Package 'HiMC'

June 26, 2018

Version 0.1.1.5

Title High-Throughput Mitochondrial Haplogroup Classification

Description Assign high-level mitochondrial haplogroups given SNP information from standard PLINK *.map* and *.ped* files. A reduced phylotree[1] is used as a classification tree to determine samples' haplogroup assignment. Please see the included package vignette ``Hi-MC: Overview Guide'' for a detailed algorithm description and usage examples. [1] Mitchell SL, Goodloe R, Brown-Gentry K, Pendergrass SA, Murdock DG, Crawford DC (2014) <doi:10.1007/s00439-014-1421-9>.

Depends R (>= 3.1.2)

Imports stringr, methods

VignetteBuilder knitr

Suggests knitr

License GPL-3

LazyData false

RoxygenNote 6.0.1

NeedsCompilation no

Author Eric Farber-Eger [aut, cre],
Dana Crawford [aut],
Nicholas R. Wheeler [aut] (0000-0003-2248-8919),
Sandra Smieszek [aut]

Maintainer Eric Farber-Eger <eric.h.farber-eger@vumc.org>

R topics documented:

generate_snp_data *SNP Data Generator*

Description

Takes in a plink map file and a plink ped file and creates a dataframe with headers

Usage

```
generate_snp_data(map_file, ped_file)
```

Arguments

map_file	The name of the .map file. Should be tab delimited with no header.
ped_file	The name of the .ped file. Should be space delimited with no header.

Examples

```
mapfile <- system.file("extdata","HapMap_Mito_Seq_QC_2.map",package="HiMC")
pedfile <- system.file("extdata","HapMap_Mito_Seq_QC_2.ped",package="HiMC")
generate_snp_data(mapfile,pedfile)
```

getAllPaths *Path Generator*

Description

Internal function. Takes a SNP dataframe, a node, and current path, and returns all available paths.

Usage

```
getAllPaths(df, node, path)
```

Arguments

df	SNP dataframe
node	Node to be checked
path	Current path checked

Examples

```
#internal function, not meant to be called externally
mapfile <- system.file("extdata","HapMap_Mito_Seq_QC_2.map",package="HiMC")
pedfile <- system.file("extdata","HapMap_Mito_Seq_QC_2.ped",package="HiMC")
snpfile <- generate_snp_data(mapfile,pedfile)
full_path_list <- getAllPaths(snpfile,HiMC::root,"")
```

getClassifications *getClassifications output generator*

Description

Takes in a dataframe generated by "generate_snp_data" and returns each subject classified with full classification paths

Usage

```
getClassifications(source_df)
```

Arguments

source_df	The snp_data generated dataframe

Examples

```
mapfile <- system.file("extdata","HapMap_Mito_Seq_QC_2.map",package="HiMC")
pedfile <- system.file("extdata","HapMap_Mito_Seq_QC_2.ped",package="HiMC")
snpfile <- generate_snp_data(mapfile,pedfile)
classifications <- getClassifications(snpfile)
```

getFinalPathList *getFinalPathList internal function*

Description

Takes in a nested list of prettified paths and returns those of the greatest length

Usage

```
getFinalPathList(plist)
```

Arguments

plist	A nested list of prettified paths

Examples

```
#internal function, not meant to be called externally
filtered_paths <- getFinalPathList(c("A -> B","A -> B -> C"))
```

getGroupFromPath *getFinalPathList internal function*

Description

Takes in a list of paths and returns the final classification of that path

Usage

```
getGroupFromPath(string)
```

Arguments

string	The path in question

Examples

```
#internal function, not meant to be called externally
group <- getGroupFromPath("A -> B -> C")
```

getPathList *getPathList internal function*

Description

Takes in a nested list of paths in dataframe format and returns a top-level path assignment

Usage

```
getPathList(df)
```

Arguments

df	The FASTMAP dataframe row

Examples

```
#internal function, not meant to be called externally
paths <- getPathList(data.frame())
```

maybeNode-class *A class to represent a maybe-node.*

Description

Union between classes NULL and node

missingSnps *validData boolean check*

Description

Internal function. Takes in a dataframe and a node and returns true if the dataframe row is missing any SNPs for the node in question

Usage

```
missingSnps(df, node)
```

Arguments

df	The FASTMAP dataframe row
node	The node in question

Examples

```
#internal function, not meant to be called externally
is_dataframe_row_missing_snps_from_node <- missingSnps(data.frame(),HiMC::root)
```

node-class *A class to represent a node.*

Description

A class to represent a node.

Slots

`name` A character vector

`snps` A list

`req` A list

`children` A list

node_a *A node node_a.*

Description

An instance of the node class.

Usage

```
node_a
```

Format

An object of class `node` of length 1.

`node_a2` *A node node_a2.*

Description

An instance of the node class.

Usage

```
node_a2
```

Format

An object of class `node` of length 1.

`node_b2` *A node node_b2.*

Description

An instance of the node class.

Usage

```
node_b2
```

Format

An object of class `node` of length 1.

`node_b4bde` *A node node_b4bde.*

Description

An instance of the node class.

Usage

```
node_b4bde
```

Format

An object of class `node` of length 1.

`node_b5` *A node node_b5.*

Description

An instance of the node class.

Usage

```
node_b5
```

Format

An object of class `node` of length 1.

`node_c` *A node node_c.*

Description

An instance of the node class.

Usage

```
node_c
```

Format

An object of class `node` of length 1.

`node_d` *A node node_d.*

Description

An instance of the node class.

Usage

```
node_d
```

Format

An object of class `node` of length 1.

`node_d1` *A node node_d1.*

Description

An instance of the node class.

Usage

```
node_d1
```

Format

An object of class `node` of length 1.

`node_d2` *A node node_d2.*

Description

An instance of the node class.

Usage

```
node_d2
```

Format

An object of class `node` of length 1.

`node_d4` *A node node_d4.*

Description

An instance of the node class.

Usage

```
node_d4
```

Format

An object of class `node` of length 1.

`node_h` *A node node_h.*

Description

An instance of the node class.

Usage

```
node_h
```

Format

An object of class `node` of length 1.

`node_h2` *A node node_h2.*

Description

An instance of the node class.

Usage

```
node_h2
```

Format

An object of class `node` of length 1.

`node_h2a` *A node node_h2a.*

Description

An instance of the node class.

Usage

```
node_h2a
```

Format

An object of class `node` of length 1.

`node_h2a2a` *A node node_h2a2a.*

Description

An instance of the node class.

Usage

```
node_h2a2a
```

Format

An object of class `node` of length 1.

`node_hv` *A node node_hv.*

Description

An instance of the node class.

Usage

```
node_hv
```

Format

An object of class `node` of length 1.

`node_i` *A node node_i.*

Description

An instance of the node class.

Usage

```
node_i
```

Format

An object of class `node` of length 1.

`node_j` *A node node_j.*

Description

An instance of the node class.

Usage

```
node_j
```

Format

An object of class `node` of length 1.

`node_jt` *A node node_jt.*

Description

An instance of the node class.

Usage

```
node_jt
```

Format

An object of class `node` of length 1.

`node_jt_alt` *A node node_jt_alt.*

Description

An instance of the node class.

Usage

```
node_jt_alt
```

Format

An object of class `node` of length 1.

`node_k` *A node node_k.*

Description

An instance of the node class.

Usage

```
node_k
```

Format

An object of class `node` of length 1.

`node_k1` *A node node_k1.*

Description

An instance of the node class.

Usage

```
node_k1
```

Format

An object of class `node` of length 1.

`node_k_alt` *A node node_k_alt.*

Description

An instance of the node class.

Usage

```
node_k_alt
```

Format

An object of class `node` of length 1.

node_l0 *A node node_l0.*

Description

An instance of the node class.

Usage

```
node_l0
```

Format

An object of class `node` of length 1.

node_l1 *A node node_l1.*

Description

An instance of the node class.

Usage

```
node_l1
```

Format

An object of class `node` of length 1.

node_l1b *A node node_l1b.*

Description

An instance of the node class.

Usage

```
node_l1b
```

Format

An object of class `node` of length 1.

`node_l2` *A node node_l2.*

Description

An instance of the node class.

Usage

```
node_l2
```

Format

An object of class `node` of length 1.

`node_l23456` *A node node_l23456.*

Description

An instance of the node class.

Usage

```
node_l23456
```

Format

An object of class `node` of length 1.

`node_l2346` *A node node_l2346.*

Description

An instance of the node class.

Usage

```
node_l2346
```

Format

An object of class `node` of length 1.

`node_l3` *A node node_l3.*

Description

An instance of the node class.

Usage

```
node_l3
```

Format

An object of class `node` of length 1.

`node_l134` *A node node_l34.*

Description

An instance of the node class.

Usage

```
node_l134
```

Format

An object of class `node` of length 1.

`node_m` *A node node_m.*

Description

An instance of the node class.

Usage

```
node_m
```

Format

An object of class `node` of length 1.

node_n *A node node_n.*

Description

An instance of the node class.

Usage

```
node_n
```

Format

An object of class `node` of length 1.

node_n1a1 *A node node_n1a1.*

Description

An instance of the node class.

Usage

```
node_n1a1
```

Format

An object of class `node` of length 1.

node_n1a1b *A node node_n1a1b.*

Description

An instance of the node class.

Usage

```
node_n1a1b
```

Format

An object of class `node` of length 1.

`node_n_alt` *A node node_n_alt.*

Description

An instance of the node class.

Usage

```
node_n_alt
```

Format

An object of class `node` of length 1.

`node_r` *A node node_r.*

Description

An instance of the node class.

Usage

```
node_r
```

Format

An object of class `node` of length 1.

`node_r_alt` *A node node_r_alt.*

Description

An instance of the node class.

Usage

```
node_r_alt
```

Format

An object of class `node` of length 1.

`node_t` *A node node_t.*

Description

An instance of the node class.

Usage

```
node_t
```

Format

An object of class `node` of length 1.

`node_t1` *A node node_t1.*

Description

An instance of the node class.

Usage

```
node_t1
```

Format

An object of class `node` of length 1.

`node_u` *A node node_u.*

Description

An instance of the node class.

Usage

```
node_u
```

Format

An object of class `node` of length 1.

`node_u8b` *A node node_u8b.*

Description

An instance of the node class.

Usage

```
node_u8b
```

Format

An object of class `node` of length 1.

`node_u8b_alt` *A node node_u8b_alt.*

Description

An instance of the node class.

Usage

```
node_u8b_alt
```

Format

An object of class `node` of length 1.

`node_u_alt` *A node node_u_alt.*

Description

An instance of the node class.

Usage

```
node_u_alt
```

Format

An object of class `node` of length 1.

`node_v` *A node node_v.*

Description

An instance of the node class.

Usage

```
node_v
```

Format

An object of class `node` of length 1.

`node_w` *A node node_w.*

Description

An instance of the node class.

Usage

```
node_w
```

Format

An object of class `node` of length 1.

`node_x` *A node node_x.*

Description

An instance of the node class.

Usage

```
node_x
```

Format

An object of class `node` of length 1.

node_x2 *A node node_x2.*

Description

An instance of the node class.

Usage

```
node_x2
```

Format

An object of class `node` of length 1.

numChildren *NumChildren*

Description

Internal function. Takes in a node object and returns the total number of that node's children

Usage

```
numChildren(node_object)
```

Arguments

`node_object`	The FASTMAP node in question

Examples

```
#internal function, not meant to be called externally
node_tail_length = numChildren(HiMC::root)
```

numReqs *NumReqs*

Description

Internal function. Takes in a node object and returns the number of SNPs that it requires for validation

Usage

```
numReqs(node_object)
```

Arguments

node_object	The FASTMAP node in question

Examples

```
#internal function, not meant to be called externally
number_of_required_snps <- numReqs(HiMC::root)
```

---

numSnps *NumSnps*

Description

Internal function. Takes in a node object and returns the number of SNPs that belong to it

Usage

```
numSnps(node_object)
```

Arguments

node_object	The FASTMAP node in question

Examples

```
#internal function, not meant to be called externally
amount_of_snps_in_node <- numSnps(HiMC::root)
```

---

root *A node root.*

Description

An instance of the node class.

Usage

```
root
```

Format

An object of class node of length 1.

validData *validData boolean check*

Description

Internal function. Takes in a dataframe and a node and returns true if the dataframe row has the required SNPs for the node

Usage

```
validData(df, node)
```

Arguments

df	The FASTMAP dataframe row
node	The node in question

Examples

```
#internal function, not meant to be called externally
df_row_meets_criteria_for_node_requirements <- validData(data.frame(),HiMC::root)
```

validPath *validPath boolean check*

Description

Internal function. Takes in a dataframe and a node and returns true if the dataframe row represents a valid path for the node

Usage

```
validPath(df, node)
```

Arguments

df	The FASTMAP dataframe row
node	The node in question

Examples

```
#internal function, not meant to be called externally
df_row_meets_criteria_for_node_path <- validPath(data.frame(),HiMC::root)
```

Index