genomes

March 24, 2012

`complete` *Complete microbial genome dates*

Description

Dates associated with complete microbial genomes at NCBI

Usage

```
data(complete)
```

Format

A data frame with 1787 observations on the following 11 variables.

`pid` genome project id

`name` taxonomy name

`released` release date in the `lproks` table

`genbank` genbank ID of the largest chromosome from the comma-separated list in the `lproks` table

`history` the revision history date associated with the genbank ID

`submitted` the submission date associated with the genbank ID

`pmid` pubmed ID of genome paper from the comma-separated list in the `lproks` table.

`published` the published date of the pubmed ID

`wgs` the WGS accession, if previously released as an assembly

`assembled` the assembly release date

`source` likely source of the `lproks` release date

Details

This table was created to check release dates in the `lproks` table. The revision history date was added using `ncbiRelease`, the submission date using `ncbiSubmit`, and publication date using the `pub` dataset. Currently, 178 complete genomes are mislabeled with the assembly release date (out of 473 that were previously released as an assembly) , the source for 204 others is unknown. and many of the first genomes released report "published" dates.

Source

See http://www.ncbi.nlm.nih.gov/projects/WGS/WGSprojectlist.cgi for a list of the 473 assembly projects superceded by a complete genome sequence.

Examples

```
data(complete)
# some early genomes use published dates from the wrong paper (eg, 2nd and 4th genomes be
complete[1:5, ]
# likely source of release dates
table2(complete$source)
# genomes previously submitted as WGS
table(is.na(complete$wgs))
subset(complete, !is.na(wgs))[1:2,]
```

---

doublingTime *Doubling time for genome projects*

Description

Calculates the doubling time of genome sequencing project releases

Usage

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

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
curdate	include the current date in calculation, if false, then default is range of release dates

Value

the doubling time

Author(s)

Chris Stubben

Examples

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

`enaExperiment` *ENA SRA experiment details*

Description

Return details about SRA experiments in the ENA

Usage

```
enaExperiment(accs, batchsize = 100)
```

Arguments

`accs`	a vector of SRA experiments or a range of accessions with prefix SRX, ERX, DRX, etc.
`batchsize`	number of accs to include in a single comma-separated url string

Details

Parses some of the tags and values in the XML experiment report

Value

a data.frame with platform, model and library details like name, layout, source and selection

Author(s)

Chris Stubben

References

http://www.ebi.ac.uk/ena/about/browser#sra_xml

See Also

`sra` and `enaSRA`

Examples

```
# compare to http://www.ebi.ac.uk/ena/data/view/ERX007105
 enaExperiment("ERX007105")

# chimps
pan<-enaSRA(9596)
head(pan)
# first experiment in sample
pan2<-enaExperiment(substr(pan$experiment, 1,9))
head(pan2)
table2(pan2$model)
```

`enaFiles` *ENA SRA submitted or fastq files*

Description

Retrieve a list of SRA submitted files or generated fastq files at the ENA

Usage

```
enaFiles(acc, file = "submitted")
```

Arguments

`acc`	a vector of SRA accession numbers
`file`	return submitted (default) or fastq file names

Value

a data.frame with experiment details and files names

Note

Only a single accession number is allowed in the URL string, so retrieving files from multiple accessions will be slow

Author(s)

Chris Stubben

References

http://www.ebi.ac.uk/ena/about/browser#sra_submitted_files

Examples

```
enaFiles("ERP000141")
enaFiles("ERP000141", "fastq")
```

enaProject *ENA projects*

Description

Search for projects at ENA using a taxonomy name or id

Usage

```
enaProject(tax, limit = 1000, refseq = TRUE)
```

Arguments

tax	a taxonomy ID or name
limit	total number of projects to return
refseq	include RefSeq projects

Details

Searches the project data from the taxonomy portal at ENA.

Value

a data.frame listing projects with submission details

Note

URL strings at ENA require a taxonomy ID, so searching by name uses a ncbiTaxonomy ID lookup at NCBI.

Author(s)

Chris Stubben

References

http://www.ebi.ac.uk/ena/about/browser#taxonomy_portal_options

See Also

enaSRA to search for SRA samples.

Examples

```
pan<-enaProject(9596)
head(pan)
table(pan$type, pan$method)
```

enaSRA *ENA sequence read archive*

Description

Search for SRA samples at the ENA using a taxonomy name or id

Usage

```
enaSRA(tax, limit = 5000)
```

Arguments

tax	a taxonomy ID or name
limit	total number of samples to return

Details

Searches the sra_sample data from the taxonomy portal at ENA.

Value

a data.frame listing SRA samples

Note

URL strings at ENA require a taxonomy ID, so searching by name uses a ncbiTaxonomy ID lookup.

Author(s)

Chris Stubben

References

http://www.ebi.ac.uk/ena/about/browser#taxonomy_portal_options

See Also

sra for all microbial SRA samples and a description of columns. Also see enaTaxonomy to check the total number of SRAs before downloading

Examples

```
# chimps
pan<-enaSRA(9596)   # or pan<-enaSRA("Pan")
head(pan)
nrow(pan)
table2(pan$center)
bases(sum(pan$bases, na.rm=TRUE))
bases(sum(pan$reads, na.rm=TRUE), round=1)
```

`enaStudy` *ENA SRA study details*

Description

Return details about SRA studies in the ENA

Usage

```
enaStudy(accs, batchsize = 100)
```

Arguments

`accs`	a vector of SRA studies or a range of accessions with prefix SRP, ERP, DRP, etc.
`batchsize`	number of accs to include in a single comma-separated url string

Value

a data.frame with study title, type, description, analysis

Note

only a few studies have secondary analysis

Author(s)

Chris Stubben

See Also

enaSRA

Examples

```
# compare to http://www.ebi.ac.uk/ena/data/view/ERP000054
enaStudy("ERP000054")

# chimps
pan<-enaSRA(9596)
head(pan)
pan2 <-enaStudy(pan$study)
head( pan2,2)
pan2[,1:2]
table2(pan2$type)
```

enaSubmission *ENA SRA submission dates*

Description

Return details about SRA submissions in the ENA

Usage

```
enaSubmission(accs, batchsize = 100)
```

Arguments

accs	a vector of SRA submissions with prefix SRA, ERA, DRA, etc, or a range of accessions
batchsize	number of accs to inlcude in a single comma-separated url string

Details

Parses the submission date attribute in the submission tag

Value

a data.frame with acc number, title and submitted date

Author(s)

Chris Stubben

References

http://www.ebi.ac.uk/ena/about/browser#sra_xml

See Also

sra and enaSRA

Examples

```
#compare to http://www.ebi.ac.uk/ena/data/view/ERA000746
enaSubmission("ERA000746")
# or ranges
# enaSubmission("SRA000600-SRA000610")

# chimps
pan<-enaSRA(9596)
head(pan)
enaSubmission(pan$submission)
```

enaTaxonomy *ENA taxonomy statistics*

Description

The number of linked records and total size in the taxonomy portal view at the European Nucleotide Archive (ENA)

Usage

```
enaTaxonomy(tax, h = TRUE, round = 0)
```

Arguments

tax	a taxonomy ID or name
h	return bases in human-readable format
round	number of digits to round bases

Value

a data.frame listing direct and subtree records in eight data classes: Assembled Nucleotide Sequences (release), Annotated Nucleotide Sequence update (std_update), Whole Genome Shotgun Sequence update (wgs_update), Genomic Contig Sequence update (con_update), Protein-coding Sequences (cds), Trace Archive (trace), SRA samples (sra_sample) and Projects (project).

Note

The ENA urls require a taxonomy ID and therefore searching by a taxonomy name will be slower since a separate query to the NCBI taxonomy database is needed.

Author(s)

Chris Stubben

References

see http://www.ebi.ac.uk/ena/about/browser#taxonomy_portal_options for details

See Also

ncbiTaxonomy

Examples

```
# COMPARE to http://www.ebi.ac.uk/ena/data/view/display=html&Taxon:2
enaTaxonomy("Bacteria")
# common names
enaTaxonomy("human")
# root
enaTaxonomy(1)
```

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

```
## top 20 Genus by year
data(lproks)
z<-table2(genus(lproks$name), year(lproks$released), n=20)
 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. A few manual updates were also added: 725 missing released dates from GenBank assemblies were added, 178 complete genomes with assembly released dates were corrected (see `complete`), and genome size outliers were removed.

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).

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=.7, lbty='n')
```

`ncbiGenome` *NCBI Genome Database*

Description

Search Entrez Genome at NCBI and retrieve genomes or linked neighbors in the Nucleotide database

Usage

```
ncbiGenome(term, neighbors=FALSE, derived = TRUE, fulltable = FALSE)
```

Arguments

`term`	Any valid combination of Entrez search terms
`neighbors`	Search for genome neighbors in Entrez Nucleotide
`derived`	If searching neighbors, also include the GenBank assembly that the Reference sequence was derived from
`fulltable`	Return all summary fields

Details

Returns summaries in Entrez Genome or if neighbor is TRUE, then links to Entrez Nucleotide using genome_nuccore_samespecies (Other genomes for species) and genome_nuccore (Assembly). See `ncbiInfo("genome", "links")`

Value

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

Note

Viral sequences typically have only one reference sequence per species in Entrez genome, 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/.

Examples

```
ncbiGenome('Nipah virus[orgn]')
x<-ncbiGenome('Nipah virus[ORGN]', neighbors = TRUE)
x[1:2,]
summary(x)
```

`ncbiInfo` *NCBI Entrez databases and fields*

Description

Return a list all Entrez databases or the indexing fields and available links for a single database

Usage

```
ncbiInfo(db, list = "field")
```

Arguments

`db`	a valid Entrez database , if missing then all databases are listed
`list`	list database fields or links

Details

Runs EInfo and parses XML results

Value

A data.frame listing databases, fields, or links

Author(s)

Chris Stubben

References

http://www.ncbi.nlm.nih.gov/books/NBK25499

Examples

```
ncbiInfo()
ncbiInfo("bioproject")
ncbiInfo("bioproject", "link")
```

`ncbiNucleotide` *NCBI Nucleotide database*

Description

Search Entrez Nucleotide at NCBI and retrieve summary tables

Usage

```
ncbiNucleotide(term, fulltable = FALSE)
```

Arguments

`term`	Any valid combination of Entrez search terms or a vector of accessions numbers
`fulltable`	Return all summary fields

Details

Returns a summary from Entrez Nucleotide.

Value

A genomes data frame with acc, name, released, taxid, size and gi.

Author(s)

Chris Stubben

References

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

See Also

ncbiGenome

Examples

```
ncbiNucleotide("AL117189,AL109969,AL117211")
# Exclude Patents and Refseq which are usually duplicates
marb <- ncbiNucleotide( "Marburgvirus[ORGN] NOT gbdiv_pat[PROP] NOT srcdb_refseq[PROP]")
marb
# two peaks in size distribution (partial and complete sequences)
hist(marb$size, col="blue", br=30, main="Marburg virus sequences", xlab="Length (bp)")

# Compare to NCBI Genomes (1 reference and 19 neighbors= 20)
marbg <-ncbiGenome("Marburgvirus[ORGN]",  neighbor=TRUE)
# Remove "nucleoprotein (NP)... genes" from 3 long deflines for display
marb$name <- gsub("(.*)( nucleoprotein.*)(, complete.*)", "\1\3",  marb$name)
# 13 genomes out of 33 missing links to Entrez Genome
data.frame(subset(marb, size > 16000 & !acc %in% marbg$acc))
```

ncbiProject *NCBI BioProject database*

Description

Search the Entrez BioProject (Genome Project) at NCBI and retrieve a project summary table

Usage

```
ncbiProject(term, refseq = TRUE)
```

Arguments

term	any valid combination of Entrez search terms
refseq	include RefSeq and Overview projects, if false then only primary submissions excluding RefSeq.

Details

Searches the new BioProject database using the ESearch utility

Value

A genomes data frame with 32 summary fields columns

Author(s)

Chris Stubben

References

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

See Also

ncbiGenome

Examples

```
#ncbiProject("Pan[ORGN]")
x <- ncbiProject("Yersinia[ORGN]", refseq=FALSE)
x
t(x[2,])  #second row
summary(x)
```

`ncbiPubmed` *NCBI PubMed database*

Description

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

Usage

```
ncbiPubmed(term)
```

Arguments

`term`	Any valid combination of Entrez search terms or a vector of pubmed IDs

Details

The function uses either Epost (for numeric pubmed IDs) or Esearch to query the PubMed database and then parses the XML summary to return a short citation

Value

A data.frame with 9 columns

`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)

Author(s)

Chris Stubben

See Also

`pub` for complete microbial genome publications

Examples

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

`ncbiRelease` *NCBI revision history*

Description

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

Usage

```
ncbiRelease(ids, db="nuccore", common=TRUE, random=20)
```

Arguments

`ids`	A vector or comma-separated list of sequence accessions or GI numbers
`db`	Entrez sequence database to search, default nuccore
`common`	If replaced sequences are found, search for the earliest date in the common revision history
`random`	The number of replaced sequences to search

Details

Searches the revision history display and parses the line listing the date a sequence was *first seen at NCBI*. In some cases, a sequence replaces earlier IDs and if the `common` option is TRUE, the earliest date of the replaced sequences is returned instead. Also, since a sequence accession may replace 500 or more ids, a random sample of the replaced sequences will be checked.

Value

A data frame listing the accession, release date, and whether replaced sequences are found

Author(s)

Chris Stubben

Examples

```
#Yersinia pestis - 1 chromosome and 3 plasmids
ncbiRelease("AL590842,AL117189,AL109969,AL117211")
# or skip common revision history
ncbiRelease("AL590842", common=FALSE)
# Protein acc
ncbiRelease("CAA21395", db="protein")
```

`ncbiSubmit` *NCBI submission dates*

Description

Returns the date a sequence was submitted to NCBI using the Direct Submission line in the GenBank file

Usage

```
ncbiSubmit(term, db = "nuccore", retmax = 1000)
```

Arguments

`term`	Any valid combination of Entrez search terms or a vector of accessions numbers
`db`	Entrez sequence database to search, default nuccore
`retmax`	Number of records to return

Details

Searches an Entrez sequence database, downloads GenBank files and parses the JOURNAL line containing a submitted date, for example, `JOURNAL Submitted (03-SEP-1999) ...`.

Value

a data.frame with accession, definition, and submitted date

Note

If more than two submitted dates are found, then the earliest date is returned

Author(s)

Chris Stubben

See Also

`ncbiRelease`)

Examples

```
#Yersinia pestis in Genome database
ncbiSubmit("Yersinia pestis CO92[ORGN]", db="genome")

# Virus in nucleotide database
ebola<- ncbiSubmit("Ebolavirus[ORGN] NOT gbdiv_pat[PROP] NOT refseq[FILTER]")
head(ebola)
# a few early submissions may be missing
subset(ebola, is.na(submitted))
table(year(ebola$submit))
```

ncbiTaxonomy *NCBI taxonomy database*

Description

Search the Entrez taxonomy database at NCBI and return summaries using Esummary or Efetch

Usage

```
ncbiTaxonomy(term, results = "summary", full = FALSE)
```

Arguments

term	either a valid Entrez search term or a vector of taxonomy IDs (passed to Epost) or names (joined using OR)
results	return a summary using Esummary (default) or lineage using Efetch
full	return all 14 Esummary fields, default is 9

Details

The function uses either Epost (for numeric taxonomy IDs) or Esearch to query the taxonomy database and then passes the results to either Esummary or Efetch. Esummary returns the id, name, rank, division and number of linked records in various Entrez databses. Efetch returns the id, name, rank, lineage and date.

Value

a data.frame

Author(s)

Chris Stubben

References

NCBI taxonomy database http://www.ncbi.nlm.nih.gov/sites/entrez?db=taxonomy

See Also

ncbiInfo for a list of fields in the taxonomy database

Examples

```
ncbiTaxonomy("cellular organisms[Next Level]")
# new Hantavirus species in 2011
ncbiTaxonomy("Hantavirus[subtree] AND 2011[date] AND species[rank]")
# difference between summary and lineage (Efetch) results
ncbiTaxonomy (1000587)
ncbiTaxonomy (1000587, "lineage")
```

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,  curdate=TRUE, abbrev = TRUE, flip = NA,
 legend = "topleft", lbty = "o", lcol = 1, ltitle = NULL, lcex = 1,
 lsort = TRUE, cex = 1, inset=0, 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	plot a single line with labeled points using genome name column
curdate	include the current date on x-axis, if false, then default is range of release dates
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
inset	inset legend distances(s)
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=.7, lbty='n')
```

---

pub *Complete microbial genome publications*

Description

Complete microbial genome publications at NCBI

Usage

```
data(pub)
```

Format

A data frame with 1000 observations on the following 10 variables.

`pmid` PubMed id

`date` published date

`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 file was created by selecting 1160 complete microbial genomes with publications in the `lproks` table and downloading the unique citations using `ncbiPubmed`. The 113 genomes with two or more listed publications were checked to identify the likely genome paper from the list of comma-separated pubmed IDs (the genome paper was the first pubmed ID in 75 of the 113 projects). The published date was added by formatting the pubdate column, except for 237 papers with only a year listed - in these cases the artdate column was used.

Source

The lproks table at `http://www.ncbi.nlm.nih.gov/genomes/lproks.cgi`

Examples

```
data(pub)
pub[1:2,]
z<-table2(pub$journal, pub$year, n=15)
image2(z[,-ncol(z)], sort=TRUE, mar=c(1,10,3,1), cex=.8, log=TRUE)
```

`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))
```

---

sra *Microbial SRA samples at the ENA*

Description

Next-generation sequencing projects from microbes in the Sequence Read Archive (SRA) at the European Nucleotide Archive (ENA).

Usage

```
data(sra)
```

Format

A data frame with 18279 observations on the following 13 variables.

`taxid` taxonomy id

`name` scientific name (if missing, then title)

`alias` name qualifier from alias attribute

`sample` SRA sample

`submission` SRA submission

`study` SRA study

`experiment` SRA experiment

`center` sequencing center

`bases` number of bases

`reads` number of reads

`submit` submission date

`model` model of sequencer

`type` study type

Details

Downloaded from ENA on Oct 27, 2011. Created by joining `enaSRA("Bacteria")` and `enaSRA("Archaea")` and adding submission dates using `enaSubmission`, model using `enaExperiment` and study type using `enaStudy`. Microbes represent ~6% of the total bases in the SRA.

Source

SRA sample portal at ENA

Examples

```
data(sra)

table2(species( sra$name))
table2(sra$center)
table2(sra$model)
table2(sra$study)

#Average read lengths by model
data.frame(read=round(tapply(sra$bases/sra$reads, list(sra$model ), mean, na.rm=TRUE), 1)

# image plot by model and year
y <- tapply(sra$bases, list(sra$model, year( sra$submit ) ), sum, na.rm=TRUE)
image2( y / 1e9, mar=c(1,11, 4,1) , log=TRUE, round=1)
title("Total microbial bases submitted per year (billions)", cex.main=1, line=2)
```

---

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"))
```

---

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 `neighbor` option in the `ncbiGenome` 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 (and exclude Genbank acc that RefSeq was derived from)
ncbiGenome("Monkeypox virus[orgn]", neighbor=TRUE, derived=FALSE)

## 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