Package ‘phyloscannerR’

October 17, 2017

Title Phylogenetics between and within hosts at once, all along the genome

Version 1.3.0

Description An R package for the second half of phyloscanner (tree analysis).

Depends R (>= 3.4.1)

Imports ape, data.table, dplyr, dtplyr, ff, GGally, ggplot2, ggtree, grid, gridExtra, gtable, kimisc, network, pegas, phangorn, phytools, prodlim, RColorBrewer, reshape, reshape2, scales ,sna

License GPL

Encoding UTF-8

LazyData true

RoxygenNote 6.0.1

R topics documented:

`phyloscanner.analyse.trees`

Perform a phyloscanner analysis on a set of trees

Description

This function performs a parsimony reconstruction and classification of pairwise host relationships.

Usage

```
phyloscanner.analyse.trees(tree.directory,
  tree.file.regex = "RAxML_bestTree.InWindow_([0-9]+_to_[0-9]+)\\.tree",
  splits.rule = c("s", "r", "f"), sankoff.k = 0,
  sankoff.unsampled.switch.threshold = 0,
  continuation.unsampled.proximity.cost = Inf, outgroup.name = NULL,
  multifurcation.threshold = 0, guess.multifurcation.threshold = F,
  user.blacklist.directory = NULL, user.blacklist.file.regex = NULL,
  duplicate.file.directory = NULL, duplicate.file.regex = NULL,
  recombination.file.directory = NULL,
 recombination.file.regex = "RecombinantReads_InWindow_([0-9]+_to_[0-9]+).csv",
  tip.regex = "^(.*)_read_([0-9]+)_count_([0-9]+)$",
  file.name.regex = "^\\D*([0-9]+)_to_([0-9]+)\\D*$",
  seed = sample(1:1e+07, 1), norm.ref.file.name = NULL,
  norm.standardise.gag.pol = F, norm.constants = NULL,
  parsimony.blacklist.k = 0, raw.blacklist.threshold = 0,
  ratio.blacklist.threshold = 0, do.dual.blacklisting = F,
  max.reads.per.host = Inf, blacklist.underrepresented = F, use.ff = F,
  prune.blacklist = F, read.counts.matter.on.zero.length.tips = F,
  verbose = F)
```

Arguments

`tree.directory` The directory containing all input trees.

`tree.file.regex`
A regular expression identifying every file in `tree.directory` that is to be included in the analysis. The first capture group, if present, gives a unique string identifying each tree. If this is NULL then `phyloscanner` will attempt to open every file in `tree.directory`.

`splits.rule` The rules by which the sets of hosts are split into groups in order to ensure that all groups can be members of connected subgraphs without causing conflicts. Options: s=Sankoff with optional within-host diversity penalty (slow, rigorous, recommended), r=Romero-Severson (quick, less rigorous with >2 hosts), f=Sankoff with continuation costs (experimental).

`sankoff.k` For `splits.rule = s` or `f` only. The *k* parameter in the Sankoff reconstruction, representing the within-host diversity penalty.

`sankoff.unsampled.switch.threshold`
For `splits.rule = s` only. Threshold at which a lineage reconstructed as infecting a host will transition to the unsampled state, if it would be equally parsimonious to remain in that host.

`continuation.unsampled.proximity.cost`
For `splits.rule = f` only. The branch length at which an node is reconstructed as unsampled if all its neighbouring nodes are a greater distance away. If infinite (the default), such a node will never receive the unsampled state.

`outgroup.name` The name of the tip in the phylogeny/phylogenies to be used as outgroup (if unspecified, trees will be assumed to be already rooted). This should be sufficiently distant to any sequence obtained from a host that it can be assumed that the MRCA of the entire tree was not a lineage present in any sampled individual.

`multifurcation.threshold`
If specified, branches shorter than this in the input tree will be collapsed to form multifurcating internal nodes. This is recommended; many phylogenetics packages output binary trees with short or zero-length branches indicating multifurcations.

`guess.multifurcation.threshold`
Whether to guess the multifurcation threshold from the branch lengths of the trees and the width of the genomic window (if that information is available). It is recommended that trees are examined by eye to check that they do appear to have multifurcations if using this option.

`user.blacklist.directory`
An optional path for a folder containing pre-existing blacklist files. These tips are specified by the user to be excluded from the analysis.

`user.blacklist.file.regex`

A regular expression identifying every file in `blacklist.directory` that contains a blacklist. If a capture group is specified then its contents will uniquely identify the tree it belongs to, which must matches the IDs found by `user.blacklist.regex`. If these IDs cannot be identified then matching will be attempted using genome window coordinates.

`duplicate.file.directory`

An optional path for a folder containing information on duplicate reads, to be used for duplicate blacklisting. Normally this is produced by `phyloscanner_make_trees.py`.

`duplicate.file.regex`

A regular expression identifying every file in `duplicate.file.directory` that contains a duplicates file. If a capture group is specified then its contents will uniquely identify the tree it belongs to, which must matches the IDs found by `duplicate.file.regex`. If these IDs cannot be identified then matching will be attempted using genome window coordinates.

`recombination.file.directory`

An optional path for a folder containing results of the `phyloscanner_make_trees.py` recombination metric analysis.

`recombination.file.regex`

A regular expression identifying every file in `recombination.file.directory` that contains a recombination file. If a capture group is specified then its contents will uniquely identify the tree it belongs to, which must matches the IDs found by `recombintion.file.regex`. If these IDs cannot be identified then matching will be attempted using genome window coordinates.

`tip.regex` Regular expression identifying tips from the dataset. This expects up to three capture groups, for host ID, read ID, and read count (in that order). If the latter two groups are missing then read information will not be used. The default matches input from the phyloscanner pipeline where the host ID is the BAM file name.

`file.name.regex`

Regular expression identifying window coordinates. Two capture groups: start and end; if the latter is missing then the first group is a single numerical identifier for the window. The default matches input from the phyloscanner pipeline.

`seed` Random number seed; used by the downsampling process, and also ties in some parsimony reconstructions can be broken randomly.

`norm.ref.file.name`

Name of a file giving a normalisation constant for every genome position. Cannot be used simultaneously with `norm.constants`. If neither is given then no normalisation will be performed.

`norm.standardise.gag.pol`

Use only if `norm.ref.file.name` is given. An HIV-specific option: if true, the normalising constants are standardised so that the average on gag+pol equals 1. Otherwise they are standardised so the average on the whole genome equals 1.

`norm.constants` Either the path of a CSV file listing the file name for each tree (column 1) and the respective normalisation constant (column 2) or a single numerical normalisation constant to be applied to every tree. Cannot be used simultaneously with `norm.ref.file.name`. If neither is given then no normalisation will be performed.

`parsimony.blacklist.k`

The *k* parameter of the single-host Sankhoff parsimony reconstruction used to identify probable contaminants. A value of 0 is equivalent to not performing parsimony blacklisting.

`raw.blacklist.threshold`
: Used to specify a read count to be used as a raw threshold for duplicate or parsimony blacklisting. Use with `parsimony.blacklist.k` or `duplicate.file.regex` or both. Parsimony blacklisting will blacklist any subgraph with a read count strictly less than this threshold. Duplicate blacklisting will black list any duplicate read with a count strictly less than this threshold. The default value of 0 means nothing is blacklisted.

`ratio.blacklist.threshold`
: Used to specify a read count ratio (between 0 and 1) to be used as a threshold for duplicate or parsimony blacklisting. Use with `parsimony.blacklist.k` or `duplicate.file.regex` or both. Parsimony blacklisting will blacklist a subgraph if the ratio of its read count to the total read count from the same host is strictly less than this threshold. Duplcate blacklisting will blacklist a duplicate read if the ratio of its count to the count of the duplicate (from another host) is strictly less than this threshold.

`do.dual.blacklisting`
: Blacklist all reads from the minor subgraphs for all hosts established as dual by parsimony blacklisting (which must have been done for this to do anything).

`max.reads.per.host`
: Used to turn on downsampling. If given, reads will be blacklisted such that read counts (or tip counts if no read counts are identified) from each host are equal (although see `blacklist.underrepresented`.

`blacklist.underrepresented`
: If TRUE and `max.reads.per.host` is given, blacklist hosts from trees where their total tip count does not reach the maximum.

`use.ff`
: Use the `ff` package to store parsimony reconstruction matrices. Use if you run out of memory.

`prune.blacklist`
: If TRUE, all blacklisted and reference tips (except the outgroup) are pruned away before starting parsimony-based reconstruction.

`read.counts.matter.on.zero.length.tips`
: If TRUE, read counts on tips will be taken into account in parsimony reconstructions at the parents of zero-length terminal branches. Not applicable for the Romero-Severson-like reconstruction method.

`verbose`
: Give verbose output.

Value

A list of class `phyloscanner.trees`.

Index