Package 'genomes'

April 10, 2015

Type Package

Title Genome sequencing project metadata

Version 2.12.0

Date 2014-09-04

Author Chris Stubben

Maintainer Chris Stubben <stubben@lanl.gov>

License Artistic-2.0

Depends R (>= 2.11), XML, RCurl, GenomicRanges, IRanges, Biostrings

biocViews Annotation, Genetics

Description Collects genome sequencing project data from NCBI

R topics documented:

`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(proks)
doublingTime(proks)
doublingTime(proks, status == Contig, time=months)
```

---

efetch *Entrez database downloads*

Description

Retrieve Entrez database records at NCBI in a variety of formats

Usage

```
efetch(id, db = "pubmed", rettype = "", retmode = "text", showURL = FALSE, destfile, ...)
```

Arguments

id	An EntrezHistory object or vector of Ids
db	An Entrez database, default pubmed
rettype	Retrieval type, see note for details
retmode	Retrieval mode, see note for details
showURL	display URL string
destfile	location to save downloaded file using download.file. If missing, the url is loaded into R using readLines
...	Other key-value pairs passed to the efetch url string, e.g seq_stop

Value

A character vector for the given retrieval type and mode.

Note

See Table 1 http://www.ncbi.nlm.nih.gov/books/NBK25499/table/chapter4.chapter4_table1 for a list of valid retrieval types and modes.

If EntrezHistory results are the input, then the database listed in that object is used. If using a vector of Ids, the database option must be included. Also, do not pass more than 200 Ids to the url (use the History or see the NCBI help pages for other suggestions).

Author(s)

Chris Stubben

References

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

Examples

```
## Not run:
# abstracts from recent bioC articles - use ids to limit the number
x <- esearch("bioconductor[TITLE]", usehistory="n", retmax=5, reldate=360 )
x
efetch(x, rettype="abstract")
# only first 500 bases
efetch( esearch( "Yersinia pestis CO92[ORGN] AND refseq[FILTER] AND plasmid[Filter]", "nuccore"), rettype="fasta",
efetch(16082679, "nuccore", "fasta")

## End(Not run)
```

---

einfo *Entrez database information*

Description

List all Entrez databases at NCBI or the indexing fields and available links for a specific database

Usage

```
einfo(db, links=FALSE)
```

Arguments

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

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

```
## Not run:
einfo()
einfo("bioproject")
einfo("bioproject", TRUE)

## End(Not run)
```

---

elink *Entrez database links*

Description

Find links between Entrez databases at NCBI

Usage

```
elink(id, cmd = "neighbor_history", parse = TRUE, showURL = FALSE, ...)
```

Arguments

id	An EntrezHistory object or vector of Ids
cmd	Command mode
parse	Parse results into an EntrezHistory object (default) or vector of linked Ids (if cmd="neighbor"). All other cmd options return XML
showURL	display URL string
...	Other key-value pairs such as dbfrom, db, linkname passed to the elink url string

Details

See einfo to find available links

Value

Same as esearch

Note

If EntrezHistory results are the input, then the database listed in that object is used as the dbfrom key. Some additional checks are needed to catch timeout and other errors returned by the NCBI servers.

Author(s)

Chris Stubben

References

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

Examples

```
## Not run:
elink("15718680,157427902", dbfrom="protein", db="gene")
elink("15718680,157427902", dbfrom="protein", db="gene", cmd="neighbor")

# list linknames
einfo("genome", TRUE)[, 1:2]
x <- esearch("Nipah virus", "genome")
# dbfrom is set to "genome" and default link is "genome_nuccore"
y <- elink(x, db="nuccore")
y
# Links to reference AND genbank sequence the reference was derived from
esummary(y)
# OR link to Other genomes for Species
esummary( elink(x, db="nuccore", linkname="genome_nuccore_samespecies"))

## End(Not run)
```

---

esearch *Entrez database search*

Description

Search Entrez databases at NCBI

Usage

```
esearch(term, db = "pubmed", usehistory = "y", parse = TRUE, verbose=TRUE, showURL=FALSE, ...)
```

Arguments

term	Any valid combination of Entrez search terms or a vector of accessions
db	An Entrez database, default pubmed
usehistory	Save results to History server for subsequent calls
parse	If false, the XML output is returned
verbose	Print number of results found
showURL	Print url string
...	Other key-value pairs passed to esearch url string

Details

See `einfo()` for a list of valid Entrez database names and search fields. If usehistory="n", the default number of ids returned is 20 (set a retmax option to increase the default limit). If a vector of accessions are input, the terms are pasted together in a comma-separated list for searching by Primary Acession.

Value

Either an EntrezHistory data.frame listing the database, query_key and WebEnv (default), a vector of Ids if usehistory="n", or the raw XML output if parse=FALSE. The default EntrezHistory object may be passed directly to the other E-utilities.

Author(s)

Chris Stubben

References

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

Examples

```
## Not run:
# EntrezHistory object
esearch("bioconductor[TITLE]", showURL=TRUE)
# taxonomy IDs
esearch("mouse", db="taxonomy", usehistory="n")
esearch("AE017223 OR ACBJ00000000", db="nuccore")
# comma-separated (or vector) to search Primary accessions
esummary( esearch("AE017223,ACBJ00000000", db="nuccore"))

## End(Not run)
```

---

esummary *Entrez database summaries*

Description

Summaries of Entrez database records at NCBI

Usage

```
esummary(id, db = "pubmed", parse = TRUE, ...)
```

Arguments

id	An EntrezHistory object or vector of Ids
db	An Entrez database, default pubmed
parse	Parse the XML results into a data.frame
...	Other key-value pairs passed to the esummary url string

Value

A data.frame or XML results if parse=FALSE

Note

If EntrezHistory results are the input, then the database listed in that object is used. If using a vector of Ids, the database option must be included. Also, do not pass more than 200 Ids to the url (use the History or see the NCBI help pages for other suggestions).

Some records may be missing fields and then constructing a data.frame will return warnings. For example, the DOI field is missing in many Pubmed records. You can also set the version="2.0" to return the version 2.0 ESummary XML.

Author(s)

Chris Stubben

References

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

Examples

```
## Not run:
# BioC articles published in the last year
x <- esearch("bioconductor[TITLE]", reldate=360)
y <- esummary(x, version="2.0")
y[, c(1, 42, 6, 3, 8, 10)]

# Y. pestis CO92 refseqs
x <- esearch( "Yersinia pestis CO92[ORGN] AND refseq[FILTER]", "nuccore")
y <- esummary(x)
y[, c(2,3,5,10)]
# Taxonomy database
esummary(esearch("Mouse[Subtree]", db="taxonomy"))

## End(Not run)
```

euks *Eukaryotic genomes at NCBI*

Description

Eukaryotic genome sequencing projects at NCBI

Usage

```
data(euks)
```

Format

A genomes data frame with observations on the following 21 variables.

`acc` BioProject id

`name` Organism name

`status` Sequencing status

`released` First public sequence release

`taxid` Taxonomy id

`acc` BioProject Accession number

`group` Phylum

`subgroup` Class level

`size` Total length of DNA (Mb)

`gc` Percent GC (guanine or cytosine)

`assembly` Name of the genome assembly (from NCBI Assembly database)

`chromosomes` Number of chromosomes

`organelles` Number of organelles

`plasmids` Number of plasmids

`wgs` Four-letter Accession prefix followed by version

`scaffolds` Number of scaffolds

`genes` Number of genes

`proteins` Number of proteins

`modified` Last modification date

`center` Sequencing center

`biosample` BioSample Accession number

Details

Excludes projects that represent only organelles

Source

downloaded from ftp.ncbi.nlm.nih.gov/genomes/GENOME_REPORTS/eukaryotes.txt

Examples

```
data(euks)
euks
t(euks[1,])
plot(euks)
summary(euks)
table2(euks$subgroup)
```

---

ftpList *List FTP files and directories*

Description

List FTP files and directories from NCBI and other hosts

Usage

```
ftpList(ftp, fileonly = FALSE)
```

Arguments

ftp	ftp directory
fileonly	only list files

Value

a data.frame

Author(s)

Chris Stubben

Examples

```
## Not run:
 # all Y. pestis files
 ftp<- "ftp.ncbi.nih.gov/genomes/Bacteria/Yersinia_pestis_CO92_uid57621"
 x<-ftpList(ftp)
 x
#all genome directories
 ftp<-"ftp.ncbi.nih.gov/genomes/Bacteria"
 x <- ftpList(ftp)
```

```
## End(Not run)
```

---

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(proks)
proks
summary(proks)
plot(proks)
## Not run: update(proks)
```

---

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(proks)
data(euks)

plot(proks, log=y, las=1, lty=3)
lines(euks, col="red", lty=2)
```

---

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, 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, default is date column name
ylab	y-axis label
type	type of plot, default is a blue line
col	color
...	additional arguments passed to plot

Details

Requires a released, created or submitted date column (and plots first column found)

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(proks)
plot(proks)
plot(proks, subset = name %like% Yersinia*, ylab="Yersinia genomes")
```

---

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(euks)
summary(euks)
summary(euks, 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(proks)
## Not run: update(proks)

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

---

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(proks)
x <- table2(genus(proks$name))[1:10,]
dotchart(rev(x), xlab="Genomes", pch=16)
```

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(proks)
z<-table2(genus(proks$name), year(proks$released), n=20)
 image2(z[,-ncol(z)], sort=TRUE, mar=c(1,10,3,1), cex=.8)
```

---

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

---

ncbiGenome *NCBI Genome links to the Nucleotide database*

Description

Search Entrez Genome at NCBI and retrieves linked genomes in the Nucleotide database

Usage

```
ncbiGenome(term, refseq=FALSE)
```

Arguments

term	Any valid combination of Entrez search terms
refseq	Include RefSeq genomes, default is GenBank submissions

Details

Searches Entrez Genome and finds linked sequences in Entrez Nucleotide using genome_nuccore (Assembly) and then finds related sequences using nuccore_nuccore_samespecies_rsgb (Other INSDC Genome Sequences). The genome_nuccore link includes the Reference and Genbank acc that Reference was derived from (and refseq option is used to exclude duplicate RefSeq from results).

Value

A genomes data frame with acc, name, created, taxid, size, gi and other fields.

Author(s)

Chris Stubben

References

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

Examples

```
## Not run:
ncbiGenome(Nipah virus[orgn])
ncbiGenome(Nipah virus[orgn], refseq=TRUE)

## End(Not run)
```

---

ncbiNucleotide *NCBI Nucleotide database*

Description

Search Entrez Nucleotide at NCBI and retrieve summary tables

Usage

```
ncbiNucleotide(term)
```

Arguments

term	Any valid combination of Entrez search terms or a vector of accessions numbers

Details

Returns a summary from Entrez Nucleotide.

Value

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

Author(s)

Chris Stubben

References

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

See Also

ncbiGenome

Examples

```
## Not run:
ncbiNucleotide("AL117189,AL109969,AL117211")[,1:6]

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

## End(Not run)
```

`ncbiProject` *NCBI BioProject database*

Description

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

Usage

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

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

```
## Not run:
x <- ncbiProject("Yersinia[ORGN]")
x
summary(x)

#Metagenomes
metag <- ncbiProject("metagenome[Project Data Type]")
metag2 <- ncbiProject("metagenomes[Orgn]")

## End(Not run)
```

---

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, abstract = FALSE)
```

Arguments

term	Any valid combination of Entrez search terms or a vector of pubmed IDs
abstract	Include abstract in result table, default FALSE

Details

The function searches the PubMed database and parses the efetch XML summary to return a short citation

Value

A data.frame with 9 or 10 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)
abstract	abstract

Author(s)

Chris Stubben

Examples

```
## Not run:
ncbiPubmed( c(7542800, 7569993))
# OR ncbiPubmed("7542800,7569993")

## End(Not run)
```

---

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

```
## Not run:
#Yersinia pestis - 1 chromosome and 3 plasmids
ncbiRelease("AL590842,AL117189,AL109969,AL117211")
# or skip common revision history
ncbiRelease("AL590842", common=FALSE)

## End(Not run)
# 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")
```

Arguments

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

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. This script uses E-fetch, so retrievals to the genome and other database will not work.

Author(s)

Chris Stubben

See Also

ncbiRelease

Examples

```
## Not run:
#Yersinia pestis  reference sequences
ncbiSubmit("Yersinia pestis CO92[ORGN] AND refseq[FILTER]")
# Ebola virus - no patents or references
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))

## End(Not run)
```

---

ncbiTaxonomy *NCBI taxonomy database*

Description

Search the Entrez taxonomy database at NCBI

Usage

```
ncbiTaxonomy(term, summary=TRUE)
```

Arguments

term	either a valid Entrez search term or a vector of taxonomy Ids or names
summary	return results using Esummary (default) or Efetch

Details

This function uses either Esummary or Efetch to return taxonomy data from NCBI. The Efetch XML include parent ids and lineage tags not found in Esummary XML. The term may be also be a vector of taxonomy Ids (joined using a comma) or taxonomy names (joined using "OR").

Value

a data.frame

Author(s)

Chris Stubben

References

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

See Also

einfo for a list of fields in the taxonomy database.

Examples

```
## Not run:
ncbiTaxonomy("Yersinia pestis")
ncbiTaxonomy("Yersinia pestis", summary=FALSE)
ncbiTaxonomy(c("Bacillus anthracis", "Yersinia pestis"))
ncbiTaxonomy("cellular organisms[Next Level]")
# new Hantavirus species added in 2012
ncbiTaxonomy("Hantavirus[subtree] AND 2012[date] AND species[rank]")

# can also use Lineage field with esummary
ncbiTaxonomy("Necocli virus[Lineage]")
# compare to efetch results
ncbiTaxonomy (1145238, FALSE)

## End(Not run)
```

---

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(proks)
# default group is status
plotby(proks, top=2)

## groupby can be a vector
plotby(proks, genus(proks$name), log=y, lcex=.7)

# OR plot labels
plotby(proks, subset=name %like% Haemophilus influenzae*, labels=TRUE, cex=.7, lbty=n)
```

---

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(proks)`)

Author(s)

Chris Stubben

Examples

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

---

proks *Prokaryotic genomes at NCBI*

Description

Prokaryotic genome sequencing projects at NCBI.

Usage

```
data(proks)
```

Format

A genomes data frame with observations on the following 23 variables.

`pid` BioProject id
`name` Organism name
`status` Sequencing status
`released` First public sequence release
`taxid` Taxonomy id
`acc` BioProject Accession number
`group` Phylum
`subgroup` Class level
`size` Total length of DNA (Mb)
`gc` Percent GC (guanine or cytosine)
`refseq` Refseq chromosome sequence accessions
`insdc` GenBank chromosome sequence accessions
`plasmid.refseq` Refseq plasmid sequence accessions
`plasmid.insdc` GenBank plasmid sequence accessions
`wgs` Four-letter WGS Accession prefix followed by version
`scaffolds` Number of scaffolds/contigs
`genes` Number of genes
`proteins` Number of proteins
`modified` Last modification date
`center` Sequencing center
`biosample` BioSample Accession number
`assembly` Assembly Accession number
`reference` Reference or representative genome

Details

BioProject IDs are no longer unique and the table was modified on Nov 1, 2013 to include BioSample and Assembly accessions. See email on NCBI announcement regarding bacterial strain-level TaxID management for details

Source

downloaded from ftp.ncbi.nlm.nih.gov/genomes/GENOME_REPORTS/prokaryotes.txt

Examples

```
data(proks)
proks
#single row
t(proks[1,])
class(proks)
attributes(proks)[c("date","url")]
summary(proks)
## check for missing release dates
table2(proks$status,!is.na(proks$wgs), dnn=list("Status", "Has WGS acc?"))
plot(proks)
plotby(proks, log=y, las=1, top=2)
hist(proks$size[proks$size<15], br=50, main="", col="blue", xlab="Size (Mb)")

## download recent table from NCBI
## Not run: update(proks)
```

---

read.genemark *Read a GeneMark output file*

Description

Read a GeneMark HMM version 2.6 file from NCBI (version 3)

Usage

```
read.genemark(file)
```

Arguments

file	GeneMark HMM file

Details

GeneMark HMM files are available from the NCBI genomes ftp directory, ftp://ftp.ncbi.nih.gov/genomes.

Value

GRanges with 2 elementMetadata columns: id and class.

Note

Two GeneMark predictions are available from teh NCBI genomes ftp. This function currently reads the HMM version 2.6 files only

Author(s)

Chris Stubben

References

see http://exon.gatech.edu for details about GeneMark

See Also

read.ncbi.ftp

Examples

```
file <- "ftp://ftp.ncbi.nih.gov/genomes/Bacteria/Yersinia_pestis_CO92_uid57621/NC_003132.GeneMarkHMM-2.6r"
 x <-read.genemark(file)
 x
 metadata(x)
```

---

read.gff *Read a GFF file from NCBI*

Description

Read a GFF file from NCBI genomes ftp (version 3)

Usage

```
read.gff(file,  locus.tags = TRUE, nrows = -1)
```

Arguments

file	a GFF file
locus.tags	only return genes with locus tags
nrows	number of rows to read

Details

GFF files are available from the NCBI genomes ftp directory, ftp://ftp.ncbi.nih.gov/genomes.

Value

GRanges with 4 elementMetadata columns: locus, feature, description and gene name. If all rows are returned (`locus.tags=FALSE`), then score, phase and tags are included. The seqid and source are saved in metadata.

Note

By default, the GFF file is parsed to return only features with locus_tag keys. Gene types, products and names are assigned from child records by matching Parent tags.

The function is intended to load GFF files from NCBI only. GFF files from other sources have not been tested and may not parse.

Author(s)

Chris Stubben

References

see http://www.sequenceontology.org/gff3.shtml for details about Generic Feature Format

See Also

read.ncbi.ftp

Examples

```
file<-"ftp://ftp.ncbi.nih.gov/genomes/Bacteria/Yersinia_pestis_CO92_uid57621/NC_003132.gff"
x <-read.gff(file)
x
metadata(x)
```

`read.glimmer` *Read a Glimmer output file*

Description

Read a Glimmer3 gene output file from NCBI

Usage

```
read.glimmer(file)
```

Arguments

`file`	Glimmer3 file

Details

Glimmer files are available from the NCBI genomes ftp directory, ftp://ftp.ncbi.nih.gov/genomes.

Value

GRanges with 3 elementMetadata columns: id, frame and score

Author(s)

Chris Stubben

References

Details about Glimmer3 are available at http://www.cbcb.umd.edu/software/glimmer

See Also

read.ncbi.ftp

Examples

```
file<-"ftp://ftp.ncbi.nih.gov/genomes/Bacteria/Yersinia_pestis_CO92_uid57621/NC_003132.Glimmer3"
x <-read.glimmer(file)
x
metadata(x)
table(values(x)$frame)
```

---

read.ncbi.ftp *Read files from the NCBI genomes FTP*

Description

Read files from the NCBI genomes FTP

Usage

```
read.ncbi.ftp(org, filePattern = "ptt$|rnt$", ftp = "genomes/ASSEMBLY_BACTERIA", ...)
```

Arguments

org	organism directory (new FTP requires species and refseq assembly)
filePattern	load files matching a specific pattern, default is protein and rna tables
ftp	name of base FTP directory
...	other options passed to read functions

Details

This function reads files in the genomes FTP and loads sequence files (faa=protein, fna=genone, ffn=gene, frn=rna) using Biostring functions or converts coordinate files (gff, ptt, rnt, GeneMarkHMM, Glimmer, Prodigal) to GRanges

Value

a Biostring or GRanges object

Note

The genomes FTP site was updated on Aug 2014 and the default ftp is now the new location. Please use ftp = "genomes/Bacteria" for the old genomes FTP. This does not read asn, gbk, val, GeneMark 2.5 and rpt files.

Author(s)

Chris Stubben

See Also

read.gff, read.ptt, read.genemark, read.glimmer, read.prodigal

Examples

```
## Not run:

#NEW FTP uses species and RefSeq assembly

org <- "Burkholderia_pseudomallei/GCF_000011545"
read.ncbi.ftp(org)         # Protein and rna tables
read.ncbi.ftp(org, "gff")  # GFF
read.ncbi.ftp(org, "fna")  # Genome sequences

# OLD genomes FTP uses strain and reseq project ID
# ALSO includes prodigal, genemark and glimmer predictions (not on new site)

org <- "Burkholderia_pseudomallei_K96243_uid57733"
read.ncbi.ftp(org, "Prod", ftp = "genomes/Bacteria") # Prodigal annotations
read.ncbi.ftp(org, "Glim", ftp = "genomes/Bacteria") # Glimmer


## End(Not run)
```

`read.prodigal` *Read a Prodigal gene finding output file*

Description

Read a gff formatted Prodigal gene output file from NCBI (version 2.5)

Usage

```
read.prodigal(file, allScores = FALSE)
```

Arguments

`file`	Prodigal gff output file
`allScores`	include all scores

Details

Prodigal output files are available from the NCBI genomes ftp directory, `ftp://ftp.ncbi.nih.gov/genomes`.

Value

GRanges with 7 elementMetadata columns: id, partial flag for genes continuing off the edge of a contig, start codon, RBS motif, RBS spacer, coding potential/score and start score.

If allScores is TRUE, then four additional score columns are included: total score (sum of coding and start score) and RBS motif score, upstream region score, and codon type score (which usually sum to start score). See the README file in the Prodigal distribution for complete details.

Author(s)

Chris Stubben

References

Prodigal is a microbial gene finding program developed at University of Tennessee and Oak Ridge National Laboratory. See `http://prodigal.ornl.gov` for details

See Also

`read.ncbi.ftp`

Examples

```
file<-"ftp://ftp.ncbi.nih.gov/genomes/Bacteria/Yersinia_pestis_CO92_uid57621/NC_003143.Prodigal-2.50"
x <-read.prodigal(file)
x
metadata(x)
table2(values(x)$start_type)
table2(values(x)$rbs_motif)
hist(values(x)$sscore, br=40, col="blue", main="", xlab="Start score")
```

`read.ptt` *Read a NCBI protein or RNA feature table*

Description

Read a protein or RNA table from NCBI genomes ftp.

Usage

```
read.ptt(file)
```

Arguments

`file`	a protein table

Details

Protein and RNA table (*.ptt and */rnt) are available in the NCBI genomes ftp directory at ftp://ftp.ncbi.nih.gov/genomes

Value

GRanges with 6 elementMetadata columns including locus tag id, length (aa), genbank ID, gene name, cog id and product.

Note

Protein tables downloaded from Entrez Genome overview pages have a different format

Author(s)

Chris Stubben

See Also

read.ptt

Examples

```
file<-"ftp://ftp.ncbi.nih.gov/genomes/Bacteria/Yersinia_pestis_CO92_uid57621/NC_003143.ptt"
x <-read.ptt(file)
x
table2(substr(values(x)$cog, 1,7), n=6)
```

`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(proks)
x <- table2(species(proks$name))[1:10,]
dotchart(rev(x), xlab="Genomes", pch=16)
## abbreviate genus name
x <- subset(proks, name %like% Bacillus*)
x <- table2(species(x$name))[1:10, ]
names(x) <- species(names(x), TRUE)
dotchart(rev(x), xlab=expression(italic(Bacillus) ~ genomes), pch=16)
```

---

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(euks)
table(euks$subgroup)
table2(euks$subgroup)
## to display all rows, use NA or a large number...
table2(euks$subgroup, n=100)
# 2-d table
table2(euks$group, year(euks$released))
```

`virus` *Virus genomes at NCBI*

Description

Viral reference genome sequencing projects at NCBI

Usage

```
data(virus)
```

Format

A genomes data frame with the following 13 variables.

`acc` BioProject id

`name` Organism name

`status` Sequencing status

`released` First public sequence release

`taxid` Taxonomy id

`acc` BioProject Accession number

`group` Phylum

`subgroup` Class level

`size` Total length of DNA (Mb)

`gc` Percent GC (guanine or cytosine)

`host` Natural host of a virus

`segments` Number of segments

`genes` Number of genes

`proteins` Number of proteins

`modified` Sequence modification date

Details

Includes only data represented in the RefSeq dataset.

Source

downloaded from ftp.ncbi.nlm.nih.gov/genomes/GENOME_REPORTS/viruses.txt

Examples

```
data(virus)
plot(virus)
summary(virus)
table2(virus$host)
## most common phages
table2(species(grep("phage", virus$name, value=TRUE)))
## Not run:
# TABLE only includes RefSeq genomes - see ncbiGenome for links
subset(virus, name=="Nipah virus")
ncbiGenome(Nipah virus[ORGN])

## End(Not run)
```

---

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(proks)
table(year(proks$released))
```

Index