.zeta.per.pulse`, `num.changes`, and `flip`, if list, then all list elements are concatenated to form a single vector, with the ordering within list elements and then between list elements preserved (*e.g.* for list of length = 2, with first list element of length = 2 and second list element of length = 1, the order from first to last is: 1) first vector element in first list element; 2) second vector element in first list element; 3) sole vector element in second list element).

For `tau.psi.prior`, `epsilon.psi.prior`, `NE.psi.prior`, `tau.zeta.prior`, `tau2.zeta.prior`, `epsilon.zeta.prior`, `epsilon2.zeta.prior`, `NE.zeta.prior`, `tau.zeta.total.prior`, `tau2.zeta.total.prior`, `epsilon.zeta.total.prior`, `epsilon2.zeta.total.prior`, `NE.zeta.total.prior`, `tau.shared.prior`, `tau2.shared.prior`, `epsilon.shared.prior`, `epsilon2.shared.prior`, `NE.shared.prior`, `tau.idio.prior`, `tau2.idio.prior`, `epsilon.idio.prior`, `epsilon2.idio.prior`, `NE.idio.prior`, `linked.param.prior`, `anchor.prior`, `change.prior`, `exponential.growth.rate.prior`, `exponential.growth.rate.prior2`, and `mut.rate.prior`, each list element contains an entire individual discrete distribution; if vector, then converted to list of length = 1 with all vector elements comprising the entirety of a single discrete distribution. Per list element, vector elements within (*i.e.* the discrete distribution) do not need to be in any particular order. Relatedly, each vector element is treated as an independent value, thus weighted distributions (*i.e.* not uniform) may be employed by duplicating values (*e.g.* a distribution of `c(0,1,1,1)` signifies 75% probability of drawing “1” and 25% probability of drawing “0”), allowing the specification of any discretized distribution (*e.g.* gamma, beta, log-uniform). Accordingly, a uniform distribution with no gaps for integer values would be of length = range of distribution.

For `num.taxa`, `num.haploid.samples`, `num.ind.sites`, `num.SNPs`, `length.seq`, `folded`, `sampling.times`, `gen.times` (except when vector of length = *n*), `num.changes`, and `flip`, the order of vector elements corresponds to the order of partitions, and for `tau.zeta.prior`, `tau2.zeta.prior`, `epsilon.zeta.prior`, `epsilon2.zeta.prior`, `NE.zeta.prior`, `tau.idio.prior`, `tau2.idio.prior`, `epsilon.idio.prior`, `epsilon2.idio.prior`, `NE.idio.prior`, `change.prior`, `exponential.growth.rate.prior`, `exponential.growth.rate.prior2`, and `mut.rate.prior`, the order of list elements corresponds to the order of partitions. Additionally, if length = 1 and `num.partitions` > 1, then the sole element is used for all partitions. Similarly, if length < `num.partitions`, then the first element is used for all partitions while ignoring any remaining elements. Except for `gen.times`, if length >

num.partitions, then the remaining elements beyond length = num.partitions are ignored; a caution is provided when the length does not equal 1 or num.partitions. For gen.times, if length > num.partitions and length < *n*, then the remaining elements beyond length = num.partitions are ignored, and if length > *n*, then the remaining elements beyond length = *n* are ignored; a caution is provided when the length does not equal 1, num.partitions, or *n*.

For min.net.tau.zeta.per.pulse, min.net.tau2.zeta.per.pulse, min.net.epsilon.zeta.per.pulse, min.net.epsilon2.zeta.per.pulse, min.net.NE.zeta.per.pulse, max.net.tau.zeta.per.pulse, max.net.tau2.zeta.per.pulse, max.net.epsilon.zeta.per.pulse, max.net.epsilon2.zeta.per.pulse, and max.net.NE.zeta.per.pulse, the order of vector elements corresponds to the temporal order, from most recent to most ancient, of pulses (as specified by the corresponding psi.prior), and for tau.shared.prior, tau2.shared.prior, epsilon.shared.prior, epsilon2.shared.prior, NE.shared.prior, and anchor.prior, the order of list elements corresponds to the temporal order, from most recent to most ancient, of pulses (as specified by the corresponding psi.prior). Additionally, if length = 1 and maximum value in corresponding psi.prior > 1, then the sole element is used for all pulses. Similarly, if length < maximum value in corresponding psi.prior, then the first element is used for all pulses while ignoring any remaining elements. For the zeta.per.pulse arguments, if length > maximum value in corresponding psi.prior, then the remaining elements beyond length = maximum value in corresponding psi.prior are ignored; a caution is provided when the length does not equal 1 or maximum value in corresponding psi.prior. For the shared.prior arguments and anchor.prior, there may be additional list elements for idiosyncratic distributions, such that any total length = 1, maximum value in corresponding psi.prior, maximum value in corresponding psi.prior + 1, or maximum value in corresponding psi.prior + num.partitions, are allowed; for any other lengths, a caution is provided and excess elements beyond the highest acceptable length are ignored. See below for more information about adding idiosyncratic distributions to these arguments.

If num.partitions=*n*, then rearrangement of bins across taxa within allele frequency classes based on descending order of the relative SNP proportions is not performed to construct the aSFS and taxon-specific inference of demographic parameters is possible. However, in general, more partitions results in more parameter space with respect to taxa samples that must be explored due to a decrease in order-independence and assumed exchangeability, thus multiple-fold more simulations must be conducted to achieve comparable accuracy in hyperparameter estimation as without partitioning.

For num.ind.sites, num.SNPs, and length.seq, Multi-DICE first looks in num.ind.sites, then num.SNPs, and finally length.seq. For num.ind.sites, specified values equate to the number of genealogies simulated by the FREQ simulation model in fastsimcoal2. For num.SNPs, specified values equate to the number of independent loci (chromosomal structure is not considered) simulated by the SNP simulation model in fastsimcoal2. For length.seq, specified values equate to the number of loci within a linkage block (*i.e.* sites, or length of sequence; recombination is

not considered) simulated by the SNP simulation model in fastsimcoal2; to utilize length.seq, mut.rate.prior must also be specified. For both the FREQ and SNP simulation models, infinite sites are assumed by fastsimcoal2. Both num.SNPs and length.seq may be specified to simulate monomorphic sites and linked SNPs for the SFS, *i.e.* derive aSFS from genomic-scale whole-sequence information.

If gen.times is specified, output τ values are in units of years. If generation times are equivalent across all *n* taxa, then it is not critical to specify gen.times, since output τ values may be multiplied by the generation time scalar to covert units to years if gen.times is not specified. Hence, if generation time is 1 year, then output τ values are in units of years as well.

For tau.psi.prior, epsilon.psi.prior, and NE.psi.prior, distinguishing between Ψ and ψ is accomplished via the corresponding zeta.prior, zeta.total.prior, idiosyncratic setting, and/or min/max.net zeta.total/per.pulse, except for values of 0, which are explicitly for $\psi = 0$ and thus indicate full idiosyncrasy. If list length > 2 for τ and ε or list length > 1 for N_E, then a caution is provided and the remaining elements beyond length = 2 for τ and ε and length = 1 for N_E are ignored. Applies across all partitions, such that it is regardless of partitioning.

For tau.zeta.prior, tau2.zeta.prior, epsilon.zeta.prior, epsilon2.zeta.prior, and NE.zeta.prior, if num.partitions > 1, may be necessary to include “0.0” as a value, but can control ζ_S and ζ_T across partitions via corresponding zeta.total.prior and/or min/max.net zeta.total/per.pulse. Attributes to each partition individually, but proportion values are out of the entirety of taxa dataset, thus if num.partitions > 1, then the upper bound for each partition should be the number of taxa within that partition divided by *n*. When $\psi = \{0, 1\}$, equivalent to hyperprior distribution for ζ_T. If identical across all partitions, it is more computationally efficient to specify only one *i.e.* list of length = 1, or a vector.

For tau.zeta.total.prior, tau2.zeta.total.prior, epsilon.zeta.total.prior, epsilon2.zeta.total.prior, and NE.zeta.total.prior, if length > 1, then remaining elements beyond the first are ignored and a caution is provided.

To build a hyperprior, Multi-DICE first looks if the corresponding zeta.total.prior is specified, then if dirichlet.process=T, and if neither is such case, then all possible combinations of corresponding psi.prior and zeta.prior draws are equally weighted. Therefore, if num.partitions=1, corresponding psi.prior=1, and dirichlet.process=F, then tau.zeta.total.prior, tau2.zeta.total.prior, epsilon.zeta.total.prior, epsilon2.zeta.total.prior, and NE.zeta.total.prior equivalent to corresponding zeta.prior and thus unnecessary to specify.

For tau.shared.prior, tau2.shared.prior, epsilon.shared.prior, epsilon2.shared.prior, and NE.shared.prior, each successive list element/distribution must have a greater minimum and maximum value than its preceding list elements/distributions. If multiple distributions are utilized for every potential pulse and these distributions overlap in their bounds,

running time may slow since, in this case, draws are made from all the distributions independently and then checked if abiding by ordering and buffering, with re-draws if not.

For each of `tau`, `tau2`, `epsilon`, `epsilon2`, and `NE`, if hyperparameterized via its corresponding `psi.prior` while `idiosyncratic=T` or a 0 value is in said `psi.prior`, then a corresponding idiosyncratic prior is required. `Multi-DICE` first looks in `change.prior` (for $\tau 2$ only), then the corresponding `idio.prior`, and finally the corresponding `shared.prior`. If a `shared.prior` argument is utilized, the additional list elements beyond length = maximum value in corresponding `psi.prior` are considered; if the number of list elements is $\leq$ maximum value in corresponding `psi.prior`, then only the first list element is considered. These list elements undergo the same specifications as aforementioned for the corresponding `idio.prior`.

For each of `tau`, `tau2`, `epsilon`, `epsilon2`, and `NE`, if not hyperparameterized yet part of the specified model (*i.e.* `tau`, `epsilon`, and `NE` are always part of the model, and `tau2` and `epsilon2` are part of the model when `num.changes=2`), then a corresponding nuisance prior is required. A nuisance prior differs from an idiosyncratic prior in that variation in the respective nuisance parameter is being considered while not of interest with respect to hyperparameterization (*i.e.* variability in values across taxa governed hierarchically), whereas an idiosyncratic prior is still governed by hyperparameters in coordination with the corresponding shared prior. `Multi-DICE` first looks in `anchor.prior` (for $\tau 2$ only), next `change.prior` (for $\tau 2$ only), afterward `linked.param.prior` (if applicable; see below for more information), then the corresponding `idio.prior`, and finally the corresponding `shared.prior`. If `linked.param.prior` is utilized for a particular nuisance parameter, either the corresponding `idio.prior` or `shared.prior` must still be specified even if all taxa are always covered by `linked.param.prior` across all simulations. If a `shared.prior` argument is utilized, it undergoes the same specifications as aforementioned for the corresponding `idio.prior`.

It is highly recommended that prior distribution bounds are as far apart as possible between $\tau 1$ and $\tau 2$, ideally mutually exclusive/non-overlapping. Otherwise, there could be a stop of operation due to an invalid/incompatible draw, computational lag, and/or statistical bias on prior distributions.

For `mut.rate.prior`, utilized only if `length.seq` is utilized as well. Values of 0 may technically be included, though this results in all polymorphic sites. If `keep.taxa.draws=T`, draws are outputted as a list element/matrix named `mut.rate` in the `sim.specs` list element of the final output. Each row represents an individual simulation, each column represents a simulated independent taxon, and each cell is the according mutation rate value used for simulation. Rows and columns correspond to other matrices *i.e.* the same row number refers to the same simulation and the same column number refers to the same simulated independent taxon.

For `tau.buffer`, `tau2.buffer`, `epsilon.buffer`, `epsilon2.buffer`, and `NE.buffer`, β buffers are applied to draws of shared pulse values, thus affecting the corresponding shared prior; the corresponding idiosyncratic prior, from which draws are subsequently made, is accordingly affected by the shared pulse buffers. For `tau.idiosyncratic.buffer`, `tau2.idiosyncratic.buffer`, `epsilon.idiosyncratic.buffer`, `epsilon2.idiosyncratic.buffer`, and

NE.idiosyncratic.buffer, β_i idiosyncratic buffers are then applied to idiosyncratic draws, which additionally affect the corresponding idiosyncratic prior that had already been initially transformed by the shared pulse buffers. In other words, β buffers idiosyncratic taxa from shared pulse values, and β_i buffers idiosyncratic taxa from each other. For example, if n = 10, $\psi_{\tau 1}$ = 2, $\zeta_{\tau 1,T}$ = 0.8, β = 10,000, β_i = 1,000, $\tau 1 \sim U(1{,}000, 100{,}000)$, $\tau 1_S$ = {11,000, 12,001}, and $\tau 1_i$ = {99,000} after the first idiosyncratic draw, then given that the synchronous/shared pulse buffers result in invalid draws from $\sim U(1{,}000, 13{,}001)$ and the first idiosyncratic draw buffer results in invalid draws from $\sim U(98{,}000, 100{,}000)$, the remaining second idiosyncratic draw would be from the resulting transformed prior distribution $\tau 1_i \sim U(13{,}002, 97{,}999)$. A caution is provided if an idiosyncratic.buffer corresponding to a specified psi.prior is not specified. Buffers cannot be deployed for nuisance draws (though see below when ψ = 0).

For num.changes, if < 1, then there is a stop of operation, and if > 2, then converted to num.changes=2 and a caution is provided.

If flip=T, $\tau 2$ and $\varepsilon 2$ still refers to the second specified event, which in this case is the more recent one.

There cannot be a filename of “dice.sims.fsc2.template” within output.directory; output.directory='.' to specify current working directory.

If append.sims=F, then cannot have a filename of “seed.txt”, “MRCAs.txt”, “dice.log”, “dice.simulations” with a numerical suffix of “1” through *n*, or anything with the prefix “dice.sims” within output.directory. If append.sims=T, temporary intermediate files may cause files that have the prefix “dice.sims” within output.directory to be deleted.

When ψ = 0 *i.e.* full idiosyncrasy, the arguments idiosyncratic, min.net.tau.zeta.total, min.net.tau2.zeta.total, min.net.epsilon.zeta.total, min.net.epsilon2.zeta.total, min.net.NE.zeta.total, min.net.tau.zeta.per.pulse, min.net.tau2.zeta.per.pulse, min.net.epsilon.zeta.per.pulse, min.net.epsilon2.zeta.per.pulse, min.net.NE.zeta.per.pulse, and idiosyncratic.rule are ignored, and the arguments tau.idiosyncratic.buffer, tau2.idiosyncratic.buffer, epsilon.idiosyncratic.buffer, epsilon2.idiosyncratic.buffer, and NE.idiosyncratic.buffer are activated (even if idiosyncratic=F).

Multi-DICE proceeds only if a valid draw is allowed given corresponding num.taxa, psi.prior, zeta.prior (*e.g.* minimum value in psi.prior * minimum value in zeta.prior $\leq$ minimum value in num.taxa), zeta.total.prior, shared prior, idiosyncratic prior, idiosyncratic setting, min/max.net zeta.total/per.pulse, buffer, and idiosyncratic.buffer. Importantly, there is no check on if all values in psi.prior have a valid draw, only if there is a valid draw among any of the values. Additionally, the range of a shared prior must be greater than its corresponding $(\Psi/\psi - 1)(2\beta + 1)$, and the range of an idiosyncratic prior ought to be greater than its corresponding $(\Psi/\psi)(2\beta + 1) + (\sigma - 1)(2\beta_i + 1)$ across every possible combination of Ψ/ψ and σ and with consideration

given to idiosyncratic.rule. For tau2.shared.prior and tau2.idio.prior, consideration must also be given to any overlap with tau.shared.prior and tau.idio.prior, respectively.

Value

Returned value is identical to that of play.dice, except if keep.taxa.draws=F, in which case the list element sim.specs is omitted, or if keep.taxa.draws=T and both length.seq and mut.rate.prior are utilized, in which case μ draws are also added to sim.specs (see above for more information). Additionally, output simulation files in simple text format are deposited into output.directory: the filenames of "dice.simulations" with a numerical suffix of 1 through *n* (*i.e.* "dice.simulations1", "dice.simulations2", etc.) contain per-taxon summary statistics, tab delimited; the filenames of "dice.sims.hyper.draws." with a suffix corresponding to the names of hyperparameters and parameter summaries (*i.e.* names of roll.object list elements *e.g.* "psi.tau" for $\Psi_{\tau 1}/\psi_{\tau 1}$ as specified by tau.psi.prior, "zeta.tau.1" for $\zeta_{\tau 1}/\zeta_{\tau 1,s}$ of the first partition $\zeta_{\tau 1}.1/\zeta_{\tau 1,s}.1$, "net.zeta.total.tau" for $\zeta_{\tau 1,T}$, "net.zeta.per.pulse.tau" for $\zeta_{\tau 1}/\zeta_{\tau 1,s}$, "pulse.values.tau" for $\tau 1_s$ (or $\tau 1$ if tau.psi.prior specifies $\Psi_{\tau 1}$), "disp.index.tau" for $\Omega(\tau 1)$, etc.) contain the according values of interest for estimation, space delimited; the filenames of "dice.sims.taxa.draws." with a suffix corresponding to the names of taxon-specific parameters (*i.e.* names of sim.specs list elements: tau, tau2, epsilon, epsilon2, NE, exponential.growth.rate.prior, exponential.growth.rate.prior2, mut.rate) contain the according nuisance values used for simulation across taxa, space delimited; if keep.fsc2.files=T, the filenames of "dice.sims" with a suffix of 1 through *n* attached to either .seed and .fsc2 (*i.e.* "dice.sims1.seed", "dice.sims1.fsc2", "dice.sims2.seed", "dice.sims2.fsc2", etc.) contain, per simulated independent taxon, the fastsimcoal2 outputs seed.txt and .lhood, respectively. Each row per file represents an individual simulation, and rows across files correspond to each other *i.e.* the same row number refers to the same simulation across files. For "dice.sims.taxa.draws." files, each column per file represents each simulated independent taxon, and column order across these files and the numeric order implied by the names of the "dice.simulations", ".seed", and ".fsc2" files correspond to each other *i.e.* the same column number and filename number refer to the same simulated independent taxon. For "dice.simulations" files, each column per file represents an SFS allele frequency class bin (in the order described by the manual for fastsimcoal2) or single-sequence summary statistic (in the order of number of haplotypes, haplotype diversity, nucleotide diversity, and Tajima's *D*), depending on data type simulated, for that respective simulated independent taxon.

Author(s)

Alexander T. Xue

References

Chan YL, Schanzenbach D, Hickerson MJ (2014) Detecting concerted demographic response across community assemblages using hierarchical approximate Bayesian computation. *Molecular Biology and Evolution*, **31**, 2501–2515.

Excoffier L, Dupanloup I, Huerta-Sánchez E, Sousa VC, Foll M (2013) Robust demographic inference from genomic and SNP data. *PLoS genetics*, **9**, e1003905.
Excoffier L (2014) fastsimcoal2 manual.
Hickerson MJ, Stahl E, Takebayashi N (2007) msBayes: Pipeline for testing comparative phylogeographic histories using hierarchical approximate Bayesian computation. *BMC bioinformatics*, **8**, 268.
Huang W, Takebayashi N, Qi Y, Hickerson MJ (2011) MTML-msBayes: approximate Bayesian comparative phylogeographic inference from multiple taxa and multiple loci with rate heterogeneity. *BMC bioinformatics*, **12**, 1.
Xue AT (2017) Multi-DICE Manual.
Xue AT, Hickerson MJ (2015) The aggregate site frequency spectrum for comparative population genomic inference. *Molecular Ecology*, **24**, 6223–6240.
Xue AT, Hickerson MJ (*submitted*) Multi-DICE: R package for comparative population genomic inference under multi-taxa hierarchical co-demographic models.

See Also

`build.dice`, `roll.dice`, `play.dice`, `dice.aSFS`, `dice.sumstats`

Examples

```
#simplest execution akin to approach in Xue and Hickerson (2015)
dice.sims(num.sims=5, num.taxa=10, num.haploid.samples=10,
num.ind.sites=2000, tau.psi.prior=c(1), tau.zeta.prior=c(1:10)/10,
tau.shared.prior=c(1000:1000000), epsilon.idio.prior=c(1000:10000)/100000,
NE.idio.prior=c(1000:100000), fsc2path='example', output.directory='.')

#simplest execution akin to approach in Xue and Hickerson (2015); ln U
distribution applied on tau.shared.prior, with 100,000 intervals
discretized uniformly across ln(tau.shared.prior)
dice.sims(num.sims=5, num.taxa=10, num.haploid.samples=10,
num.ind.sites=2000, tau.psi.prior=c(1), tau.zeta.prior=c(1:10)/10,
tau.shared.prior=exp(c((log(1000)*100000):(log(1000000)*100000))/100000),
epsilon.idio.prior=c(1000:10000)/100000, NE.idio.prior=c(1000:100000),
fsc2path='example', output.directory='.')

#simplest execution akin to approach in Xue and Hickerson (2015); assuming
play.dice was previously performed and the output was directed to object
play.object
dice.sims(num.sims=5, num.taxa=10, num.haploid.samples=10,
num.ind.sites=2000, fsc2path='example', output.directory='.',
play.object=play.object)

#simplest execution akin to approach in software package msBayes
dice.sims(num.sims=5, num.taxa=10, num.haploid.samples=10,
num.ind.sites=2000, tau.psi.prior=c(1:10), tau.zeta.prior=c(1:10)/10,
```

```
tau.shared.prior=c(1000:1000000), epsilon.idio.prior=c(1000:10000)/100000,
NE.idio.prior=c(1000:100000), idiosyncratic=F, fsc2path='example',
output.directory='.')
```

Constructing aSFS from per-taxon SFS simulations/data

Description

`dice.aSFS` transforms per-taxon SFS, either simulated by `dice.sims` or empirically produced (*i.e.* observed/collected data), into the aSFS, following the procedure described in Xue and Hickerson (2015).

Usage

```
dice.aSFS(num.sims, num.taxa, num.partitions=1, num.haploid.samples,
folded=T, remove.afclasses=NULL, output.directory='.',
input.directory=NULL, input.base=NULL, input.files=NULL)
```

Arguments

`num.sims`	Positive integer. Number of simulations. Required. See also `Details`.
DATA	
`num.taxa`	List, vector of length = `num.partitions`, or positive integer. Number of taxa per partition. Total sum across partitions equals the total number of taxa *n* in dataset. See also `Details`. Required.
`num.partitions`	Positive integer. Number of partitions for taxa in dataset. Allows differential data and model specifications across user-specified taxa groupings, including sampling size of individuals, generation times, demographic syndrome, and taxon-specific nuisance parameter prior distributions. See also `Details`.
`num.haploid.samples`	List, vector of length = `num.partitions`, or positive integer. Number of haploid samples per partition. See also `Details`. Required.
`folded`	List, vector of length = `num.partitions`, or logical value. Activates folding of the SFS per partition. See also `Details`.
`remove.afclasses`	List of length = `num.partitions` or 1, vector, or non-negative integer. Allele frequency classes to be removed from the SFS per partition prior to aSFS construction, with 0 referring to

monomorphic bins, 1 referring to singleton bins, 2 referring to doubleton bins, etc. By default, monomorphic bins are removed. See also `Details`.

SIMULATION SPECIFICATIONS

`output.directory`, `input.directory`

Character string of path(s) for directory where SFS files are located. May or may not end with "/". May be absolute path (*i.e.* beginning with "/") or relative path from current working directory. If `output.directory` is specified, then it is assumed that there is only one directory with output SFS filenames as produced by `dice.sims`. To specify multiple directories or different filenames, `input.directory` must be specified, which may be a list or vector. See also `Details`.

`input.base`

Character string of prefix for SFS filenames. Assumed all SFS files within `input.directory` begin with prefix and end with a numerical suffix of `1` through *n*. See also `Details`.

`input.files`

List, vector = *n*, or character string of SFS filenames within `input.directory`. See also `Details`.

Arguments from other `Multi-DICE` functions may be included here and are ignored if not applicable.

Details

For more information, see `Multi-DICE` Manual.

`Multi-DICE` cannot currently accommodate data for more than one population per taxon.

For `num.taxa`, `num.haploid.samples`, `folded`, `input.directory`, and `input.files`, if list, then all list elements are concatenated to form a single vector, with the ordering within list elements and then between list elements preserved (*e.g.* for list of length = 2, with first list element of length = 2 and second list element of length = 1, the order from first to last is: 1) first vector element in first list element; 2) second vector element in first list element; 3) sole vector element in second list element).

For `remove.afclasses`, if vector, then converted to list of length = 1. Per list element, vector elements within do not need to be in any particular order. Relatedly, only unique vector elements are considered, with duplicated values ignored.

For `num.taxa`, `num.haploid.samples`, and `folded`, the order of vector elements corresponds to the order of partitions, and for `remove.afclasses`, the order of list elements corresponds to the order of partitions. Additionally, if length = 1 and `num.partitions` > 1, then the sole element is used for all partitions. Similarly, if length < `num.partitions`, then the first element is used for all

partitions while ignoring any remaining elements, and if length > num.partitions, then the remaining elements beyond length = num.partitions are ignored; a caution is provided when the length does not equal 1 or num.partitions.

If empirical data are converted to aSFS format, then num.sims = number of multi-taxa comparative datasets (most likely 1). SFS files must be in similar format to output SFS files produced by dice.sims, such that each taxon SFS is within a separate tab-delimited file, number of rows = num.sims, number of columns = number of SFS allele frequency classes (*i.e.* number of haploid samples + 1 (number of haploid samples – 1 if monomorphic bins are already removed), or if folded, then may be (number of haploid samples/2) + 1, rounded up, with + 1 omitted if monomorphic bins are already removed), and no headers or labels. Values are converted to proportions out of the total of polymorphic bins by default, thus values do not need to be user-converted.

If num.partitions=*n*, then rearrangement of bins across taxa within allele frequency classes based on descending order of the relative SNP proportions is not performed to construct the aSFS and taxon-specific inference of demographic parameters is possible. However, in general, more partitions results in more parameter space with respect to taxa samples that must be explored due to a decrease in order-independence and assumed exchangeability, thus multiple-fold more simulations must be conducted to achieve comparable accuracy in hyperparameter estimation as without partitioning.

For remove.afclasses, monomorphic bins are not removed by default if specified, thus 0 must be included if desired to be removed. To include the monomorphic frequency class without removing any bins, the integer of num.haploid.samples + 1 (or any value(s) > num.haploid.samples) must be specified.

To determine input directory, Multi-DICE first looks in input.directory, then output.directory. If input.directory is utilized, may be of any length, with order corresponding to arrangement of output aSFS matrix rows. Filenames are assumed to be consistent across input.directory directories. To determine filename format, Multi-DICE first looks in input.base, then input.files. If input.base is list or vector of length > 1, then only first vector element is considered. For input.files, length must be $\geq n$; if > *n*, then the remaining elements beyond length = *n* are ignored and a caution is provided. If neither input.files nor input.base is specified, or if output.directory is utilized, then filenames assumed to follow the same format as ouput SFS files produced by dice.sims (*i.e.* "dice.simulations1", "dice.simulations2", etc.). To accommodate parallelized runs across multiple directories, input.directory ought to be utilized.

Value

Returned value is aSFS matrix, with each row representing an individual simulation and in the same order as input files (followed then by the user-specified order of input directories if applicable), and each column representing an aSFS bin.

Author(s)

Alexander T. Xue

References

Xue AT (2017) Multi-DICE Manual.
Xue AT, Hickerson MJ (2015) The aggregate site frequency spectrum for comparative population genomic inference. *Molecular Ecology*, **24**, 6223–6240.
Xue AT, Hickerson MJ (*submitted*) Multi-DICE: R package for comparative population genomic inference under multi-taxa hierarchical co-demographic models.

See Also

`build.dice`, `roll.dice`, `play.dice`, `dice.sims`

Examples

```
#simplest execution
dice.aSFS(num.sims=5, num.taxa=10, num.haploid.samples=10,
output.directory='example')
```

Constructing multi-taxa single-sequence summary statistic vector from per-taxon summary statistics simulations/data

Description

`dice.sumstats` transforms per-taxon single-sequence summary statistics, either simulated by `dice.sims` or empirically produced (*i.e.* observed/collected data), into the multi-taxa single-sequence summary statistic vector, following the procedure described in Chan *et al.* (2015). Empirical sequence data may also be inputted. If `convert.sequences=T`, then `bash` commands are called upon here and `dice.sumstats` can run only within a `bash` terminal environment (*e.g.* Mac, Linux).

Usage

```
dice.sumstats(num.sims, num.taxa, num.partitions=1, num.haploid.samples,
convert.sequences=F, output.directory='.', input.directory=NULL,
input.base=NULL, input.files=NULL)
```

Arguments

`num.sims`	Positive integer. Number of simulations. Required. See also `Details`.
DATA	
`num.taxa`	List, vector of length = `num.partitions`, or positive integer. Number of taxa per partition. Total sum across partitions equals the total number of taxa *n* in dataset. See also `Details`. Required.
`num.partitions`	Positive integer. Number of partitions for taxa in dataset. Allows differential data and model specifications across user-specified taxa groupings, including sampling size of individuals, generation times, demographic syndrome, and taxon-specific nuisance parameter prior distributions. See also `Details`.
`num.haploid.samples`	List, vector of length = `num.partitions`, or positive integer. Number of haploid samples per partition. See also `Details`. Required if `convert.sequences=T`.
`convert.sequences`	Logical value. Activates sequence data input. See also `Details`.

SIMULATION SPECIFICATIONS

`output.directory`, `input.directory`	Character string of path(s) for directory where SFS files are located. May or may not end with "/". May be absolute path (*i.e.* beginning with "/") or relative path from current working directory. If `output.directory` is specified, then it is assumed that there is only one directory with output SFS filenames as produced by `dice.sims`. To specify multiple directories or different filenames, `input.directory` must be specified, which may be a list or vector. See also `Details`.
`input.base`	Character string of prefix for SFS filenames. Assumed all SFS files within `input.directory` begin with prefix and end with a numerical suffix of `1` through *n*. See also `Details`.
`input.files`	List, vector = *n*, or character string of SFS filenames within `input.directory`. See also `Details`.

Arguments from other `Multi-DICE` functions may be included here and are ignored if not applicable.

Details

For more information, see `Multi-DICE` Manual.

`Multi-DICE` cannot currently accommodate data for more than one population per taxon.

For `num.taxa`, `num.haploid.samples`, `input.directory`, and `input.files`, if list, then all list elements are concatenated to form a single vector, with the ordering within list elements and then between list elements preserved (*e.g.* for list of length = 2, with first list element of length = 2 and second list element of length = 1, the order from first to last is: 1) first vector element in first list element; 2) second vector element in first list element; 3) sole vector element in second list element).

For `num.taxa` and `num.haploid.samples`, the order of vector elements corresponds to the order of partitions. Additionally, if length = 1 and `num.partitions` > 1, then the sole element is used for all partitions. Similarly, if length < `num.partitions`, then the first element is used for all partitions while ignoring any remaining elements, and if length > `num.partitions`, then the remaining elements beyond length = `num.partitions` are ignored; a caution is provided when the length does not equal 1 or `num.partitions`.

If empirical data are converted, then `num.sims` = number of multi-taxa comparative datasets (most likely 1). If `convert.sequences=F`, files must be in similar format to output simulation files produced by `dice.sims`, such that each taxon single-sequence summary statistics are within a separate tab-delimited file, number of rows = `num.sims`, number of columns = 4, no headers or

labels, and columns are in order of: number of haplotypes, haplotype diversity, nucleotide diversity, Tajima's *D*.

If `num.partitions=`*n*, then distribution moments (*i.e.* transforming into multi-taxa single-sequence summary statistic vector) are irrelevant (mean = per-taxon value; variance, skewness, kurtosis set to 0) and taxon-specific inference of demographic parameters is possible. However, in general, more partitions results in more parameter space with respect to taxa samples that must be explored due to a decrease in order-independence and assumed exchangeability, thus multiple-fold more simulations must be conducted to achieve comparable accuracy in hyperparameter estimation as without partitioning.

If `convert.sequences=T`, assumed that: each taxon single-sequence data are within a separate file; sequence format is in 1/0 (with missing data = 9), AGTC (with missing data = N or -), or IUPAC (with missing data = N or -); each row corresponds to a haploid (diploid for IUPAC format) sample such that number of rows = `num.haploid.samples`; no headers or labels; each multi-taxa comparative dataset is within a separate directory such that length of directories = `num.sims` and `num.sims=1` if `output.directory` is utilized; sequence format is consistent across all directories. To calculate number of haplotypes per taxon, missing data in otherwise monomorphic sites are converted to the according monomorphic base and sites that are missing across all samples are removed, then unique haplotypes are discovered (while ignoring missing sites) from the sequences in order from least to most missing data (ties result in original user-specified order). Additionally, there cannot be a filename of "dice.sims.fsc2.template" within any directory.

To determine input directory, `Multi-DICE` first looks in `input.directory`, then `output.directory`. If `input.directory` is utilized, may be of any length, with order corresponding to arrangement of output matrix rows. Filenames are assumed to be consistent across `input.directory` directories. To determine filename format, `Multi-DICE` first looks in `input.base`, then `input.files`. If `input.base` is list or vector of length > 1, then only first vector element is considered. For `input.files`, length must be $\geq$ *n*; if > *n*, then the remaining elements beyond length = *n* are ignored and a caution is provided. If neither `input.files` nor `input.base` is specified, or if `output.directory` is utilized, then filenames assumed to follow the same format as ouput files produced by `dice.sims` (*i.e.* "`dice.simulations1`", "`dice.simulations2`", etc.). To accommodate parallelized runs across multiple directories, `input.directory` ought to be utilized.

Value

Returned value is matrix of multi-taxa single-sequence summary statistic vectors, with each row representing an individual simulation and in the same order as input files (followed then by the user-specified order of input directories if applicable), and each column representing an element of the multi-taxa single-sequence summary statistic vector and in the order of mean, variance, skewness, and kurtosis, per single-taxon summary statistic (following same order as before: number of haplotypes, haplotype diversity, nucleotide diversity, Tajima's *D*), per partition.

Author(s)

Alexander T. Xue

References

Chan YL, Schanzenbach D, Hickerson MJ (2014) Detecting concerted demographic response across community assemblages using hierarchical approximate Bayesian computation. *Molecular Biology and Evolution*, **31**, 2501–2515.
Xue AT (2017) Multi-DICE Manual.
Xue AT, Hickerson MJ (*submitted*) Multi-DICE: R package for comparative population genomic inference under multi-taxa hierarchical co-demographic models.

See Also

`build.dice`, `roll.dice`, `play.dice`, `dice.sims`

Examples

```
#simplest execution
dice.sumstats(num.sims=5, num.taxa=10, num.haploid.samples=10,
output.directory='example')
```