Package 'Reol'

July 2, 2014

Version 1.55

Date 2013-04-18

Title R interface to the Encyclopedia of Life

Author Barb Banbury <darwinthesun@gmail.com>, Brian O'Meara <bomeara@utk.edu>

Maintainer Barb Banbury <darwinthesun@gmail.com>

BugReports http://r-forge.r-project.org/tracker/?group_id=1523

Depends R (>= 2.13.0)

Imports XML, RCurl, ape

Suggests geiger

LazyData yes

Description An R interface to the Encyclopedia of Life API. Includes functions for downloading and extracting information off the EOL pages.

License GPL (>= 2)

Repository CRAN

Repository/R-Forge/Project reol

Repository/R-Forge/Revision 178

Repository/R-Forge/DateTimeStamp 2014-04-18 19:45:57

Date/Publication 2014-04-19 08:13:16

NeedsCompilation no

R topics documented:

`Reol-package` *Reol*

Description

R interface to the Encyclopedia of Life

Details

Package:	Reol
Type:	Package
Version:	1.21
Date:	2013-9-9
License:	GPL (>= 2)

Author(s)

Barb Banbury, Brian O'Meara Maintainer: Barb Banbury <bbanbury@utk.edu>

APItaxon *Encode Taxon Name*

Description

This function will take a taxon name and encode spaces, and underscores to + for the EOL API.

Usage

```
APItaxon(taxon)
```

Arguments

taxon	Single taxonomic name

Value

Returns an encoded taxon name.

Examples

```
APItaxon("Ursus americanus")
```

CollectDataforWeb *Collect Data from EOL Pages for Website*

Description

These functions will read and scrape content off the downloaded EOL pages. This is for building the data table on eoldata.org.

Usage

```
CollectDataforWeb(MyEOL, do.higher.taxonomy)
CNCount(res)
DOCount(res)
providerCount(res)
DataProcessing(res, do.higher.taxonomy)
```

Arguments

MyEOL	A single filename for downloaded EOL pages
res	XML object
do.higher.taxonomy	TRUE/FALSE on whether to download higher level taxonomy

Value

Appends EOL data to a table.

Examples

```
data(MyEOLs)
CollectDataforWeb(MyEOLs[1])
```

CombineHierarchyInfo *Combine Hierarchy Info*

Description

This function combines different hierarchy dataframes into one. Mostly an internal function for tree and edge label building.

Usage

```
CombineHierarchyInfo(MyHiers)
```

Arguments

MyHiers	A vector of hier pages to combine

Value

Returns a combined data frame.

Examples

```
data(MyHiers)
CombineHierarchyInfo(MyHiers)
```

DataObjectOverview *Data Object Information*

Description

These functions gather information about the number of kinds of data objects. For example, it will tell you that there are 3 gifs 12 jpgs, and 34 text objects.

Usage

```
GatherDataObjectInformation(MyEOL)
CombineDataObjectInformation(MyEOLs, verbose=TRUE)
DataObjectOverview(MyEOLs, verbose=TRUE)
```

Arguments

MyEOL	A single EOL file or R object
MyEOLs	A vector of filenames or a list of XMLs for downloaded EOL pages
verbose	An optional print statement while combining

Value

GatherDataObjectInformation will return information from one single EOL file all of the data objects and their associated accession info. CombineDataObjectInformation will combine the data gathered singly into a huge matrix. Printing the results from either of these functions may hang up R or cause memory issues, so it isnt recommended (though you can pull of pieces of it). DataObjectOverview returns a data frame where each row is a single taxon and columns are a type of data object.

See Also

GetRichnessScores GetCommonNames GetIUCNStat GetReferences

Examples

```
data(MyEOLs)
DataObjectOverview(MyEOLs[1], verbose=FALSE)
## Not run: DataObjectOverview(MyEOLs, verbose=FALSE)

#Data from the first data object from the first EOL page on the list
## Not run: GatherDataObjectInformation(MyEOLs[1])[1,]

#See searchable variables for data objects using names and view contents
## Not run: names(GatherDataObjectInformation(MyEOLs[1]))
## Not run: GatherDataObjectInformation(MyEOLs[1])$source

#One way to get all the english text back from a single EOL file
## Not run:
```

```
  DOI <- GatherDataObjectInformation(MyEOLs[1])
  text <- which(which(DOI$mimeType == "text/html")
    "en"))
  DOI$description[text]

## End(Not run)
```

DownloadHierarchy *Download Hierarchy Content From Contributors*

Description

This function will use the EOL pages to find the provider information (ie which providers contribute information to the pages). Hierarchies get downloaded as hier pages rather than eol pages. Providers contribute information on taxonomic rankings, hierarchical tree structure, synonyms, etc. Some providers only contribute certain kinds of information; for example not all will provide synonyms or taxonomic hierarchy information.

Usage

```
ProviderCount(MyEOLs,verbose=FALSE)
BestProvider(MyEOLs)
DownloadHierarchy(MyEOLs, to.file=TRUE, database=NULL, verbose=TRUE, ...)
```

Arguments

MyEOLs	A vector of filenames for downloaded EOL pages
to.file	Whether to download data to a file
database	Provider hierarchy information to use
verbose	An optional print statement during download
...	further arguments to be passed to DownloadHierarchy

Value

ProviderCount will give a named vector back of all of the providers that contribute to the set of MyEOLs taxa, and the number of taxa they contribute. The maximum number of species a provider can contribute will be the number of taxa in MyEOLs. Usually, ITIS, GBIF, and NCBI are top contributors (but this varies depending on scale). BestProvider will choose the top provider from this list (even if there are ties). DownloadHierarchy will download the XML information either to a file (to.file=TRUE) or as a single R object as a list (to.file=FALSE).

See Also

DownloadEOLpages DownloadSearchedTaxa

Examples

```
data(MyEOLs)
## Not run: ProviderCount(MyEOLs[6], verbose=TRUE)
BestProvider(MyEOLs[6])
## End(Not run)
DownloadHierarchy(MyEOLs[6], FALSE, database="NCBI Taxonomy")

#Download data from whichever provider has the most coverage
## Not run: MyHiers <- DownloadHierarchy(MyEOLs, to.file=TRUE, BestProvider(MyEOLs))

#Or download from a specific provider
## Not run: MyHiers <- DownloadHierarchy(MyEOLs, FALSE, database="NCBI Taxonomy")
```

DownloadSearchedTaxa *Download Page Content From EOL*

Description

These functions will take a string of EOL IDs or taxonomic names and search EOL database for the pages. If it finds a match, it will download the EOL page.

Usage

```
DownloadEOLpages(pages, to.file=TRUE, MyKey=NULL, verbose=TRUE, ...)
DownloadSearchedTaxa(ListOfTaxa, to.file=TRUE, MyKey=NULL, exact=TRUE, verbose=TRUE, ...)
```

Arguments

pages	EOL page numbers to download
to.file	Whether to download data to a file
MyKey	An optional user identification key to identify yourself to EOL
verbose	An optional print statement during download
ListOfTaxa	List of EOL taxa to search and download EOL pages
exact	Should taxon name match exactly in EOL or fuzzy match
...	further arguments to be passed to DownloadEOLpages and DownloadSearchedTaxa

Details

DownloadEOLpages will download EOL pages based on the EOL unique identifyer number (EOL ID). Each taxon is associated with a unique identifier. These numbers are used to match EOL pages with hierarchy pages and keep track of taxonomic changes. If you are unsure of these numbers you can use DownloadSearchedTaxa, which will search for either an exact taxonomic match (exact=TRUE) or use fuzzy name matching to catch spelling errors (exact=FALSE). This will automatically recover the matching EOL ID and download or save the XML data accordingly.

To generate an api key (MyKey), register with EOL and find it under your profile.

Value

Either an XML file(s) downloaded to working directory or as an R object saved in the workspace.

See Also

DownloadHierarchy

Examples

```
#Download taxa files to working directory in R
## Not run: DownloadEOLpages(c(1,2,3), to.file=TRUE, MyKey)
## Not run: DownloadSearchedTaxa(c("Anolis_carolinensis",
"Anolis garmani"), to.file=TRUE, exact=TRUE)
## End(Not run)

#Save data as an R object rather than download files

MyTaxa <- c("Camelus dromedarius")
MyEOLs <- DownloadSearchedTaxa(MyTaxa, to.file=FALSE)
#save(MyEOLs, file="MyEOLs.rda")
```

FirstTwo *Get Binomial Nomenclature*

Description

This function strips a taxon name to the first two words. Many times, and EOL species name will come in with additional data (such as the describer, subspecies, variety, etc). This will prune it down to just two names.

Usage

```
FirstTwo(name)
```

Arguments

name	any taxonomic unit

Value

Returns a name

Examples

```
FirstTwo("Galanthus cilicicus Baker")
FirstTwo("Galanthus peshmenii A.P.Davis & C.D.Brickell")
FirstTwo("Galanthus")
```

GatherProviderDataFrame *Gather EOL Provider Information*

Description

This function will read in the downloaded xml files and amass a data frame with provider information. Information will include taxon name, eol number, and information from each provider in the following order: provider taxon, provider taxon ID, provider scientific name, provider taxonomy rank.

Usage

```
GatherProviderDataFrame(MyEOLs, extended.output=FALSE)
```

Arguments

MyEOLs	A vector of filenames or a list of XMLs for downloaded EOL pages
extended.output	If set to TRUE, then a large dataframe will be returned with information from each provider in the following order: provider taxon, provider taxon ID, provider scientific name, provider taxonomy rank.

Value

Returns a data frame with presence/absence of provider information or if extended.output=TRUE ID values for each provider.

Examples

```
data(MyEOLs)
## Not run: GatherProviderDataFrame(MyEOLs, extended.output=FALSE)

GatherProviderDataFrame(MyEOLs[1], extended.output=FALSE)
```

GatherSynonyms *Gathers A List Of Taxonomic Synonyms*

Description

This function goes through the hierarchy pages to collect taxonomic synonyms.

Usage

```
GatherSynonyms(MyHiers, output = c("detail", "counts"))
```

Arguments

MyHiers	A vector of filenames for downloaded hierarchy pages
output	Detail will return a data frame with each taxon and their taxonomic synonyms; counts will return a dataframe with each species and synonym counts.

Value

When output is set to detail, it returns a data frame where each row is the taxonomic unit that has a synonym, the hierarchy ID, and the synonym. When output="counts", each taxon is included whether it has a synonym or not. Rather than report the synonyms, it reports the number of synonyms for each.

Examples

```
data(MyHiers)
GatherSynonyms(MyHiers, output="detail")
GatherSynonyms(MyHiers, "c")
```

GetCommonNames *Gather Common Name Information*

Description

This function gathers each species common names and the language with which they are associated.

Usage

```
GetCommonNames(MyEOLs, output = c("detail", "counts"))
```

Arguments

MyEOLs	A vector of filenames or a list of XMLs for downloaded EOL pages
output	Detail will return a data frame with common names and their language; counts will return a dataframe with each taxon and language counts for common names.

Value

Returns a data frame with taxon, eol ID, common name, and language.

See Also

GetRichnessScores GetIUCNStat GetReferences DataObjectOverview

Examples

```
data(MyEOLs)

GetCommonNames(MyEOLs[1], "d")

## Not run: GetCommonNames(MyEOLs, output="detail")
## Not run: GetCommonNames(MyEOLs, "c")
```

GetHierID *Gets Hierarchy page ID*

Description

This function uses the name of the file to return numerical ID.

Usage

```
GetHierID(MyHier)
```

Arguments

MyHier	A single filename for downloaded hierarchy page

Value

Returns a hierarchical concept ID.

Examples

```
#Works with a single page
GetHierID("hier51323249.xml")

#Or works with a list or vector of names
data(MyHiers)
GetHierID(MyHiers)
```

GetIUCNStat *Gather IUCN Status for EOL Pages*

Description

This function gathers IUCN threat statuses from EOL pages if they are reported.

Usage

```
GetIUCNStat(MyEOLs)
```

Arguments

MyEOLs	A vector of filenames or a list of XMLs for downloaded EOL pages

Value

Returns a data frame with taxon, eol ID, and IUCN status.

See Also

GetRichnessScores GetCommonNames GetReferences DataObjectOverview

Examples

```
data(MyEOLs)
## Not run: GetIUCNStat(MyEOLs)
GetIUCNStat(MyEOLs[3])
```

GetReferences *Gather EOL References*

Description

This function gathers the references on the EOL pages (not the references on the provider pages).

Usage

```
GetReferences(MyEOLs, output = c("detail", "counts"))
```

Arguments

MyEOLs	A vector of filenames or a list of XMLs for downloaded EOL pages
output	Detail will return a data frame with eolID and reference; counts will return a dataframe with eol taxon name, eol ID, and number of references.

Value

Returns a data frame with taxon, eol ID, common name, and language.

See Also

GetRichnessScores GetCommonNames GetIUCNStat DataObjectOverview

Examples

```
data(MyEOLs)
## Not run: GetReferences(MyEOLs, output="detail")
GetReferences(MyEOLs[1], "c")
```

GetRichnessScores *Gather EOL Richness Score Information*

Description

EOL hosts richness scores for each species ranging from 0-100. These are calculated on how much information data is available for that species, based on the amount of text on a page, how many multimedia files are present, how many different topics are covered, how many sources contribute, and whether the information has been reviewed.

Usage

```
GetRichnessScores(MyEOLs)
```

Arguments

MyEOLs	A vector of filenames for downloaded EOL pages OR a list of XML data stored as an R object

Value

Returns a data frame with taxon, eol ID and numerical richness score.

See Also

GetCommonNames GetIUCNStat GetReferences DataObjectOverview

Examples

```
data(MyEOLs)
GetRichnessScores(MyEOLs[1:2])
```

MakeEdgeLabels *Creates Edge Labels for Hierarchy Trees*

Description

These functions will create edge labels for hierarchy trees.

Usage

```
getTipList(phy)
whichEdge(phy, taxa)
node.leaves(phy, node)
node.offspring(phy, node)
WhatToDoWithDuplicateEdgeNames(edgeLabels, duplicateEdgeLabels)
MakeEdgeLabels(MyHiers, label="all", missingData=NULL, duplicateEdgeLabels="oldest")
```

Arguments

phy	A tree in the class phylo
node	Any node number in the tree
MyHiers	A vector of hier pages OR a list of XMLs as an R object
taxa	Vector of tip taxa
edgeLabels	Vector of edge labels with the same associated branch
duplicateEdgeLabels	Choice of which edge label to prefer: recent, oldest, or combined
missingData	If tip taxa are not all the same taxonomic rank, should Reol cleave out taxa or hierarchical rank first
label	Which hierarchical units should be included in the edge labels

Details

Note that edges are slightly different than node labels, in that edges are plotted along the center of the branch rather than at a node. Plotting both is redundant, but one or the other may look better aesthetically. Also edges in the edgeLabels function is not the actual edge number, but the row. Our functions reflect this. There will likely be cases where edges have more than one hierarchical name (for example, the branch to camel species will have the genus Camelus and the family Camelidae). When making edge labels, you have the choice of using the most recent hierarchical name (genus in the camel example) or the oldest (family in camels), or you can choose to combine the names so that you can see all of them (Camilidae.Camelus).

Value

`MakeEdgeLabels` returns a vector of edges and their clade names to be used in apes edgeLabels function. `getTipList`, `WhatToDoWithDuplicateEdgeNames`, and `whichEdge` are internal functions for `MakeEdgeLabel`.

See Also

`MakeHierarchyTree`

Examples

```
library(ape)
data(MyHiers)
Tree <- MakeHierarchyTree(MyHiers, includeNodeLabels=FALSE)
edges <- MakeEdgeLabels(MyHiers)
plot(Tree, show.node.label=FALSE)
edgelabels(text=names(edges), edge=edges)


## Not run:
edges <- MakeEdgeLabels(MyHiers, missingData="pruneTaxa", duplicateEdgeLabels="recent")
plot(Tree, show.node.label=FALSE)
edgelabels(text=names(edges), edge=edges)

## End(Not run)
```

MakeHierarchyTree *Creates Hierarchical Trees*

Description

These functions will create a taxonomic tree (dendrogram) based on ranking from EOLs provider (hierarchy) pages.

Usage

```
MakeTreeData(MyHiers)
AutofillTaxonNames(TreeData)
MakeHierarchyTree(MyHiers, missingData=NULL, includeNodeLabels=TRUE, userRanks=NULL)
ReturnTaxSet(Taxon, TreeData)
NodeLabelList(MyHiers, label="all", missingData)
MergeTaxonomies(i, j)
```

Arguments

MyHiers	A vector of filenames or a list of XMLs for downloaded EOL pages
TreeData	A dataframe of taxonomic hierarchy information out of MakeTreeData function
missingData	If tip taxa are not all the same taxonomic rank, should Reol cleave out taxa or hierarchical rank first
includeNodeLabels	Option to write node labels to phylogenetic tree (Note, this can also be done separately using NodeLabelList
userRanks	Option for the user to define their own hierarchical pattern to make a tree. This will define which ranked classifications to include in the final tree. If left NULL, it will try to keep as much information as possible.
Taxon	Taxonomic group that contains subunits
label	Which hierarchical units should be included in the node labels
i	A Hierarchical Taxonomy (ex: Kingdom, Class, Species)
j	A Hierarchical Taxonomy (ex: Class, Genus, Species)

Details

This tree displays taxonomic structuring only and is not the result of a phylogenetic analysis. Alos note that not all providers return hierarchy information, if errors or no tree is returned it is likely that information is missing and you may have to use another provider.

Value

MakeTreeData returns a data frame with data for use in MakeHierarchyTree function, but can also be used independently to examine the hierarchical structure. AutofillTaxonNames is an internal function that deals with missing internal data in TreeData. If taxa are of varying hierarchical ranks (for example a mix of genera and species) then tips can not be aligned in the tree. You will need to select what data you would like to drop from the analysis; for example, either taxa with missing species information OR all species names that will then make a tree of genera. MakeHierarchyTree returns taxonomic tree in the class phylo. ReturnTaxSet will return the tree tips for the taxonomic group requested, this is mostly an internal function for creating node labels. MakeNodeLabels will return a list of tip labels per internal node. This can be used to create node labels to plot on the tree. MergeTaxonomies will merge two taxonomies into a single list that preserves hierarchical structure. It is mostly for internal use.

See Also

ProviderCount DownloadHierarchy MakeEdgeLabels

Examples

```
data(MyHiers)
TreeData <- MakeTreeData(MyHiers)
Tree <- MakeHierarchyTree(MyHiers, includeNodeLabels=TRUE)
labels <- NodeLabelList(MyHiers, "all")
plot(Tree, show.node.label=TRUE)

plot(Tree, "c", show.node.label=TRUE, adj=0.5, font=3, edge.color="gray",
tip.color=rainbow(10))
```

MatchHierPageToEOLdata

Match Data Functions

Description

These functions match hierarchy concept IDs with EOL page data and/or taxonomic tree tips.

Usage

```
MatchHierPageToEOLdata(MyHiers, EOLdata)
MatchDataToTreeTips(Tree, Data)
```

Arguments

MyHiers	A vector of filenames for downloaded hierarchy pages or a list of XML data stored as an R object
EOLdata	Any output dataframe from Reol functions
Tree	Taxonomic tree in the class phylo
Data	Data output from one of the EOL or Hierarchy gathering functions

Value

MatchHierPageToEOLdata returns a data frame where each row is a taxonomic unit. This is especially helpful if you wish to plot EOL data on a taxonomic tree (see example), because it matches the hierarchy taxon names with the EOL data (since taxon names will not always match). MatchDataToTreeTips will match the tip labels on a tree with those from a data frame. It orders the data in the same way as is in the trees, so that you can plot data on a tree without distortion (either in number of data points, or in taxonomic name).

Examples

```
#plotting richness scores
## Not run: data(MyEOLs)
 library(ape)
     data(MyHiers)
     Tree <- MakeHierarchyTree(MyHiers, includeNodeLabels=FALSE)
     MyData <- MatchHierPageToEOLdata(MyHiers, GetRichnessScores(MyEOLs))
     plot(Tree, label.offset=1, x.lim=10)
     tiplabels(round(as.numeric(MyData[,3])), 1:6, col="Blue", frame="none",
bg="clear",adj = -0.5)
     title(main="Richness scores")

## End(Not run)
```

MatchTaxatoEOLID *Matches Taxa to and eolID*

Description

This function will take a sting of taxa and search EOL database for the EOL ID. These IDs can then be used with the DownloadEOLpages function.

Usage

```
MatchTaxatoEOLID(ListOfTaxa, exact=TRUE, ...)
```

Arguments

ListOfTaxa	List of EOL taxa to search and download EOL pages
exact	Should taxon name match exactly in EOL or fuzzy match
...	further arguments to be passed to MatchTaxatoEOLID

Value

Returns a table with the submitted taxon name, returned taxon name, and eol ID number.

See Also

DownloadEOLpages DownloadSearchedTaxa

Examples

```
bears<-c("Ursus americanus", "Ursus thibetanus", "Ursus maritimus")
MatchTaxatoEOLID(bears, exact=TRUE)
```

`MyEOLs` *Example EOL and Hierarchy pages*

Description

MyEOLs is an example of a subset of EOL files that were saved as an R object. It is a list of six animal species (Camelus bactrianus, Camelus dromedarius, Hippopotamus amphibius, Rattus rattus, Rana cascadae, Bufo bufo), each item in the list is a separate species XML. The names of items in the list are the associated EOL numbers. You can use these numbers to download the pages directly, or go to the EOL website and see the interactive page (for example, http://eol.org/pages/344581/overview).

MyHiers is an example of the hierarchy pages that you can get from the EOL API, saved as an R object. This download uses NCBI hierarchy information (see DownloadHierarchy help pages for an example). The taxa in MyHiers is the same as in MyEOLs. Note that any identification numbers or even taxon names may differ between these files, since EOL and NCBI are independent.

Format

XML, R

`OneFileHierarchy` *Creates A Taxonomic Hierarchy for a single Taxon*

Description

This function uses a provider hierarchy file to gather information about all taxonomic levels and their associated IDs.

Usage

```
OneFileHierarchy(MyHier)
```

Arguments

`MyHier`	A single of filename for downloaded hierarchy page

Value

Returns a data frame where each row is a hierarchical taxonomic unit.

Examples

```
data(MyHiers)
OneFileHierarchy(MyHiers[1])
```

PageProcessing *Process XML Data into tree format*

Description

This function will read in the XML data and parse it into a tree structure for R to read.

Usage

```
PageProcessing(MyEOL, ...)
RemoveNAFiles(MyFiles)
```

Arguments

MyEOL	A filename or R object for downloaded EOL pages
MyFiles	A vector of filenames or R objects for EOL or Hier pages
...	further arguments to be passed to PageProcessing

Value

PageProcessing returns XML tree as an R object. RemoveNAFiles is an internal function that will take a vector of filenames and check to make sure they all have data associated (sometimes requested pages will be empty and contain no information). If they are empty, they are cleaved from the analysis.

Examples

```
#Reads in a file
#PageProcessing("eol1.xml")

#or reads in an R object
data(MyEOLs)
PageProcessing(MyEOLs[1])
```

PingAPI *Test the working order of the EOL API*

Description

This function will send out a ping to the EOL API to make sure it is in working order. The EOL server will return either a message of success or failure.

Usage

```
PingAPI(MyKey=NULL)
```

Arguments

MyKey	An optional user identification key to identify yourself to EOL

Value

Returns a message with success or failure.

Examples

```
PingAPI()
#PingAPI(MyKey)
```

RepeatDataToDrop *Repeat Data To Drop*

Description

These internal functions determine columns or rows of data that are either all the same (thus providing no hierarchical information) or contain NAs. DropADim walks through the TreeData table and deletes either a row or a column that has the most amount of missing data. It continues this stepwise until there are no longer any NAs in TreeData. This preserves as much data as possible. Then RepeatDataToDrop will delete any columns of TreeData that have all the same information (these can not be used with ape tree plotting)

Usage

```
RepeatDataToDrop(TreeData)
DropADim(TreeData)
```

Arguments

TreeData	A data frame from

Value

DropADim returns a new TreeData table with no NAs. RepeatDataToDrop returns boolean response of whether to drop data or not.

subsetDataForHierTrees *Subset Data For Hier Trees*

Description

This internal function removes repeat taxonomic units, undefined clades, and NAs from hierarchy information.

Usage

```
subsetDataForHierTrees(oneFileHier, HierID)
```

Arguments

oneFileHier	A dataframe of taxonomic hierarchy information out of OneFileHierarchy function
HierID	Hierarchy ID value of interest

Value

Returns a subset data frame with only parent nodes from HierID up for building taxonomic trees.

TaxonChildren *Gathers A List Of Taxonomic Parents and Offspring*

Description

This function goes through the hierarchy pages to collect taxonomic offspring and parantage.

Usage

```
TaxonChildren(MyHiers)
TaxonParents(MyHier)
```

Arguments

MyHiers	A vector or single filename for downloaded hierarchy pages
MyHier	A single filename for downloaded hierarchy pages

Value

TaxonChildren will report the primary offspring of a taxon if the hierarchy page reports this information. TaxonParents will return the list of taxonomic parentage.

See Also

MakeTreeData

Examples

```
#simple example using Reol data:
data(MyHiers)
TaxonChildren(MyHiers)
TaxonParents(MyHiers[1])


#Species of Anolis off NCBI
data(MyHiers)
TaxonChildren(MyHiers)

#Example to get all Anolis species from NCBI
## Not run:
eolAnolis <- DownloadSearchedTaxa("Anolis", to.file=FALSE)
hierAnolis <- DownloadHierarchy(eolAnolis, to.file=FALSE, database="NCBI
     Taxonomy")
TaxonChildren(hierAnolis)

#Species of Anolis off The Reptile Database
eolAnolis <- DownloadSearchedTaxa("Anolis", to.file=FALSE)
repdbAnolis <- DownloadHierarchy(eolAnolis, to.file=FALSE, database="The Reptile
     Database")
TaxonChildren(repdbAnolis)

## End(Not run)
```

Index