Package 'palmid'

October 15, 2021

Title RdRP Analysis Suite

Description R Analysis suite for viral RNA dependent RNA polymerase (RdRP). Statistical and meta-data analysis of 'palmscan' output and 'palmDB'/ 'DIAMOND' alignment files. Cross reference an input RNA virus against 145,000 RdRP identified in the Serratus project.

Version 0.0.3

URL https://serratus.io/palmid

BugReports https://github.com/ababaian/palmid/issues

License AGPL-3

Encoding UTF-8

LazyData true

RoxygenNote 7.1.2

Suggests sf, rnaturalearth, rnaturalearthdata

Imports dbplyr, downloadthis, DBI, DT, ggplotify, ggwordcloud, ggExtra, grDevices, gridExtra (>= 2.3), htmltools, htmlwidgets, leaflet, methods, plotly, rmarkdown, RPostgreSQL, scales, stats, utils, viridisLite

Depends R (>= 3.5.0), dplyr, ggplot2 (>= 3.3.3)

NeedsCompilation no

Author Artem Babaian [aut, cre]

Maintainer Artem Babaian <artem@rRNA.ca>

Repository CRAN

Date/Publication 2021-10-15 07:50:04 UTC

R topics documented:

`fev2df` *Convert a palmscan Field-Equals-Value (FEV) column into a dataframe*

Description

Convert a palmscan Field-Equals-Value (FEV) column into a dataframe

Usage

```
fev2df(fev.col)
```

Arguments

fev.col	character vector of FEV

Value

A 1-column data.frame

Examples

```
# fev.df <- as.data.frame( apply(fev.tsv, 2, fev2df) )
```

---

geoFilter *Conversion between run_ids and geo objects often contain NA/NULL values This removes NA-containing rows*

Description

Conversion between run_ids and geo objects often contain NA/NULL values This removes NA-containing rows

Usage

```
geoFilter(sra.geo, wobble = FALSE, wradius = 0.005)
```

Arguments

sra.geo	data.frame, output df from get.sraGeo()
wobble	boolean, add a wobble to each point to prevent overplotting [FALSE]
wradius	numeric, maximum magnitude of wobble [0.005]

Value

data.frame

`geoFilter2` *Conversion between run_ids and geo objects often contain NA/NULL values This removes NA-containing rows*

Description

Conversion between run_ids and geo objects often contain NA/NULL values This removes NA-containing rows

Usage

```
geoFilter2(palm.sra, wobble = FALSE, wradius = 0.005)
```

Arguments

`palm.sra`	data.frame, output df from get.sraGeo()
`wobble`	boolean, add a wobble to each point to prevent overplotting [FALSE]
`wradius`	numeric, maximum magnitude of wobble [0.005]

Value

modified palm.sra data.frame

Examples

```
data("waxsys.palm.sra")
# waxsys.palm.sra
# -- 4159 rows

waxsys.geo.filtered <- geoFilter2(waxsys.palm.sra)
# -- 2300 rows

# 2300 / 4159 SRA-libraries have associated geospatial meta-data
```

`get.palmSra` *A wrapper of several get* functions to create a palm.sra data.frame*

Description

A wrapper of several get* functions to create a palm.sra data.frame

Usage

```
get.palmSra(pro.df, con = SerratusConnect())
```

Arguments

pro.df	data.frame, imported diamond pro df. use get.pro()
con	pq-connection, use SerratusConnect()

Value

palm.sra data.frame

Examples

```
data("waxsys.pro.df")
con <- SerratusConnect()


get.palmSra( waxsys.pro.df, con )
```

---

get.proTax *A wrapper for get.tax() specific for 'pro.df' input and returns a populated the "tspe", "tfam", and "tphy" columns of 'pro.df' based on the "sseqid" column*

Description

A wrapper for get.tax() specific for 'pro.df' input and returns a populated the "tspe", "tfam", and "tphy" columns of 'pro.df' based on the "sseqid" column

Usage

```
get.proTax(pro.df, con = SerratusConnect())
```

Arguments

pro.df	data.frame, imported diamond pro df. use get.pro()
con	pq-connection, use SerratusConnect()

Value

pro.df data.frame

Examples

```
## Prepare data
# data("waxsys.pro.df")
# con <- SerratusConnect()

## Generate Report
# geoSRA <- PlotGeoReport( waxsys.pro.df )
```

`get.sOTU` *get.sOTU*

Description

Retrieve the parent sOTU for a 'palm_id' in palmdb or the set of all children 'palm_id' within an species/sOTU

Usage

```
get.sOTU(palm_ids, con, get_childs = FALSE, ordinal = FALSE)
```

Arguments

`palm_ids`	character, set of 'palm_id' to lookup in palmdb
`con`	pq-connection, use SerratusConnect()
`get_childs`	boolean, return all children 'palm_id' instead of parent sOTU [FALSE]
`ordinal`	boolean, return an ordered sOTU vector based on input 'palm_ids'

Value

character, unique 'palm_id' sOTU or sOTU-children

Examples

```
## R Code Example

con <- SerratusConnect()
get.sOTU(c("u1337"), con, get_childs = TRUE)


## Non-Running Example to demonstrate sOTU Relationships
# palm_id    sOTU
# u1         u3
# u2         u3
# u3         u3
```

```
# u4         u4

# Retrieve the parent sOTU for an input of palm_ids
# get.sOTU(c("u1","u2",u4"), con, get_childs = FALSE)
# -- returns c("u3","u4")

# Return an ordinal list of sOTU for iput
# get.sOTU(c("u2","u4","u2","u1"), con, ordinal = TRUE)
# -- returns c("u3", "u4", "u3", "u3")

# Return all children palm_id within an sOTU
# get.sOTU(c("u2"), con, get_childs = TRUE)
# -- returns c("u1", u2", "u3")
```

---

get.sra *get.sra*

Description

Retrieve the SRA runs containing a 'palm_id'-matching contig.

Usage

```
get.sra(palm_ids, con, ret_df = FALSE, ret_contig.df = FALSE, qc = TRUE)
```

Arguments

palm_ids	character or list, set of 'palm_id' to lookup in palmdb
con	pq-connection, use SerratusConnect()
ret_df	boolean, return a 'palm_id' 'run_id', 'coverage', 'qsequence' data.frame [F]
ret_contig.df	boolean, return a data.frame of matching contigs [F]
qc	boolean, require 85-percent palmprint coverage and e-value < 1e-6

Value

character, de-duplicated run_ids with a potential match for 'palm_ids'

Examples

```
palmid_of_interest <- c("u1337")
con <- SerratusConnect()

palm.sra   <- get.sra(palmid_of_interest, con)
```

get.sraBio *get.sraBio*

Description

Retrieve the "BioSample" field for a set of SRA 'run_id'

Usage

```
get.sraBio(run_ids, con, ordinal = FALSE)
```

Arguments

run_ids	character, SRA 'run_id'
con	pq-connection, use SerratusConnect()
ordinal	boolean, return 'run_ids' ordered vector [FALSE]

Value

data.frame, run_id, biosample character vectors

Examples

```
# SRA Library of interest
con <- SerratusConnect()
library.bioSample   <- get.sraBio( 'SRR9968562' , con)
```

get.sraDate *get.sraDate*

Description

Retrieve the "Load_Date" for a set of SRA 'run_id'

Usage

```
get.sraDate(run_ids, con, ordinal = FALSE, as.df = FALSE)
```

Arguments

run_ids	character, SRA 'run_id'
con	pq-connection, use SerratusConnect()
ordinal	boolean, return 'run_ids' ordered vector [F]
as.df	boolean, return run_id, date data.frame [F]

Value

POSIXct, date object vector

Examples

```
con <- SerratusConnect()
palm.date   <- get.sraDate("SRR9968562", con)
```

`get.sraGeo` *get.sraGeo*

Description

Retrieve the geo_coordinates for a set of SRA 'run_id'

Usage

```
get.sraGeo(run_ids = NULL, biosample_ids = NULL, con, ordinal = FALSE)
```

Arguments

`run_ids`	character, SRA 'run_id'
`biosample_ids`	character, BioSample 'biosample_id'
`con`	pq-connection, use SerratusConnect()
`ordinal`	boolean, return 'run_ids' ordered vector [F]

Value

data.frame, lon and lat numeric vectors

Examples

```
con <- SerratusConnect()
palm.geo   <- get.sraGeo(run_ids = "SRR9968562", con = con)
```

get.sraOrgn *get.sraOrgn*

Description

Retrieve the "scientific_name" for a set of SRA 'run_id'

Usage

```
get.sraOrgn(run_ids, con, ordinal = FALSE, as.df = FALSE)
```

Arguments

run_ids	character, SRA 'run_id'
con	pq-connection, use SerratusConnect()
ordinal	boolean, return 'run_ids' ordered vector [FALSE]
as.df	boolean, return run_id, date data.frame [FALSE]

Value

character, string vector

Examples

```
 # Retrive a single "scientific_name"
con <- SerratusConnect()
palm.orgn   <- get.sraOrgn('SRR9968562', con)

# Retrieve an ordered vector of "scientific_name"
data( waxsys.palm.sra)
waxsys_runs <- waxsys.palm.sra$run_id

waxsys_orgn <- get.sraOrgn(waxsys_runs, con, ordinal = TRUE)
```

get.tax *get.tax*

Description

Retrieve the taxonomic identifiers for a set of 'palm_id' for a given rank.

Usage

```
get.tax(palm_ids, con, rank = "family", ordinal = FALSE)
```

Arguments

palm_ids	character, set of 'palm_id' to lookup in palmdb
con	pq-connection, use SerratusConnect()
rank	character, taxonomic rank to retrieve. One of "species", "genus", "family" (Default), "phylum"
ordinal	boolean, return an ordered vector based on input 'palm_ids'

Value

character, unique 'tax_id' vector (i.e. "Coronaviridae")

Examples

```
con <- SerratusConnect()

# Return species-identifiers for a set of palmprints (uxxx)
get.tax(c("u2", "u1337"), con, rank = "species")
```

identityWordcount *identityWordcount*

Description

Create frequency-count table from a set of characters which are assigned a standardized rank-order scores from 10.0 to 1.0.

Usage

```
identityWordcount(orgn.df, ntop = 50)
```

Arguments

orgn.df	data.frame, scientific_name, pident
ntop	numeric, return only N top words [50]

Value

table, identity-count table with 100

---

linkBLAST *Parse an input sequence into a BLAST-able HTML link*

Description

Parse an input sequence into a BLAST-able HTML link

Usage

```
linkBLAST(header, aa.seq, label = "[BLAST]")
```

Arguments

header	character, header for blast search
aa.seq	character, query sequence (amino acid)
label	character, Display string["BLAST"]

Value

character, html link for click to search

Examples

```
blast.link <- linkBLAST("u1337", "MYAASTRING", "BLAST")
```

make_bg_data *Read a multiple-FEV file to create a background set of palmprints Standard approach is to use palmDB*

Description

Read a multiple-FEV file to create a background set of palmprints Standard approach is to use palmDB

Usage

```
make_bg_data(fev.path, dataset.id = NULL, return.data = TRUE)
```

Arguments

fev.path	Path to multiple fev file
dataset.id	Name for output dataset.
return.data	Boolean. Return data.frame instead of writing file [TRUE]

Value

NULL: will write an RData file to data/<fev.path>.RData

Examples

```
#' # palmscan example fev file
ps.fev.path <- system.file( "extdata", "waxsys.fev", package = "palmid")
example_bg <- make_bg_data(fev.path = ps.fev.path, dataset.id = 'example_bg')

## Documentation on Making Background Dataset from palmDB
##  i.e. load("palmdb")
##
## Download palmDB to make background set
# system("git clone https://github.com/rcedgar/palmdb.git")
#
## Generate the palmprint-FEV with palmscan
# system("palmscan -search_pp palmdb/2021-03-02/otu_centroids.fa \
#         -all -rdrp -fevout data/palmdb210302.fev")
#
## Create R object (data.frame)
# make_bg_data(fev.path = "data/palmdb210302.fev",
#              dataset.it = "palmdb",
#              return.data = FALSE)
#
# load("palmdb")
```

`normalizeWordcount` *normalizeWordCount Create frequency-count table from a set of characters which are normalized as percentage of total corpus*

Description

normalizeWordCount Create frequency-count table from a set of characters which are normalized as percentage of total corpus

Usage

```
normalizeWordcount(words, ntop = 50, logTwo = FALSE)
```

Arguments

`words`	character, description of vector [Default]
`ntop`	numeric, return only N top words [50]
`logTwo`	boolean, apply a log2 transformation [FALSE]

Value

table, frequency-count table

`palmdb` *palmdb*

Description

Palmprint database - initial version (210302) containing GenBank v241 RNA dependent RNA polymerase palmprints

Usage

```
data(palmdb)
```

Format

data.frame with 15016 obs of 18 variables

Details

#’

- score. palmscan PSSM motif score. 20+ is high confidence RdRP
- query. input fasta query name
- gene. One of "RdRP" or "RT"
- order. Order of catalytic motifs on input. "ABC" or "CAB"
- confidence. "high" or "low" confidence classification
- qlen. input query length. Is truncated on long input sequences
- pp_start. Start coordinate of palmprint within input sequence
- pp_end. End coordinate of palmprint within input sequence
- pp_length. Length of palmprint sequence (in AA)
- v1_length. Length of v1 region of palmprint
- v2_length. Length of v2 region of palmprint
- pssm_total_score. Raw total score of A,B,C motifs
- pssm_min_score. Lowest scoring PSSM score in palmprint
- motifs. Isolated A,B,C motifs from palmprint.
- super. Isolated super-motif residues from A,B,C (catalytic sites)
- group. Guess of sequence phylum.
- comments. Summary statement on palmprint QC.

Source

palmDB repo

`PlotDistro` *Plot a value relative to a background distribution from a palmprint data.frame.*

Description

Plot a value relative to a background distribution from a palmprint data.frame.

Usage

```
PlotDistro(
  pp,
  pp.bg,
  plotValue,
  distrocol = "skyblue",
  set.ylab = "palmDB density"
)
```

Arguments

pp	A palmprint.df row to use for the plot "value"
pp.bg	A multiple palmprint.df for the plot background "distribution"
plotValue	Which numeric column to use for pp and pp.bg. One of "score", "pssm_total_score", "pp_length", "v1_length", "v2_length"
distrocol	Color to use for distribution ["skyblue"]
set.ylab	Label for y-axis ["palmDB density"]

Value

A ggplot2 object

Examples

```
data("waxsys.palmprint")
data("palmdb")

PlotDistro( pp = waxsys.palmprint, pp.bg = palmdb, "score", "black")
PlotDistro( pp = waxsys.palmprint, pp.bg = palmdb, "pp_length", "skyblue")
```

PlotGeo *A multi-plot wrapper to convert a list of SRA 'run_ids' into a geographic world-map and timeline.*

Description

A multi-plot wrapper to convert a list of SRA 'run_ids' into a geographic world-map and timeline.

Usage

```
PlotGeo(run_ids, con = SerratusConnect())
```

Arguments

run_ids	character, vector of SRA run_id
con	pq-connection, use SerratusConnect()

Value

A grid-table object. Dimension standard is 800 x 600 px.

Examples

```
NULL
```

PlotGeo2 *Create a rich plotly geo map from a palm.sra data.frame*

Description

Create a rich plotly geo map from a palm.sra data.frame

Usage

```
PlotGeo2(palm.sra)
```

Arguments

palm.sra	data.frame, created with get.palmSra(pro.df, con)

Value

A plotly map

Examples

```
# Waxsystermes example data
data("waxsys.palm.sra")
geoSRA <- PlotGeo2( waxsys.palm.sra )
```

PlotGeoReport *A multi-plot wrapper to convert a list of SRA 'run_ids' into a geographic world-map and timeline.*

Description

A multi-plot wrapper to convert a list of SRA 'run_ids' into a geographic world-map and timeline.

Usage

```
PlotGeoReport(run_ids, con = SerratusConnect())
```

Arguments

run_ids	character, vector of SRA run_id
con	pq-connection, use SerratusConnect()

Value

A grid-table object. Dimension standard is 800 x 600 px.

PlotID *Plot Percent-identity vs. E-value of a pro file*

Description

Plot Percent-identity vs. E-value of a pro file

Usage

```
PlotID(pro, html = TRUE)
```

Arguments

pro	data.frame, A diamond-aligned pro file
html	boolean, include additional parsing for htmlwidget display [TRUE]

Value

A scatterplot as a ggplot2 object

Examples

```
data("waxsys.pro.df")
PlotID(waxsys.pro.df)
```

PlotLengths *A wrapper for PlotDistro() for "pp_length", "v1_length", "v2_length".*

Description

A wrapper for PlotDistro() for "pp_length", "v1_length", "v2_length".

Usage

```
PlotLengths(pp, pp.bg, set.ylab = "palmDB density")
```

Arguments

pp	A palmprint.df row to use for the plot "value"
pp.bg	A multiple palmprint.df for the plot background "distribution"
set.ylab	Label for y-axis ["palmDB density"]

Value

A grid-table object of "pp_length", "v1_length", "v2_length"

Examples

```
data("waxsys.palmprint")
data("palmdb")

ppLen <- PlotLengths(pp = waxsys.palmprint, pp.bg = palmdb)
plot(ppLen)
```

PlotOrgn *Plot a wordcloud of the organisms in a palm.sra object or orgn.vec*

Description

Plot a wordcloud of the organisms in a palm.sra object or orgn.vec

Usage

```
PlotOrgn(palm.sra = NULL, orgn.vec = NULL, freq = T)
```

Arguments

palm.sra	data.frame, created from get.palmSra() [NULL]
orgn.vec	character, vector of "scientific_name" from sra run table [NULL]
freq	boolean, scale words by frequency, else by identity [T]

Value

A ggwordcloud object of the "ntop" frequent terms

Examples

```
# Retrive organism identifiers from SRA Run Info Table
# palm.orgn <- get.sraOrgn(run_ids, con)

# Load Waxsystermes Exampel data
data("waxsys.palm.sra")

# Create wordcloud of organism terms
# using column "scientific_name" in data.frame

# Scaled by frequency of organism term in all of data.frame
PlotOrgn( waxsys.palm.sra )

# Scaled by proximity of organism tag to input sequence (pident)
PlotOrgn( waxsys.palm.sra , freq = FALSE)
```

PlotPP *Plot the palmprint-diagram for a palmscan df object*

Description

Plot the palmprint-diagram for a palmscan df object

Usage

```
PlotPP(ps)
```

Arguments

ps	A palmscan data.frame containing one palmprint

Value

A gene-diagram as a ggplot2 object

Examples

```
data("waxsys.palmprint")

palmprint.diagram <- PlotPP(waxsys.palmprint)

plot(palmprint.diagram)
```

PlotProReport *Create PlotID and PlotTax grid-plot*

Description

Create PlotID and PlotTax grid-plot

Usage

```
PlotProReport(pro, html = FALSE)
```

Arguments

pro	data.frame, pro.df object
html	boolean, generate htmlWidget instead of ggplot [FALSE]

Value

A grid-table object. Dimension standard is 800 x 400 px.

Examples

```
data("waxsys.pro.df")

proPlot <- PlotProReport(waxsys.pro.df)

plot(proPlot)
```

PlotReport *A wrapper for palmid Plot* functions to create a standard "report"*

Description

A wrapper for palmid Plot* functions to create a standard "report"

Usage

```
PlotReport(pp, pp.bg)
```

Arguments

pp	A palmprint.df row to use for the plot "value"
pp.bg	A multiple palmprint.df for the plot background "distribution"

Value

A grid-table object. Dimension standard is 800 x 400 px.

Examples

```
data("waxsys.palmprint")
data("palmdb")

ppRep <- PlotReport(waxsys.palmprint, palmdb)

plot(ppRep)
```

PlotTax *Plot a taxonomic-classifier based histogram*

Description

Plot a taxonomic-classifier based histogram

Usage

```
PlotTax(pro, html = TRUE)
```

Arguments

pro	data.frame, A diamond-aligned pro file
html	boolean, include additional parsing for htmlwidget display [TRUE]

Value

A histogram ggplot2

Examples

```
data("waxsys.pro.df")

proTax <- PlotTax(waxsys.pro.df, html = TRUE)

plot(proTax)
```

PlotTaxHist *Plot Percent-identity, factored on taxonomic strings of a pro df*

Description

Plot Percent-identity, factored on taxonomic strings of a pro df

Usage

```
PlotTaxHist(pro.pident, pro.tax, rank = NA)
```

Arguments

pro.pident	numeric, pident column from pro.df
pro.tax	character, tax column from pro.df (use: get.tax)
rank	character, string of tax-rank to label graph

Value

A histogram as a ggplot2 object

Examples

```
data("waxsys.pro.df")

taxHist <- PlotTaxHist(pro.pident = waxsys.pro.df$pident,
                       pro.tax    = waxsys.pro.df$tfam,
                       rank       = "family")
```

`PlotTaxReport` *A multi-plot wrapper to convert a list of SRA 'run_ids' into a geographic world-map and timeline.*

Description

A multi-plot wrapper to convert a list of SRA 'run_ids' into a geographic world-map and timeline.

Usage

```
PlotTaxReport(pro.df)
```

Arguments

`pro.df`	data.frame, imported diamond pro df. use get.pro()

Value

A grid-table object. Dimension standard is 800 x 600 px.

Examples

```
data("waxsys.pro.df")

proTax <- PlotTaxReport( waxsys.pro.df )
```

PlotTimeline *PlotTimeline Create a timeline of*

Description

PlotTimeline Create a timeline of

Usage

```
PlotTimeline(run_ids = NULL, sra.dates = NULL, con = SerratusConnect())
```

Arguments

run_ids	character, vector of sra run_ids
sra.dates	POSIXct, set of dates
con	pq-connection, use SerratusConnect()

Value

ggplot2, timeline of SRA load dates

read.fev *Reads a .fev file created by 'palmscan'*

Description

Reads a .fev file created by 'palmscan'

Usage

```
read.fev(fev.path, FIRST = FALSE)
```

Arguments

fev.path	relative system path to .fev file
FIRST	read only the first palmscan-line in .fev [FALSE]

Value

A palmscan data.frame object

Examples

```
# palmscan fev file
ps.fev.path <- system.file( "extdata", "waxsys.fev", package = "palmid")
palmprint <- read.fev(ps.fev.path, FIRST = TRUE)
```

read.pro *Reads a .pro file created by 'diamond'*

Description

Reads a .pro file created by 'diamond'

Usage

```
read.pro(pro.path)
```

Arguments

pro.path	relative system path to .fev file

Value

A diamond-pro data.frame object

Examples

```
# palmDB Alignment file (.pro)
pro.path <- system.file( "extdata", "waxsys.pro", package = "palmid")
pro.df <- read.pro(pro.path)
```

SerratusConnect *SerratusConnect*

Description

Connection information for reaching the Serratus postgreSQL database See also: https://github.com/ababaian/serratus/wiki/S
Schema

Usage

```
SerratusConnect()
```

Value

con PostgreSQLConnection

Examples

```
con <- SerratusConnect()
```

`standardizeWordcount` *standardizeWordcount Create frequency-count table from a set of characters which are assigned a standardized rank-order scores from 10.0 to 1.0.*

Description

standardizeWordcount Create frequency-count table from a set of characters which are assigned a standardized rank-order scores from 10.0 to 1.0.

Usage

```
standardizeWordcount(words, ntop = 50)
```

Arguments

`words`	character, description of vector [Default]
`ntop`	numeric, return only N top words [50]

Value

table, frequency-count table with 10:1 standard

Examples

```
NULL
```

`waxsys.palm.sra` *waxsys.palm.sra*

Description

Waxsystermes Virus example data. Sequencing libraries in the Sequence Read Archive (SRA) for which Waxsys or related viruses have been identified in the Serratus Database. Including associated sequencing library meta-data.

Usage

```
data(waxsys.palm.sra)
```

Format

data.frame with 4159 obs. of 13 variables

Details

#'

- run_id. Sequencing library identifier in the SRA
- palm_id. PalmDB unique identifier for RdRP sequence in library
- coverage. Read coverage of contig containing RdRP
- sOTU. Species-like Operational Taxonomic Unit to which palm_id belong
- qseqid. Input query sequence name
- pident. Identity between input sequence and respective sOTU
- evalue. Expectence value for alignment
- sra_sequence. Matching sequence in the sequence library
- biosample_id. Corresponding identifier for the biosample database
- scientific_name. Provided organism meta-data
- date. Load date for sequencing library
- lng. Longitude meta-data (if available)
- lat. Latitude meta-data (if available)

Source

Waxsystermes Virus Tutorial

`waxsys.palmprint` *waxsys.palmprint*

Description

Waxsystermes Virus example data. Palmprint RdRP-barcode information generated by 'palmscan' and imported via read.fev().

Usage

```
data(waxsys.palmprint)
```

Format

data.frame with 1 obs. of 17 variables

Details

#'

- score. palmscan PSSM motif score. 20+ is high confidence RdRP
- query. input fasta query name
- gene. One of "RdRP" or "RT"
- order. Order of catalytic motifs on input. "ABC" or "CAB"
- confidence. "high" or "low" confidence classification
- qlen. input query length. Is truncated on long input sequences
- pp_start. Start coordinate of palmprint within input sequence
- pp_end. End coordinate of palmprint within input sequence
- pp_length. Length of palmprint sequence (in AA)
- v1_length. Length of v1 region of palmprint
- v2_length. Length of v2 region of palmprint
- pssm_total_score. Raw total score of A,B,C motifs
- pssm_min_score. Lowest scoring PSSM score in palmprint
- motifs. Isolated A,B,C motifs from palmprint.
- super. Isolated super-motif residues from A,B,C (catalytic sites)
- group. Guess of sequence phylum.
- comments. Summary statement on palmprint QC.

Source

Waxsystermes Virus Tutorial

`waxsys.pro.df` *waxsys.pro.df*

Description

Protein-alignment data.frame from the Waxsystermes Virus example. Generated by read.pro() from DIAMOND-alignment of an input sequence against PalmDB.

Usage

```
data(waxsys.pro.df)
```

Format

data.frame with 223 obs. of 15 variables

Details

#’

- qseqid. Query or input sequence name ("SRR9968562_waxsystermes_virus_microassembly")
- qstart. Coordinate of alignment start, 1 based
- qend. Coordinate of alignment end, 1 based
- qlen. Length of aligned sequence on query
- sseqid. Subject (palmDB) sequence identifier, sOTU used.
- sstart. Coordinate of alignment start, 1 based
- send. Coordinate of alignment end, 1 based
- slen. Length of aligned sequence on query
- pident. Percent AA-identity between query and subject. (0–100)
- evalue. Expectance value for alignment
- cigar. CIGAR alignment string for query alignment
- full_sseq. Complete subject sequence (including non-aligned)
- tspe. Taxonomic species of the palmDB match if available, else "."
- tfam. Taxonomic family of the palmDB match if available, else "."
- tphy. Taxonomic phylum of the palmDB match if available, else "."

Source

Waxsystermes Virus Tutorial

Index