Package: hybridclock (via r-universe)

October 23, 2026

Title How Long Have Hybrids Been Forming? Hybrid Population Histories from Diagnostic Markers

Version 0.1.1.9000

Description Estimates the history of hybrid populations from diagnostic markers, with or without a genetic map: the average age of the hybrids, how long ago the rarer species' ancestry entered, the number of backcrosses, and whether pure animals of the rarer species are still arriving. Population samples of hybrid index and interspecies heterozygosity, as defined by Fitzpatrick (2012) <doi:10.1186/1471-2148-12-131>, are compared with forward simulations that include drift, small breeding groups, sibling mating and separate breeding pools, so that inbreeding is not mistaken for age. Estimates come from 'Cubist' models with split-conformal intervals. Recent hybrids are also classified individually, and the number of markers needed to detect backcrossed hybrids follows Boecklen and Howard (1997) <doi:10.1890/0012-9658(1997)078[2611:GAOHZN]2.0.CO;2>.

License MIT + file LICENSE

URL https://github.com/paulmaier/hybridclock, https://paulmaier.github.io/hybridclock/

BugReports https://github.com/paulmaier/hybridclock/issues

Depends R (>= 4.1)

Imports Cubist, graphics, grDevices, parallel, ranger, stats

Suggests knitr, rmarkdown, testthat (>= 3.0.0)

VignetteBuilder knitr

Config/testthat/edition 3

Encoding UTF-8

Roxygen list(markdown = TRUE)

Config/roxygen2/version 8.1.0

LazyData true

Config/pak/sysreqs libicu-dev

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

Date/Publication 2026-10-22 17:52:13 UTC

RemoteUrl https://github.com/paulmaier/hybridclock

RemoteRef HEAD

RemoteSha e2b25b9537c2987acf77faa3a27c50d44b4ae4c9

Contents

bettongs . . . 3
build_reference . . . 4
carp . . . 7
check_markers . . . 8
chickens . . . 9
class_settings . . . 10
classification_power . . . 11
classify_hybrids . . . 12
classify_hybrids_microsat . . . 14
diagnostic_markers . . . 15
effective_markers . . . 16
estimate_history . . . 16
hybrid_scores . . . 18
hybrid_scores_microsat . . . 19
hybridclock_example . . . 20
marker_information . . . 20
marker_power . . . 21
microsat_markers . . . 22
orient_genotypes . . . 22
plot_triangle . . . 23
possums . . . 24
predict.hybridclock_model . . . 25
random_map . . . 26
sample_summaries . . . 26
shared_loci . . . 28
simulate_hybrid_zone . . . 29
train_clock . . . 32
wolves . . . 34

Index **36**

`bettongs` *Burrowing bettongs: two subspecies mixed in a fenced reserve*

Description

Burrowing bettongs, or boodies (*Bettongia lesueur*), from two subspecies were released together into a predator-proof fenced reserve in Western Australia in 2010: animals from Barrow Island and Shark Bay animals from a breeding facility at Dryandra. The reserve is closed, so no animals arrived later. The founders and animals born in the reserve, sampled each year from 2010 to 2018, were genotyped by RAD sequencing.

Usage

```
bettongs
```

Format

A list with:

genotypes Integer matrix, 215 animals x 500 SNPs, counting Shark Bay (species 2) alleles (0, 1, 2; `NA` missing). The SNPs are a random 500 of the 1,233 that are fixed between the two founder groups, each on a separate RAD locus. They are not mapped.

groups `"ref1"` (Barrow Island founders, 22), `"ref2"` (Shark Bay founders, 24) and `"Matuwa2010"` to `"Matuwa2018"` (animals born in the reserve, by year of first capture; 4 to 37 per year).

truth Data frame, one row per group: `group`, `n`, `year`, `years_since_mixing`, `max_hybrid_age` (an upper bound) and `generations_at_3yr`.

animals Data frame, one row per animal: `id`, `group`, `year_sampled` and `published_class` (the NewHybrids class published by Rick et al. 2019, Table S6: `"BWI"`, `"DRY"`, `"F1"`, `"F2"`, `"F1BWI"`, `"F1DRY"`, or founder).

description A short description.

Truth

Mixing started in 2010. Bettongs can breed in their first year, so an animal first caught in year Y is at most Y - 2010 generations from the pure founders (`max_hybrid_age`); this upper bound is the only firm fact. With the average generation length of three years, the expected age is about (Y - 2010) / 3 (`generations_at_3yr`). Pure animals of both sources kept breeding inside the fence, so new F1s and first backcrosses appear every year, and most hybrids are much younger than the bound. Year is the year of first capture, not of birth.

About a quarter of heterozygotes are called homozygous (F1s show H of about 0.75 instead of 1), so use `het_dropout = 0.25` in `classify_hybrids()`. The published classes are model-based, not known pedigrees; a few disagree with the genotypes (D6056 and D6057 are listed as F1 but are pure Barrow Island; D5893 is listed as Barrow Island but has a hybrid index of 0.38).

Licence

The genotypes are from Mendeley Data and were released under CC BY 4.0, which requires attribution: cite Rick et al. (2019) and the dataset (see Source). The founder source and sampling year of each animal, and the published classes, come from Table S6 of Rick et al. (2019), an open access article under CC BY 4.0. Changes made: 500 SNPs were selected, genotypes were recoded to count Shark Bay alleles, and only the animal labels listed above were kept. No locality data are included.

Source

Rick K (2019) SNP dataset and Capture-Mark-Recapture dataset used in the analysis for the paper Rick et al.: Population genomics of Bettongia lesueur: admixing increases genetic diversity with no evidence of outbreeding depression. Mendeley Data, version 1, doi:10.17632/vg983472v3.1 (file "SNP dataset of Bettongia lesueur at Matuwa from STACKS pipeline.vcf"). Licensed under CC BY 4.0; a subset of SNPs was taken and genotypes recoded (see data-raw/bettongs.R).

References

Rick K, Ottewell K, Lohr C, Thavornkanlapachai R, Byrne M, Kennington WJ (2019) Population genomics of *Bettongia lesueur*: admixing increases genetic diversity with no evidence of outbreeding depression. Genes 10:851. doi:10.3390/genes10110851

Examples

```
mk <- diagnostic_markers(bettongs$genotypes, bettongs$groups, "ref1", "ref2")
sc <- hybrid_scores(bettongs$genotypes, mk)
cl <- classify_hybrids(sc, het_dropout = 0.25)
table(bettongs$groups, cl$class)
```

---

build_reference *Simulate a reference set of hybrid histories*

Description

Runs `simulate_hybrid_zone()` many times with settings drawn at random from the ranges given, and stacks every generation of every run. The ranges act as the prior of the estimator trained on the result, so they should cover what is plausible for the study system. The breeding population size matters most: past about five generations of a closed hybrid swarm, drift is the only clock left, and its speed depends on population size.

Usage

```
build_reference(
  n_sims = 4000,
  generations = 50,
  breeders = c(10, 500),
  start_mix = c(0.01, 0.99),
```

```
  arrival_max = 0.1,
  kinds = c(closed = 0.2, one = 0.4, both = 0.4),
  sib_mating = c(0, 0.9),
  pools = 4,
  n_markers = 1000,
  shared_freq = NULL,
  sample_size = 20,
  missing = 0,
  founders = "pure",
  map = NULL,
  map_scale = c(1, 1),
  assort = c(0, 0.8),
  het_dropout = c(0, 0.3),
  freq = NULL,
  microsat = NULL,
  founder_impurity = c(0, 0),
  founder_samples = FALSE,
  class_settings = NULL,
  cores = 1,
  seed = 1
)
```

Arguments

`n_sims` Number of simulated histories.

`generations` Generations per history.

`breeders` Range of breeding population sizes (drawn on a log scale). An effective size (Ne) range is usually the most relevant figure.

`start_mix` Range of the founding share of species 2.

`arrival_max` Largest share of breeders replaced by arrivals per generation.

`kinds` Probabilities of closed, one-species and two-species arrival histories.

`sib_mating` Range of sibling-mating probability; half the runs use 0.

`pools` Largest number of breeding pools; half the runs use 1.

`n_markers`, `shared_freq`, `sample_size`, `missing` Passed to `simulate_hybrid_zone()`. Match these to the observed data.

`founders` Founding designs to draw from (see `simulate_hybrid_zone()`): `"pure"` for natural contact, or `"F1"`, `"BC1_1"`, `"BC1_2"` for experimental or hatchery populations.

`map` Linkage: `NULL` for unlinked markers; a marker map (see `random_map()`) used by every history; or a list with ranges `chromosomes` and `length_cM`, from which each history draws its own random map, so that an unknown genome enters the reference as uncertainty. A fixed map should hold the observed markers' real positions, in the order of the diagnostic markers; then genotype switches along chromosomes are summarized too (see `sample_summaries()`).

`map_scale` With a fixed map, a range by which positions are multiplied in each history, for an uncertain recombination rate (for example, positions in Mb with `map_scale = c(0.5, 3)` for 0.5-3 cM per Mb).

`assort` Range of the probability of mating by ancestry (see `simulate_hybrid_zone()`); half the runs use 0.

`het_dropout` Range of the probability that a heterozygote is called homozygous; half the runs use 0 unless the range excludes 0. Give `c(0, 0)` for genotyping methods without allele dropout.

`freq` Optional allele frequencies for markers that are not fixed (see `simulate_hybrid_zone()`); observed scores must then come from `hybrid_scores(method = "likelihood")`.

`microsat` Optional microsatellite markers (see `microsat_markers()`); observed scores must then come from `hybrid_scores_microsat()`, and `n_markers` is set to the number of loci.

`founder_impurity` Range of the share of diagnostic alleles that founders carry from the other species (see `simulate_hybrid_zone()`), drawn for each history.

`founder_samples` With F1 or backcross founders, also include samples of the founders themselves (generation 0), for experimental families of F1s or first backcrosses (see `simulate_hybrid_zone()`).

`class_settings` Settings of the classes behind the class shares in the summaries (see `class_settings()`): genotyping error, heterozygote dropout, linked markers, and reference allele frequencies for markers that are not fixed. They are stored with the reference and used again by `estimate_history()`, so that the data are classified the same way.

`cores` Number of processes (forked; 1 on Windows).

`seed` Base random seed; run `i` uses `seed + i`.

Details

Each run is one of three kinds of history, with probabilities `kinds`: a closed swarm (no arrivals after founding), arrivals of one species only (repeated backcrossing), or arrivals of both species (ongoing hybridization). Arrivals of each species are, with equal probability, constant, stop at a random generation, or start at a random generation, so that the reference includes hybridization that has ended and hybridization that began late.

Value

A data.frame of all generations of all runs, with the run's settings (`run`, `breeders`, `start_mix`, `arrival1`, `arrival2` and their `timing1`, `timing2` and `switch1`, `switch2` generations, `kind`, `sib_mating`, `pools`), the true history and the summaries. Attribute `settings` records the arguments.

Examples

```
ref <- build_reference(n_sims = 20, generations = 10, n_markers = 200)
dim(ref)
```

`carp` *Bighead and silver carp: an invasive hybrid zone*

Description

Bighead carp (*Hypophthalmichthys nobilis*) and silver carp (*H. molitrix*) were imported from China to the United States in the early 1970s, escaped, and now hybridize widely in the Mississippi River Basin. Native-range fish from China serve as references, with fish from three invaded sites sampled in 2009-2011, typed at 57 species-diagnostic SNPs.

Usage

```
carp
```

Format

A list with:

genotypes Integer matrix, 1,354 fish x 57 SNPs, counting silver carp (species 2) alleles (0, 1, 2; `NA` missing). The SNPs are on separate assays and are not mapped.

groups `"ref1"` (bighead carp from the Pearl and Yangtze rivers, 16), `"ref2"` (silver carp from the same rivers, 20), `"IMAR"` (Illinois River, Marseilles Reach, the invasion front, 545), `"PL26"` (upper Mississippi River, Pool 26, 365) and `"MKY"` (lower Mississippi River, Kentucky, 408).

truth Data frame, one row per group: `group`, `n`, `site`, `sampled` and `max_hybrid_age` (an upper bound).

animals Data frame, one row per fish: `id`, `group`, `published_class` (the class published by Lamer et al. 2015 from NewHybrids and manual checks: `"BH"`, `"SV"`, `"F1"`, `"F2"`, backcrosses `"BxBH"` to `"Bx4SV"`, and later `"FxBH"`, `"FxSV"`) and `birth_year` (capture year minus the age read from the postcleithrum bone).

description A short description.

Truth

Both species were imported in 1973 and were in open waters by the early 1980s. With a generation time of at least 3 years, hybrids sampled in 2009-2011 are at most about 12 generations old (37 years / 3), and probably 6-12; each site was colonized later than the basin as a whole, so its own hybrids are likely younger. Hybridization is ongoing: F1s were born as late as 2006. The published classes are inferred, not known pedigrees. The references are partly circular, because some Yangtze fish were used to choose the SNPs, so they look perfectly fixed. Nets targeted fish over 400 mm, so recent cohorts are under-represented.

Licence

The genotypes are from two Dryad packages, released under CC0 1.0 (public domain dedication). Three of the nine invaded sites were kept, and genotypes were recoded to count silver carp alleles.

Source

Data package of Lamer et al. (2015) on Dryad, doi:10.5061/dryad.kp6j2 (invaded sites). CC0 1.0.

Data package of Lamer et al. (2013) on Dryad, doi:10.5061/dryad.5j2v6 (native-range reference fish). CC0 1.0.

References

Lamer JT, Ruebush BC, Arbieva ZH, McClelland MA, Epifanio JM, Sass GG (2015) Diagnostic SNPs reveal widespread introgressive hybridization between introduced bighead and silver carp in the Mississippi River Basin. Molecular Ecology 24:3931. doi:10.1111/mec.13285

Lamer JT, Sass GG, Boone JQ, Arbieva ZH, Green SJ, Epifanio JM (2013) Restriction site-associated DNA sequencing generates high-quality single nucleotide polymorphisms for assessing hybridization between bighead and silver carp in the United States and China. Molecular Ecology Resources. doi:10.1111/17550998.12152

Examples

```
mk <- diagnostic_markers(carp$genotypes, carp$groups, "ref1", "ref2")
sc <- hybrid_scores(carp$genotypes, mk)
aggregate(cbind(S, H) ~ group, data.frame(sc, group = carp$groups), mean)
table(carp$groups, classify_hybrids(sc)$class)
```

---

check_markers — *Check that markers are diagnostic in the parents that actually bred*

Description

Markers chosen in reference samples can still be shared by the animals that founded a hybrid population, for example when a release strain or cross parent differs from the reference sample. First-generation hybrids expose such markers: at a truly diagnostic marker every F1 is heterozygous. This function finds animals that look like F1s (hybrid index near 0.5 and high interspecies heterozygosity) and reports, for each marker, how often they are heterozygous there.

Usage

```
check_markers(
  geno,
  markers,
  scores,
  S_range = c(0.4, 0.6),
  min_H = 0.7,
  flag = 0.7
)
```

Arguments

geno	Genotype matrix as in diagnostic_markers().
markers	Output of diagnostic_markers().
scores	Output of hybrid_scores() for the same animals.
S_range	Range of hybrid index for likely F1s.
min_H	Smallest interspecies heterozygosity for likely F1s.
flag	Markers heterozygous in fewer than this share of likely F1s are flagged.

Value

A data.frame with one row per marker: name, n_f1 (likely F1s genotyped), het_f1 (their share of heterozygotes) and flagged. Attribute n_f1 gives the number of likely F1s; with fewer than about 5, the check says little.

Examples

```
ex <- hybridclock_example()
mk <- diagnostic_markers(ex$genotypes, ex$groups, "species1", "species2", locus = ex$locus)
chk <- check_markers(ex$genotypes, mk, hybrid_scores(ex$genotypes, mk))
attr(chk, "n_f1")
```

---

chickens — *Chickens: an advanced intercross after eight generations*

Description

Generation F8 of an advanced intercross line between Red Junglefowl and White Leghorn chickens: one Red Junglefowl male crossed with three White Leghorn females, then kept as a closed line of about 100 birds per generation with no backcrossing. Every bird is eight generations from the pure parents.

Usage

```
chickens
```

Format

A list with:

genotypes Integer matrix, 572 birds x 514 autosomal SNPs, counting White Leghorn (species 2) alleles (0, 1, 2; NA missing).

groups "F8" for every bird. No pure birds were genotyped.

map Data frame with snp, chromosome, position (cM per meiosis, approximate: the authors' F8 map divided by 3.09, their genome-wide map expansion over eight generations) and position_f8 (cM on the authors' F8 map).

truth Data frame with group, n, generation, hybrid_age (8), backcrosses (0) and breeders.

description A short description.

Truth

All birds are F8, so the hybrid age is 8, with no backcrosses and no arrivals. The line was kept at about 100 birds per generation from F3 to F7 and produced the F8 from 122 F7 parents; the mating scheme is not reported. The loss of heterozygosity (mean H about 0.47 instead of 0.5) suggests about 50 effective breeders. The founders were outbred, so not every marker was fixed between them, which also lowers H. Which allele is White Leghorn is not stated by the authors; it was inferred from the Z chromosome (excluded here), where White Leghorn alleles make up one third of copies because of the direction of the founding cross.

Licence

The genotypes are from figshare and were released under CC0 1.0 (public domain dedication). Markers with partial calls, on the Z chromosome or on an unplaced scaffold were dropped, and genotypes were recoded.

Source

Johnsson M (2017) Open field phenotypes and genotypes (R/qtl csv format). figshare, doi:10.6084/m9.figshare.1265060. CC0 1.0.

References

Johnsson M, Williams MJ, Jensen P, Wright D (2016) Genetical genomics of behavior: a novel chicken genomic model for anxiety behavior. Genetics 202:327-340. doi:10.1534/genetics.115.179010

Johnsson M, Gustafson I, Rubin C-J, et al. (2012) A sexual ornament in chickens is affected by pleiotropic alleles at HAO1 and BMP2, selected during domestication. PLoS Genetics 8:e1002914. doi:10.1371/journal.pgen.1002914

Examples

```
# every SNP already counts White Leghorn alleles
g <- chickens$genotypes
mk <- data.frame(column = seq_len(ncol(g)), name = colnames(g), flip = FALSE,
                 freq1 = 0, freq2 = 1)
sc <- hybrid_scores(g, mk)
colMeans(sc[, c("S", "H")])
```

---

class_settings *Settings for the classes that choose between the two tiers*

Description

The share of hybrids in each early class is part of the summaries the clock learns from, and it decides whether a population is dated by its individual classes or by the clock (see `estimate_history()`). Those classes must be made the same way in the simulated reference and in the data, so the settings are given once, to `build_reference()`, and are then used for both.

Usage

```
class_settings(
  error = 0.005,
  het_dropout = 0,
  effective_markers = NULL,
  markers = NULL,
  alpha = 0.01
)
```

Arguments

`error`	Genotyping error rate assumed by the classes.
`het_dropout`	Share of heterozygotes called homozygous (see `classify_hybrids()`); for RAD or GBS data where F1s show clearly less than full heterozygosity.
`effective_markers`	Optional effective number of independent markers (see `effective_markers()`), for many linked markers.
`markers`	Optional output of `diagnostic_markers()` (columns `freq1`, `freq2`) for markers that are not fixed; with it, animals are classified from their genotype counts at the markers.
`alpha`	Significance level of the fit test for "later".

Details

With markers that are not fixed, class probabilities are averaged over markers using the reference allele frequencies, which needs only a few summary moments of the frequencies and keeps the classification fast enough to run on every simulated sample.

Value

A list of class `hybridclock_class_settings`.

Examples

```
class_settings(het_dropout = 0.2)
```

`classification_power` *Expected accuracy of individual classification*

Description

Simulates animals of each class from the reference allele frequencies of the diagnostic markers, adds genotyping error, and classifies them with `classify_hybrids()`. The result shows how often each class can be called correctly with these markers, and what it is mistaken for.

Usage

```
classification_power(markers, n = 1000, error = 0.005, alpha = 0.01, seed = 1)
```

Arguments

markers	Output of diagnostic_markers() (columns freq1, freq2).
n	Animals simulated per class.
error	Genotyping error rate used to simulate and classify.
alpha	Passed to classify_hybrids().
seed	Random seed.

Value

A matrix of proportions: true classes in rows, called classes in columns. The diagonal is the expected accuracy.

Examples

```
ex <- hybridclock_example()
mk <- diagnostic_markers(ex$genotypes, ex$groups, "species1", "species2", locus = ex$locus)
round(classification_power(mk[1:15, ], n = 200), 2)
```

---

classify_hybrids *Classify recent hybrids individually*

Description

Tests each animal against the pure, first- and second-generation hybrid classes and second backcrosses. Each class is a pair of parents, each passing on species-2 alleles with a set probability: 0 (pure species 1), 1 (pure species 2), 0.5 (F1 or F2 parent), 0.25 or 0.75 (first backcross parent). For each animal the likelihood of its genotypes under every class is compared with the best fit under any pair of parental ancestries. If no class fits (likelihood-ratio test at alpha), the animal is called a later-generation hybrid.

Usage

```
classify_hybrids(
  scores,
  alpha = 0.01,
  error = 0.005,
  genotypes = NULL,
  markers = NULL,
  effective_markers = NULL,
  het_dropout = 0
)
```

Arguments

- `scores` — Output of `hybrid_scores()` (`S`, `H`, `n_markers`).
- `alpha` — Significance level of the fit test.
- `error` — Genotyping error rate, or `"estimate"` (with `genotypes`). An estimate that reaches the top of its range (0.1) gives a warning: with many linked markers, extra variation in heterozygosity can be taken for genotyping error, and a fixed rate is then safer.
- `genotypes` — Optional genotypes at the diagnostic markers, counting species-2 alleles, as from `orient_genotypes()`, one row per row of `scores`.
- `markers` — Output of `diagnostic_markers()` matching the columns of `genotypes` (columns `freq1` and `freq2` are used).
- `effective_markers` — Optional number of effectively independent markers. Markers on the same chromosome are inherited in blocks, so they carry less independent information than their number suggests. With this, log-likelihoods are scaled by `effective_markers / n_markers` (a composite-likelihood adjustment). The best class is unchanged, but animals are no longer rejected from every class just because their markers come in blocks. See `effective_markers()`.
- `het_dropout` — Probability that a heterozygote is called homozygous (allele dropout, common in reduced-representation sequencing). It is not estimated, because heterozygosity lost to dropout cannot be told apart from heterozygosity lost to inbreeding one animal at a time; set it from replicate genotyping or known error rates.

Details

With `genotypes` and `markers`, the likelihood uses each marker's allele frequencies in the two reference groups, so markers that are nearly but not perfectly fixed are handled correctly, as in NewHybrids (Anderson and Thompson 2002) and HIest (Fitzpatrick 2012). A parent with species-2 ancestry `a` passes on the species-2 allele at marker `l` with probability `a * freq2[l] + (1 - a) * freq1[l]`. The genotyping error rate can be estimated from the data (`error = "estimate"`). Without them, markers are treated as fixed and only the genotype counts implied by `S` and `H` are used.

Several third-generation classes are indistinguishable from earlier ones at unlinked markers whatever the number of markers: F1 x F2 and F2 x F2 look like F2, and F2 x pure looks like a first backcross. Inbreeding (related parents) lowers heterozygosity and moves animals off every class, so they are called later-generation hybrids. Use `estimate_history()` for populations of advanced hybrids. Markers must be diagnostic in the populations that actually hybridized, not only in the reference samples: a marker at which one source population carries the "wrong" allele makes its hybrids look like no class.

Value

`scores` with columns `class` (best class or `"later"`), `probability` (of the best class among the tested classes) and `fit_p` (p-value of the fit test). With `error = "estimate"`, attribute `error` holds the estimated rate.

References

Anderson EC, Thompson EA (2002) A model-based method for identifying species hybrids using multilocus genetic data. Genetics 160:1217-1229.

Examples

```
ex <- hybridclock_example()
mk <- diagnostic_markers(ex$genotypes, ex$groups, "species1", "species2", locus = ex$locus)
sc <- hybrid_scores(ex$genotypes, mk)
table(ex$groups, classify_hybrids(sc)$class)
cl <- classify_hybrids(sc, error = "estimate", genotypes = orient_genotypes(ex$genotypes, mk),
                       markers = mk)
attr(cl, "error")
```

---

classify_hybrids_microsat

Classify recent hybrids from microsatellites

Description

The multi-allelic version of `classify_hybrids()`: a parent with species-2 ancestry `a` passes on allele `x` with probability `a * freq2(x) + (1 - a) * freq1(x)`. Each animal is tested against the pure, F1, F2, first- and second-backcross classes and against the best fit under any pair of parental ancestries; if no class fits (likelihood-ratio test at `alpha`), it is called `"later"`.

Usage

```
classify_hybrids_microsat(alleles, ms, alpha = 0.01, error = 0.01)
```

Arguments

`alleles`	Allele matrix as in `microsat_markers()`.
`ms`	Output of `microsat_markers()`.
`alpha`	Significance level of the fit test. Many later-generation animals (for example backcrosses of backcrosses) have genotypes close to an early class and cannot be told apart one at a time. A larger `alpha` calls more of them "later" at the cost of rejecting some true early hybrids: in a bighorn sheep pedigree with 187 microsatellites, `alpha = 0.01` kept 91% of early calls right and called 20% of later-generation sheep "later"; `alpha = 0.1` gave 85% and 44%. Population ages from `estimate_history()` do not depend on this choice.
`error`	Probability that a called allele is replaced by a random allele of that locus (genotyping error, null alleles).

Value

A data.frame with `class`, `probability` and `fit_p`, one row per animal.

Examples

```
set.seed(1)
a <- cbind(L1.a1 = c(rep(100, 10), rep(104, 10)), L1.a2 = c(rep(100, 10), rep(104, 10)))
g <- rep(c("sp1", "sp2"), each = 10)
ms <- microsat_markers(a, g, "sp1", "sp2")
classify_hybrids_microsat(rbind(a, c(100, 104)), ms)
```

---

diagnostic_markers *Find diagnostic markers between two reference groups*

Description

Keeps SNPs that are nearly fixed for different alleles in two reference groups of (presumed) pure animals: at most `threshold` frequency in one group and at least `1 - threshold` in the other. With `locus`, only the first such SNP per locus is kept, so markers are effectively unlinked.

Usage

```
diagnostic_markers(geno, groups, ref1, ref2, threshold = 0.01, locus = NULL)
```

Arguments

`geno`	Genotype matrix, animals in rows and SNPs in columns, counting copies of one allele (0, 1, 2; `NA` for missing).
`groups`	Group label of each animal.
`ref1`, `ref2`	Labels of the two reference groups. `ref2` defines species 2, whose alleles the hybrid index counts.
`threshold`	Largest frequency allowed for the minor allele in each reference group (0.01 keeps markers within 1% of fixation).
`locus`	Optional locus name of each SNP.

Value

A data.frame with one row per marker: `column` (index in `geno`), `name`, `flip` (`TRUE` when the counted allele belongs to species 1), and the counted allele's frequency in each reference group (`freq1`, `freq2`).

Examples

```
ex <- hybridclock_example()
mk <- diagnostic_markers(ex$genotypes, ex$groups, "species1", "species2", locus = ex$locus)
nrow(mk)
```

effective_markers *Effective number of independent markers*

Description

Markers on one chromosome travel together through a meiosis, broken only by crossovers. An early-generation hybrid's genotype is shaped by the recombinant gamete(s) of its hybrid parent(s): one for a backcross, two for an F2. The number of independently inherited pieces is about `meioses * (chromosomes + map l` The default of one meiosis suits backcrosses, the classes most affected; in the bear pedigree truth set it let every known backcross be classified correctly while animals with old introgression were still called later-generation.

Usage

```
effective_markers(chromosomes, length_cM = 100 * chromosomes, meioses = 1)
```

Arguments

`chromosomes`	Number of chromosomes (haploid).
`length_cM`	Total map length in centimorgans. If unknown, about 50-150 cM per chromosome is typical.
`meioses`	Number of recombinant gametes shaping the genotype (1 for a backcross, 2 for an F2).

Value

A number to pass to `classify_hybrids()` as `effective_markers`.

Examples

```
effective_markers(chromosomes = 37, length_cM = 2500)
```

estimate_history *Estimate the hybrid history of sampled populations*

Description

Gives one answer per population, from one of two tiers:

- **Individual classes.** When most hybrids confidently fit an early class (F1, F2, first or second backcross; see `classify_hybrids()`) and those are mostly F1s and backcrosses rather than F2-like animals, the ages follow directly from the classes: F1 = 1, first backcross = 1.5, second backcross = 1.75, F2 = 2 generations (hybrid age, both species), and likewise for the rare species' lines and the number of backcrosses. Intervals come from resampling animals, widened to at least the error of class-based ages in the reference at a similar share of early hybrids.
- **Population clock.** Otherwise the trained clock is used: the sample's summaries are compared with the simulated reference.

Usage

```
estimate_history(
  clock,
  scores,
  population,
  diag = NULL,
  shared = NULL,
  markers = NULL,
  positions = NULL,
  effective_markers = NULL,
  het_dropout = 0,
  repeats = 30,
  seed = 1
)
```

Arguments

`clock` Output of `train_clock()`.

`scores` Output of `hybrid_scores()` (columns `S`, `H`, `n_markers`).

`population` Population label of each animal.

`diag` Genotypes at the diagnostic markers (`orient_genotypes()`), needed if the reference used them (it always does).

`shared` Genotypes at shared loci, needed if the reference had shared loci.

`markers`, `effective_markers`, `het_dropout`
No longer used, kept so that older scripts run. The classes that decide between the two tiers are made exactly as in the reference (from S, H and the number of markers, with default settings), so that the rule learned on the reference applies; use `classify_hybrids()` with these settings for the individual calls.

`positions` Marker positions matching the columns of `diag` (`chromosome`, `position`), needed if the reference used a fixed map.

`repeats` Number of random subsets for larger populations.

`seed` Random seed.

Details

F2-like animals are not trusted as second-generation hybrids on their own: in a swarm that mates among itself, F3, F4 and later generations have the same expected hybrid index and heterozygosity as F2s, so one animal cannot show its generation. Only the population can, through linkage blocks that shrink and drift that accumulates over generations, which is what the clock reads.

The share of hybrids in early classes at which the answer switches to the classes was chosen when the clock was trained, as the point above which the classes' large errors (90th percentile) on held-out simulated histories were no worse than the clock's, or under about 30%. With few or weak markers, later-generation animals still fit early classes, so the classes are never used. Column `method` says which tier gave the answer; the other tier's estimates are kept in columns starting `clock_` and `classes_`.

Populations sampled more heavily than the reference are summarized in random subsets of the reference sample size and the results averaged (on the model's scale), so that every summary is computed exactly as in training.

Value

A data.frame with one row per population: `n`, `n_hybrids` (animals not classified as pure), `rare_species` (the species with less ancestry in the hybrids), `rare_share` (its mean share in the hybrids), `method`, `early_share` (share of hybrids in early classes), an estimate and interval (`_lower`, `_upper`) for `hybrid_age_both`, `hybrid_age_rare` and `backcrosses`, the arrival probability `arriving` and rate `arrival_rate` (always from the clock), and notes. `arriving_note` flags young swarms (hybrid age under 5 generations), where pure founders of the rarer species still form new hybrids without any arrivals and the arrival call was right only 63-67% of the time in validation. `information` is 1 minus the width of the hybrid-age interval relative to the spread of ages in the reference (log scale): near 1 the data pin the age down, near 0 the answer mostly reflects the reference's age range (few or weak markers), and `note` says so. `note` also flags populations with too few hybrids.

Examples

```
ex <- hybridclock_example()
mk <- diagnostic_markers(ex$genotypes, ex$groups, "species1", "species2", locus = ex$locus)
sc <- hybrid_scores(ex$genotypes, mk)
ref <- build_reference(n_sims = 150, generations = 12, n_markers = nrow(mk),
                       breeders = c(20, 100))
clock <- train_clock(ref, targets = "hybrid_age_both", committees = 5)
hyb <- ex$groups %in% c("site_A", "site_B")
estimate_history(clock, sc[hyb, ], ex$groups[hyb], diag = orient_genotypes(ex$genotypes, mk)[hyb, ])
```

`hybrid_scores` *Hybrid index and interspecies heterozygosity*

Description

For each animal, the hybrid index S is the share of species-2 alleles at the diagnostic markers, and the interspecies heterozygosity H is the share of markers carrying one allele from each species. With markers that are fixed or nearly fixed, these simple counts match the maximum-likelihood estimates of Fitzpatrick (2012) closely.

Usage

```
hybrid_scores(geno, markers, method = c("count", "likelihood"))
```

Arguments

`geno`	Genotype matrix as in `diagnostic_markers()`.
`markers`	Output of `diagnostic_markers()`.

method	"count" (simple counts; for fixed or nearly fixed markers) or "likelihood" (maximum-likelihood estimates that use each marker's allele frequencies in the two reference groups, for markers that are not fixed; Fitzpatrick 2012).

Value

A data.frame with S, H and n_markers (markers genotyped).

References

Fitzpatrick BM (2012) Estimating ancestry and heterozygosity of hybrids using molecular markers. BMC Evolutionary Biology 12:131.

Examples

```
ex <- hybridclock_example()
mk <- diagnostic_markers(ex$genotypes, ex$groups, "species1", "species2", locus = ex$locus)
head(hybrid_scores(ex$genotypes, mk))
```

---

hybrid_scores_microsat

Hybrid index and heterozygosity from microsatellites

Description

Maximum-likelihood hybrid index (S) and interspecies heterozygosity (H) for each animal from multi-allelic markers, following Fitzpatrick (2012): at each locus an animal carries two species-1 copies, one of each, or two species-2 copies, with probabilities 1 - S - H/2, H and S - H/2, and each copy's allele is drawn from that species' frequencies. Alleles not seen in either reference are treated as missing.

Usage

```
hybrid_scores_microsat(alleles, ms)
```

Arguments

alleles	Allele matrix as in microsat_markers().
ms	Output of microsat_markers().

Value

A data.frame with S, H and n_markers (loci genotyped).

Examples

```
set.seed(1)
a <- cbind(L1.a1 = c(rep(100, 10), rep(104, 10)), L1.a2 = c(rep(100, 10), rep(104, 10)))
g <- rep(c("sp1", "sp2"), each = 10)
ms <- microsat_markers(a, g, "sp1", "sp2")
hybrid_scores_microsat(rbind(a, c(100, 104)), ms)
```

---

hybridclock_example *Example data set*

Description

Simulates a small data set: 20 animals from each of two pure reference groups and two hybrid sites. Site A is a young hybrid swarm (4 generations, no arrivals); site B has been backcrossing into species 1 for 15 generations. Animals carry 300 diagnostic SNPs, 40 SNPs that are not quite fixed, and 150 shared SNPs, on 490 separate loci. The data are simulated afresh with a fixed seed, so every call returns the same data.

Usage

```
hybridclock_example()
```

Value

A list with `genotypes` (animals x SNPs, counts of the reference allele), `groups` (`"species1"`, `"species2"`, `"site_A"`, `"site_B"`), `locus` (locus of each SNP) and `truth` (the simulated histories of the two sites at the time of sampling).

Examples

```
ex <- hybridclock_example()
table(ex$groups)
ex$truth
```

---

marker_information *Information carried by a set of markers*

Description

The sum over markers of the squared allele-frequency difference between the two reference groups. A fixed marker contributes 1, so the total is roughly the number of fixed markers the set is worth. In simulations, accuracy of hybrid ages fell gradually with it: typical errors were about 20% at 300, 26% at 126 and 36-39% at 7-12, with honest intervals throughout.

Usage

```
marker_information(markers)
```

Arguments

markers	Output of diagnostic_markers() (columns freq1, freq2).

Value

A number.

Examples

```
ex <- hybridclock_example()
marker_information(diagnostic_markers(ex$genotypes, ex$groups, "species1", "species2"))
```

---

marker_power — *Chance that a backcrossed hybrid looks pure*

Description

After k generations of backcrossing to one species, each fixed diagnostic marker is heterozygous with probability 0.5^k, so the chance that none of n markers is heterozygous (and the hybrid passes as pure) is (1 - 0.5^k)^n (Boecklen and Howard 1997).

Usage

```
marker_power(n_markers, generations = 1:10)

markers_needed(generations = 1:10, error = 0.05)
```

Arguments

n_markers	Number of diagnostic markers (vector allowed).
generations	Generations of backcrossing (1 is a first backcross).
error	Acceptable chance of missing a hybrid.

Value

A data.frame of generations, n_markers and p_missed.

For markers_needed(), the smallest number of markers that keeps the chance of missing a hybrid of each backcross generation below error.

References

Boecklen WJ, Howard DJ (1997) Genetic analysis of hybrid zones: numbers of markers and power of resolution. Ecology 78:2611-2616.

Examples

```
marker_power(c(100, 1000), generations = 1:8)
markers_needed(1:10)
```

microsat_markers *Microsatellite markers*

Description

Allele frequencies of multi-allelic markers (such as microsatellites) in two reference groups, for hybrid_scores_microsat() and for simulating reference histories (microsat in simulate_hybrid_zone() and build_reference()). Microsatellites rarely have alleles fixed between species, so each animal's hybrid index and interspecies heterozygosity are estimated by likelihood from these frequencies.

Usage

```
microsat_markers(alleles, groups, ref1, ref2, prior = 0.5)
```

Arguments

alleles	Integer matrix of allele sizes, two columns per locus (named locus.a1, locus.a2), animals in rows; NA for missing.
groups	Group label of each animal.
ref1, ref2	Labels of the two reference groups.
prior	Pseudo-count added to every allele seen in either group, so that an allele absent from one reference sample is rare, not impossible, in that species.

Value

A list with loci (locus names), alleles (per locus, the allele sizes), and freq1, freq2 (per locus, their frequencies in each reference group).

Examples

```
set.seed(1)
a <- cbind(L1.a1 = c(rep(100, 10), rep(104, 10)), L1.a2 = c(rep(100, 10), rep(104, 10)))
ms <- microsat_markers(a, rep(c("sp1", "sp2"), each = 10), "sp1", "sp2")
ms$freq1
```

orient_genotypes *Count species-2 alleles at diagnostic markers*

Description

Count species-2 alleles at diagnostic markers

Usage

```
orient_genotypes(geno, markers)
```

Arguments

geno	Genotype matrix as in diagnostic_markers().
markers	Output of diagnostic_markers().

Value

An integer matrix (animals x markers) counting species-2 alleles.

Examples

```
ex <- hybridclock_example()
mk <- diagnostic_markers(ex$genotypes, ex$groups, "species1", "species2", locus = ex$locus)
dim(orient_genotypes(ex$genotypes, mk))
```

---

plot_triangle *Plot animals on the hybrid triangle*

Description

Plots interspecies heterozygosity (H) against hybrid index (S). Possible values fill a triangle with pure animals at the bottom corners and F1s at the top. The expected positions of first- and second-generation classes and second backcrosses are marked. Animals below the dashed curve `H = 2S(1 - S)` are less heterozygous than any offspring of unrelated parents can be on average, a sign of inbreeding.

Usage

```
plot_triangle(scores, group = NULL, col = NULL, classes = TRUE, ...)
```

Arguments

scores	Output of hybrid_scores().
group	Optional grouping for colors.
col	Colors, one per group.
classes	Mark the expected class positions.
...	Passed to graphics::points().

Value

NULL, invisibly.

Examples

```
ex <- hybridclock_example()
mk <- diagnostic_markers(ex$genotypes, ex$groups, "species1", "species2", locus = ex$locus)
plot_triangle(hybrid_scores(ex$genotypes, mk), ex$groups)
```

possums *Mountain pygmy possums: genetic rescue at 24 microsatellites*

Description

The mountain pygmy possum (*Burramys parvus*) population at Mount Buller was small and inbred. In September 2011 six males from Mount Higginbotham, in the central part of the range (isolated from Mount Buller for at least 20,000 years), were released there, and four of them sired young that season. More males, from Timms Spur, were released in September 2014. Possums were trapped every spring and genotyped at 24 microsatellites.

Usage

```
possums
```

Format

A list with:

alleles Integer matrix, 536 rows x 48 columns: allele sizes at 24 microsatellites, two columns per locus (`locus.a1`, `locus.a2`; `NA` missing), as used by `microsat_markers()`.

groups `"ref1"` (Mount Buller 2010, before the rescue, 29), `"ref2"` (Mount Higginbotham 2012, the source population, 104), `"Buller2011"` to `"Buller2015"` (Mount Buller by year; 40, 43, 67, 103 and 138), `"Higginbotham_males_2011"` (the six males released in 2011) and `"TimmsSpur_males_2014"` (the six released in 2014).

truth Data frame, one row per group: `group`, `n`, `year`, `years_since_rescue`, `hybrid_age` (expected mean hybrid age of the hybrids, where known), `max_hybrid_age` (upper bound) and `note`.

animals Data frame, one row per row of `alleles`: `id`, `tag`, `group`, `year` and `new_this_year` (first year the tag was caught at Mount Buller).

description A short description.

Truth

Possums breed as one-year-olds, so the oldest possible hybrid generation equals the number of years since the 2011 breeding season. Buller 2011 was trapped as the first males were released and is pure. In 2012 every hybrid is an F1 (hybrid age 1; the authors found 13). In 2013 the new adults were F2s and backcrosses with no new F1s (6 F2 and 23 backcrosses), and surviving F1s returned, so the hybrids average about 1.5 generations. In 2014 hybrids are at most 3 generations old, but new F1s also appeared (one 2011 male first bred in 2013, and captive-bred animals with some Higginbotham ancestry were released after the 2013 trapping); in 2015 they are at most 4 generations old, with new F1s from the Timms Spur males. The mean generation time is 1.54 years. Animals caught in several years appear once per year, so year groups overlap. Alleles from Timms Spur that Mount Higginbotham lacks appear in the 2015 sample.

Licence

The genotypes are from Dryad and were released under CC0 1.0 (public domain dedication). Rows were reordered and labelled; allele sizes are unchanged. No locality data are included.

Source

Data package of Weeks et al. (2017) on Dryad, doi:10.5061/dryad.7g988 (file `Burramys_parvus_genotypic_data_Mt_Bull` CC0 1.0.

References

Weeks AR, Heinze D, Perrin L, Stoklosa J, Hoffmann AA, van Rooyen A, Kelly T, Mansergh I (2017) Genetic rescue increases fitness and aids rapid recovery of an endangered marsupial population. Nature Communications 8:1071. doi:10.1038/s41467017011823

Examples

```
ms <- microsat_markers(possums$alleles, possums$groups, "ref1", "ref2")
y12 <- possums$groups == "Buller2012"
table(classify_hybrids_microsat(possums$alleles[y12, ], ms)$class)
sc <- hybrid_scores_microsat(possums$alleles, ms)
aggregate(cbind(S, H) ~ group, data.frame(sc, group = possums$groups), mean)
```

`predict.hybridclock_model` *Predict histories from sample summaries*

Description

Predict histories from sample summaries

Usage

```
## S3 method for class 'hybridclock_model'
predict(object, newdata, ...)
```

Arguments

`object`	Output of `train_clock()`.
`newdata`	A data.frame (or matrix) of summaries, one row per sample, as from `sample_summaries()` or `build_reference()`.
`...`	Not used.

Value

A data.frame with an estimate and interval (`_lower`, `_upper`) for each target.

Examples

```
ref <- build_reference(n_sims = 150, generations = 12, n_markers = 200, breeders = c(20, 100))
clock <- train_clock(ref, targets = "hybrid_age_both", committees = 5)
head(predict(clock, ref))
```

---

random_map *Random marker map*

Description

Places markers at random positions on chromosomes, for simulating linkage when no genetic map is available. Use the species' chromosome number and, if known, its map length; otherwise give `build_reference()` a range so that the uncertainty is part of the reference.

Usage

```
random_map(n_markers, chromosomes, length_cM)
```

Arguments

`n_markers`	Number of markers.
`chromosomes`	Number of chromosomes.
`length_cM`	Total map length in centimorgans, shared equally among chromosomes.

Value

A data.frame with `chromosome` and `position` (cM), sorted.

Examples

```
head(random_map(500, chromosomes = 11, length_cM = 1500))
```

---

sample_summaries *Summarize a population sample of hybrid scores*

Description

Turns the hybrid index (S) and interspecies heterozygosity (H) of a sample of animals, and optionally their genotypes, into the summary statistics that `train_clock()` learns from. The same function is used for simulated and observed samples, so the two are always comparable.

Usage

```
sample_summaries(
  S,
  H,
  diag = NULL,
  shared = NULL,
  n_markers = NULL,
  positions = NULL,
  class_settings = NULL
)
```

Arguments

S, H	Hybrid index and interspecies heterozygosity of each animal, as from hybrid_scores().
diag	Optional genotype matrix (animals x diagnostic markers) counting species-2 alleles, as from orient_genotypes(). Missing values allowed.
shared	Optional genotype matrix (animals x shared loci) of allele counts. Missing values allowed.
n_markers	Optional number of markers genotyped in each animal, as from hybrid_scores().
positions	Optional marker positions matching the columns of diag: a data.frame with chromosome and position. With them, the rate at which each hybrid's genotype switches between neighbouring markers along a chromosome is summarized (switch_rate, switch_rate_sd). Switches are ancestry junctions plus genotyping noise; they accumulate with every generation of recombination (Janzen et al. 2018), so they date early generations even when animals look alike in S and H.
class_settings	Settings of the classes behind the class shares (see class_settings()); the same settings must be used for the reference and the data.

Details

The sample is first oriented so that the rarer species among its hybrids counts as species 2 (S below 0.5). The summaries then describe:

- the spread of animals over the S-H triangle (means, percentiles, and the share of animals in each cell of a 10 x 6 grid);
- heterozygosity relative to its maximum for each animal's ancestry (2 * min(S, 1 - S)) and to the minimum possible for unrelated parents (2 * S * (1 - S)). Values below the latter point to inbreeding;
- with diag, how unevenly the rarer species' alleles are spread across markers (drift);
- with shared, inbreeding (shared_fis) and excess linkage disequilibrium (shared_ld) at loci that do not differ between species;
- with n_markers, the share of animals in each class of classify_hybrids() (pure, F1, F2, first and second backcrosses toward the common or the rare species, and later generations).

Value

A named numeric vector.

Examples

```
set.seed(1)
S <- c(0, 0, 0.12, 0.25, 0.3, 0.06, 0.5)
H <- c(0, 0, 0.22, 0.48, 0.41, 0.12, 1)
round(sample_summaries(S, H)[1:12], 3)
```

---

shared_loci *Find loci shared by both species*

Description

Keeps SNPs that are common in both reference groups and differ little between them. Inbreeding and linkage disequilibrium measured at these loci help separate inbreeding from hybrid age (see `sample_summaries()`).

Usage

```
shared_loci(
  geno,
  groups,
  ref1,
  ref2,
  min_maf = 0.05,
  max_diff = 0.1,
  locus = NULL
)
```

Arguments

`geno`	Genotype matrix, animals in rows and SNPs in columns, counting copies of one allele (0, 1, 2; `NA` for missing).
`groups`	Group label of each animal.
`ref1`, `ref2`	Labels of the two reference groups. `ref2` defines species 2, whose alleles the hybrid index counts.
`min_maf`	Smallest minor allele frequency required in each group.
`max_diff`	Largest allele frequency difference allowed between groups.
`locus`	Optional locus name of each SNP.

Value

A data.frame with `column`, `name` and the pooled allele frequency `freq`.

Examples

```
ex <- hybridclock_example()
nrow(shared_loci(ex$genotypes, ex$groups, "species1", "species2", locus = ex$locus))
```

`simulate_hybrid_zone` *Simulate a hybrid population forward in time*

Description

Simulates a population founded by two species that interbreed for a number of generations, with optional ongoing arrival of pure animals of either species, mating between siblings, and separate breeding pools. Each generation, a sample of offspring (for example tadpoles) is drawn from that generation's matings, scored at unlinked diagnostic markers, and summarized with `sample_summaries()`. The true history of every sampled animal is tracked alongside, so the output can be used to train and test estimators.

Usage

```
simulate_hybrid_zone(
  generations = 20,
  breeders = 100,
  start_mix = 0.5,
  arrivals = c(0, 0),
  sib_mating = 0,
  pools = 1,
  pool_crossing = 0.1,
  n_markers = 1000,
  shared_freq = NULL,
  sample_size = 20,
  missing = 0,
  sample_at = NULL,
  seed = NULL,
  founders = c("pure", "F1", "BC1_1", "BC1_2"),
  map = NULL,
  assort = 0,
  het_dropout = 0,
  junctions = FALSE,
  freq = NULL,
  microsat = NULL,
  founder_impurity = 0,
  sample_founders = FALSE,
  class_settings = NULL
)
```

Arguments

`generations`	Number of generations to simulate.
`breeders`	Number of breeding adults per generation.
`start_mix`	Share of species 2 among the founders (pure founders only).
`arrivals`	Share of breeders replaced each generation by newly arrived pure animals of species 1 and species 2: a length-2 vector for constant rates, or a matrix with one row per generation and two columns for rates that change over time (arrivals that start or stop).
`sib_mating`	Probability that an animal mates with a sibling (shared mother) when one is available.
`pools`	Number of breeding pools. Animals mate within their pool except with probability `pool_crossing`.
`pool_crossing`	Probability that a mating crosses pools.
`n_markers`	Number of diagnostic markers.
`shared_freq`	Allele frequencies of shared loci, or `NULL` for none.
`sample_size`	Number of offspring sampled each generation.
`missing`	Per-animal missing-data rates to draw from (a single value or a vector of observed rates).
`sample_at`	Generations at which to return the sampled animals' genotypes (attribute `samples`).
`seed`	Optional random seed. The global random state is restored afterwards.
`founders`	Who founds the population: `"pure"` animals of both species (in proportions `start_mix`), or all `"F1"`, or all first backcrosses to species 1 (`"BC1_1"`) or species 2 (`"BC1_2"`), as in experimental and hatchery populations.
`map`	Optional marker map for linkage: a data.frame with columns `chromosome` and `position` (cM), one row per diagnostic marker, as from `random_map()`. Markers are simulated in map order, and recombination between neighbours on a chromosome follows Haldane's map function. `NULL` (the default) treats all markers as unlinked. Shared loci are always unlinked.
`assort`	Probability that a mating is assortative by ancestry: the mate is the closest in ancestry among five random candidates (for example through mate choice or different breeding times).
`het_dropout`	Probability that a heterozygous genotype is called as homozygous (either allele), as with allele dropout in reduced- representation sequencing. Applied to the observed genotypes.
`junctions`	Also summarize genotype switches along chromosomes (requires `map`; see `sample_summaries()`). Use this only when the observed data have marker positions too.
`freq`	Optional allele frequencies for markers that are not fixed: a list or data.frame with `freq1` and `freq2`, the frequency of the counted allele at each diagnostic marker in species 1 and species 2 (as from `diagnostic_markers()`). Pure animals then draw their alleles from these frequencies, and S and H are maximum-likelihood estimates (see `hybrid_scores()`).

`microsat` Optional output of `microsat_markers()`: the diagnostic markers are then microsatellites. Each copy's species of origin is simulated, observed alleles are drawn from that species' frequencies, and S and H are estimated with `hybrid_scores_microsat()`. `n_markers` is set to the number of loci.

`founder_impurity` Share of diagnostic alleles that founders carry from the other species, for founders that were not quite pure (for example, cross parents with some old introgression). Arrivals are always pure.

`sample_founders` With F1 or backcross founders, also sample the founders themselves (generation 0), so that the output covers families of F1s or first backcrosses as well as their descendants.

`class_settings` Settings of the classes in the summaries (see `class_settings()`).

Details

Genotypes count species-2 alleles: 0 is a pure species-1 animal and 2 a pure species-2 animal at every diagnostic marker. Shared loci (optional) are polymorphic in both species at the same frequencies and are used to measure inbreeding and linkage disequilibrium.

Two hybrid ages and a backcross count are tracked for each hybrid:

- `hybrid_age_both` (hybrid age, both species): every line of the pedigree is followed back to its first pure ancestor, of either species, and the number of generations is averaged over lines. An F1 is 1, a first backcross 1.5, an F2 2. Recent backcrossing keeps it low.
- `hybrid_age_rare` (hybrid age, rare species): only the lines leading back to pure animals of the rarer species in the sample, weighted by how much ancestry each contributes: how long ago, on average, the rarer species' genes entered.
- `backcrosses`: the number of matings with a pure animal since the first cross (averaged over the two parents when both are hybrids).

Value

A data.frame with one row per generation: `generation`, the true history of the sampled hybrids (`hybrid_age_both`, `hybrid_age_rare`, `backcrosses`, `recent_share` for the share that are F1, F2 or first backcrosses, `rare_species`, the current `arrival_rate` of that species (its mean over the three generations before sampling) and `arriving` (whether that rate is above zero)), and the summaries from `sample_summaries()`.

Examples

```
sim <- simulate_hybrid_zone(generations = 8, breeders = 60, start_mix = 0.2,
                            arrivals = c(0.05, 0), n_markers = 300, seed = 1)
sim[, c("generation", "hybrid_age_both", "hybrid_age_rare", "backcrosses", "mean_S", "mean_H")]
```

`train_clock` *Train the hybrid clock on a reference set*

Description

Fits one Cubist model (Quinlan 1993; Kuhn and Quinlan 2023) per target, from the summaries of each simulated sample to its true history, and gives each estimate an interval by split-conformal calibration (Vovk et al. 2005; Lei et al. 2018). The simulated runs are divided into three sets so that no run informs more than one step: one to fit the models, one to calibrate the intervals, and one to test the finished estimator.

Usage

```
train_clock(
  reference,
  targets = c("hybrid_age_both", "hybrid_age_rare", "backcrosses", "arriving",
    "arrival_rate"),
  level = 0.9,
  split = c(fit = 0.6, calibrate = 0.2, test = 0.2),
  max_rows = 40000,
  committees = 20,
  bins = 8,
  intervals = c("normalized", "binned", "cqr"),
  cores = 1,
  seed = 1
)
```

Arguments

`reference`	Output of `build_reference()`.
`targets`	Which parts of the history to estimate. See `simulate_hybrid_zone()` for definitions. `arriving` and `arrival_rate` describe new hybridization: whether pure animals of the rarer species are still arriving, and if so what share of breeders they make up per generation.
`level`	Interval coverage.
`split`	Shares of runs used to fit, calibrate and test.
`max_rows`	Most samples used to fit each model.
`committees`	Number of Cubist committees.
`bins`	Number of ranges of the estimate within which intervals are calibrated.
`intervals`	How intervals are made: `"binned"` (split-conformal residuals within ranges of the estimate and by sample type) or `"cqr"` (conformalized quantile regression: a quantile forest gives each sample its own interval from its summaries, widened on calibration runs to the stated coverage; Romano et al. 2019), or `"normalized"` (intervals centred on the estimate, scaled by a forest's prediction of each sample's typical error, and calibrated on held-out runs; Papadopoulos et al. 2011).

`cores`	Number of processes (targets are fitted in parallel).
`seed`	Random seed.

Details

New hybridization is estimated in two parts. `arriving` is the probability that pure animals of the rarer species are still arriving (over the three generations before sampling), from a probability forest (Wright and Ziegler 2017) trained on equal numbers of histories with and without arrivals, so the prior odds are even. Its probabilities are then recalibrated on the calibration runs by logistic regression on the forest's log-odds, the estimated hybrid age and their interaction (Platt 1999), because the same summaries mean different things in young and old swarms: in a young swarm, pure founders still form new hybrids without any arrivals. `arrival_rate` is the rate if they are, from a Cubist model trained only on histories with arrivals.

Ages and rates are modeled on a log scale (`log(age)`, `log(1 + backcrosses)`, `log(arrival_rate)`). Fitting and calibration sets are balanced so that every range of the target is equally common; otherwise the most common ages in the reference set pull estimates toward them. Intervals are calibrated separately within ranges of the estimate (Mondrian conformal prediction), so that young and old estimates each get intervals of the right width. Coverage is guaranteed on average over histories like those in the reference set, which is why its settings should match the study system.

Value

An object of class `hybridclock_model`: the fitted models, the interval calibration, the features used, the reference settings and a `validation` table of accuracy on the test runs.

References

Kuhn M, Quinlan R (2023) Cubist: Rule- and instance-based regression modeling. R package.

Lei J, G'Sell M, Rinaldo A, Tibshirani RJ, Wasserman L (2018) Distribution-free predictive inference for regression. Journal of the American Statistical Association 113:1094-1111.

Quinlan JR (1993) Combining instance-based and model-based learning. Proceedings of the Tenth International Conference on Machine Learning, 236-243.

Vovk V, Gammerman A, Shafer G (2005) Algorithmic Learning in a Random World. Springer.

Platt J (1999) Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In: Advances in Large Margin Classifiers, MIT Press, 61-74.

Wright MN, Ziegler A (2017) ranger: A fast implementation of random forests for high dimensional data in C++ and R. Journal of Statistical Software 77:1-17.

Examples

```
ref <- build_reference(n_sims = 150, generations = 12, n_markers = 200, breeders = c(20, 100))
clock <- train_clock(ref, targets = "hybrid_age_both", committees = 5)
clock
```

`wolves` *Mexican wolves: crosses between two captive lineages*

Description

Captive Mexican wolves (*Canis lupus baileyi*) from two lineages that were kept apart for 15-20 years, McBride and Ghost Ranch, and wolves from the crosses made after the lineages were merged in 1995: first-generation crosses (F1), first backcrosses to McBride and second backcrosses to McBride. A young hybrid population with a known pedigree for every animal.

Usage

```
wolves
```

Format

A list with:

genotypes Integer matrix, 56 wolves x 486 SNPs, counting Ghost Ranch (species 2) alleles (0, 1, 2; `NA` missing). The SNPs are fixed between the two reference groups and typed in every reference wolf, and are spread over the 38 autosomes in proportion to their length.

groups Group of each wolf: `"ref1"` (pure McBride, 31), `"ref2"` (pure Ghost Ranch, 6), `"F1"` (5), `"BC1_MB"` (11) and `"BC2_MB"` (3).

map Data frame with `snp`, `chromosome`, `position` (cM, approximate: Mb x 0.97, the genome-wide rate of the dog linkage map) and `bp` (CanFam2).

truth Data frame, one row per group: `group`, `n`, `class` (as named by `classify_hybrids()`), `description` and `hybrid_age`.

animals Data frame, one row per wolf: `id`, `group`, `class` and `hybrid_age` from the pedigree.

description A short description.

Truth

Every wolf's class comes from the pedigree and was checked against its genotypes. Hybrid age is the number of generations back to the nearest pure ancestor, averaged over the pedigree lines: F1 = 1, first backcross = 1.5, second backcross = 1.75. The sample labelled `"547"` is a re-identified museum voucher and is an F1. The two lineages differ only by drift from very few founders, the Ghost Ranch reference is only six related wolves (including parents of the hybrids), and several hybrids are full siblings. With at most two meioses since the F1, ancestry blocks are long, so markers on one chromosome are far from independent (see `effective_markers()`).

Licence

The genotypes are from Dryad and were released under CC0 1.0 (public domain dedication). They were subset to the markers and wolves described above and recoded.

Source

Data package of Fitak et al. (2018) on Dryad, doi:10.5061/dryad.g68k008 (files `MW.ped` and `MW.map`). CC0 1.0.

References

Fitak RR, Rinkevich SE, Culver M (2018) Genome-wide analysis of SNPs is consistent with no domestic dog ancestry in the endangered Mexican wolf (*Canis lupus baileyi*). Journal of Heredity. doi:10.1093/jhered/esy009

Examples

```
mk <- diagnostic_markers(wolves$genotypes, wolves$groups, "ref1", "ref2")
sc <- hybrid_scores(wolves$genotypes, mk)
cl <- classify_hybrids(sc)
table(truth = wolves$animals$class, called = cl$class)
```

Index

∗ **datasets**
bettongs, 3
carp, 7
chickens, 9
possums, 24
wolves, 34

bettongs, 3
build_reference, 4
build_reference(), *10*, *22*, *25, 26*, *32*

carp, 7
check_markers, 8
chickens, 9
class_settings, 10
class_settings(), *6*, *27*, *31*
classification_power, 11
classify_hybrids, 12
classify_hybrids(), *3*, *11, 12*, *14*, *16, 17*, *27*, *34*
classify_hybrids_microsat, 14

diagnostic_markers, 15
diagnostic_markers(), *9*, *11–13*, *18*, *21*, *23*, *30*

effective_markers, 16
effective_markers(), *11*, *13*, *34*
estimate_history, 16
estimate_history(), *6*, *10*, *13, 14*

graphics::points(), *23*

hybrid_scores, 18
hybrid_scores(), *9*, *13*, *17*, *23*, *27*, *30*
hybrid_scores_microsat, 19
hybrid_scores_microsat(), *6*, *22*, *31*
hybridclock_example, 20

marker_information, 20
marker_power, 21
markers_needed *(*marker_power*)*, 21
microsat_markers, 22
microsat_markers(), *6*, *14*, *19*, *24*, *31*

orient_genotypes, 22
orient_genotypes(), *13*, *17*, *27*

plot_triangle, 23
possums, 24
predict.hybridclock_model, 25

random_map, 26
random_map(), *5*, *30*

sample_summaries, 26
sample_summaries(), *5*, *25*, *28–31*
shared_loci, 28
simulate_hybrid_zone, 29
simulate_hybrid_zone(), *4–6*, *22*, *32*

train_clock, 32
train_clock(), *17*, *25, 26*

wolves, 34