Package: CRBHits (via r-universe)

May 11, 2026

Title Conditional reciprocal best hits (CRBHits) in R

Version 0.0.10

Description CRBHits calculates conditional reciprocal best hit (CRBHit) pairs and subsequently can classify tandem duplicates, assign syntney groups and calculate Ka/Ks values.

License MIT + file LICENSE

Encoding UTF-8

LazyData false

Depends R (>= 4.5.0)

Imports Biostrings, MSA2dist (>= 1.12.0), curl, doParallel, dplyr, foreach, ggplot2, graphics, gridExtra, grDevices, methods, parallel, readr, rlang, stats, stringr, tidyr, utils

Suggests ape, rmarkdown, knitr, devtools, seqinr, stringi, testthat, tibble

VignetteBuilder knitr

biocViews Software, Alignment, Genetics

NeedsCompilation yes

SystemRequirements LAST (https://anaconda.org/bioconda/last), DAGchainer (https://anaconda.org/bioconda/dagchainer), DIAMOND (optional, https://anaconda.org/bioconda/diamond), MMSEQS2 (optiona, https://anaconda.org/bioconda/mmseqs2), LAMBDA (optional, https://anaconda.org/bioconda/lambda), GNU make, libcurl (deb: libcurl4-openssl-dev or CentOS: libcurl-devel), openssl (deb: libssl-dev or CentOS: openssl-devel), libxml2 (deb: libxml2-dev or CentOS: libxml2-devel), libGLU (deb: libglu1-mesa-dev or CentOS: mesa-libGLU-devel), libgit2-devel (deb: libgit2-dev or CentOS: libgit2-devel)

URL https://github.com/kullrich/CRBHits, https://kullrich.github.io/CRBHits/

BugReports https://github.com/kullrich/CRBHits/issues

RoxygenNote 7.3.2

Config/pak/sysreqs make libgit2-dev libicu-dev libxml2-dev libssl-dev libx11-dev zlib1g-dev

Repository https://kullrich.r-universe.dev

Date/Publication 2025-05-21 16:54:01 UTC

RemoteUrl https://github.com/kullrich/crbhits

RemoteRef HEAD

RemoteSha 158ea52a96bbcb0928e384ac15015597aaa987bc

Contents

---

aa2rbh *aa2rbh*

Description

This function calculates (conditional-)reciprocal best hit (CRBHit) pairs from two AA `AAStringSet`'s. Conditional-reciprocal best hit pairs were introduced by *Aubry S, Kelly S et al. (2014)*. Sequence searches are performed with **last** *Kiełbasa, SM et al. (2011)* [default] or with **mmseqs2** *Steinegger, M and Soeding, J (2017)* or with **diamond** *Buchfink, B et al. (2021)*. If one specifies aa1 and aa2 as the same input a selfblast is conducted.

Usage

```
aa2rbh(
  aa1,
  aa2,
  dbfile1 = NULL,
  dbfile2 = NULL,
  searchtool = "last",
  lastpath = file.path(find.package("CRBHits"), "extdata", "last-1639", "bin"),
  lastD = 1e+06,
  lastm = 10,
  mmseqs2path = NULL,
  mmseqs2sensitivity = 5.7,
  mmseqs2maxseqs = 300,
  diamondpath = NULL,
  diamondsensitivity = "--sensitive",
  diamondmaxtargetseqs = 0,
  lambda3path = NULL,
  lambda3sensitivity = "sensitive",
  lambda3nummatches = 25,
  outpath = "/tmp",
  crbh = TRUE,
  keepSingleDirection = FALSE,
  eval = 0.001,
  qcov = 0,
  tcov = 0,
  pident = 0,
  alnlen = 0,
  rost1999 = FALSE,
  filter = NULL,
  plotCurve = FALSE,
  fit.type = "mean",
  fit.varweight = 0.1,
  fit.min = 5,
  threads = 1,
```

```
  remove = TRUE,
  remove.db = TRUE
)
```

Arguments

aa1	aa1 sequences as AAStringSet [mandatory]
aa2	aa2 sequences as AAStringSet [mandatory]
dbfile1	aa1 db file [optional]
dbfile2	aa2 db file [optional]
searchtool	specify sequence search algorithm last, mmseqs2, diamond or lambda3 [default: last]
lastpath	specify the PATH to the last binaries [default: /extdata/last-1639/bin/]
lastD	last option D: query letters per random alignment [default: 1e6]
lastm	last option m: maximum initial matches per query position [default: 10]
mmseqs2path	specify the PATH to the mmseqs2 binaries [default: NULL]
mmseqs2sensitivity	specify the sensitivity option of mmseqs2 [default: 5.7]
mmseqs2maxseqs	mmseqs2 option: Maximum results per query sequence allowed to pass the prefilter [default: 300]
diamondpath	specify the PATH to the diamond binaries [default: NULL]
diamondsensitivity	specify the sensitivity option of diamond [default: –sensitive]
diamondmaxtargetseqs	specify the maximum number of target sequences per query option of diamond [default: 0]
lambda3path	specify the PATH to the lambda3 binaries [default: NULL]
lambda3sensitivity	specify the sensitivity option of lambda3 [default: sensitive]
lambda3nummatches	specify the number of matches per query option of lambda3 [default: 25]
outpath	specify the output PATH [default: /tmp]
crbh	specify if conditional-reciprocal hit pairs should be retained as secondary hits [default: TRUE]
keepSingleDirection	specify if single direction secondary hit pairs should be retained [default: FALSE]
eval	evalue [default: 1e-3]
qcov	query coverage [default: 0.0]
tcov	target coverage [default: 0.0]
pident	percent identity [default: 0.0]
alnlen	alignment length [default: 0.0]
rost1999	specify if hit pairs should be filter by equation 2 of Rost 1999 [default: FALSE]

`filter`	specify additional custom filters as list to be applied on hit pairs [default: NULL]
`plotCurve`	specify if crbh fitting curve should be plotted [default: FALSE]
`fit.type`	specify if mean or median should be used for fitting [default: mean]
`fit.varweight`	factor for fitting function to consider neighborhood [default: 0.1]
`fit.min`	specify minimum neighborhood alignment length [default: 5]
`threads`	number of parallel threads [default: 1]
`remove`	specify if last result files should be removed [default: TRUE]
`remove.db`	specify if last db files should be removed [default: TRUE]

Value

List of three (crbh=FALSE)
1: $crbh.pairs
2: $crbh1 matrix; query > target
3: $crbh2 matrix; target > query

List of four (crbh=TRUE)
1: $crbh.pairs
2: $crbh1 matrix; query > target
3: $crbh2 matrix; target > query
4: $rbh1_rbh2_fit; evalue fitting function

Author(s)

Kristian K Ullrich

References

Aubry S, Kelly S et al. (2014) Deep Evolutionary Comparison of Gene Expression Identifies Parallel Recruitment of Trans-Factors in Two Independent Origins of C4 Photosynthesis. *PLOS Genetics*, **10(6)** e1004365.

Kiełbasa, SM et al. (2011) Adaptive seeds tame genomic sequence comparison. *Genome research*, **21(3)**, 487-493.

Rost B. (1999). Twilight zone of protein sequence alignments. *Protein Engineering*, **12(2)**, 85-94.

See Also

`cds2rbh`

Examples

```
## compile last-1639 within CRBHits
CRBHits::make_last()
## load example sequence data
data("ath", package="CRBHits")
data("aly", package="CRBHits")
ath.aa <- MSA2dist::cds2aa(ath)
```

```
aly.aa <- MSA2dist::cds2aa(aly)
## get CRBHit pairs
ath_aly_crbh <- aa2rbh(
    aa1=ath.aa,
    aa2=aly.aa,
    plotCurve=TRUE)
dim(ath_aly_crbh$crbh.pairs)
## get classical reciprocal best hit (RBHit) pairs
ath_aly_rbh <- aa2rbh(
    aa1=ath.aa,
    aa2=aly.aa,
    crbh=FALSE)
dim(ath_aly_rbh$crbh.pairs)
## selfblast
ath_selfblast_crbh <- aa2rbh(
    aa1=ath.aa,
    aa2=ath.aa,
    plotCurve=TRUE)
## see ?cds2rbh for more examples
```

`aadir2orthofinder` *aadir2orthofinder*

Description

This function calculates (conditional-)reciprocal best hit (CRBHit) pairs for all possible comparison including self comparison from a directory of AA fasta files. Sequence searches are performed with **last** *Kiełbasa, SM et al. (2011)* [default] or with **mmseqs2** *Steinegger, M and Soeding, J (2017)* or with **diamond** *Buchfink, B et al. (2021)*.

Usage

```
aadir2orthofinder(
  dir,
  file_ending = "*",
  searchtool = "last",
  lastpath = file.path(find.package("CRBHits"), "extdata", "last-1639", "bin"),
  lastD = 1e+06,
  lastm = 10,
  mmseqs2path = NULL,
  mmseqs2sensitivity = 5.7,
  mmseqs2maxseqs = 300,
  diamondpath = NULL,
  diamondsensitivity = "--sensitive",
  diamondmaxtargetseqs = 0,
  lambda3path = NULL,
  lambda3sensitivity = "sensitive",
  lambda3nummatches = 25,
```

```
  outpath = "/tmp",
  crbh = TRUE,
  keepSingleDirection = FALSE,
  eval = 0.001,
  qcov = 0,
  tcov = 0,
  pident = 0,
  alnlen = 0,
  rost1999 = FALSE,
  filter = NULL,
  fit.type = "mean",
  fit.varweight = 0.1,
  fit.min = 5,
  threads = 1,
  remove = TRUE
)
```

Arguments

dir	directory containing AA fasta files [mandatory]
file_ending	define file ending to consider [default: *]
searchtool	specify sequence search algorithm last, mmseqs2, diamond or lambda3 [default: last]
lastpath	specify the PATH to the last binaries [default: /extdata/last-1639/bin/]
lastD	last option D: query letters per random alignment [default: 1e6]
lastm	last option m: maximum initial matches per query position [default: 10]
mmseqs2path	specify the PATH to the mmseqs2 binaries [default: NULL]
mmseqs2sensitivity	specify the sensitivity option of mmseqs2 [default: 5.7]
mmseqs2maxseqs	mmseqs2 option: Maximum results per query sequence allowed to pass the prefilter [default: 300]
diamondpath	specify the PATH to the diamond binaries [default: NULL]
diamondsensitivity	specify the sensitivity option of diamond [default: –sensitive]
diamondmaxtargetseqs	specify the maximum number of target sequences per query option of diamond [default: 0]
lambda3path	specify the PATH to the lambda3 binaries [default: NULL]
lambda3sensitivity	specify the sensitivity option of lambda3 [default: sensitive]
lambda3nummatches	specify the number of matches per query option of lambda3 [default: 25]
outpath	specify the output PATH [default: /tmp]
crbh	specify if conditional-reciprocal hit pairs should be retained as secondary hits [default: TRUE]

keepSingleDirection	specify if single direction secondary hit pairs should be retained [default: FALSE]
eval	evalue [default: 1e-3]
qcov	query coverage [default: 0.0]
tcov	target coverage [default: 0.0]
pident	percent identity [default: 0.0]
alnlen	alignment length [default: 0.0]
rost1999	specify if hit pairs should be filter by equation 2 of Rost 1999 [default: FALSE]
filter	specify additional custom filters as list to be applied on hit pairs [default: NULL]
fit.type	specify if mean or median should be used for fitting [default: mean]
fit.varweight	factor for fitting function to consider neighborhood [default: 0.1]
fit.min	specify minimum neighborhood alignment length [default: 5]
threads	number of parallel threads [default: 1]
remove	specify if last result files should be removed [default: TRUE]

Value

List of three (crbh=FALSE)

Author(s)

Kristian K Ullrich

References

Aubry S, Kelly S et al. (2014) Deep Evolutionary Comparison of Gene Expression Identifies Parallel Recruitment of Trans-Factors in Two Independent Origins of C4 Photosynthesis. *PLOS Genetics*, **10(6)** e1004365.

Kiełbasa, SM et al. (2011) Adaptive seeds tame genomic sequence comparison. *Genome research*, **21(3)**, 487-493.

Rost B. (1999). Twilight zone of protein sequence alignments. *Protein Engineering*, **12(2)**, 85-94.

See Also

aafile2rbh

Examples

```
## compile last-1639 within CRBHits
CRBHits::make_last()
```

aafile2rbh *aafile2rbh*

Description

This function calculates (conditional-)reciprocal best hit (CRBHit) pairs from two AA fasta files. Conditional-reciprocal best hit pairs were introduced by *Aubry S, Kelly S et al. (2014)*. Sequence searches are performed with **last** *Kiełbasa, SM et al. (2011)* [default] or with **mmseqs2** *Steinegger, M and Soeding, J (2017)* or with **diamond** *Buchfink, B et al. (2021)*. If one specifies aafile1 and aafile2 as the same input a selfblast is conducted.

Usage

```
aafile2rbh(
  aafile1,
  aafile2,
  dbfile1 = NULL,
  dbfile2 = NULL,
  searchtool = "last",
  lastpath = file.path(find.package("CRBHits"), "extdata", "last-1639", "bin"),
  lastD = 1e+06,
  lastm = 10,
  mmseqs2path = NULL,
  mmseqs2sensitivity = 5.7,
  mmseqs2maxseqs = 300,
  diamondpath = NULL,
  diamondsensitivity = "--sensitive",
  diamondmaxtargetseqs = 0,
  lambda3path = NULL,
  lambda3sensitivity = "sensitive",
  lambda3nummatches = 25,
  outpath = "/tmp",
  crbh = TRUE,
  keepSingleDirection = FALSE,
  eval = 0.001,
  qcov = 0,
  tcov = 0,
  pident = 0,
  alnlen = 0,
  rost1999 = FALSE,
  filter = NULL,
  plotCurve = FALSE,
  fit.type = "mean",
  fit.varweight = 0.1,
  fit.min = 5,
  threads = 1,
  aafile2tmp = TRUE,
```

```
  remove = TRUE,
  remove.db = TRUE
)
```

Arguments

aafile1	aa1 fasta file [mandatory]
aafile2	aa2 fasta file [mandatory]
dbfile1	aa1 db file [optional]
dbfile2	aa2 db file [optional]
searchtool	specify sequence search algorithm last, mmseqs2, diamond or lambda3 [default: last]
lastpath	specify the PATH to the last binaries [default: /extdata/last-1639/bin/]
lastD	last option D: query letters per random alignment [default: 1e6]
lastm	last option m: maximum initial matches per query position [default: 10]
mmseqs2path	specify the PATH to the mmseqs2 binaries [default: NULL]
mmseqs2sensitivity	specify the sensitivity option of mmseqs2 [default: 5.7]
mmseqs2maxseqs	mmseqs2 option: Maximum results per query sequence allowed to pass the prefilter [default: 300]
diamondpath	specify the PATH to the diamond binaries [default: NULL]
diamondsensitivity	specify the sensitivity option of diamond [default: –sensitive]
diamondmaxtargetseqs	specify the maximum number of target sequences per query option of diamond [default: 0]
lambda3path	specify the PATH to the lambda3 binaries [default: NULL]
lambda3sensitivity	specify the sensitivity option of lambda3 [default: sensitive]
lambda3nummatches	specify the number of matches per query option of lambda3 [default: 25]
outpath	specify the output PATH [default: /tmp]
crbh	specify if conditional-reciprocal hit pairs should be retained as secondary hits [default: TRUE]
keepSingleDirection	specify if single direction secondary hit pairs should be retained [default: FALSE]
eval	evalue [default: 1e-3]
qcov	query coverage [default: 0.0]
tcov	target coverage [default: 0.0]
pident	percent identity [default: 0.0]
alnlen	alignment length [default: 0.0]
rost1999	specify if hit pairs should be filter by equation 2 of Rost 1999 [default: FALSE]

`filter`	specify additional custom filters as list to be applied on hit pairs [default: NULL]
`plotCurve`	specify if crbh fitting curve should be plotted [default: FALSE]
`fit.type`	specify if mean or median should be used for fitting [default: mean]
`fit.varweight`	factor for fitting function to consider neighborhood [default: 0.1]
`fit.min`	specify minimum neighborhood alignment length [default: 5]
`threads`	number of parallel threads [default: 1]
`aafile2tmp`	specify if aa input files sequenceIDs should be reduced to the first word and be written to a temporary aa file [default: TRUE]
`remove`	specify if last result files should be removed [default: TRUE]
`remove.db`	specify if last db files should be removed [default: TRUE]

Value

List of three (crbh=FALSE)
1: $crbh.pairs
2: $crbh1 matrix; query > target
3: $crbh2 matrix; target > query

List of four (crbh=TRUE)
1: $crbh.pairs
2: $crbh1 matrix; query > target
3: $crbh2 matrix; target > query
4: $rbh1_rbh2_fit; evalue fitting function

Author(s)

Kristian K Ullrich

References

Aubry S, Kelly S et al. (2014) Deep Evolutionary Comparison of Gene Expression Identifies Parallel Recruitment of Trans-Factors in Two Independent Origins of C4 Photosynthesis. *PLOS Genetics*, **10(6)** e1004365.

Kiełbasa, SM et al. (2011) Adaptive seeds tame genomic sequence comparison. *Genome research*, **21(3)**, 487-493.

Rost B. (1999). Twilight zone of protein sequence alignments. *Protein Engineering*, **12(2)**, 85-94.

See Also

`aa2rbh`

Examples

```
## compile last-1639 within CRBHits
CRBHits::make_last()
## load example sequence data
athfile <- system.file("fasta", "ath.aa.fasta.gz", package="CRBHits")
```

```
alyfile <- system.file("fasta", "aly.aa.fasta.gz", package="CRBHits")
## get CRBHit pairs
ath_aly_crbh <- aafile2rbh(
    aafile1=athfile,
    aafile2=alyfile,
    plotCurve=TRUE)
dim(ath_aly_crbh$crbh.pairs)
## get classical reciprocal best hit (RBHit) pairs
ath_aly_rbh <- aafile2rbh(
    aafile1=athfile,
    aafile2=alyfile,
    crbh=FALSE)
dim(ath_aly_rbh$crbh.pairs)
## selfblast
ath_selfblast_crbh <- aafile2rbh(
    aafile1=athfile,
    aafile2=athfile,
    plotCurve=TRUE)
## see ?cds2rbh for more examples
```

---

aafile2rbhplus *aafile2rbhplus*

Description

This function calculates (conditional-)reciprocal best hit (CRBHit) pairs from two AA fasta files. Conditional-reciprocal best hit pairs were introduced by *Aubry S, Kelly S et al. (2014)*. Sequence searches are performed with **last** *Kiełbasa, SM et al. (2011)* [default] or with **mmseqs2** *Steinegger, M and Soeding, J (2017)* or with **diamond** *Buchfink, B et al. (2021)*. If one specifies aafile1 and aafile2 as the same input a selfblast is conducted.

Usage

```
aafile2rbhplus(
  aafile1,
  aafile2,
  dbfile1 = NULL,
  dbfile2 = NULL,
  searchtool = "last",
  lastpath = file.path(find.package("CRBHits"), "extdata", "last-1639", "bin"),
  lastD = 1e+06,
  lastm = 10,
 mafconvertpath = paste0(find.package("CRBHits"), "/extdata/maf-convert-crbhits"),
  mmseqs2path = NULL,
  mmseqs2sensitivity = 5.7,
  mmseqs2maxseqs = 300,
  diamondpath = NULL,
  diamondsensitivity = "--sensitive",
```

```
  diamondmaxtargetseqs = 0,
  lambda3path = NULL,
  lambda3sensitivity = "sensitive",
  lambda3nummatches = 25,
  outpath = "/tmp",
  crbh = TRUE,
  keepSingleDirection = FALSE,
  eval = 0.001,
  qcov = 0,
  tcov = 0,
  pident = 0,
  alnlen = 0,
  rost1999 = FALSE,
  filter = NULL,
  plotCurve = FALSE,
  fit.type = "mean",
  fit.varweight = 0.1,
  fit.min = 5,
  threads = 1,
  aafile2tmp = TRUE,
  remove = TRUE,
  remove.db = TRUE
)
```

Arguments

aafile1	aa1 fasta file [mandatory]
aafile2	aa2 fasta file [mandatory]
dbfile1	aa1 db file [optional]
dbfile2	aa2 db file [optional]
searchtool	specify sequence search algorithm last, mmseqs2, diamond or lambda3 [default: last]
lastpath	specify the PATH to the last binaries [default: /extdata/last-1639/bin/]
lastD	last option D: query letters per random alignment [default: 1e6]
lastm	last option m: maximum initial matches per query position [default: 10]
mafconvertpath	specify the PATH to the maf-convert script [default: /extdata/maf-convert-crbhits]
mmseqs2path	specify the PATH to the mmseqs2 binaries [default: NULL]
mmseqs2sensitivity	specify the sensitivity option of mmseqs2 [default: 5.7]
mmseqs2maxseqs	mmseqs2 option: Maximum results per query sequence allowed to pass the prefilter [default: 300]
diamondpath	specify the PATH to the diamond binaries [default: NULL]
diamondsensitivity	specify the sensitivity option of diamond [default: –sensitive]

`diamondmaxtargetseqs`
: specify the maximum number of target sequences per query option of diamond [default: 0]

`lambda3path`
: specify the PATH to the lambda3 binaries [default: NULL]

`lambda3sensitivity`
: specify the sensitivity option of lambda3 [default: sensitive]

`lambda3nummatches`
: specify the number of matches per query option of lambda3 [default: 25]

`outpath`
: specify the output PATH [default: /tmp]

`crbh`
: specify if conditional-reciprocal hit pairs should be retained as secondary hits [default: TRUE]

`keepSingleDirection`
: specify if single direction secondary hit pairs should be retained [default: FALSE]

`eval`
: evalue [default: 1e-3]

`qcov`
: query coverage [default: 0.0]

`tcov`
: target coverage [default: 0.0]

`pident`
: percent identity [default: 0.0]

`alnlen`
: alignment length [default: 0.0]

`rost1999`
: specify if hit pairs should be filter by equation 2 of Rost 1999 [default: FALSE]

`filter`
: specify additional custom filters as list to be applied on hit pairs [default: NULL]

`plotCurve`
: specify if crbh fitting curve should be plotted [default: FALSE]

`fit.type`
: specify if mean or median should be used for fitting [default: mean]

`fit.varweight`
: factor for fitting function to consider neighborhood [default: 0.1]

`fit.min`
: specify minimum neighborhood alignment length [default: 5]

`threads`
: number of parallel threads [default: 1]

`aafile2tmp`
: specify if aa input files sequenceIDs should be reduced to the first word and be written to a temporary aa file [default: TRUE]

`remove`
: specify if last result files should be removed [default: TRUE]

`remove.db`
: specify if last db files should be removed [default: TRUE]

Value

List of three (crbh=FALSE)
1: $crbh.pairs
2: $crbh1 matrix; query > target
3: $crbh2 matrix; target > query

List of four (crbh=TRUE)
1: $crbh.pairs
2: $crbh1 matrix; query > target
3: $crbh2 matrix; target > query
4: $rbh1_rbh2_fit; evalue fitting function

Author(s)

Kristian K Ullrich

References

Aubry S, Kelly S et al. (2014) Deep Evolutionary Comparison of Gene Expression Identifies Parallel Recruitment of Trans-Factors in Two Independent Origins of C4 Photosynthesis. *PLOS Genetics*, **10(6)** e1004365.

Kiełbasa, SM et al. (2011) Adaptive seeds tame genomic sequence comparison. *Genome research*, **21(3)**, 487-493.

Rost B. (1999). Twilight zone of protein sequence alignments. *Protein Engineering*, **12(2)**, 85-94.

See Also

aa2rbh

Examples

```
## compile last-1639 within CRBHits
CRBHits::make_last()
## load example sequence data
athfile <- system.file("fasta", "ath.aa.fasta.gz", package="CRBHits")
alyfile <- system.file("fasta", "aly.aa.fasta.gz", package="CRBHits")
## get CRBHit pairs
ath_aly_crbh <- aafile2rbhplus(
    aafile1=athfile,
    aafile2=alyfile,
    plotCurve=TRUE)
dim(ath_aly_crbh$crbh.pairs)
## get classical reciprocal best hit (RBHit) pairs
ath_aly_rbh <- aafile2rbhplus(
    aafile1=athfile,
    aafile2=alyfile,
    crbh=FALSE)
dim(ath_aly_rbh$crbh.pairs)
## selfblast
ath_selfblast_crbh <- aafile2rbhplus(
    aafile1=athfile,
    aafile2=athfile,
    plotCurve=TRUE)
## see ?cds2rbh for more examples
```

---

aly-data *aly-data*

Description

Example cds sequences from Arabidopsis lyrata as DNAStringSet.

Usage

```
data(aly)
```

Format

an object of class DNAStringSet see XStringSet-class

Examples

```
data("aly", package="CRBHits")
```

---

ath-data *ath-data*

Description

Example cds sequences from Arabidopsis thaliana as DNAStringSet.

Usage

```
data(ath)
```

Format

an object of class DNAStringSet see XStringSet-class

Examples

```
data("ath", package="CRBHits")
```

---

ath_aly_crbh-data *ath_aly_crbh-data*

Description

Example conditional-reciprocal best hit (CRBHit) pair results.

Usage

```
data(ath_aly_crbh)
```

Format

an object of class list

Examples

```
data("ath_aly_crbh", package="CRBHits")
```

ath_aly_ncbi_dagchainer-data
ath_aly_ncbi_dagchainer-data

Description

Example of DAGchainer results between Arabidopsis thalina and Arabidopsis lyrata obtained from NCBI.

Usage

```
data(ath_aly_ncbi_dagchainer)
```

Format

an object of class `list`

Examples

```
data("ath_aly_ncbi_dagchainer", package="CRBHits")
```

ath_aly_ncbi_kaks-data
ath_aly_ncbi_kaks-data

Description

Example of Ka/Ks values between Arabidopsis thalina and Arabidopsis lyrata obtained from NCBI.

Usage

```
data(ath_aly_ncbi_kaks)
```

Format

an object of class `list`

Examples

```
data("ath_aly_ncbi_kaks", package="CRBHits")
```

cds2genepos *cds2genepos*

Description

This function extracts the gene position from either NCBI or ENSEMBL CDS input.

Usage

```
cds2genepos(cds, source = "NCBI", keep.names = NULL)
```

Arguments

cds	DNAStringSet [mandatory]
source	source indicating either NCBI or ENSEMBL [default: NCBI]
keep.names	vector indicating gene ids to be kept before chromosomal position assignment [default: NULL]

Value

matrix 1: $gene.seq.id
2: $gene.chr
3: $gene.start
4: $gene.end
5: $gene.mid
6: $gene.strand
7: $gene.idx

Author(s)

Kristian K Ullrich

See Also

isoform2longest

Examples

```
## load example sequence data
data("ath", package="CRBHits")
ath.genepos <- cds2genepos(
    cds=ath,
    source="ENSEMBL")
## Not run:
## load example sequence data
## set EnsemblPlants URL
ensemblPlants <- "ftp://ftp.ensemblgenomes.org/pub/plants/release-48/fasta/"
```

```
## set Arabidopsis thaliana CDS URL
ARATHA.cds.url <- paste0(ensemblPlants,
    "arabidopsis_thaliana/cds/Arabidopsis_thaliana.TAIR10.cds.all.fa.gz")
ARATHA.cds.file <- tempfile()
## download CDS
download.file(ARATHA.cds.url, ARATHA.cds.file, quiet=FALSE)
ARATHA.cds <- Biostrings::readDNAStringSet(ARATHA.cds.file)
## get gene position
ARATHA.cds.genepos <- cds2genepos(ARATHA.cds, "ENSEMBL")

## End(Not run)
```

---

cds2rbh *cds2rbh*

Description

This function calculates (conditional-)reciprocal best hit (CRBHit) pairs from two CDS `DNAStringSet`'s. CRBHit pairs were introduced by *Aubry S, Kelly S et al. (2014)*. Sequence searches are performed with **last** *Kiełbasa, SM et al. (2011)* [default] or with **mmseqs2** *Steinegger, M and Soeding, J (2017)* or with **diamond** *Buchfink, B et al. (2021)*. If one specifies cds1 and cds2 as the same input a selfblast is conducted.

Usage

```
cds2rbh(
  cds1,
  cds2,
  dbfile1 = NULL,
  dbfile2 = NULL,
  searchtool = "last",
  lastpath = file.path(find.package("CRBHits"), "extdata", "last-1639", "bin"),
  lastD = 1e+06,
  lastm = 10,
  mmseqs2path = NULL,
  mmseqs2sensitivity = 5.7,
  mmseqs2maxseqs = 300,
  diamondpath = NULL,
  diamondsensitivity = "--sensitive",
  diamondmaxtargetseqs = 0,
  lambda3path = NULL,
  lambda3sensitivity = "sensitive",
  lambda3nummatches = 25,
  outpath = "/tmp",
  crbh = TRUE,
  keepSingleDirection = FALSE,
  eval = 0.001,
  qcov = 0,
```

```
  tcov = 0,
  pident = 0,
  alnlen = 0,
  rost1999 = FALSE,
  filter = NULL,
  plotCurve = FALSE,
  fit.type = "mean",
  fit.varweight = 0.1,
  fit.min = 5,
  longest.isoform = FALSE,
  isoform.source = "NCBI",
  threads = 1,
  remove = TRUE,
  remove.db = TRUE,
  shorten1 = FALSE,
  frame1 = 1,
  framelist1 = NULL,
  genetic.code1 = NULL,
  shorten2 = FALSE,
  frame2 = 1,
  framelist2 = NULL,
  genetic.code2 = NULL
)
```

Arguments

cds1	cds1 sequences as DNAStringSet [mandatory]
cds2	cds2 sequences as DNAStringSet [mandatory]
dbfile1	aa1 db file [optional]
dbfile2	aa2 db file [optional]
searchtool	specify sequence search algorithm last, mmseqs2, diamond or lambda3 [default: last]
lastpath	specify the PATH to the last binaries [default: /extdata/last-1639/bin/]
lastD	last option D: query letters per random alignment [default: 1e6]
lastm	last option m: maximum initial matches per query position [default: 10]
mmseqs2path	specify the PATH to the mmseqs2 binaries [default: NULL]
mmseqs2sensitivity	specify the sensitivity option of mmseqs2 [default: 5.7]
mmseqs2maxseqs	mmseqs2 option: Maximum results per query sequence allowed to pass the prefilter [default: 300]
diamondpath	specify the PATH to the diamond binaries [default: NULL]
diamondsensitivity	specify the sensitivity option of diamond [default: –sensitive]
diamondmaxtargetseqs	specify the maximum number of target sequences per query option of diamond [default: 0]

`lambda3path` specify the PATH to the lambda3 binaries [default: NULL]

`lambda3sensitivity` specify the sensitivity option of lambda3 [default: sensitive]

`lambda3nummatches` specify the number of matches per query option of lambda3 [default: 25]

`outpath` specify the output PATH [default: /tmp]

`crbh` specify if conditional-reciprocal hit pairs should be retained as secondary hits [default: TRUE]

`keepSingleDirection` specify if single direction secondary hit pairs should be retained [default: FALSE]

`eval` evalue [default: 1e-3]

`qcov` query coverage [default: 0.0]

`tcov` target coverage [default: 0.0]

`pident` percent identity [default: 0.0]

`alnlen` alignment length [default: 0.0]

`rost1999` specify if hit pairs should be filter by equation 2 of Rost 1999 [default: FALSE]

`filter` specify additional custom filters as list to be applied on hit pairs [default: NULL]

`plotCurve` specify if crbh fitting curve should be plotted [default: FALSE]

`fit.type` specify if mean or median should be used for fitting [default: mean]

`fit.varweight` factor for fitting function to consider neighborhood [default: 0.1]

`fit.min` specify minimum neighborhood alignment length [default: 5]

`longest.isoform` specify if cds sequences should be reduced to the longest isoform (only possible if data was accessed from NCBI or ENSEMBL) [default: FALSE]

`isoform.source` specify cds sequences source (either NCBI or ENSEMBL) [default: NCBI]

`threads` number of parallel threads [default: 1]

`remove` specify if last result files should be removed [default: TRUE]

`remove.db` specify if last db files should be removed [default: TRUE]

`shorten1` shorten all sequences to multiple of three [default: FALSE]

`frame1` indicates the first base of the first codon [default: 1]

`framelist1` supply vector of frames for each entry [default: NULL]

`genetic.code1` The genetic code to use for the translation of codons into Amino Acid letters [default: NULL]

`shorten2` shorten all sequences to multiple of three [default: FALSE]

`frame2` indicates the first base of the first codon [default: 1]

`framelist2` supply vector of frames for each entry [default: NULL]

`genetic.code2` The genetic code to use for the translation of codons into Amino Acid letters [default: NULL]

Value

List of three (crbh=FALSE)
1: $crbh.pairs
2: $crbh1 matrix; query > target
3: $crbh2 matrix; target > query

List of four (crbh=TRUE)
1: $crbh.pairs
2: $crbh1 matrix; query > target
3: $crbh2 matrix; target > query
4: $rbh1_rbh2_fit; evalue fitting function

Author(s)

Kristian K Ullrich

References

Aubry S, Kelly S et al. (2014) Deep Evolutionary Comparison of Gene Expression Identifies Parallel Recruitment of Trans-Factors in Two Independent Origins of C4 Photosynthesis. *PLOS Genetics*, **10(6)** e1004365.

Kiełbasa, SM et al. (2011) Adaptive seeds tame genomic sequence comparison. *Genome research*, **21(3)**, 487-493.

Rost B. (1999). Twilight zone of protein sequence alignments. *Protein Engineering*, **12(2)**, 85-94.

See Also

cdsfile2rbh, isoform2longest

Examples

```
## compile last-1639 within CRBHits
CRBHits::make_last()
## load example sequence data
data("ath", package="CRBHits")
data("aly", package="CRBHits")
## get CRBHit pairs
ath_aly_crbh <- cds2rbh(
    cds1=ath,
    cds2=aly,
    plotCurve=TRUE)
dim(ath_aly_crbh$crbh.pairs)
## get classical reciprocal best hit (RBHit) pairs
ath_aly_rbh <- cds2rbh(
    cds1=ath,
    cds2=aly,
    crbh=FALSE)
dim(ath_aly_rbh$crbh.pairs)
## filter for evalue 1e-100
ath_aly_crbh.eval100 <- cds2rbh(
```

```
    cds1=ath,
    cds2=aly,
    eval=1e-100)
dim(ath_aly_crbh.eval100$crbh.pairs)
## filter for query coverage
ath_aly_crbh.qcov <- cds2rbh(
    cds1=ath,
    cds2=aly,
    qcov=0.5)
dim(ath_aly_crbh.qcov$crbh.pairs)
## custom filter for e.g. bit score (column 12)
## Note: multiple filters can be given in filter param as list
myfilter <- function(rbh, value=500.0){
    return(rbh[as.numeric(rbh[, 12])>=value, , drop=FALSE])
}
ath_aly_crbh.custom <- cds2rbh(
    cds1=ath,
    cds2=aly,
    filter=list(myfilter))
dim(ath_aly_crbh.custom$crbh.pairs)
## selfblast
ath_selfblast_crbh <- cds2rbh(
    cds1=ath,
    cds2=ath,
    plotCurve=TRUE)
```

`cdsdir2orthofinder` *cdsdir2orthofinder*

Description

This function calculates (conditional-)reciprocal best hit (CRBHit) pairs for all possible comparison including self comparison from a directory of CDS fasta files. Sequence searches are performed with **last** *Kiełbasa, SM et al. (2011)* [default] or with **mmseqs2** *Steinegger, M and Soeding, J (2017)* or with **diamond** *Buchfink, B et al. (2021)*.

Usage

```
cdsdir2orthofinder(
  dir,
  file_ending = "*",
  searchtool = "last",
  lastpath = file.path(find.package("CRBHits"), "extdata", "last-1639", "bin"),
  lastD = 1e+06,
  lastm = 10,
  mmseqs2path = NULL,
  mmseqs2sensitivity = 5.7,
  mmseqs2maxseqs = 300,
  diamondpath = NULL,
```

```
  diamondsensitivity = "--sensitive",
  diamondmaxtargetseqs = 0,
  lambda3path = NULL,
  lambda3sensitivity = "sensitive",
  lambda3nummatches = 25,
  outpath = "/tmp",
  dbpath = "/tmp",
  crbh = TRUE,
  keepSingleDirection = FALSE,
  eval = 0.001,
  qcov = 0,
  tcov = 0,
  pident = 0,
  alnlen = 0,
  rost1999 = FALSE,
  filter = NULL,
  fit.type = "mean",
  fit.varweight = 0.1,
  fit.min = 5,
  threads = 1,
  remove = FALSE,
  remove.db = FALSE
)
```

Arguments

`dir`	directory containing CDS fasta files [mandatory]
`file_ending`	define file ending to consider [default: *]
`searchtool`	specify sequence search algorithm last, mmseqs2, diamond or lambda3 [default: last]
`lastpath`	specify the PATH to the last binaries [default: /extdata/last-1639/bin/]
`lastD`	last option D: query letters per random alignment [default: 1e6]
`lastm`	last option m: maximum initial matches per query position [default: 10]
`mmseqs2path`	specify the PATH to the mmseqs2 binaries [default: NULL]
`mmseqs2sensitivity`	specify the sensitivity option of mmseqs2 [default: 5.7]
`mmseqs2maxseqs`	mmseqs2 option: Maximum results per query sequence allowed to pass the prefilter [default: 300]
`diamondpath`	specify the PATH to the diamond binaries [default: NULL]
`diamondsensitivity`	specify the sensitivity option of diamond [default: –sensitive]
`diamondmaxtargetseqs`	specify the maximum number of target sequences per query option of diamond [default: 0]
`lambda3path`	specify the PATH to the lambda3 binaries [default: NULL]

`lambda3sensitivity`
: specify the sensitivity option of lambda3 [default: sensitive]

`lambda3nummatches`
: specify the number of matches per query option of lambda3 [default: 25]

`outpath`
: specify the output PATH [default: /tmp]

`dbpath`
: specify the output PATH [default: /tmp]

`crbh`
: specify if conditional-reciprocal hit pairs should be retained as secondary hits [default: TRUE]

`keepSingleDirection`
: specify if single direction secondary hit pairs should be retained [default: FALSE]

`eval`
: evalue [default: 1e-3]

`qcov`
: query coverage [default: 0.0]

`tcov`
: target coverage [default: 0.0]

`pident`
: percent identity [default: 0.0]

`alnlen`
: alignment length [default: 0.0]

`rost1999`
: specify if hit pairs should be filter by equation 2 of Rost 1999 [default: FALSE]

`filter`
: specify additional custom filters as list to be applied on hit pairs [default: NULL]

`fit.type`
: specify if mean or median should be used for fitting [default: mean]

`fit.varweight`
: factor for fitting function to consider neighborhood [default: 0.1]

`fit.min`
: specify minimum neighborhood alignment length [default: 5]

`threads`
: number of parallel threads [default: 1]

`remove`
: specify if last result files should be removed [default: TRUE]

`remove.db`
: specify if last db files should be removed [default: TRUE]

Value

List of three (crbh=FALSE)

Author(s)

Kristian K Ullrich

References

Aubry S, Kelly S et al. (2014) Deep Evolutionary Comparison of Gene Expression Identifies Parallel Recruitment of Trans-Factors in Two Independent Origins of C4 Photosynthesis. *PLOS Genetics*, **10(6)** e1004365.

Kiełbasa, SM et al. (2011) Adaptive seeds tame genomic sequence comparison. *Genome research*, **21(3)**, 487-493.

Rost B. (1999). Twilight zone of protein sequence alignments. *Protein Engineering*, **12(2)**, 85-94.

See Also

`aafile2rbh`

Examples

```
## compile last-1639 within CRBHits
CRBHits::make_last()
```

`cdsfile2aafile` *cdsfile2aafile*

Description

This function translates a cds fasta file into an aa fasta file.

Usage

```
cdsfile2aafile(
  infile,
  outfile,
  shorten = FALSE,
  frame = 1,
  framelist = NULL,
  genetic.code = NULL
)
```

Arguments

`infile`	cds fasta file [mandatory]
`outfile`	aa fasta file [mandatory]
`shorten`	shorten all sequences to multiple of three [default: FALSE]
`frame`	indicates the first base of a the first codon [default: 1]
`framelist`	supply vector of frames for each entry [default: NULL]
`genetic.code`	The genetic code to use for the translation of codons into Amino Acid letters [default: NULL]

Value

aa fasta file

Author(s)

Kristian K Ullrich

See Also

`cds2aa`

Examples

```
## define file path
cdsfile <- system.file("fasta", "ath.cds.fasta.gz", package="CRBHits")
## create empty temp file
aafile <- tempfile()
## convert input CDS fasta file into AA fasta file
cdsfile2aafile(cdsfile, aafile)
aa <- Biostrings::readAAStringSet(aafile)
aa
```

`cdsfile2rbh` *cdsfile2rbh*

Description

This function calculates (conditional-)reciprocal best hit (CRBHit) pairs from two CDS fasta files. CRBHit pairs were introduced by *Aubry S, Kelly S et al. (2014)*. Sequence searches are performed with **last** *Kiełbasa, SM et al. (2011)* [default] or with **mmseqs2** *Steinegger, M and Soeding, J (2017)* or with **diamond** *Buchfink, B et al. (2021)*. If one specifies cdsfile1 and cdsfile2 as the same input a selfblast is conducted.

Usage

```
cdsfile2rbh(
  cdsfile1,
  cdsfile2,
  dbfile1 = NULL,
  dbfile2 = NULL,
  searchtool = "last",
  lastpath = file.path(find.package("CRBHits"), "extdata", "last-1639", "bin"),
  lastD = 1e+06,
  lastm = 10,
  mmseqs2path = NULL,
  mmseqs2sensitivity = 5.7,
  mmseqs2maxseqs = 300,
  diamondpath = NULL,
  diamondsensitivity = "--sensitive",
  diamondmaxtargetseqs = 0,
  lambda3path = NULL,
  lambda3sensitivity = "sensitive",
  lambda3nummatches = 25,
  outpath = "/tmp",
  crbh = TRUE,
  keepSingleDirection = FALSE,
  eval = 0.001,
  qcov = 0,
  tcov = 0,
```

```
  pident = 0,
  alnlen = 0,
  rost1999 = FALSE,
  filter = NULL,
  plotCurve = FALSE,
  fit.type = "mean",
  fit.varweight = 0.1,
  fit.min = 5,
  longest.isoform = FALSE,
  isoform.source = "NCBI",
  threads = 1,
  remove = TRUE,
  remove.db = TRUE,
  shorten1 = FALSE,
  frame1 = 1,
  framelist1 = NULL,
  genetic.code1 = NULL,
  shorten2 = FALSE,
  frame2 = 1,
  framelist2 = NULL,
  genetic.code2 = NULL
)
```

Arguments

cdsfile1	cds1 fasta file [mandatory]
cdsfile2	cds2 fasta file [mandatory]
dbfile1	aa1 db file [optional]
dbfile2	aa2 db file [optional]
searchtool	specify sequence search algorithm last, mmseqs2, diamond or lambda3 [default: last]
lastpath	specify the PATH to the last binaries [default: /extdata/last-1639/bin/]
lastD	last option D: query letters per random alignment [default: 1e6]
lastm	last option m: maximum initial matches per query position [default: 10]
mmseqs2path	specify the PATH to the mmseqs2 binaries [default: NULL]
mmseqs2sensitivity	specify the sensitivity option of mmseqs2 [default: 5.7]
mmseqs2maxseqs	mmseqs2 option: Maximum results per query sequence allowed to pass the prefilter [default: 300]
diamondpath	specify the PATH to the diamond binaries [default: NULL]
diamondsensitivity	specify the sensitivity option of diamond [default: –sensitive]
diamondmaxtargetseqs	specify the maximum number of target sequences per query option of diamond [default: 0]

`lambda3path` specify the PATH to the lambda3 binaries [default: NULL]

`lambda3sensitivity` specify the sensitivity option of lambda3 [default: sensitive]

`lambda3nummatches` specify the number of matches per query option of lambda3 [default: 25]

`outpath` specify the output PATH [default: /tmp]

`crbh` specify if conditional-reciprocal hit pairs should be retained as secondary hits [default: TRUE]

`keepSingleDirection` specify if single direction secondary hit pairs should be retained [default: FALSE]

`eval` evalue [default: 1e-3]

`qcov` query coverage [default: 0.0]

`tcov` target coverage [default: 0.0]

`pident` percent identity [default: 0.0]

`alnlen` alignment length [default: 0.0]

`rost1999` specify if hit pairs should be filter by equation 2 of Rost 1999 [default: FALSE]

`filter` specify additional custom filters as list to be applied on hit pairs [default: NULL]

`plotCurve` specify if crbh fitting curve should be plotted [default: FALSE]

`fit.type` specify if mean or median should be used for fitting [default: mean]

`fit.varweight` factor for fitting function to consider neighborhood [default: 0.1]

`fit.min` specify minimum neighborhood alignment length [default: 5]

`longest.isoform` specify if cds sequences should be removed to the longest isoform (only possible if data was accessed from NCBI or ENSEMBL) [default: FALSE]

`isoform.source` specify cds sequences source (either NCBI or ENSEMBL) [default: NCBI]

`threads` number of parallel threads [default: 1]

`remove` specify if last result files should be removed [default: TRUE]

`remove.db` specify if last db files should be removed [default: TRUE]

`shorten1` shorten all sequences to multiple of three [default: FALSE]

`frame1` indicates the first base of the first codon [default: 1]

`framelist1` supply vector of frames for each entry [default: NULL]

`genetic.code1` The genetic code to use for the translation of codons into Amino Acid letters [default: NULL]

`shorten2` shorten all sequences to multiple of three [default: FALSE]

`frame2` indicates the first base of the first codon [default: 1]

`framelist2` supply vector of frames for each entry [default: NULL]

`genetic.code2` The genetic code to use for the translation of codons into Amino Acid letters [default: NULL]

Value

List of three (crbh=FALSE)
1: $crbh.pairs
2: $crbh1 matrix; query > target
3: $crbh2 matrix; target > query

List of four (crbh=TRUE)
1: $crbh.pairs
2: $crbh1 matrix; query > target
3: $crbh2 matrix; target > query
4: $rbh1_rbh2_fit; evalue fitting function

Author(s)

Kristian K Ullrich

References

Aubry S, Kelly S et al. (2014) Deep Evolutionary Comparison of Gene Expression Identifies Parallel Recruitment of Trans-Factors in Two Independent Origins of C4 Photosynthesis. *PLOS Genetics*, **10(6)** e1004365.

Kiełbasa, SM et al. (2011) Adaptive seeds tame genomic sequence comparison. *Genome research*, **21(3)**, 487-493.

Steinegger, M and Soeding, J. (2017). MMseqs2 enables sensitive protein sequence searching for the analysis of massive data sets. *Nature Biotechnology*, **35**, 1026-1028.

Rost B. (1999). Twilight zone of protein sequence alignments. *Protein Engineering*, **12(2)**, 85-94.

See Also

cds2rbh, isoform2longest

Examples

```
## compile last-1639 within CRBHits
CRBHits::make_last()
## load example sequence data
athfile <- system.file("fasta", "ath.cds.fasta.gz", package="CRBHits")
alyfile <- system.file("fasta", "aly.cds.fasta.gz", package="CRBHits")
## get CRBHit pairs
ath_aly_crbh <- cdsfile2rbh(
    cdsfile1=athfile,
    cdsfile2=alyfile,
    plotCurve=TRUE)
dim(ath_aly_crbh$crbh.pairs)
## get classical reciprocal best hit (RBHit) pairs
ath_aly_rbh <- cdsfile2rbh(
    cdsfile1=athfile,
    cdsfile2=alyfile,
    crbh=FALSE)
```

```
dim(ath_aly_rbh$crbh.pairs)
## selfblast
ath_selfblast_crbh <- cdsfile2rbh(
    cdsfile1=athfile,
    cdsfile2=athfile,
    plotCurve=TRUE)
## see ?cds2rbh for more examples
```

`check_ext_install` *check_ext_install*

Description

This function checks if external binary exists

Usage

```
check_ext_install(ext_name, ext_dir, binary_name = ext_name)
```

Arguments

`ext_name`	external tool name [mandatory]
`ext_dir`	external tool directory [mandatory]
`binary_name`	external binary name (per default ext_name) [mandatory]

Value

path of prerequisites

Author(s)

Kristian K Ullrich

`col2transparent` *col2transparent*

Description

R color to transparent conversion.

Usage

```
col2transparent(col, alpha.perc = 0)
```

Arguments

col	R color name or R color pal [mandatory]
alpha.perc	an alpha-transparency level in percent [default: 0]

Value

A character vector with elements of 9 characters, "#" followed by the red, blue, green and alpha values in hexadecimal (after rescaling to 0 ... 255).

Author(s)

Kristian K Ullrich

See Also

palette

Examples

```
col2transparent("green", 50)
## Demonstrate the colors 1:8 in different palettes using a custom matplot()
sinplot <- function(main=NULL) {
    x <- outer(
        seq(-pi, pi, length.out = 50),
        seq(0, pi, length.out = 9),
        function(x, y) sin(x - y)
    )
    matplot(x, type = "l", lwd = 4, lty = 1, col = 1:9, ylab = "", main=main)
}
my.palette <- c(
    "#CBC106", "#27993C", "#1C6838",
    "#8EBCB5", "#389CA7", "#4D83AB",
    "#CB7B26", "#BF565D", "#9E163C")
palette(my.palette); sinplot("my.palette")
## 25% transparent
palette(col2transparent(palette(my.palette), 25));
    sinplot("my.palette - transparent")
## 50% transparent
palette(col2transparent(palette(my.palette), 50));
    sinplot("my.palette - transparent")
## 75% transparent
palette(col2transparent(palette(my.palette), 75));
    sinplot("my.palette - transparent")
```

CRBHitsColors *CRBHitsColors*

Description

CRBHits specific color palette.

Usage

```
CRBHitsColors(n, alpha.perc = 0)
```

Arguments

n	The number of colors (greater than or equal to 1) to be in the palette [mandatory]
alpha.perc	an alpha-transparency level in percent [default: 0]

Value

CRBHits specific color palette

Author(s)

Kristian K Ullrich

See Also

palette

Examples

```
plot(1:9, pch=20, col=CRBHitsColors(9), cex=3)
## use alpha
plot(1:9, pch=20, col=CRBHitsColors(9, 50), cex=3)
```

filter_alnlen *filter_alnlen*

Description

This function filters BLAST-like tabular output according to alignment length.

Usage

```
filter_alnlen(rbh, alnlen = 0, inverse = FALSE)
```

Arguments

rbh	BLAST-like tabular matrix [mandatory]
alnlen	alignment length [default: 0.0]
inverse	specify if filter should keep the removed values [default: FALSE]

Value

rbh matrix

Author(s)

Kristian K Ullrich

Examples

```
## load crbh data
data(ath_aly_crbh)
dim(ath_aly_crbh$crbh1)
dim(filter_alnlen(
    rbh=ath_aly_crbh$crbh1,
    alnlen=75))
```

---

filter_eval *filter_eval*

Description

This function filters BLAST-like tabular output according to evalue.

Usage

```
filter_eval(rbh, eval = 0.001, inverse = FALSE)
```

Arguments

rbh	BLAST-like tabular matrix [mandatory]
eval	evalue [default: 1e-3]
inverse	specify if filter should keep the removed values [default: FALSE]

Value

rbh matrix

Author(s)

Kristian K Ullrich

Examples

```
## load crbh data
data(ath_aly_crbh)
dim(ath_aly_crbh$crbh1)
dim(filter_eval(
    rbh=ath_aly_crbh$crbh1,
    eval=1e-100))
```

`filter_pident` *filter_pident*

Description

This function filters BLAST-like tabular output according to protein identity.

Usage

```
filter_pident(rbh, pident = 0, inverse = FALSE)
```

Arguments

`rbh`	BLAST-like tabular matrix [mandatory]
`pident`	percent identity [default: 0.0]
`inverse`	specify if filter should keep the removed values [default: FALSE]

Value

rbh matrix

Author(s)

Kristian K Ullrich

Examples

```
## load crbh data
data(ath_aly_crbh)
dim(ath_aly_crbh$crbh1)
dim(filter_pident(
    rbh=ath_aly_crbh$crbh1,
    pident=75))
```

filter_qcov *filter_qcov*

Description

This function filters BLAST-like tabular output according to percentage query coverage.

Usage

```
filter_qcov(rbh, qcov = 0, inverse = FALSE)
```

Arguments

rbh	BLAST-like tabular matrix [mandatory]
qcov	query coverage [default: 0.0]
inverse	specify if filter should keep the removed values [default: FALSE]

Value

rbh matrix

Author(s)

Kristian K Ullrich

Examples

```
## load crbh data
data(ath_aly_crbh)
dim(ath_aly_crbh$crbh1)
dim(filter_qcov(
    rbh=ath_aly_crbh$crbh1,
    qcov=0.75))
```

filter_rost1999 *filter_rost1999*

Description

This function filters BLAST-like tabular output according to equation 2 of Rost 1999.

Usage

```
filter_rost1999(rbh, inverse = FALSE)
```

Arguments

rbh	BLAST-like tabular matrix [mandatory]
inverse	specify if filter should keep the removed values [default: FALSE]

Value

rbh matrix

Author(s)

Kristian K Ullrich

References

Rost B. (1999). Twilight zone of protein sequence alignments. *Protein Engineering*, **12(2)**, 85-94.

Examples

```
## load crbh data
data(ath_aly_crbh)
dim(ath_aly_crbh$crbh1)
dim(filter_rost1999(
    rbh=ath_aly_crbh$crbh1))
```

---

filter_tcov *filter_tcov*

Description

This function filters BLAST-like tabular output according to percentage target coverage.

Usage

```
filter_tcov(rbh, tcov = 0, inverse = FALSE)
```

Arguments

rbh	BLAST-like tabular matrix [mandatory]
tcov	target coverage [default: 0.0]
inverse	specify if filter should keep the removed values [default: FALSE]

Value

rbh matrix

Author(s)

Kristian K Ullrich

Examples

```
## load crbh data
data(ath_aly_crbh)
dim(ath_aly_crbh$crbh1)
dim(filter_tcov(
    rbh=ath_aly_crbh$crbh1,
    tcov=0.75))
```

---

gff2longest *gff2longest*

Description

This function extracts the gene position from GFF3 input and optional extracts the longest isoform.

Usage

```
gff2longest(gff3file, cds = NULL, removeNonCoding = TRUE, source = "NCBI")
```

Arguments

gff3file	GFF3 path [mandatory]
cds	DNAStringSet [optional]
removeNonCoding	specify if NonCoding transcripts should be removed
source	source indicating either NCBI or ENSEMBL [default: NCBI]

Value

list

Author(s)

Kristian K Ullrich

See Also

XStringSet-class

Examples

```
## Not run:
## load example sequence data
## set NCBI GFF3 URL
NCBI <- "ftp://ftp.ncbi.nlm.nih.gov/genomes/all/"
ARATHA.NCBI.gff3.url <- paste0(NCBI,
    "GCF/000/001/735/GCF_000001735.4_TAIR10.1/",
    "GCF_000001735.4_TAIR10.1_genomic.gff.gz")
```

```
ARATHA.NCBI.gff3.file <- tempfile()
## download GTF file
download.file(ARATHA.NCBI.gff3.url, ARATHA.NCBI.gff3.file, quiet=FALSE)
## set NCBI CDS URL
ARATHA.NCBI.cds.url <- paste0(NCBI,
    "GCF/000/001/735/GCF_000001735.4_TAIR10.1/",
    "GCF_000001735.4_TAIR10.1_cds_from_genomic.fna.gz")
ARATHA.NCBI.cds.file <- tempfile()
## download CDS file
download.file(ARATHA.NCBI.cds.url, ARATHA.NCBI.cds.file, quiet=FALSE)
## load CDS
ARATHA.NCBI.cds <- Biostrings::readDNAStringSet(ARATHA.NCBI.cds.file)
## get genepos and longest isoform
ARATHA.NCBI.gff3.longest <- gff2longest(gtffile=ARATHA.NCBI.gff3.file,
    cds=ARATHA.NCBI.cds, source="NCBI")
ARATHA.NCBI.gff3.longest$genepos
ARATHA.NCBI.gff3.longest$cds
## set ENSEMBL GFF3 URL
ensembl <- "http://ftp.ensemblgenomes.org/pub/plants/release-52/"
ARATHA.ENSEMBL.gff3.url <- paste0(ensembl,
    "gff3/arabidopsis_thaliana/Arabidopsis_thaliana.TAIR10.52.gff3.gz")
ARATHA.ENSEMBL.gff3.file <- tempfile()
## download GTF file
download.file(ARATHA.ENSEMBL.gff3.url, ARATHA.ENSEMBL.gff3.file, quiet=FALSE)
## set ENSEMBL CDS URL
ARATHA.ENSEMBL.cds.url <- paste0(ensembl,
    "fasta/arabidopsis_thaliana/cds/",
    "Arabidopsis_thaliana.TAIR10.cds.all.fa.gz")
ARATHA.ENSEMBL.cds.file <- tempfile()
## download CDS file
download.file(ARATHA.ENSEMBL.cds.url, ARATHA.ENSEMBL.cds.file, quiet=FALSE)
ARATHA.ENSEMBL.cds <- Biostrings::readDNAStringSet(ARATHA.ENSEMBL.cds.file)
## get genepos and longest isoform
ARATHA.ENSEMBL.gff3.longest <- gff2longest(gff3file=ARATHA.ENSEMBL.gff3.file,
    cds=ARATHA.ENSEMBL.cds)
ARATHA.ENSEMBL.gff3.longest$genepos
ARATHA.ENSEMBL.gff3.longest$cds

## End(Not run)
```

---

gtf2longest *gtf2longest*

Description

This function extracts the gene position from GTF input and optional extracts the longest isoform.

Usage

```
gtf2longest(gtffile, cds = NULL, removeNonCoding = TRUE, source = "NCBI")
```

Arguments

gtffile	GTF path [mandatory]
cds	DNAStringSet [optional]
removeNonCoding	specify if NonCoding transcripts should be removed
source	source indicating either NCBI or ENSEMBL [default: NCBI]

Value

list

Author(s)

Kristian K Ullrich

See Also

XStringSet-class

Examples

```
## Not run:
## load example sequence data
## set NCBI GTF URL
NCBI <- "ftp://ftp.ncbi.nlm.nih.gov/genomes/all/"
ARATHA.NCBI.gtf.url <- paste0(NCBI,
    "GCF/000/001/735/GCF_000001735.4_TAIR10.1/",
    "GCF_000001735.4_TAIR10.1_genomic.gtf.gz")
ARATHA.NCBI.gtf.file <- tempfile()
## download GTF file
download.file(ARATHA.NCBI.gtf.url, ARATHA.NCBI.gtf.file, quiet=FALSE)
## set NCBI CDS URL
ARATHA.NCBI.cds.url <- paste0(NCBI,
    "GCF/000/001/735/GCF_000001735.4_TAIR10.1/",
    "GCF_000001735.4_TAIR10.1_cds_from_genomic.fna.gz")
ARATHA.NCBI.cds.file <- tempfile()
## download CDS file
download.file(ARATHA.NCBI.cds.url, ARATHA.NCBI.cds.file, quiet=FALSE)
## load CDS
ARATHA.NCBI.cds <- Biostrings::readDNAStringSet(ARATHA.NCBI.cds.file)
## get genepos and longest isoform
ARATHA.NCBI.gtf.longest <- gtf2longest(gtffile=ARATHA.NCBI.gtf.file,
    cds=ARATHA.NCBI.cds, source="NCBI")
ARATHA.NCBI.gtf.longest$genepos
ARATHA.NCBI.gtf.longest$cds
## set ENSEMBL GTF URL
ensembl <- "http://ftp.ensemblgenomes.org/pub/plants/release-52/"
ARATHA.ENSEMBL.gtf.url <- paste0(ensembl,
    "gtf/arabidopsis_thaliana/Arabidopsis_thaliana.TAIR10.52.gtf.gz")
ARATHA.ENSEMBL.gtf.file <- tempfile()
```

```
## download GTF file
download.file(ARATHA.ENSEMBL.gtf.url, ARATHA.ENSEMBL.gtf.file, quiet=FALSE)
## set ENSEMBL CDS URL
ARATHA.ENSEMBL.cds.url <- paste0(ensembl,
    "fasta/arabidopsis_thaliana/cds/",
    "Arabidopsis_thaliana.TAIR10.cds.all.fa.gz")
ARATHA.ENSEMBL.cds.file <- tempfile()
## download CDS file
download.file(ARATHA.ENSEMBL.cds.url, ARATHA.ENSEMBL.cds.file, quiet=FALSE)
ARATHA.ENSEMBL.cds <- Biostrings::readDNAStringSet(ARATHA.ENSEMBL.cds.file)
## get genepos and longest isoform
ARATHA.ENSEMBL.gtf.longest <- gtf2longest(gtffile=ARATHA.ENSEMBL.gtf.file,
    cds=ARATHA.ENSEMBL.cds)
ARATHA.ENSEMBL.gtf.longest$genepos
ARATHA.ENSEMBL.gtf.longest$cds

## End(Not run)
```

`hom_pan_ensembl_kaks-data` *hom_pan_ensembl_kaks-data*

Description

Example of Ka/Ks values between Homo sapiens and Pan troglodytes obtained from NCBI.

Usage

```
data(hom_pan_ensembl_kaks)
```

Format

an object of class `list`

Examples

```
data("hom_pan_ensembl_kaks", package="CRBHits")
```

`isoform2longest` *isoform2longest*

Description

This function extracts the longest isoform from either NCBI or ENSEMBL CDS input.

Usage

```
isoform2longest(cds, source = "NCBI")
```

Arguments

cds	DNAStringSet [mandatory]
source	source indicating either NCBI, ENSEMBL or WORMBASE [default: NCBI]

Value

DNAStringSet

Author(s)

Kristian K Ullrich

See Also

XStringSet-class

Examples

```
## Not run:
## load example sequence data
## set NCBI URL
NCBI <- "ftp://ftp.ncbi.nlm.nih.gov/genomes/all/"
HOMSAP.NCBI.cds.url <- paste0(NCBI,
    "GCF/000/001/405/GCF_000001405.39_GRCh38.p13/",
    "GCF_000001405.39_GRCh38.p13_cds_from_genomic.fna.gz")
HOMSAP.NCBI.cds.file <- tempfile()
## download CDS file
download.file(HOMSAP.NCBI.cds.url, HOMSAP.NCBI.cds.file, quiet=FALSE)
HOMSAP.NCBI.cds <- Biostrings::readDNAStringSet(HOMSAP.NCBI.cds.file)
## get longest isoform
HOMSAP.NCBI.cds.longest <- isoform2longest(HOMSAP.NCBI.cds, "NCBI")
length(HOMSAP.NCBI.cds)
length(HOMSAP.NCBI.cds.longest)
## set ENSEMBL URL
ensembl <- "ftp://ftp.ensembl.org/pub/release-101/fasta/"
## set Homo sapiens CDS URL
HOMSAP.ENSEMBL.cds.url <- paste0(ensembl,
    "homo_sapiens/cds/Homo_sapiens.GRCh38.cds.all.fa.gz")
HOMSAP.ENSEMBL.cds.file <- tempfile()
## download CDS file
download.file(HOMSAP.ENSEMBL.cds.url, HOMSAP.ENSEMBL.cds.file, quiet=FALSE)
HOMSAP.ENSEMBL.cds <- Biostrings::readDNAStringSet(HOMSAP.ENSEMBL.cds.file)
## get longest isoform
HOMSAP.ENSEMBL.cds.longest<-isoform2longest(HOMSAP.ENSEMBL.cds, "ENSEMBL")
length(HOMSAP.ENSEMBL.cds)
length(HOMSAP.ENSEMBL.cds.longest)

## End(Not run)
```

make_dagchainer *make_dagchainer*

Description

This function tries to build the DAGchainer from source code forked within CRBHits

Usage

```
make_dagchainer(build_dir = file.path(find.package("CRBHits"), "extdata"))
```

Arguments

build_dir	directory to build DAGchainer [mandatory]

Value

compile DAGchainer and return binary path

Author(s)

Kristian K Ullrich

References

Haas BJ et al. (2004) DAGchainer: a tool for mining segmental genome duplications and synteny. **Bioinformatics 20 (18)**, 3643-6.

make_last *make_last*

Description

This function tries to build the prerequisite last-1639 from source code forked within CRBHits

Usage

```
make_last(build_dir = file.path(find.package("CRBHits"), "extdata"))
```

Arguments

build_dir	directory to build last [mandatory]

Value

compile last and return binary path

Author(s)

Kristian K Ullrich

References

Kiełbasa SM et al. (2011) Adaptive seeds tame genomic sequence comparison. **Genome Res. 21 (3)**, 487-93.

`make_vignette` *make_vignette*

Description

This function tries to build the prerequisite last-1639, and DAGchainer from source code forked within CRBHits

Usage

```
make_vignette()
```

Value

path of prerequisites

Author(s)

Kristian K Ullrich

References

Kiełbasa SM et al. (2011) Adaptive seeds tame genomic sequence comparison. **Genome Res. 21 (3)**, 487-93.

Wang D, Zhang Y et al. (2010) KaKs_Calculator 2.0: a toolkit incorporating gamma-series methods and sliding window strategies. *Genomics Proteomics Bioinformatics.* **8(1)**, 77-80.

Haas BJ et al. (2004) DAGchainer: a tool for mining segmental genome duplications and synteny. **Bioinformatics 20 (18)**, 3643-6.

plot_dagchainer *plot_dagchainer*

Description

This function plots DAGchainer (http://dagchainer.sourceforge.net/) results obtained via ‘rbh2dagchainer()‘ function.

Usage

```
plot_dagchainer(
  dag,
  DotPlotTitle = "DAGchainer results",
  colorBy = "none",
  kaks = NULL,
  ka.max = 5,
  ks.max = 5,
  ka.min = 0,
  ks.min = 0,
  select.chr = NULL
)
```

Arguments

dag	specify DAGchainer results as obtained via ‘rbh2dagchainer()‘ [mandatory]
DotPlotTitle	specify DotPlot title [default: DAGchainer results]
colorBy	specify if dagchainer groups should be colored by "Ka", "Ks", "Ka/Ks" or "none" [default: none]
kaks	specify Ka/Ks input obtained via ‘rbh2kaks()‘ [default: NULL]
ka.max	specify max Ka to be filtered [default: 5]
ks.max	specify max Ks to be filtered [default: 5]
ka.min	specify min Ka to be filtered [default: 0]
ks.min	specify min Ks to be filtered [default: 0]
select.chr	filter results for chromosome names [default: NULL]

Value

synteny plot

Author(s)

Kristian K Ullrich

Examples

```
## load example sequence data
data("ath_aly_ncbi_dagchainer", package="CRBHits")
## plot DAGchainer results - default
plot_dagchainer(
    dag=ath_aly_ncbi_dagchainer)
## chromosome subset
plot_dagchainer(
    dag=ath_aly_ncbi_dagchainer,
    select.chr=c("NC_000932.1", "NC_034379.1"))
```

---

plot_kaks *plot_kaks*

Description

This function plots Ka/Ks results obtained via 'rbh2kaks()' function or 'MSA2dist::dnastring2kaks()' function.

Usage

```
plot_kaks(
  kaks,
  dag = NULL,
  gene.position.cds1 = NULL,
  gene.position.cds2 = NULL,
  tandem.dups.cds1 = NULL,
  tandem.dups.cds2 = NULL,
  PlotTitle = "Ka/Ks results",
  PlotType = "h",
  binw = 0.05,
  splitByChr = FALSE,
  colorBy = "none",
  ka.max = 5,
  ks.max = 5,
  ka.min = 0,
  ks.min = 0,
  select.chr = NULL,
  doPlot = TRUE
)
```

Arguments

kaks	specify Ka/Ks input obtained via 'rbh2kaks()' [mandatory]
dag	specify DAGchainer results as obtained via 'rbh2dagchainer()' [default: NULL]
gene.position.cds1	specify gene position for cds1 sequences (see cds2genepos) [default: NULL]

`gene.position.cds2`
: specify gene position for cds2 sequences (see `cds2genepos`) [default: NULL]

`tandem.dups.cds1`
: specify tandem duplicates for cds1 sequences (see `tandemdups`) [default: NULL]

`tandem.dups.cds2`
: specify tandem duplicates for cds2 sequences (see `tandemdups`) [default: NULL]

`PlotTitle`
: specify Plot title [default: Ka/Ks results]

`PlotType`
: specify Plot type: "h" histogram or "d" dotplot [default: h]

`binw`
: specify binwidth (see `geom_histogram`) [default: 0.05]

`splitByChr`
: specify if plot should be split by chromosome [default: FALSE]

`colorBy`
: specify if Ka/Ks gene pairs should be colored by "rbh_class", "dagchainer", "tandemdups" or "none" [default: rbh_class]

`ka.max`
: specify max Ka to be filtered [default: 5]

`ks.max`
: specify max Ks to be filtered [default: 5]

`ka.min`
: specify min Ka to be filtered [default: 0]

`ks.min`
: specify min Ks to be filtered [default: 0]

`select.chr`
: filter results for chromosome names [default: NULL]

`doPlot`
: specify plot [default: TRUE]

Value

Ka/Ks plot

Author(s)

Kristian K Ullrich

Examples

```
## load example sequence data
data("ath_aly_ncbi_kaks", package="CRBHits")
## plot Ka/Ks values - default
g <- plot_kaks(ath_aly_ncbi_kaks)
## Calculate Ka/Ks values based on MSA
data("hiv", package="MSA2dist")
hiv_kaks <- MSA2dist::dnastring2kaks(hiv)
g <- plot_kaks(hiv_kaks)
```

rbh2dagchainer *rbh2dagchainer*

Description

This function runs DAGchainer (http://dagchainer.sourceforge.net/) given CRBHit pairs and gene positions for both cds1 and cds2. The default options are set to not compare gene positions in base pairs but instead using gene order (gene.idx).

Usage

```
rbh2dagchainer(
  rbhpairs,
  selfblast1 = NULL,
  selfblast2 = NULL,
  gene.position.cds1 = NULL,
  gene.position.cds2 = NULL,
  dagchainerpath = file.path(find.package("CRBHits"), "extdata", "dagchainer"),
  gap_open_penalty = 0,
  gap_extension_penalty = -3,
  gap_length = 10000,
  max_match_score = 50,
  max_dist_allowed = 2e+05,
  max_evalue = 0.001,
  ignore_tandem = TRUE,
  only_tandem = FALSE,
  min_number_aligned_pairs = 5,
  type = "bp",
  plotDotPlot = FALSE,
  DotPlotTitle = "DAGchainer results",
  colorBy = "none",
  kaks = NULL,
  ka.max = 5,
  ks.max = 5,
  ka.min = 0,
  ks.min = 0,
  select.chr = NULL
)
```

Arguments

rbhpairs	(conditional-)reciprocal best hit (CRBHit) pair result (see cds2rbh) [mandatory]
selfblast1	(conditional-)reciprocal best hit (CRBHit) pair selfblast result for cds1 sequences (see cds2rbh) [optional]
selfblast2	(conditional-)reciprocal best hit (CRBHit) pair selfblast result for cds2 sequences (see cds2rbh) [optional]

`gene.position.cds1`
: specify gene position for cds1 sequences (see `cds2genepos`) [default: NULL]

`gene.position.cds2`
: specify gene position for cds2 sequences (see `cds2genepos`) [default: NULL]

`dagchainerpath`
: specify the PATH to the DAGchainer directory containing the dagchainer binaries [default: /extdata/dagchainer/]

`gap_open_penalty`
: gap open penalty [default: 0]

`gap_extension_penalty`
: gap extension penalty [default: -3]

`gap_length`
: length of a gap (avgerage distance expected between two syntenic genes); if type is set to "idx" use 1 [default: 10000]

`max_match_score`
: Maximum match score [default: 50]

`max_dist_allowed`
: maximum distance allowed between two matches; if type is set to "idx" use 20 [default: 200000]

`max_evalue`
: Maximum E-value [default: 1e-3]

`ignore_tandem`
: ignore tandem duplicates [default = TRUE]

`only_tandem`
: only tandem alignments [default = FALSE]

`min_number_aligned_pairs`
: Minimum number of Aligned Pairs [default: 5]

`type`
: specify if gene order index "idx" or gene base pair position "bp" should be extracted and used with DAGchainer [default: bp]

`plotDotPlot`
: specify if dotplot should be plotted [default: FALSE]

`DotPlotTitle`
: specify DotPlot title [default: 'DAGchainer results']

`colorBy`
: specify if dagchainer groups should be colored by "Ka", "Ks", "Ka/Ks" or "none" [default: none]

`kaks`
: specify Ka/Ks input obtained via 'rbh2kaks()' [default: NULL]

`ka.max`
: specify max Ka to be filtered [default: 5]

`ks.max`
: specify max Ks to be filtered [default: 5]

`ka.min`
: specify min Ka to be filtered [default: 0]

`ks.min`
: specify min Ks to be filtered [default: 0]

`select.chr`
: filter results for chromosome names [default: NULL]

Value

`DAGchanier` results
1: $gene1.chr
2: $gene1.seq.id
3: $gene1.start
4: $gene1.end
5: $gene1.mid

6: $gene1.idx
7: $gene2.chr
8: $gene2.seq.id
9: $gene2.start
10: $gene2.end
11: $gene2.mid
12: $gene2.idx
13: $evalue
14: $score

Author(s)

Kristian K Ullrich

References

Haas BJ et al. (2004) DAGchainer: a tool for mining segmental genome duplications and synteny. *Bioinformatics.* **20(18)**, 3643-3646.

See Also

plot_dagchainer, cds2genepos, tandemdups, rbh2kaks

Examples

```
## compile dagchainer
CRBHits::make_dagchainer()
## load example sequence data
data("ath", package="CRBHits")
## get selfhits CRBHit pairs
ath_selfhits_crbh <- cds2rbh(
    cds1=ath,
    cds2=ath,
    plotCurve=TRUE)
## get gene position
ath.genepos <- cds2genepos(
    cds=ath,
    source="ENSEMBL")
## get DAGchainer results
ath_selfblast_crbh.dagchainer <- rbh2dagchainer(
    rbhpairs=ath_selfhits_crbh,
    gene.position.cds1=ath.genepos,
    gene.position.cds2=ath.genepos)
head(ath_selfblast_crbh.dagchainer)
## plot dagchainer
plot_dagchainer(
    dag=ath_selfblast_crbh.dagchainer)
```

rbh2kaks *rbh2kaks*

Description

This function calculates Ka/Ks (dN/dS; accoring to *Li (1993)* or *Yang and Nielson (2000)* for each (conditional-)reciprocal best hit (CRBHit) pair. The names of the `rbh` columns must match the names of the corresponding `cds1` and `cds2` `DNAStringSet` vectors.

Usage

```
rbh2kaks(
  rbhpairs,
  cds1,
  cds2,
  model = "Li",
  plotHistPlot = FALSE,
  plotDotPlot = FALSE,
  dag = NULL,
  gene.position.cds1 = NULL,
  gene.position.cds2 = NULL,
  tandem.dups.cds1 = NULL,
  tandem.dups.cds2 = NULL,
  colorBy = "none",
  threads = 1,
  ...
)
```

Arguments

`rbhpairs`	(conditional-)reciprocal best hit (CRBHit) pair result (see `cds2rbh`) [mandatory]
`cds1`	cds1 sequences as `DNAStringSet` or `url` for first crbh pairs column [mandatory]
`cds2`	cds2 sequences as `DNAStringSet` or `url` for second crbh pairs column [mandatory]
`model`	specify codon model either "Li" or "NG86" or one of KaKs_Calculator2 model "NG", "LWL", "LPB", "MLWL", "MLPB", "GY", "YN", "MYN", "MS", "MA", "GNG", "GLWL", "GLPB", "GMLWL", "GMLPB", "GYN", "GMYN" [default: Li]
`plotHistPlot`	specify if histogram should be plotted [default: TRUE]
`plotDotPlot`	specify if dotplot should be plotted (mandatory to define `gene.position.cds1` and `gene.position.cds1`) [default: FALSE]
`dag`	specify DAGchainer results as obtained via 'rbh2dagchainer()' [default: NULL]
`gene.position.cds1`	specify gene position for cds1 sequences (see `cds2genepos`) [default: NULL]

`gene.position.cds2`
: specify gene position for cds2 sequences (see `cds2genepos`) [default: NULL]

`tandem.dups.cds1`
: specify tandem duplicates for cds1 sequences (see `tandemdups`) [default: NULL]

`tandem.dups.cds2`
: specify tandem duplicates for cds2 sequences (see `tandemdups`) [default: NULL]

`colorBy`
: specify if Ka/Ks gene pairs should be colored by "rbh_class", dagchainer", "tandemdups" or "none" [default: none]

`threads`
: number of parallel threads [default: 1]

`...`
: other codon alignment parameters (see `cds2codonaln`) and other plot_kaks parameters (see `plot_kaks`)

Value

Ka/Ks values

Author(s)

Kristian K Ullrich

References

Li WH. (1993) Unbiased estimation of the rates of synonymous and nonsynonymous substitution. *J. Mol. Evol.*, **36**, 96-99.

Wang D, Zhang Y et al. (2010) KaKs_Calculator 2.0: a toolkit incorporating gamma-series methods and sliding window strategies. *Genomics Proteomics Bioinformatics.* **8(1)**, 77-80.

Yang Z and Nielson R. (2000) Estimating synonymous and nonsynonymous substitution rates under realistic evolutionary models. *Mol. Biol. Evol.*, **17(1)**, 32-43.

See Also

`dnastring2kaks`, `isoform2longest`, `cds2genepos`, `plot_kaks`, `rbh2dagchainer`, `tandemdups`

Examples

```
## load example sequence data
data("ath", package="CRBHits")
data("aly", package="CRBHits")
## load example CRBHit pairs
data("ath_aly_crbh", package="CRBHits")
## only analyse subset of CRBHit pairs
ath_aly_crbh$crbh.pairs <- head(ath_aly_crbh$crbh.pairs)
ath_aly_crbh.kaks <- rbh2kaks(
    rbhpairs=ath_aly_crbh,
    cds1=ath,
    cds2=aly,
    model="Li")
head(ath_aly_crbh.kaks)
## plot kaks
g.kaks <- plot_kaks(ath_aly_crbh.kaks)
```

seqid *seqid*

Description

This function returns the first whitespace separated names value of a XStringSet.

Usage

```
seqid(x)
```

Arguments

x	XStringSet [mandatory]

Value

chr

Author(s)

Kristian K Ullrich

Examples

```
## load example sequence data
data("ath", package="CRBHits")
seqid(ath)
```

tandemdups *tandemdups*

Description

This function assigns tandem duplicates given (conditional-)reciprocal best hit (CRBHit) pairs and their chromosomal gene position. The function is ported into R from Haug-Baltzell et al. (2017) https://github.com/LyonsLab/coge/blob/master/bin/dagchainer/tandems.py

Usage

```
tandemdups(rbhpairs, genepos, dupdist = 5)
```

Arguments

rbhpairs	(conditional-)reciprocal best hit (CRBHit) pair result (see cds2rbh) [mandatory]
genepos	Gene position matrix as obtained from cds2genepos [mandatory]
dupdist	Maximum distance between 2 gene positions on the same chromosome which will call them as a pair of local duplicates. If they are farther apart than dupdist, but share a common hit within dupdist of both, then they will still be in the same set of local duplicates. [default: 5]

Value

matrix 1: $gene.seq.id
2: $gene.chr
3: $gene.start
4: $gene.end
5: $gene.mid
6: $gene.strand
7: $gene.idx
8: $tandem_group

Author(s)

Kristian K Ullrich

References

Haug-Beltzell A et al. (2017) SynMap2 and SynMap3D: web-based whole-genome synteny browsers. *Bioinformatics* **33(14)**, 2197-2198.

See Also

isoform2longest

Examples

```
## load example sequence data
data("ath", package="CRBHits")
## get selfhits CRBHit pairs
ath_selfhits_crbh <- cds2rbh(
    cds1=ath,
    cds2=ath,
    plotCurve=TRUE)
## get gene position
ath.genepos <- cds2genepos(
    cds=ath,
    source="ENSEMBL")
## get tandem duplicate results
ath_selfblast_crbh.tandemdups <- tandemdups(
    rbhpairs=ath_selfhits_crbh,
    genepos=ath.genepos)
```

```
head(ath_selfblast_crbh.tandemdups)
```

Index