

genomes

October 5, 2010

doublingTime *Doubling time for genome projects*

Description

Calculates the doubling time of genome sequencing project submissions

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

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

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")
```

gold

Genomes OnLine Database (GOLD) table

Description

Genome sequencing projects listed at GOLD

Usage

```
data(gold)
```

Format

A genomes data frame with 7000+ observations on the following 105 variables.

```
goldstamp GOLD project id
name project name (was ORGANISM NAME)
status status (was SEQUENCING STATUS)
released released date (was COMPLETION DATE)
superkingdom a character vector
phylum a character vector
class a character vector
order a character vector
family a character vector
genus a character vector
species a character vector
domain a character vector
strain a character vector
serovar.biovar a character vector
common.name a character vector
project.name a character vector
culture.collection a character vector
type.strain a character vector
old.goldstamp a character vector
project.type a character vector
project.status a character vector
availability a character vector
sequencing.centers a character vector
```

funding a character vector
contact.name a character vector
contact.email a character vector
contact.url a character vector
locus.tag a character vector
publication a character vector
project.relevance a character vector
taxon.id a numeric vector
project.id a numeric vector
archive.id a numeric vector
short.reads.archive.id a character vector
gcat_id a character vector
hmp.id a numeric vector
homd.id a character vector
straininfo.id a numeric vector
greengenes.object.id a character vector
img.object.id a numeric vector
genome.data a character vector
sequencing.quality a character vector
comments a character vector
library.method a character vector
reads.count a character vector
vector a character vector
assembly.method a character vector
sequencing.depth a character vector
gene.calling.method a character vector
contig.count a numeric vector
size.kb a numeric vector
orfs a numeric vector
chromosome.count a numeric vector
plasmid.count a numeric vector
gc.content a numeric vector
sequencing.method a character vector
sequencing.country a character vector
isolation.site a character vector
isolation.source a character vector
isolation.comments a character vector
collection.date a character vector
isolation.country a character vector
isolation.pubmed.id a numeric vector

geographic.location a character vector
latitude a character vector
longitude a character vector
altitude a character vector
depth a character vector
host.name a character vector
host.taxon.id a numeric vector
host.gender a character vector
host.age a character vector
host.race a character vector
host.health a character vector
host.medication a character vector
host.specificity a logical vector
body.sample.site a character vector
body.product a character vector
body.sample.subsite a character vector
host.comments a character vector
oxygen.requirement a character vector
cell.shape a character vector
sporulation a character vector
pressure a character vector
temperature.range a character vector
salinity a character vector
ph a character vector
cell.diameter a character vector
cell.length a character vector
color a character vector
gram.staining a character vector
biotic.relationships a character vector
symbiotic.interaction a character vector
symbiotic.relationship a character vector
symbiont.name a character vector
symbiont.taxon.id a numeric vector
cell.arrangement a character vector
disease a character vector
habitat a character vector
temperature a character vector
metabolism a character vector
motility a character vector
phenotype a character vector
energy.source a character vector

Details

The column names match the positions and names in the source file except for the first four columns. Note that released dates are listed for complete genomes only.

Source

<http://www.genomesonline.org/DBs/goldtable.xls>

References

<http://nar.oxfordjournals.org/cgi/content/full/gkp848v1>

Examples

```
data(gold)
gold
# single row, long format
t(gold[1,])
## release date is completion date
table(!is.na(gold$released), gold$status)
## genomes with Entrez project ID
table(!is.na(gold$project.id), !is.na(gold$released),
      dnn=list("Has project ID?", "Is complete?"))
plotby(gold, "project.type", log='y')
table2(gold$project.type)
plotby(gold, "domain", project.type == 'Genome-Isolate', log='y')
```

hmp

Human Microbiome Project (HMP) data

Description

Genome sequencing projects listed by the Human microbiome project

Usage

```
data(hmp)
```

Format

A genomes data frame with 18 columns.

```
pid HMP project ID
name project name
status project status
released date draft sequencing completed
entrezid Entrez genome project id
taxid NCBI taxonomy id
phylum phylum
site primary body sample site
```

subsite body sample subsite
 goal finishing goal
 submission NCBI submission status
 center sequencing center name
 assembler assembler
 reads total number of reads
 coverage coverage
 contigs number of contigs
 platform sequencing platform
 comment comments

Details

see the HMP Project Catalog at <http://www.hmpdacc.org/> . The full table has 34 columns and only 18 columns are displayed in this dataset

Source

<http://durian.jgi-psf.org/~kliolios/HMP/hmp.xls>

Examples

```

data(hmp)
hmp
  t(hmp[1,])
summary(hmp)
plotby(hmp, "site", main='Human microbiome project', lbtty='n', log='y', lcex=.7)
plotby(hmp)
table(hmp$status, !is.na(hmp$released),
      dnn=list("status", "released?"))

```

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, Assembly, or In Progress genomes
released released date, complete and WGS genomes only
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.

Nearly all of the Assembly genomes released after Sept 2009 are missing release dates. If needed, these dates are available from Entrez Genomes (see the example in [wgs](#)).

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)
## many Assembly genomes are now 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, lbtty='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", lbtty = "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

lbtty	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, lbtty='n')
```

species	<i>Extract the species name</i>
---------	---------------------------------

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 epithet)

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	<i>Format and sort a contingency table</i>
--------	--

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 <code>table</code>
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	<i>Retrieve taxonomy names from NCBI</i>
-------------	--

Description

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

Usage

```
taxid2names(ids)
```

Arguments

ids an NCBI taxonomy id

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	<i>Retrieve genome neighbors from NCBI</i>
---------------	--

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_nuclide_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 (define), 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 (define), 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*')

##In progress microbial genomes with data in the Short Read Archive
x <- term2summary( 'inprogress[Sequencing status] AND genomeprj sra[Filter] NOT eukaryota')
x

## 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 8 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

Details

Please refer to the Viral genomes page at NCBI <http://www.ncbi.nlm.nih.gov/genomes/GenomesHome.cgi?taxid=10239&hopt=aboutsitesite> 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)))
```

wgs

Bacterial assembly sequences

Description

Bacterial whole genome shotgun (wgs) assemblies in the Entrez Genome database

Usage

```
data(wgs)
```

Format

A genomes data frame with 1000+ observations on the following 9 variables.

```
acc accession number without trailing 00000000
name taxonomy name
status status, always Assembly
```

released release date
size size (bp)
proteins proteins
rna RNA
genes genes
updated update date

Details

See the left side bar at the Genomes home page at <http://www.ncbi.nlm.nih.gov/sites/entrez?db=genome>

Source

<http://www.ncbi.nlm.nih.gov/genomes/genlist.cgi?taxid=2&type=3>

Examples

```
data(wgs)
wgs
summary(wgs)[[1]]
data(lproks)
## missing dates in genome project - paste 8 zeros to acc and match to refseq without dat
x <- subset(wgs, paste(wgs$acc, "00000000", sep="") %in% lproks$refseq[is.na(lproks$rel
      c(name, released, status)])
x
table(format(x$released, "%Y-%m"))
## better plot than
## plotby(lproks, lsort=FALSE)
plotby(rbind(x, lproks[,c("name", "released", "status")]), log='y')
```

Index

*Topic **datasets**

gold, 8
hmp, 11
lenvs, 12
leuks, 13
lproks, 15
virus, 25
wgs, 26

*Topic **hplot**

genomes-lines, 2
genomes-plot, 3
plotby, 17

*Topic **manip**

like, 14

*Topic **methods**

doublingTime, 1
genomes-subset, 4
genomes-summary, 5
genomes-update, 6
genus, 7
print.genomes, 4
species, 19
table2, 20
taxid2names, 21
term2neighbor, 21
term2summary, 23
top, 24

*Topic **package**

genomes, 7
%like% (*like*), 14

doublingTime, 1

genomes, 7
genomes-plot, 6
genomes-summary, 6
genomes-lines, 2
genomes-plot, 3
genomes-subset, 4
genomes-summary, 5
genomes-update, 6
genus, 7, 19
glob2rx, 15
gold, 8

grep, 15

hmp, 11

lenvs, 12
leuks, 13
like, 14, 25
lines.genomes (*genomes-lines*), 2
lproks, 15

matplotlib, 18

plot.genomes, 5, 18
plot.genomes (*genomes-plot*), 3
plotby, 2, 3, 17
print.genomes, 4

species, 7, 19
subset, 15, 24
subset.genomes (*genomes-subset*), 4
summary.genomes
 (*genomes-summary*), 5

table, 20

table2, 20

taxid2names, 21

term2neighbor, 21, 23, 26

term2summary, 22, 23

top, 24

update.genomes (*genomes-update*), 6

virus, 22, 25

wgs, 16, 26