genomes

October 25, 2011

`acc2date` *Retrieve release dates from NCBI's revision history*

Description

Returns the date a sequence was first seen at NCBI using the revision history.

Usage

```
acc2date(ids, common=TRUE)
```

Arguments

`ids`	a vector or comma-separated list of sequence accessions or GI numbers
`common`	if present, use the common revision history link

Details

Searches the sequence revision history at NCBI `http://www.ncbi.nlm.nih.gov/sviewer/girevhist.cgi` and parses the line listing the date a sequence was *first seen at NCBI*. In many cases, a sequence replaces earlier IDs and will therefore include a common revision history link. If this option is set, each common history link is searched and the earliest release date from the collection of sequence ids is returned instead.

Value

A data frame listing the sequence identifier, release date, and if common revision history was used.

Author(s)

Chris Stubben

See Also

`revhist`

Examples

```
data(lproks)
yp<-subset(lproks, name %like% 'Yersinia*CO92')
yp$genbank
# 1 chromosome and 3 plasmids
acc2date(yp$genbank)

acc2date(yp$genbank, common=FALSE)
```

`doublingTime` *Doubling time for genome projects*

Description

Calculates the doubling time of genome sequencing project releases

Usage

```
doublingTime(x, subset, time = "days")
```

Arguments

`x`	genomes data frame with class 'genomes'
`subset`	logical vector indicating rows to keep
`time`	return doubling time in days (default), months, or years

Value

the doubling time

Author(s)

Chris Stubben

Examples

```
data(lproks)
doublingTime(lproks)
doublingTime(lproks, status == 'Complete', time='months')
```

genomes-lines *Add lines to a genomes plot*

Description

Add lines representing the cumulative number of genomes by released date to a genome plot.

Usage

```
## S3 method for class 'genomes'
lines(x, subset,  ...)
```

Arguments

`x`	genomes data frame with class 'genomes'
`subset`	logical vector indicating rows to keep
`...`	additional arguments passed to lines

Details

Use `plotby` to plot multiple lines within the same genome table. This function adds new lines from different genome tables to the same plot.

Author(s)

Chris Stubben

See Also

`plotby`

Examples

```
data(lproks)
data(leuks)
data(lenvs)
plot(lproks, log='y', las=1, lty=3)
lines(leuks, col="red", lty=2)
lines(lenvs, col="green3", lty=1)
legend("topleft", c("Microbes", "Eukaryotes", "Metagenomes"),
                  bty='n',   lty=3:1, col=c("blue", "red", "green3"))
```

genomes-plot *Genome table plots by release date*

Description

Generic function for plotting the cumulative number of genomes by released date for genome tables

Usage

```
## S3 method for class 'genomes'
plot(x, subset,
 xlab = "Release Date", ylab ="Genomes",
 type= "l", col = "blue", ...)
```

Arguments

`x`	a genomes data frame with class 'genomes'
`subset`	logical vector indicating rows to keep
`xlab`	x-axis label
`ylab`	y-axis label
`type`	type of plot, default is a blue line
`col`	color
`...`	additional arguments passed to plot

Value

A plot of the cumulative total of genomes by release date.

Author(s)

Chris Stubben

See Also

`plotby` to plot release dates by any grouping column

Examples

```
data(lproks)
plot(lproks)
plot(lproks, name %like% 'Yersinia*', ylab="Yersinia genomes")
```

`print.genomes` *Print genome tables*

Description

Print method for genome tables

Usage

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

Arguments

`x`	a genomes data.frame
`...`	additional arguments ignored

Details

Prints the first four columns and first five and last row of a genomes data.frame. To view all the columns in a genome table, you can either select fewer than 7 rows or convert the object to a data.frame (`data.frame(lproks)`)

Author(s)

Chris Stubben

Examples

```
data(lproks)
lproks
## full table printed  if 6 rows or less
lproks[1,]
```

`genomes-subset` *Subset genome tables*

Description

Return subsets of a genome table.

Usage

```
 ## S3 method for class 'genomes'
subset(x, ...)
```

Arguments

`x`	a genomes data.frame
`...`	additional arguments ignored

Details

Preserves the genomes class and other attributes if name and released columns are present, otherwise the subsetting operation will return a data.frame. Update methods will not work on subsets of genome tables, but the other genome functions will work

Author(s)

Chris Stubben

Examples

```
data(lproks)
yp<-subset(lproks, name %like% 'Yersinia pest*')
yp
summary(yp)
```

`genomes-summary` *Genome table summaries*

Description

Generic function for summarizing genome tables

Usage

```
## S3 method for class 'genomes'
summary(object, subset, top = 5, ...)
```

Arguments

`object`	a genomes data frame
`subset`	logical vector indicating rows to keep
`top`	number of recently released genomes to display, default is 5
`...`	additional arguments are currently ignored

Value

A list with 2 or 3 elements: the total number of genomes, counts by status (if column is present), and a table listing recent submissions.

Author(s)

Chris Stubben

See Also

`plot.genomes`

Examples

```
data(leuks)
summary(leuks)
summary(leuks, group=='Fungi')
```

genomes-update *Genome table updates*

Description

Generic function for updating genome tables.

Usage

```
## S3 method for class 'genomes'
update(object, ...)
```

Arguments

object	a genomes data frame to update
...	additional arguments are currently ignored

Details

update will retrieve the new genome table using the update string in attr(object, 'update'). The new table will replace the existing version, *but not permanently*, since reloading the dataset using data will restore the older version. If you have write permission, one option is to use system.file to replace the data set (see the example below).

Value

Returns the updated genome table and a count of the number of new IDs added and old IDs removed. Old IDs are typically assembly genomes in NCBI tables that have been released as a single complete genome.

Author(s)

Chris Stubben

See Also

genomes-summary, genomes-plot

Examples

```
## Not run: data(lproks)
## Not run: update(lproks)

# to replace the data set permanently
x <- system.file("data", "lproks.rda", package="genomes")
x
## Not run: save(lproks, file=x)
```

genomes *Introduction to the genomes package*

Description

Genomes sequencing project statistics from prokaryotes, eukaryotes, and metagenomes.

Author(s)

Chris Stubben <stubben@lanl.gov>

Examples

```
data(lproks)
lproks
summary(lproks)
plot(lproks)
## Not run: update(lproks)
```

genus *Extract the genus name*

Description

Extracts the genus name from a scientific name (latin binomial)

Usage

```
genus(x)
```

Arguments

x	A vector of scientific names

Details

Returns the first word in the scientific name. For candidate species labeled *Candidatus*, then the second word is returned.

Value

A vector of genus names

Author(s)

Chris Stubben

See Also

`species`

Examples

```
genus("Bacillus anthracis Ames")
data(lproks)
x <- table2(genus(lproks$name))[1:10,]
dotplot(rev(x), xlab="Genomes")
```

image2 *Display a matrix image*

Description

Creates a grid of colored rectangles to display a matrix

Usage

```
image2(x, col = rev(heat.colors(24)), breaks, log = FALSE,
 zeroNA=TRUE, sort01=FALSE, all=FALSE, border = NA, box.offset = 0.1,
 round = 3, cex, text.cex = 1, text.col = "black", mar = c(1, 3, 3, 1),
 labels = 2:3, label.offset = 0.1, label.cex = 1)
```

Arguments

`x` A numeric matrix, typically with row and column names

`col` A vector of colors for boxes

`breaks` A numeric vector of break points or number of intervals into which `x` is to be `cut`. Default is the length of `col`

`log` Cut values in `x` using a log scale, default TRUE

`zeroNA` Set zeros to NA (and color white)

`sort01` Sort rows in descending order using the entire string of numbers

`all` Display entire matrix, default is first 50 rows and columns

`border` The border color for boxes, default is no borders

`box.offset` Percent reduction in box size (a number between 0 and 1), default is 10% reduction

`round` Number of decimal places to display values of `x` in each box

`cex` Magnification size of text and labels, if specified this will replace values in both text.cex and label.cex

`text.cex` Magnification size of text in cells only

`text.col` Color of text in cells, use NA to skip text labels

`mar` Margins on four sides of plot

`labels` A vector giving sides of the plot (1=bottom, 2=left, 3=top, 4=right) for row and column labels

`label.offset` Amount of space between label and boxes

`label.cex` Magnification size of labels

Details

Missing values (NAs) and zeroes are assigned to the color white (unless zeroNA is FALSE) and remaining values are cut into groups and colored using the assigned values.

Value

A image plot of the matrix in `x`

Author(s)

Chris Stubben

See Also

`image`

Examples

```
## Journals with most microbial genome publications,
data(pubmed)
z<-table2(pubmed$journal, pubmed$year, n=15)
image2(z[,-ncol(z)], sort=TRUE, mar=c(1,10,3,1), cex=.8)
```

`lenvs` *Metagenome sequencing projects at NCBI*

Description

Metagenome sequencing projects from the Entrez genome project at NCBI

Usage

```
data(lenvs)
```

Format

A genomes data frame with observations on the following 10 variables.

`pid` genome project id

`name` metagenome title or taxonomy name

`released` released date

`source` metagenome source

`type` metagenome type, environmental (E) or organismal (O)

`accession` comma-separated list of accession numbers

`parent` parent genome project id

`center` sequencing center

`blast` has blast page

`traces` has traces

Source

downloaded from http://www.ncbi.nlm.nih.gov/genomes/lenvs.cgi

Examples

```
data(lenvs)
lenvs
## single row
t(lenvs[1,])
plot(lenvs)
summary(lenvs)
```

leuks *Eukaryotic genome projects at NCBI*

Description

Eukaryotic genome sequencing projects at NCBI

Usage

```
data(leuks)
```

Format

A genomes data frame with observations on the following 20 variables.

`pid` genome project id

`name` taxonomy name

`status` sequencing status

`released` released date

`group` taxonomy group (animals, fungi, protists, or plants)

`subgroup` taxonomy subgroup

`taxid` taxonomy id

`size` genome size (Mbp)

`chromosomes` number of chromosomes

`method` sequencing method

`depth` depth or coverage

`center` pipe-separated list of sequencing centers

`genbank` has GenBank sequences

`pubmed` has PubMed

`refseq` has RefSeq sequences

`gene` has Gene link

`traces` has Traces

`blast` has Blast page

`mapview` has MapView

`ftp` comma-separated list of ftps

Source

downloaded from Entrez genome project at http://www.ncbi.nlm.nih.gov/genomes/leuks.cgi

Examples

```
data(leuks)
leuks
# single row, long format
t(leuks[1,])
plot(leuks)
summary(leuks)
dotplot(sort(table(leuks$subgroup)), pch=16, xlab="Genome projects")
```

---

like *Pattern matching using wildcards*

Description

Pattern matching using wildcards

Usage

```
x %like% pattern
```

Arguments

`pattern`	character string containing the pattern to be matched
`x`	values to be matched

Details

Only wildcards matching a single character '?' or zero or more characters '*' are allowed. Matches are case-insensitive. The pattern is first converted to a regular expression using `glob2rx` then matched to values in `x` using `grep`.

This is a shortcut for a commonly used expression found in the `subset` example where `nm %in% grep("^M", nm, value=TRUE)` simplifies to `nm %like% 'M*'`.

Value

A logical vector indicating if there is a match or not. This will mostly be useful in conjunction with the `subset` function.

Author(s)

Chris Stubben

See Also

`grep`, `glob2rx`, `subset`

Examples

```
data(lproks)
subset(lproks, name %like% 'Yersinia*', c(name, released))
# also works with date or numeric fields
subset(lproks, released %like% '2008-01*', c(name, released))
```

---

lproks *Microbial genome projects at NCBI*

Description

Microbial genomes from Entrez genome project at NCBI.

Usage

```
data(lproks)
```

Format

A genomes data frame with observations on the following 31 variables.

`pid` genome project id

`name` taxonomy name

`status` sequencing status, Complete, Assemby, or In Progress genomes

`released` released date, complete and WGS genomes only

`refseq_pid` RefSeq project id

`taxid` taxonomy id

`kingdom` kingdom

`group` phylum or class

`size` genome size (Mbp)

`GC` percent GC content

`chromosomes` number of chromosomes, complete genomes only

`plasmids` number of plasmids, complete genomes only

`modified` modified date, complete genomes only

`genbank` comma-separated list of GenBank accession numbers

`refseq` comma-separated list of RefSeq accession numbers

`publication` comma-separated list of PubMed ids, complete genomes only

`center` pipe-separated list of sequencing centers

`contigs` number of genome contigs. For complete genomes, contigs are the sum of chromosomes and plasmids

`cds` number of coding sequences, WGS only

`url` sequencing center url, WGS and In Progress genomes only

`gram` gram stain

`shape` shape

`arrange` arrangement

`endospore` endospores

`motility` motility

`salinity` salinity

`oxygen` oxygen requirement

`habitat` habitat

`temp` temperature preference

`range` temperature range

`pathogen` pathogenic in host

`disease` disease

Details

This table is constructed using all three tabs at `http://www.ncbi.nlm.nih.gov/genomes/lproks.cgi`. Complete genomes and In Progress tabs are combined and then joined to the Organism Info tab.

The `update(genomes)` function downloads a recent copy of the table from NCBI. The number of new project IDs are reported as well as the number of project IDs removed (which are typically Assembly genomes that are now available as a Complete sequence). Please note that NCBI is currently changing how prokaryotic genomes are managed and some changes to these tables are possible (see `http://www.ncbi.nlm.nih.gov/genomeprj` for details).

Source

downloaded from `http://www.ncbi.nlm.nih.gov/genomes/lproks.cgi`

Examples

```
data(lproks)
lproks
#single row (long format)
t(lproks[1,])
class(lproks)
## download stats
attributes(lproks)[c("stats", "date","url")]
summary(lproks)
## check for missing release dates
table2(!is.na(lproks$released), lproks$status, dnn=list("Released Date?", "Status"))
plot(lproks)
plotby(lproks, log='y', las=1)
## download recent table from NCBI
## Not run: update(lproks)
## Yersinia genomes
yp <- subset(lproks, name %like% 'Yersinia*')
yp
summary(yp)
plotby(yp, labels=TRUE, cex=.5, lbty='n')
```

`plotby` *Plot groups of genomes by release date*

Description

Plots the cumulative number of genomes by released date for different groups of genomes

Usage

```
plotby(x, groupby = "status", subset = NA, top = 5,
labels = FALSE, abbrev = TRUE, flip = NA,
 legend = "topleft", lbty = "o", lcol = 1, ltitle = NULL, lcex = 1,
 lsort = TRUE, cex = 1, ylim = NA, las = 1, lwd = 1, log = "",
xlab = "Release Date", ylab = "Genomes", type='l',
col = c("blue", "red", "green3", "magenta", "yellow"),
lty = 1:top, pch = c(15:18, 1:3), ...)
```

Arguments

`x`	a genomes data frame
`groupby`	a column name in the genomes table or a vector to group by
`subset`	logical vector indicating rows to keep
`top`	number of top groups to display
`labels`	add genome names to each point - plot a single line and
`abbrev`	abbreviated genome names
`flip`	a number indicating where to flip labels from right to left, default is middle of plot
`legend`	a legend keyword or vector of x,y coordinates, defaults to top-left corner. Use NA for no legend
`lbty`	legend box type
`lcol`	number of columns in legend
`ltitle`	legend title
`lcex`	legend size expansion
`lsort`	sort legend by decreasing order of genomes, default true
`cex`	label size expansion
`ylim`	y axis limits
`las`	rotate axis labels
`lwd`	line width
`log`	log scale
`xlab`	x axis label
`ylab`	y axis label
`type`	plot type
`col`	line or point colors
`lty`	line type
`pch`	point type
`...`	additional items passed to plot

Details

Two different plot types are available. The default is to plot multiple lines, one for each group (like matplot). If `labels=TRUE`, then a single line is drawn with different labeled points for each group.

Value

A plot of released dates by group

Author(s)

Chris Stubben

See Also

`plot.genomes`

Examples

```
data(lproks)
# default group is status
plotby(lproks)
plotby(lproks, 'habitat', top=3)

## groupby can be a vector
plotby(lproks, genus(lproks$name), log='y', lcex=.7)
plotby(lproks, factor(lproks$pathogen %in% c("No"),
   labels=c("Pathogen", "Non-pathogen")), pathogen!="")

# OR plot labels
plotby(lproks, subset=name %like% 'Yersinia pestis*', labels=TRUE, cex=.5, lbty='n')
```

`pub2date` *Retrieve the published date from NCBI's PubMed database*

Description

Searches the PubMed database at NCBI and returns a short citation with author, year, title, journal and published date.

Usage

```
pub2date(pmids)
```

Arguments

`pmids`	a vector or comma-separated list of pubmed IDs

Details

Searches the Pubmed database using EFetch and parses the XML summary to return a short citation.

Value

A data.frame with 9 columns: pmid, authors, year, title, journal, volume, pages, published date, and article date.

Note

The article date is the date an electronic copy was available. See `pubmed` for additional details about columns.

Author(s)

Chris Stubben

Examples

```
data(lproks)
yp<-subset(lproks, name %like% 'Yersinia*CO92')
# comma-separated list
yp$publication
pub2date(yp$publication)
# or vector
pub2date( c(7542800, 7569993))
```

`pubmed` *Complete microbial genome publications in PubMed*

Description

Publications for complete microbial genomes in the PubMed database at NCBI

Usage

```
data(pubmed)
```

Format

A data frame with 747 observations on the following 9 variables.

`pmid` PubMed id

`authors` first 3 author names

`year` year journal was published

`title` title

`journal` journal name

`volume` volume number

`pages` pages

`pubdate` date journal was published (from PubDate tag)

`artdate` date electronic copy was available (from ArticleDate tag)

Details

This table was created by taking the *first* pubmed ID in the lproks table (publication column) and using pub2date to return the citation for each unique pubmed ID. In some cases, the genome publication may not be the first pubmed ID in lproks and no attempt was made to correct these rows (except for deleting 4 publications before 1995).

Source

PubMed database at NCBI

Examples

```
data(pubmed)

pubmed[1:2,c(1,3,4,8)]

# Streptomyces coelicolor A3(2) should use the second pmid.
# even worse, the release date uses the wrong published date!
data(lproks)
subset(lproks, pid==242, c(1,2,4,16))
pub2date(12000953)
```

---

revhist *Complete microbial genome release dates in Revision History*

Description

Lists the date a sequence was *first seen* using genbank accessions from complete microbial genomes in the Seqeunce Revision History database at NCBI.

Usage

```
data(revhist)
```

Format

A data frame with 1485 observations on the following 3 variables.

`id` genbank accession number

`released` date sequence was first seen

`common` was common revision history link used?

Details

This table was created by taking the *first* genbank accession in the lproks table and using acc2date to return the date the sequence was first seen at NCBI. In some cases, the genome sequence may not be the first genbank ID in the list (eg, a plasmid sequence may be first) and no attempt was made to correct these rows.

Source

Sequence Revision History database at NCBI http://www.ncbi.nlm.nih.gov/sviewer/girevhist.cgi

Examples

```
data(revhist)

# sorted by release date in lproks
head(revhist)
data(lproks)
x<-subset(lproks, year(released)==1995, c(name,genbank))
x
# genome sequence BA000022 for Synechocystis is 4th id in list
acc2date(x$genbank)
```

species *Extract the species name*

Description

Extracts the species name from a scientific name

Usage

```
species(x, abbrev=FALSE, epithet=FALSE)
```

Arguments

`x`	A vector of scientific names
`abbrev`	Abbreviate the genus name
`epithet`	Return only the specific epithet (default is genus + specific epithat)

Details

Returns the species name. For candidate species labeled *Candidatus*, the qualifier is not included

Value

A vector of species names

Author(s)

Chris Stubben

See Also

`genus`

Examples

```
species("Bacillus anthracis Ames")
species("Bacillus anthracis Ames", abbrev=TRUE)
species("Bacillus anthracis Ames", epithet=TRUE)
data(lproks)
x <- table2(species(lproks$name))[1:10,]
dotplot(rev(x), xlab="Genomes")
## abbreviate genus name
x <- subset(lproks, name %like% 'Bacillus*')
x <- table2(species(x$name))[1:10, ]
names(x) <- species(names(x), TRUE)
dotplot(rev(x), xlab=expression(italic(Bacillus) ~ genomes))
```

`table2` *Format and sort a contigency table*

Description

Formats the output of `table` into an matrix ordered by total counts in descending order

Usage

```
table2(..., n = 10)
```

Arguments

`...`	one or more objects passed to `table`
`n`	number of rows to display, default 10

Details

Currently limited to 1 or 2 dimensional table arrays.

Value

A matrix, sorted by total counts in descending order. Any rows or columns with zero counts are also removed from the matrix.

Author(s)

Chris Stubben

See Also

`table`

Examples

```
data(leuks)
table(leuks$subgroup)
table2(leuks$subgroup)
## to display all rows, use NA or a large number...
table2(leuks$subgroup, n=100)
# 2-d table
table2(leuks$group, format(leuks$released, "%Y"))
```

taxid2names *Retreive taxonomy names from NCBI*

Description

Search the Entrez taxonomy database at NCBI and return names and lineages for valid taxonomy ids

Usage

```
taxid2names(ids)
```

Arguments

ids	a vector of NCBI taxonomy ids

Details

The function searches the Taxonomy database using the EFetch utility and returns an XML summary report, and then parses the name and lineage fields

Value

A dataframe listing taxonomy id, name and lineage

Author(s)

Chris Stubben

Examples

```
taxid2names(2)
x <- taxid2names(c(280855, 11595, 273349))
# remove common parents
x$lineage<- gsub("Viruses; ssRNA viruses; ssRNA negative-strand viruses; Bunyaviridae; ",
x
```

`term2neighbor` *Retrieve genome neighbors from NCBI*

Description

Search Entrez Genome at NCBI and retrieve links (other genomes for species) to the nucleotide database using Entrez programming utilities (eUtils)

Usage

```
term2neighbor(term, derived = FALSE, sortdate = FALSE, fulltable = FALSE)
```

Arguments

`term`	Any valid combination of Entrez search terms
`derived`	Include GenBank sequences that the Reference sequences were derived from (default is only the neighbors in genome_nuccore_samespecies)
`sortdate`	Sort the results by released date (default is by name)
`fulltable`	Return all 12 summary fields

Details

The functions searches the Genome database using the ESearch utility, finds links to Other Genomes for Species using ELink, returns document summary pages using ESummary, and then parses the XML fields using the `XML` package

Value

A genomes data frame with 5 columns (acc, name (defline), released date, taxid, and size). If fulltable is TRUE, then all fields are returned

Note

This function will most likely be useful for viral sequences, which typically have only one reference sequence per species, and other strains are linked as Genome Neighbors.

Author(s)

Chris Stubben

References

A description of the Entrez programming utilities is at `http://eutils.ncbi.nlm.nih.gov/`.

See Also

`term2summary` and `virus`

Examples

```
data(virus)
## Nipah virus list 7 neighbors
subset(virus, name %like% 'Nipah*')
# term2neighbor('Nipah virus[orgn]')
#  if plotting, also  include the genbank sequence that reference was derived from
x <- term2neighbor('Nipah virus[ORGN]', derived = TRUE)
x
plot(x, ylab = 'Nipah virus sequences')
```

`term2summary` *Retrieve genome summaries from NCBI*

Description

Search the Entrez Genome Project or Genome database at NCBI and retrieve a summary table using Entrez programming utilities (eUtils)

Usage

```
term2summary(term, db = 'genomeprj', sortdate = FALSE, fulltable = FALSE)
```

Arguments

`term`	Any valid combination of Entrez search terms
`db`	Database to search, either genomeprj or genome
`sortdate`	Sort the results by status and released date (default is by name)
`fulltable`	Return all 20 E-summary fields for genomeprj or 12 fields for genome.

Details

Searches either genome database using the ESearch utility, returns document summary pages using the ESummary utility, and then parses the XML fields using the `XML` package.

If searching Genome Project, then a genomes data frame with 4 columns (project id, name, status, released date) is returned. If fulltable is TRUE, then all 20 fields are returned, plus extra rows for overview genome projects (type = Top level), RefSeq genomes (type = RefSeq), and plasmid genomes (type = Plasmid genome). In many cases, recent assemblies will be listed on an overview page, a genome page (missing released date), and a RefSeq page (missing status).

If searching Genomes, then a genomes data frame with 6 columns (acc, name (defline), status, released, taxid, size) is returned, or all 12 columns if fulltable is TRUE.

Value

A genomes data frame

Author(s)

Chris Stubben

References

A description of the Entrez programming utilities is at http://eutils.ncbi.nlm.nih.gov/.

See Also

term2neighbor

Examples

```
# Genomes sequenced at Los Alamos
x <- term2summary( "Los Alamos AND Bacteria[ORGN]")
x
summary(x)
#  list of centers in lproks table are often incomplete
data(lproks)
summary(lproks, center %like% '*Los Alamos*')

## Taxonomy queries like genomes in Bacteroidetes phylum
x <- term2summary("Bacteroidetes[ORGN]")
x
plot(x, ylab = 'Bacteroidetes genomes')
```

---

top — *Find the most common values*

Description

Finds the most common values in a vector with repeating elements.

Usage

```
top(x, n = 10)
```

Arguments

x	A vector with some repeating elements
n	The number of top elements

Details

`top` returns a logical vector indicating if the element is one of the most common values in the vector

Value

A logical vector indicating if the element is one of the top values.

Note

This will mostly be useful in conjunction with the `subset` function.

Author(s)

Chris Stubben

See Also

`like`

Examples

```
x <- c("a", "b", "b", "c")
top(x, 1)
#top is a short cut for
x %in% names(sort( table(x), decreasing=TRUE))[1]

data(lproks)
x <- subset(lproks, status != 'In Progress' , c(name, status, released))
# get top 15 genera
x <- subset(x, top(genus(name), 15))
x$status[x$status == 'Assembly'] <- 'WGS'
y <- table(genus(x$name), x$status)
y <- cbind(y, Total=rowSums(y))
y <- y[order(y[ ,3]), ]    # order by total

dotplot(y ,  xlab=list("Number of genomes at NCBI",cex=.8),
        par.settings=list(superpose.symbol=list(pch=15:17)),
        auto.key=list(cex=.8, columns=3, between=.5,  between.columns=1))
```

`virus` *Virus genomes at NCBI*

Description

Viral reference genome sequencing projects at NCBI.

Usage

```
data(virus)
```

Format

A genomes data frame with the following 10 variables.

`name` virus name

`released` release date

`neighbors` number of Genome Neighbors

`segments` number of segments

`refseq` RefSeq accession number

`isolate` isolate name

`size` genome size (nt)

`proteins` number of proteins

`host` host name

`updated` modified date

Details

Please refer to the Viral genomes page at NCBI http://www.ncbi.nlm.nih.gov/genomes/GenomesHome.cgi?taxid=10239&hopt=aboutsite for details on Reference genomes. One Reference genome is selected per viral species and other strains are linked as Genome Neighbors (other complete sequences for the species). See the `term2neighbor` function to get a list of Genome neighbors.

Summing the number of segments in this table should return the total number of reference sequences; however, summing the number of genome neighbors will not return the number of linked GenBank sequences since many counts are duplicated or missing (eg, Dengue virus neighbors are listed 4 times, Influenza A and B neighbors are missing.

Source

downloaded from http://www.ncbi.nlm.nih.gov/genomes/GenomesGroup.cgi?taxid=10239&opt=Virus&sort=genome

Examples

```
data(virus)
plot(virus)
summary(virus)
sum(virus$segments)
# some neighbors repeat (others are missing)
subset(virus, name %like% 'Dengue*')
subset(virus, name %like% 'Monkey*')
# list the neighbors
term2neighbor("Monkeypox virus[orgn]")

## most common phages
table2(species(grep("phage", virus$name, value=TRUE)))
```

`year` *Parse a date string*

Description

Parses the year or month from a date

Usage

```
year(x)
month(x)
```

Arguments

`x`	a date

Details

functions are a shortcut for `as.numeric(format.Date(x, "%Y"))`

Value

the year or month

Author(s)

Chris Stubben

Examples

```
data(lproks)
table(year(lproks$released))
# just complete genomes
table(year(lproks$released[lproks$status=="Complete"]))
```

Index