

Package ‘RADami’

February 11, 2017

Type Package

Title Phylogenetic Analysis of RADseq Data

Version 1.1-2

Date 2017-02-10

Maintainer Andrew Hipp <ahipp@mortonarb.org>

Description Implements import, export, manipulation, visualization, and downstream (post-clustering) analysis of RADseq data, integrating with the 'pyRAD' package by Deren Eaton.

License GPL-3

Depends ape (>= 4.0)

Imports geiger, Biostrings, phangorn, IRanges, hierfstat

NeedsCompilation no

Repository CRAN

Date/Publication 2017-02-11 10:04:29

Author Andrew Hipp [aut, cre],
Klaus Schliep [ctb]

R topics documented:

RADami-package	2
compare.all.trees	3
consensus.pyRAD	4
filter.by	5
gen.RAD.loci.datasets	6
genotypes.pyRAD.loci	7
genTrees	9
get.raxml.siteLikelihoods	10
get.raxml.treeLikelihoods	11
grab.pyRAD.locus	11
lengths.report	12
locus.dist	13
match.lnL.to.trees	14
oak.rads	15

oak.tree	16
overlap.report	16
pairwise.fst.loci	17
plot.locus.dist	18
plot.partitionedRAD	20
plot.rankedPartitionedRAD	22
print.pyRAD.loci	24
rad2mat	24
rad2phy	25
rank.partitionedRAD	26
read.pyRAD	28
rtreePhylo	29
subset.pyRAD.loci	30
summary.pyRAD.loci	31
tidyName	32
write.DNAStringSet	33

Index	35
--------------	-----------

RADami-package	<i>Phylogenetic Analysis of RADseq Data</i>
----------------	---

Description

This package implements import, export, manipulation, visualization, and downstream (post-clustering) analysis of RADseq data, integrating with the pyRAD package by Deren Eaton. It begins with output from the pyRAD package by Deren Eataon (<http://dereneaton.com/software>; Eaton and Ree, 2013) and implements many analyses conducted in a recent phylogenetic analysis of oaks (Hipp et al., accepted).

Details

Package:	RADami
Type:	Package
Version:	1.1-2
Date:	2017-02-10
License:	GPL-3

Author(s)

Andrew Hipp

Maintainer: Andrew Hipp <ahipp@mortonarb.org>

References

- Eaton D.A.R. & Ree R.H. (2013). Inferring Phylogeny and Introgression using RADseq Data: An Example from Flowering Plants (Pedicularis: Orobanchaceae). *Systematic Biology* 62: 689-706.
- Hipp A.L., Eaton D.A.R., Cavender-Bares J., Fitzek E., Nipper R. & Manos P.S. 2014. A framework phylogeny of the American oak clade based on sequenced RAD data. *PLoS ONE* 9: e93975. <http://www.dereneaton.com/software/pyrad/>

compare.all.trees	<i>Compare all trees in a treeset, and return a matrix (lower diagonal only) indicating which pairwise comparisons are identical</i>
-------------------	--

Description

Compare all trees in a treeset, and return a matrix (lower diagonal only) indicating which pairwise comparisons are identical

Usage

```
compare.all.trees(treeset, ...)
```

Arguments

treeset	an object of class multiPhylo or a list of phylo objects
...	additional arguments to be passed along to all.equal.phylo

Details

Slow, tedious one-on-one comparison of trees, depending on the implementation in ape. As written, this function is sensitive to rooting.

Value

A matrix of TRUE and FALSE values depending on whether the compared trees are identical in topology

Author(s)

Andrew Hipp

See Also

`link{unique.multiPhylo}`

consensus.pyRAD	<i>Generate a consensus sequence for each of the loci in a pyRAD.loci object</i>
-----------------	--

Description

Uses consensusString [S4 method for signature 'DNAStringSet'] from the Biostrings package to create a consensus sequence for each locus in a pyRAD.loci object. All options in consensusString can be passed in. As a side-effect, writes a fasta-style file to writeFile.

Usage

```
consensus.pyRAD(pyIn, from = NA, to = NA, fastaNames = T,
               writeFile = 'rads.con.txt', cores = 1, ...)
```

Arguments

pyIn	an object of class pyRAD.loci
from	first locus to summarize (by number, sequentially from the beginning of the file), or NA if all loci are to be used
to	last locus to summarize (by number, sequentially from the beginning of the file), or NA if all loci are to be used
fastaNames	use TRUE to append a > to the front of each sequence title, making a fasta-style file
writeFile	filename to which to write the sequences
cores	number of cores to use; requires the parallel package
...	extra arguments to consensusString

Value

A character vector of sequences, with locus names as the vector names.

Author(s)

Andrew Hipp

Examples

```
data(oak.rads)
# provide filename for export in line below
temp = consensus.pyRAD(oak.rads, from=1, to = 10, writeFile = NA)
temp
```

filter.by	<i>Identify names of loci for which at the requested taxa are present, either all or some number of them</i>
-----------	--

Description

Returns the names of loci for which the requested taxa are present at some threshold

Usage

```
filter.by(dat, taxa, req = NULL,
         threshInds = 3, threshTaxa = 2,
         use.tidyName = FALSE, ...)
```

Arguments

dat	an object of class <code>pyRAD.loci</code> or <code>summary.pyRAD.loci</code>
taxa	a list of character vectors of taxa or individuals
req	a character vector listing taxa that are required; defaults to none
threshInds	a vector of integers corresponding to taxa, indicating the threshold number of individuals
threshTaxa	a single integer, indicating how many taxa are required at their <code>threshInds</code> level to export a locus
use.tidyName	whether to use the <code>tidyName</code> function for matching taxon names; useful if there is a mix of uppercase, lowercase, special characters etc. that are not really essential for distinguishing taxa
...	extra arguments to <code>tidyName</code>

Details

Not currently called by any other functions in this package, but useful for quickly identifying loci for which data are present in a specified number of individuals

Value

A list with two items:

loci	a character vector of loci that satisfy the filter criteria
loc.mat	a boolean matrix indicating for each locus which taxa satisfy their individuals threshold

Author(s)

Andrew Hipp

See Also

[subset.pyRAD.loci](#)

gen.RAD.loci.datasets *Generate datasets for partitioned RAD analysis*

Description

This function takes a set of trees (typically from `genTrees`), RAD loci, and some parameters about how taxa to include in the datasets, how many taxa at minimum need to be included in the datasets, and whether only variable loci will be included.

Usage

```
gen.RAD.loci.datasets(rads, trees = "none", loci = "all",
                     taxa = "all", minTaxa = 4,
                     onlyVariable = TRUE, fileBase = "DEFAULT",
                     splitInto = 1,
                     cores = 2,
                     raxSinglePath = "raxmlHPC-AVX",
                     raxMultiPath = "raxmlHPC-PTHREADS-AVX",
                     header = "#!/bin/sh")
```

Arguments

<code>rads</code>	an object of class <code>pyRAD.loci</code>
<code>trees</code>	a set of trees, or 'none' if no trees are being used (e.g., if you just want to export loci one-by-one for analysis)
<code>loci</code>	a vector of locus names to include in analysis, or 'all.' Typically, use 'all' here, because part of what this function does is subset loci for you based on criteria you provide
<code>taxa</code>	a vector of taxa to include in analysis, or 'all' for all taxa
<code>minTaxa</code>	an integer, specifying the minimum number of taxa required to export a locus
<code>onlyVariable</code>	either TRUE or FALSE depending on whether loci should be variable or not to be exported
<code>fileBase</code>	base name for the directories and files created by this function; defaults to a handy dated format
<code>splitInto</code>	the number of folders to split files into for analysis, in case you want to run several instances of RAxML to get through analysis more quickly
<code>cores</code>	the number of processing cores to use for the RAxML full-matrix analysis
<code>raxSinglePath</code>	the path to your RAxML executable, single-core installation
<code>raxMultiPath</code>	the path to your RAxML executable, multi-core installation
<code>header</code>	header for the top of your shell scripts / bash files; not necessary for a Windows batch file (.bat), but defaults to the header needed for a Linux shell script

Details

This function subsets the treeset you give it for all the loci imported, then matched loci and treesets and a batch file for calculating the likelihood of all trees for each locus. It is called for the side-effect of writing all these files.

All files are exported to a series of folders equal in number to `splitInto + 1`. As currently written, the folder with the suffix `.0` holds all batch files, an index of trees that matches the trees you hand to the function (as `trees`) with the trees it returns in each subset, and the `.info` output files that are used by `get.raxml.treeLikelihoods` as well as the site-likelihood files. The alignments and trees, as well as the reduced alignments created by RAxML, remain in the split folders (1 through `splitInto`).

This function is the workhorse of the partitioned RAD analysis and is time-consuming to run. It could be executed in parallel and should be in future versions. What is most time-consuming is that for every treeset exported, the function makes sure that all trees are unique (see `unique.multiPhylo` from `ape`).

Value

no value. Called for the side effect of the files it exports.

Author(s)

Andrew Hipp

References

Hipp A.L., Eaton D.A.R., Cavender-Bares J., Fitzek E., Nipper R. & Manos P.S. (Accepted pending revision). A framework phylogeny of the American oak clade based on sequenced RAD data. PLoS ONE.

See Also

[genTrees](#), [get.raxml.treeLikelihoods](#), [match.lnL.to.trees](#), [rank.partitionedRAD](#)

`genotypes.pyRAD.loci` *Generates genotypes from a subset.pyRAD.loci object*

Description

Makes a dataframe of SNP calls from a `pyRAD.loci` object for export to `hierstat`

Usage

```
genotypes.pyRAD.loci(dat, groups, loci = 'all', taxa = 'all',
                    useSnps = c('first', 'all'), concat = c(FALSE, TRUE),
                    use.tidyName = FALSE,
                    na.rm = c('none', 'columns', 'rows'),
                    maxAlleles = 2,
                    tidyVals = c('-', '.', '>', '_', ' '),
                    sortByGroups = TRUE,
                    variable.only = FALSE, make.dummy.column = TRUE,
                    alleleDigits = 1,
                    toInteger = TRUE, missingData = "00",
                    cores = 1)
```

Arguments

<code>dat</code>	an object of class <code>subset.pyRAD.loci</code>
<code>groups</code>	a list of vectors, each vector designating a set of individuals to count for each group
<code>loci</code>	a character vector of loci to include; defaults to all loci
<code>taxa</code>	a character vector of taxa to include; defaults to all taxa
<code>useSnps</code>	whether to use the first SNP in each locus or all
<code>concat</code>	boolean: whether to concatenate SNPs or return by locus
<code>use.tidyName</code>	choose whether or not to use <code>tidyName</code> when matching names
<code>na.rm</code>	whether to remove missin data by columns, by rows, or not at all
<code>maxAlleles</code>	maximum number of alleles per locus
<code>tidyVals</code>	characters to be stripped out by <code>tidyName</code> , if using
<code>sortByGroups</code>	boolean: whether to sort snps by groups
<code>variable.only</code>	boolean: whether to keep only variable sites
<code>make.dummy.column</code>	boolean: whether to add a dummy column when there is only one SNP (currently added as a work-around for <code>wc</code>)
<code>alleleDigits</code>	how many digits per allele
<code>toInteger</code>	boolean: whether to return SNPs as integers
<code>missingData</code>	string to use for missing data
<code>cores</code>	how many processing cores to use; if <code>cores = 1</code> , <code>lapply</code> is used throughout instead of <code>mclapply</code>

Details

current not tightly integrated with `hierfstat`, but designed to be used by it

Value

data.frame of genotypes

Author(s)

Andrew Hipp

See Also[subset.pyRAD.loci](#)

genTrees	<i>Generate permuted trees using rNNI or rtree, and check for uniqueness.</i>
----------	---

Description

This function generates a specified number of trees by permutation or random generation, then checks to make sure that they are unique before returning the original tree plus the generated trees

Usage

```
genTrees(x, N = 200, filebase = "trial", method = c("nni", "random"),
        maxmoves = 2, perms = "DEFAULT", software = c("raxml"), ...)
```

Arguments

x	an object of class "phylo"
N	number of new trees to return
filebase	base name of the file to which trees will be written
method	method of generating the trees, either nearest neighbor interchange (nni) or random
maxmoves	maximum number nni moves
perms	number of permutations per nni stratum; defaults to the number of single-step nni trees, then the remainder of N divided by the number of nni strata requested (negatives not allowed)
software	not currently used; in an earlier version, would write out analysis file for PAUP or RAxML
...	further arguments to be passed to rtree

Details

The main need for this function is to ensure that the trees you get back are unique... mainly just a wrapper to rNNI and rtree. perms defaults to:

```
c(length(nni(x)), max(1, as.integer(N - (length(nni(x))))))
```

Value

an object of class "multiPhylo"

Author(s)

Andrew Hipp

See Also

[gen.RAD.loci.datasets](#), [compare.all.trees](#)

Examples

```
data(oak.tree)
out <- genTrees(oak.tree, N = 10, maxmoves = 2, perms = c(5, 5))
plot(out[[1]]) # this is the tree you input
plot(out[[2]]) # this is the first generated tree...
plot(out[[11]]) # ... and this is the last generated tree.
```

get.raxml.siteLikelihoods

Read likelihoods from a RAxML_perSiteLLs file

Description

This function is used to import likelihoods from a RAxML site likelihoods export.

Usage

```
get.raxml.siteLikelihoods(x)
```

Arguments

x Filename of the RAxML_perSiteLLs file

Details

Not used in any downstream analyses currently in RADbuddy.

Value

A matrix with trees as rows, site lnL as columns

Author(s)

Andrew Hipp

See Also

[get.raxml.treeLikelihoods](#), [match.lnL.to.trees](#)

`get.raxml.treeLikelihoods`*Extract likelihoods from a RAxML info file*

Description

Reads the info file from a RAxML site-likelihood analysis with multiple input trees. Probably not often needed on its own, but used in `match.lnL.to.trees`.

Usage

```
get.raxml.treeLikelihoods(x, logfile = NA)
```

Arguments

<code>x</code>	file name of a RAxML .info file from site-likelihood analysis
<code>logfile</code>	name of a log file, useful for recording any files that were not successfully read in

Value

A named vector of class `double` with tree likelihoods, where the names are character equivalents of the tree numbers; or, if the file had no trees in it, the character vector "FAIL"

Author(s)

Andrew Hipp

See Also

[match.lnL.to.trees](#), [get.raxml.siteLikelihoods](#)

`grab.pyRAD.locus`*Extract sequence data for one or a few loci*

Description

This function is pretty rudimentary and I've used it but little. As much as anything, it's a reminder for how to get sequence data out of a `pyRAD.loci` object.

Usage

```
grab.pyRAD.locus(pyDat, locName,  
  dat.format = c("text", "fasta", "matrix"))
```

Arguments

pyDat	an object of class pyRAD.loci
locName	character name of the locus or loci you want
dat.format	currently takes text, fasta, or matrix, but the fasta format would still need to be written to a file

Value

A named character vector if dat.format is 'text' or 'fasta', otherwise a matrix

Author(s)

Andrew Hipp

lengths.report *Return the lengths of all loci in a pyRAD.loci object*

Description

Return the lengths of all loci in a pyRAD.loci object

Usage

```
lengths.report(dat, numtodo = 10,
               reportInterval = 2000,
               high.mem = TRUE)
```

Arguments

dat	an object of class pyRAD.loci
numtodo	an integer specifying how many loci to return the length of
reportInterval	how many loci between reports of time required to complete when using high.mem = FALSE
high.mem	forces the code to loop through instead of using sapply

Details

Called by summary.pyRAD.loci, and thus not ordinarily needed by users. If looping is used to get lengths, the function runs very slowly, and reportInterval can be useful. In early tests, I was having memory allocation issues, so high.mem was useful. I haven't needed it since then, but maybe you will.

Value

A named vector of class integer, with names corresponding to loci whose lengths are returned

Author(s)

Andrew Hipp

See Also[read.pyRAD](#), [summary.pyRAD.loci](#)**Examples**

```
data(oak.rads)
lengths.report(oak.rads, -1) # returns all loci
lengths.report(oak.rads, 20) # returns only the first 20 loci
identical(oak.rads$radSummary$locus.lengths, lengths.report(oak.rads, -1))
```

locus.dist	<i>Generate a pairwise similarity matrix for individuals based on locus-sharing patterns</i>
------------	--

Description

This function returns a pairwise similarity matrix based on how many loci or the proportion of loci shared between individuals.

Usage

```
locus.dist(pyIn, proportional = TRUE, upper = TRUE, diagonal = TRUE)
```

Arguments

pyIn	an object of class <code>pyRAD.loci</code>
proportional	set to TRUE for a proportional similarity (scaled from 0 to 1), FALSE for absolute number of loci shared
upper	set to TRUE for a symmetrical matrix, FALSE for a lower-only matrix
diagonal	set to TRUE to get the number (or percentage) of total loci reported for each individual

Details

Under the `proportional = TRUE` option, this function normalizes based on all loci in the dataset. Thus, two individuals that share 90 have sequences for only 0.9 instead of 1.0.

Value

A square matrix with individuals as the rows and columns, and proportions or whole numbers in the cells

Author(s)

Andrew Hipp

See Also[plot.locus.dist](#)**Examples**

```

data(oak.rads)
temp <- locus.dist(oak.rads, proportional = TRUE)
diag(temp) # this is how well everyone sequenced as a proportion
temp <- locus.dist(oak.rads, proportional = FALSE)
temp[c(1,28),]# how well did a successful sequencing do in comparison to a poor sequencing?
temp[6:7,] # ... and 2010 sequencing (upper row) vs. 2012 sequencing of the same library?

```

match.lnL.to.trees	<i>Match the likelihoods returned on a subset of trees for numerous loci to the trees originally input</i>
--------------------	--

Description

This function reads in tree likelihoods for a series of loci exported from `gen.RAD.loci.datasets` and matches those up to the trees originally created. Because `gen.RAD.loci.datasets` only exports trees that include the taxa in each locus exported, these likelihoods are matched up to all trees in the original dataset that are compatible with each pruned tree.

Usage

```

match.lnL.to.trees(directory = "getwd()",
                  lnLprefix = "RAxML_info.",
                  lnLsuffix = ".lnL",
                  treeIndexFile = "tree.index.lines.txt",
                  locus.names = NULL, ...)

```

Arguments

directory	path for the directory in which the 'info' and 'lnL' files were saved by RAxML
lnLprefix	prefix for the RAxML info files; generally should not be changed, unless RAxML changes this down the line
lnLsuffix	suffix appended to the files; this is currently should not be changed, as it is hard-coded in <code>gen.RAD.loci.datasets</code>
treeIndexFile	names of the tree index file; again, this file name is hard-coded in <code>gen.RAD.loci.datasets</code>
locus.names	a vector of locus names, if you want to subset by loci at the outset of analysis; typically, subsetting is just done by the function
...	additional parameters, not currently passed along to anybody

Details

The idea behind this function is that we want to let loci vote only on trees they are qualified to vote on. In an earlier implementation, each locus was forced to vote on every tree, unpruned, and the results were simply noisy. Here, we take a set of trees and, for each locus out a treeset pruned to just that taxon set for each taxon set. We then calculate the site likelihoods based on those treesets.

Every locus is, in fact, evaluating every tree, but because many trees become topologically identical after pruning, we have to set an equal score to trees that are topologically identical... and which trees are topologically identical differs among loci.

Value

A matrix of likelihoods, with loci as rows and the original trees as columns

Author(s)

Andrew Hipp

References

Hipp A.L., Eaton D.A.R., Cavender-Bares J., Fitzek E., Nipper R. & Manos P.S. (Accepted pending revision). A framework phylogeny of the American oak clade based on sequenced RAD data. PLoS ONE.

See Also

[gen.RAD.loci.datasets](#), [plot.rankedPartitionedRAD](#), [rank.partitionedRAD](#)

oak.rads

101-locus RAD dataset, in pyRAD.loci format

Description

This dataset includes 101 RAD loci from a recent phylogenetic study of oaks (Hipp et al. in review).

Usage

oak.rads

Format

An object of class `pyRAD.loci`

References

Hipp A.L., Eaton D.A.R., Cavender-Bares J., Fitzek E., Nipper R. & Manos P.S. (Accepted pending revision). A framework phylogeny of the American oak clade based on sequenced RAD data. PLoS ONE.

oak.tree	<i>28-tip oak phylogeny, in phylo format</i>
----------	--

Description

This dataset is the 28-tip oak framework oak phylogeny published in Hipp et al. (in revision). Tips in this phylogeny match up with the tips in the oak RAD dataset provided with this package.

Usage

```
oak.tree
```

Format

An object of class phylo

References

Hipp A.L., Eaton D.A.R., Cavender-Bares J., Fitzek E., Nipper R. & Manos P.S. (Accepted pending revision). A framework phylogeny of the American oak clade based on sequenced RAD data. PLoS ONE.

See Also

[oak.rads](#)

overlap.report	<i>Generates a report on locus overlap between pairs of samples named in a structured fashion</i>
----------------	---

Description

Provided a pyRAD.loci object with a subset of paired sample names that differ only by a suffix (defined here as repPattern and origPattern), this function will return a table summarizing the overlap pattern

Usage

```
overlap.report(dat, repPattern = "_re", origPattern = "_h")
```

Arguments

dat	an object of class pyRAD.loci or summary.pyRAD.loci, or just the indsMat object from a summary.pyRAD.loci object
repPattern	suffix for the first half of each pair
origPattern	suffix for the second half of each pair

Details

This may not be of use to too many people, but we used it in the article below. In our case, the samples were paired technical replicates, but you could imagine sister species, populations or anything else being the replicates of interest.

Value

a matrix with pairs of individuals as rows and overlap statistics as columns

Author(s)

Andrew Hipp

References

Hipp A.L., Eaton D.A.R., Cavender-Bares J., Fitzek E., Nipper R. & Manos P.S. (Accepted pending revision). A framework phylogeny of the American oak clade based on sequenced RAD data. PLoS ONE.

See Also

[help, ~~~](#)

Examples

```
data(oak.rads)
overlap.report(oak.rads)
```

`pairwise.fst.loci` *Returns pairwise FST list for groups of individuals*

Description

for use with RADami

Usage

```
pairwise.fst.loci(dat, group.list, to.do,
                  minInds = 3, do.hist = FALSE,
                  cores = 1, ...)
```

Arguments

dat	an object of class subset.pyRAD.loci
group.list	named list of character vectors, indicating groups of individuals
to.do	which groups to generate pairwise data for
minInds	minimum number of individuals allowed per locus
do.hist	boolean; whether a histogram of pairwise FST should be written to the working directory
cores	how many processing cores to use
...	additional parameters passed along to genotypes.pyRAD.loci

Value

A list with the following elements:

fst	pairwise fst list
groups	groups used in the analysis
genotypes	genotype data
sp.pairs	species pairs analyzed
snpLocs	location of SNPs
timestamp	when analysis was conducted

Author(s)

Andrew Hipp

See Also

[subset.pyRAD.loci](#)

plot.locus.dist	<i>Plot a phylogeny with a graphical representation of locus sharing</i>
-----------------	--

Description

Plots a phylogeny lined up with the locus-sharing similarity matrix in graphical style

Usage

```
## S3 method for class 'locus.dist'
plot(x, tr, trW = 3, plotW = 5, labelsW = 3, plotGap = 0.25,
     scalar = 1.5, barH = 1, barW = 10, bar.cex = 0.5,
     point.pch = c(21, 21),
     cols = c('black', 'red'), ...)
```

Arguments

x	output from locus.dist
tr	an object of class phylo with tips named the same as the rows of locD
trW	width of the panels holding the tree
plotW	width of the panel holding the matrix plot
labelsW	width of the panel holding the taxon labels (right side of plot)
plotGap	gap between the side panels and the central panel
scalar	the scalar for the dots in the locus distance plot
barH	height of the panel holding the bars
barW	width of individual bars; goes to lwd
bar.cex	font size for numbers above the bars
point.pch	plot character (pch) for the off-diagonal and diagonal elements of the matrix plot
cols	color for the off-diagonal and diagonal elements of the matrix plot
...	additional arguments passed along to text

Details

Use default options in locus.dist to get a sensible plot. Also, while different plotting characters can be defined for the off-diagonal and diagonal elements of the matrix plot, these may be hard to compare with each other in size; probably the diagonal elements are better distinguished by color or shading.

Currently, this function is implemented using layout, which makes it rather clunky and inflexible.

Value

A list comprising:

Xs	the x-coordinates of dots in the matrix plot
heights	heights of the bars at the top of the plot, which show the overlap of each individual averaged over all other individuals on the plot

Author(s)

Andrew Hipp

See Also

[plot.rankedPartitionedRAD](#), [locus.dist](#)

Examples

```

data(oak.rads)
data(oak.tree)
plot.locus.dist(locus.dist(oak.rads, proportional = TRUE), oak.tree)
# and now a nicer-looking plot:
plot.locus.dist(locus.dist(oak.rads, proportional = TRUE, upper = TRUE),
                ladderize(oak.tree), trW = 1, labelsW = 1, scalar = 3)

```

plot.partitionedRAD *Plot a partitionedRAD object*

Description

This function creates a plot with 1 or more panels describing the number of loci supporting, disfavoring, or ambiguous for each tree, plotted against log-likelihood

Usage

```

## S3 method for class 'partitionedRAD'
plot(x,
     fileprefix = NULL,
     lnL.break = NULL,
     regression = NULL,
     ci = NULL,
     widthScalar = .85,
     panels = c('bestMat', 'worstMat', 'doubleCountMat'),
     squareSize = switch(as.character(length(panels)), '2' = 5, '3' = 3),
     primeTreeColor = 'red',
     primeTreeCharacter = 19,
     highlight.points = NULL,
     highlight.colors = NULL,
     filebase = 'DEFAULT',
     ...)

```

Arguments

x	an object of class partitionedRAD
fileprefix	prefix of the files to save; as written, you have to provide some prefix to identify the files if you want a pdf, but the remainder of the file name defaults to a combination of parameters and date
lnL.break	a numeric vector, which should be the length of panels; if not null, the lowest tree likelihood to show on each panel
regression	a boolean vector, which should be the length of panels; if not null, whether to draw a linear fit line on each panel
ci	a numeric vector, the length of panels; what CI to show on each regression, or \emptyset for no CI

<code>widthScalar</code>	a multiplier on the width of the entire plot to get the plotting panels square or not as you prefer
<code>panels</code>	a character vector describing the order of panels; can be any of "bestMat", "worstMat", "doubleCountMat" in any order
<code>squareSize</code>	the size of the panels in inches; defaults to 5" for a 2-panel plot, 3" for a 3-panel plot, only because that works well in my trials
<code>primeTreeColor</code>	the color of the dot representing the first tree in the matrix, which should be your original tree (unpermuted) if you've been following the workflow from <code>genTrees</code> through this final and exciting plot
<code>primeTreeCharacter</code>	the character of the dot representing the first tree in the matrix
<code>highlight.points</code>	number of any trees to highlight
<code>highlight.colors</code>	colors vector or single color for highlighted points
<code>filebase</code>	the base of the file name for exported plots
<code>...</code>	additional arguments passed to <code>plot</code> and <code>rank.partitionedRAD</code>

Details

The x-axis will either be the tree likelihoods estimated by RAxML in a separate analysis (e.g., using the entire dataset), or the summed locus likelihoods from the partitioned RAD analysis if no filename is provided. This function as written prints to pdf if you enter a `fileprefix`, to the screen if you don't. Default filebase pastes `rad.partitioned` with the date and the parameter values used. If default filebase is not desired, enter something other than "DEFAULT."

Value

This function is called primarily for its side-effect of printing plots to a file. It returns, invisibly, a list of the fit objects for each panel, only relevant if you set any elements of regression to TRUE.

Author(s)

Andrew Hipp

References

Hipp A.L., Eaton D.A.R., Cavender-Bares J., Fitzek E., Nipper R. & Manos P.S. (Accepted pending revision). A framework phylogeny of the American oak clade based on sequenced RAD data. PLoS ONE.

See Also

[rank.partitionedRAD](#), [plot.locus.dist](#)

plot.rankedPartitionedRAD

Plot a rankedPartitionedRAD object

Description

This function creates a plot with 1-3 panels describing the number of loci supporting, disfavoring, or ambiguous for each tree, plotted against log-likelihood

Usage

```
## S3 method for class 'rankedPartitionedRAD'
plot(x,
     tree.lnL.file = NULL,
     fileprefix = NULL,
     lnL.break = NULL,
     regression = NULL,
     ci = NULL,
     widthScalar = 0.85,
     panels = c("bestMat", "worstMat", "doubleCountMat"),
     squareSize = switch(as.character(length(panels)), '2' = 5, '3' = 3),
     primeTreeColor = "red",
     primeTreeCharacter = 19,
     highlight.points = NULL,
     highlight.colors = NULL,
     filebase = 'DEFAULT',
     ...)
```

Arguments

x	an object of class rankedPartitionedRAD
tree.lnL.file	file name of the info file written by RAxML; optional
fileprefix	prefix of the files to save; as written, you have to provide some prefix to identify the files if you want a pdf, but the remainder of the file name defaults to a combination of parameters and date
lnL.break	a numeric vector, which should be the length of panels; if not null, the lowest tree likelihood to show on each panel
regression	a boolean vector, which should be the length of panels; if not null, whether to draw a linear fit line on each panel
ci	a numeric vector, the length of panels; what CI to show on each regression, or 0 for no CI
widthScalar	a multiplier on the width of the entire plot to get the plotting panels square or not as you prefer
panels	a character vector describing the order of panels; can be any of "bestMat", "worstMat", "doubleCountMat" in any order

<code>squareSize</code>	the size of the panels in inches; defaults to 5" for a 2-panel plot, 3" for a 3-panel plot, only because that works well in my trials
<code>primeTreeColor</code>	the color of the dot representing the first tree in the matrix, which should be your original tree (unpermuted) if you've been following the workflow from <code>genTrees</code> through this final and exciting plot
<code>primeTreeCharacter</code>	the character of the dot representing the first tree in the matrix
<code>highlight.points</code>	number of any trees to highlight
<code>highlight.colors</code>	colors vector or single color for highlighted points
<code>filebase</code>	the base of the file name for exported plots
<code>...</code>	additional arguments passed to <code>plot</code>

Details

The x-axis will either be the tree likelihoods estimated by RAxML in a separate analysis (e.g., using the entire dataset), or the summed locus likelihoods from the partitioned RAD analysis if no filename is provided. This function as written prints to pdf if you enter a `fileprefix`, to the screen if you don't. Default filebase pastes `rad.partitioned` with the date and the parameter values used. If default filebase is not desired, enter something other than "DEFAULT."

Value

This function is called primarily for its side-effect of printing plots to a file. It returns, invisibly, a list of the fit objects for each panel, only relevant if you set any elements of regression to TRUE.

Author(s)

Andrew Hipp

References

Hipp A.L., Eaton D.A.R., Cavender-Bares J., Fitzek E., Nipper R. & Manos P.S. (Accepted pending revision). A framework phylogeny of the American oak clade based on sequenced RAD data. PLoS ONE.

See Also

[rank.partitionedRAD](#), [plot.locus.dist](#)

`print.pyRAD.loci` *Print method for pyRAD.loci objects*

Description

Prints very basic summary data for a `pyRAD.loci` object rather than streaming the entire thing, which can be huge and not so useful to read on the screen

Usage

```
## S3 method for class 'pyRAD.loci'  
print(x, ...)
```

Arguments

<code>x</code>	an object of class <code>pyRAD.loci</code>
<code>...</code>	unused

Value

No value. Called for the side-effect of printing attractively to the screen

Author(s)

Andrew Hipp

See Also

[read.pyRAD](#)

`rad2mat` *Generates a matrix of sequences from a pyRAD.loci object*

Description

This function exports a matrix of sequences from a `pyRAD.loci` object, with aligned sequences in the cells, individuals as rows, and loci as columns. As a matrix, this is very easy to work with in R for subsetting and creating new datasets.

Usage

```
rad2mat(pyDat, fill.N = TRUE)
```


Arguments

pyDat an object of class pyRAD.loci
fill.N indicates whether empty cells should be filled with Ns (TRUE) or not

Details

In general, filling with Ns is a handy thing to do, even though it presumably eats up a bunch of memory. As written, RADami is not memory-friendly.

Value

A matrix of sequences from a pyRAD.loci object, with aligned sequences in the cells, individuals as rows, and loci as columns

Author(s)

Andrew Hipp

See Also

[rad2phy](#), [summary.pyRAD.loci](#)

Examples

```
data(oak.rads)
out <- rad2mat(oak.rads)
```

rad2phy

Export a rad.mat object to a phylip file

Description

This function takes output from rad2mat and writes it to a phylip file, ready for analysis by RAxML or other software

Usage

```
rad2phy(pyDat, inds = row.names(pyDat), loci = dimnames(pyDat)[[2]],
        outfile = "pyMat.out.phy", padding = 50, verbose = FALSE,
        logfile = "rad2phy.log")
```

Arguments

pyDat	an object of class rad.mat
inds	individuals to include in the export
loci	loci to include in the export
outfile	file name for export
padding	total number of characters, including sequence title, to place before each sequence
verbose	whether to report on loci being written
logfile	name of logfile, which holds the names of loci in the phylip file written

Value

No value. This function is called for the side-effect of writing a phylip file

Author(s)

Andrew Hipp

See Also

[rad2mat](#), [write.DNAStringSet](#)

rank.partitionedRAD *Bin trees into 'supported' or 'disfavored' by locus.*

Description

This function takes output from `match.lnL.to.trees` and scores trees as favored or disfavored for all loci that satisfy the criteria specified

Usage

```
rank.partitionedRAD(radMat, minTrees = 10,
                    min.overall.diff.lnL = 5,
                    threshold.lnL = 2,
                    discardDoubleCounts = TRUE)
```

Arguments

radMat	matrix output from <code>match.lnL.to.trees</code>
minTrees	integer, minimum number of unique trees required per locus
min.overall.diff.lnL	minimum log-likelihood difference required between the most poorly-supported tree and the best supported tree for each locus
threshold.lnL	the log-likelihood window for binning a tree into favored or disfavored

discardDoubleCounts

determines whether that place any tree within both the favored and disfavored bins should be counted (TRUE) or not (FALSE)

Details

This function works by first filtering loci by the minimum number of trees and the overall range in log-likelihood, then binning trees into supported if they are within threshold.lnL of the best-supported tree for each locus, and disfavored if they are within threshold.lnL of the least supported tree for each locus. Overlapping trees can occur when the range in log-likelihood is set to less than twice the threshold log-likelihood or quite near it.

Value

A list composed of 7 matrices and one numeric vector. The matrices all have loci as the rows and trees as the columns:

bestMat	boolean; which trees are supported by which loci
worstMat	boolean; which trees are disfavored by which loci
doubleCountMat	boolean; which trees are double-counted by which loci
radMat	the final likelihood matrix corresponding with bestMat and worstMat
radMat.preLnLDiff	the likelihood matrix before filtering by lnL range
radMat.preMinTrees	the likelihood matrix before filtering by minimum number of trees
radMat.preDoubleCounts	the likelihood matrix before filtering by double counts
params	the analysis parameters, in this order: minTrees, min.overall.diff.lnL, threshold.lnL, discardDoubleCounts

Author(s)

Andrew Hipp

References

Hipp A.L., Eaton D.A.R., Cavender-Bares J., Fitzek E., Nipper R. & Manos P.S. (Accepted pending revision). A framework phylogeny of the American oak clade based on sequenced RAD data. PLoS ONE.

See Also

[match.lnL.to.trees](#), [plot.rankedPartitionedRAD](#)

read.pyRAD	<i>Read in data from pyRAD</i>
------------	--------------------------------

Description

This function begins essentially all analyses in this package. It reads in the pyRAD loci file and stores it in a form to facilitate a variety of downstream analyses coded in this package.

Usage

```
read.pyRAD(filename, reportInterval = 20000,
           breakLinesSeparate = FALSE,
           doSummary = TRUE, ...)
```

Arguments

filename	name of the pyRAD loci file
reportInterval	how often to report progress while reading and processing files
breakLinesSeparate	are there separate break lines (blank) and summary lines (not blank) between loci in the pyRAD file?
doSummary	should summary.pyRAD.loci be run on the object and appended to the end?
...	extra arguments to be passed along to readLines

Details

As written, this function has a lot of summary embedded in it. The default doSummary = TRUE should be left for the other higher-level functions to work properly

Value

A list with the following items:

tips	a character vector of the tips for each sequence
seqs	a character vector of all sequences
breaks	an integer vector indexing the lines that are breaks between loci
break.vectors	a named character vector of the break lines between vectors, with names corresponding to locus names
cons	a named character vector of the lines summarizing each locus, with names corresponding to locus names; if breakLinesSeparate = F, this is identical to break.vectors
locus.index	an index corresponding to seqs and tips, indicating what locus each belongs to
file.read	name of the file read in
timestamp	timestamp for exiting the function
radSummary	an object of class summary.pyRAD.loci, if called, which it is by default

Author(s)

Andrew Hipp

See Also

[summary.pyRAD.loci](#)

rtreePhylo

Generate random trees using an input tree to get taxon names and tips

Description

A wrapper around rtree

Usage

```
rtreePhylo(x, N = 1, ...)
```

Arguments

x	an object of class phylo
N	number of trees to return
...	extra arguments to pass along to rtree

Value

an object of class multiPhylo

Author(s)

Andrew Hipp

See Also

[genTrees](#)

Examples

```
data(oak.tree)
a <- rtreePhylo(oak.tree, 4)
layout(matrix(1:4,2,2))
lapply(a, plot)
```

subset.pyRAD.loci *Subsets a pyRAD.loci object*

Description

This function returns a dataset (at this point, only in DNASTringSet format) from a pyRAD.loci object, subsetting by loci and taxa provided as arguments

Usage

```
## S3 method for class 'pyRAD.loci'
subset(x, loci = colnames(x$radSummary$inds.mat),
       taxa = row.names(x$radSummary$inds.mat),
       format = "DNASTringSet", reportInterval = 500, mins = 1,
       nucVarType = c("verystrict", "strict", "relaxed"),
       use.tidyName = FALSE,
       snpsOnly = FALSE, cores = 1, ...)
```

Arguments

x	an object of class pyRAD.loci
loci	a character vector of loci to include; defaults to all loci
taxa	a character vector of taxa to include; defaults to all taxa
format	format for DNA matrix; at this point, only DNASTringSet is supported
reportInterval	interval (number of loci) for reporting progress
mins	minimum number of individuals required to return locus data
nucVarType	nucleotide variability to consider in screening for polymorphism: 'relaxed' for all nucleotides, including ambiguities other than 'N', and counting all variable sites; 'strict' for only A,C,G,T, still counting all variable sites; and 'verystrict' for only A,C,G,T, countin only parsimony- informative sites.
use.tidyName	choose whether names are passed through tidyName
snpsOnly	choose whether you only want to get SNPs back, or whole locus sequences
cores	how many processing cores to use; if cores = 1, lapply is used throughout instead of mclapply
...	additional arguments passed to tidyName

Details

Probably it would make sense to wrap some of the functioning of gen.RAD.loci.datasets into this function, but for now they are separated.

Value

A list with values:

DNA	a list of datasets, one per locus, defined by format argument
variable	a boolean vector, indicating which loci are variable according to the criterion provided in arguments
ntaxa	an integer vector, indicating how many taxa are in each locus

Author(s)

Andrew Hipp

See Also

[gen.RAD.loci.datasets](#), [filter.by](#)

summary.pyRAD.loci *Summary method for pyRAD.loci objects*

Description

This function provides the summary stats used by many functions in RADami

Usage

```
## S3 method for class 'pyRAD.loci'
summary(object, ...)
```

Arguments

object	an object of class pyRAD.loci
...	not currently used, but maintained for consistency with default summary method

Details

This is currently called by default by read.pyRAD, creating a list object called radSummary in value for that object.

Value

A list with of the following objects:

num.loci	an integer, indicating how many loci there are in the object
tips.per.locus	a list of character vectors, enumerating the tips in each locus
break.vectors	a named character vector of the break lines between vectors, with names corresponding to locus names – completely redundant with the same object in the pyRAD.loci object

<code>seqs.per.locus</code>	a list of named character vectors, enumerating the sequences in each locus, named by tips
<code>num.inds.per.locus</code>	an integer vector, indicating how many individuals are in each locus
<code>variable.loci</code>	a boolean vector, indicating which loci are variable (only using the pyRAD break vectors)
<code>inds.mat</code>	a handy boolean matrix with individuals as the rows and loci as the columns; used in several places in RADami
<code>locus.lengths</code>	an integer vector, indicating the length of each locus

Note

Probably these functions could be teased apart to be more useful and save space, but for now they are left as is. At this point, there is actually a fair bit of redundancy.

Author(s)

Andrew Hipp

See Also

[read.pyRAD](#), [print.pyRAD.loci](#)

tidyName

Clean up a name

Description

Deletes in a character vector all the characters provided in `fixes`.

Usage

```
tidyName(x, fixes = c("_", ".", " ", "-", ">"),
        case = c("lower", "upper", "nochange"))
```

Arguments

<code>x</code>	the vector in which to make replacements
<code>fixes</code>	all characters to be replaced
<code>case</code>	change case to upper, lower, or make no changes

Details

This function currently uses no regular expressions or wildcards and is made deliberately very thorough, tedious even, as it cycles through all characters in turn.

Value

A character vector of the same length as `x`, but with characters replaced

Note

I have used this function in many projects, and duplicates of it linger in a variety of places.

Author(s)

Andrew Hipp

Examples

```
nuttyName <- "Jimmy-Smith"  
notNuttyName <- tidyName("Jimmy-Smith")
```

`write.DNAStringSet` *Writes a sequence matrix to phylip format*

Description

This function writes a Biostrings DNAStringSet object to a file. Currently, phylip and fasta formats are supported

Usage

```
write.DNAStringSet(x, format= c('phylip', 'fasta'), padding = 30,  
                  filename = "DNAStringSetOut.phy",  
                  fastaPrefix = ">")
```

Arguments

<code>x</code>	an object of class DNAStringSet
<code>format</code>	at this point, only fasta and phylip formats are supported
<code>padding</code>	total number of characters, including sequence title, to place before each sequence
<code>filename</code>	full name of file and path
<code>fastaPrefix</code>	prefix character to add to taxon label for fasta files

Value

At this point, returns a 0 to indicate successful completion

Author(s)

Andrew Hipp

See Also[rad2phy](#)

Index

*Topic **IO**

get.raxml.treeLikelihoods, 11
rad2phy, 25
read.pyRAD, 28

*Topic **datagen**

gen.RAD.loci.datasets, 6

*Topic **datasets**

oak.rads, 15
oak.tree, 16

*Topic **hplot**

plot.locus.dist, 18
plot.partitionedRAD, 20
plot.rankedPartitionedRAD, 22

*Topic **manip**

consensus.pyRAD, 4
filter.by, 5
gen.RAD.loci.datasets, 6
get.raxml.siteLikelihoods, 10
match.lnL.to.trees, 14

*Topic **package**

RADami-package, 2

compare.all.trees, 3, 10

consensus.pyRAD, 4

filter.by, 5, 31

gen.RAD.loci.datasets, 6, 10, 15, 31

genotypes.pyRAD.loci, 7

genTrees, 7, 9, 29

get.raxml.siteLikelihoods, 10, 11

get.raxml.treeLikelihoods, 7, 10, 11

grab.pyRAD.locus, 11

help, 17

lengths.report, 12

locus.dist, 13, 19

match.lnL.to.trees, 7, 10, 11, 14, 27

oak.rads, 15, 16

oak.tree, 16

overlap.report, 16

pairwise.fst.loci, 17

plot.locus.dist, 14, 18, 21, 23

plot.partitionedRAD, 20

plot.rankedPartitionedRAD, 15, 19, 22, 27

print.pyRAD.loci, 24, 32

rad2mat, 24, 26

rad2phy, 25, 25, 34

RADami (RADami-package), 2

RADami-package, 2

rank.partitionedRAD, 7, 15, 21, 23, 26

read.pyRAD, 13, 24, 28, 32

rTreePhylo, 29

subset.pyRAD.loci, 6, 9, 18, 30

summary.pyRAD.loci, 13, 25, 29, 31

tidyName, 32

write.DNAStringSet, 26, 33