Package 'phylotaR'

July 31, 2018

Type Package

Title Automated Phylogenetic Sequence Cluster Identification from 'GenBank'

Version 1.0.0

Maintainer Dom Bennett <dominic.john.bennett@gmail.com>

Description A pipeline for the identification, within taxonomic groups, of orthologous sequence clusters from 'GenBank' <https://www.ncbi.nlm.nih.gov/genbank/> as the first step in a phylogenetic analysis. The pipeline depends on a local alignment search tool and is, therefore, not dependent on differences in gene naming conventions and naming errors.

License MIT + file LICENSE

URL https://github.com/ropensci/phylotaR#readme

BugReports https://github.com/ropensci/phylotaR/issues

SystemRequirements BLAST+ (>=2.0)

Depends R (>= 3.3.0), methods,

Imports igraph, rentrez, XML, ggplot2, sys, treeman, treemapify, R.utils

Suggests testthat, knitr, rmarkdown

RoxygenNote 6.0.1

VignetteBuilder knitr

X-schema.org-isPartOf https://ropensci.org

NeedsCompilation no

Author Hannes Hettling [aut],
Rutger Vos [aut],
Alexander Zizka [aut],
Dom Bennett [aut, cre],
Alexandre Antonelli [aut]

Repository CRAN

Date/Publication 2018-07-31 15:50:03 UTC

R topics documented:

`aotus` *Example phylota object (aotus)*

Description

Final phylota object produced by running pipeline for the night monkeys, Aotus (9504).

Usage

```
data(aotus)
```

Format

`aotus` is generated from read_phylota().

Examples

```
data(aotus) # load object
str(aotus)
```

batcher *Download in batches*

Description

Run downloader function in batches for sequences or taxonomic records

Usage

```
batcher(ids, func, ps, lvl = 0)
```

Arguments

ids	Vector of record ids
func	Downloader function
ps	Parameters list, generated with parameters()
lvl	Integer, number of message indentations indicating code depth.

Value

Vector of records

vector of rentrez function results

See Also

Other run-private: blast_clstr, blast_filter, blast_setup, blast_sqs, blastcache_load, blastcache_save, blastdb_gen, blastn_run, cache_rm, cache_setup, clade_select, clstr2_calc, clstr_all, clstr_direct, clstr_sqs, clstr_subtree, clstrarc_gen, clstrarc_join, clstrrec_gen, clstrs_calc, clstrs_join, clstrs_merge, clstrs_renumber, clstrs_save, cmdln, descendants_get, download_obj_check, error, hierarchic_download, info, ncbicache_load, ncbicache_save, obj_check, obj_load, obj_save, parameters_load, parameters_setup, parent_get, progress_init, progress_read, progress_reset, progress_save, rank_get, safely_connect, search_and_cache, searchterm_gen, seeds_blast, seq_download, seqarc_gen, seqrec_augment, seqrec_convert, seqrec_gen, seqrec_get, sids_check, sids_get, sids_load, sids_save, sqs_count, sqs_load, sqs_save, stage_args_check, stages_run, tax_download, taxdict_gen, taxtree_gen, txids_get, txnds_count, warn

blastcache_load *Load BLAST results from cache*

Description

Run to load cached BLAST results.

Usage

```
blastcache_load(sids, wd)
```

Arguments

sids	Sequence IDs
wd	Working dir

Value

blast_res data.frame or NULL

See Also

Other run-private: batcher, blast_clstr, blast_filter, blast_setup, blast_sqs, blastcache_save, blastdb_gen, blastn_run, cache_rm, cache_setup, clade_select, clstr2_calc, clstr_all, clstr_direct, clstr_sqs, clstr_subtree, clstrarc_gen, clstrarc_join, clstrrec_gen, clstrs_calc, clstrs_join, clstrs_merge, clstrs_renumber, clstrs_save, cmdln, descendants_get, download_obj_check, error, hierarchic_download, info, ncbicache_load, ncbicache_save, obj_check, obj_load, obj_save, parameters_load, parameters_setup, parent_get, progress_init, progress_read, progress_reset, progress_save, rank_get, safely_connect, search_and_cache, searchterm_gen, seeds_blast, seq_download, seqarc_gen, seqrec_augment, seqrec_convert, seqrec_gen, seqrec_get, sids_check, sids_get, sids_load, sids_save, sqs_count, sqs_load, sqs_save, stage_args_check, stages_run, tax_download, taxdict_gen, taxtree_gen, txids_get, txnds_count, warn

blastcache_save *Save BLAST results to cache*

Description

Run whenever local BLAST runs are made to save results in cache in case the pipeline is run again.

Usage

```
blastcache_save(sids, wd, obj)
```

Arguments

sids	Sequence IDs
wd	Working dir
obj	BLAST result

See Also

Other run-private: batcher, blast_clstr, blast_filter, blast_setup, blast_sqs, blastcache_load, blastdb_gen, blastn_run, cache_rm, cache_setup, clade_select, clstr2_calc, clstr_all, clstr_direct, clstr_sqs, clstr_subtree, clstrarc_gen, clstrarc_join, clstrrec_gen, clstrs_calc, clstrs_join, clstrs_merge, clstrs_renumber, clstrs_save, cmdln, descendants_get, download_obj_check, error, hierarchic_download, info, ncbicache_load, ncbicache_save, obj_check, obj_load, obj_save, parameters_load, parameters_setup, parent_get, progress_init, progress_read, progress_reset, progress_save, rank_get, safely_connect, search_and_cache, searchterm_gen, seeds_blast, seq_download, seqarc_gen, seqrec_augment, seqrec_convert, seqrec_gen, seqrec_get, sids_check, sids_get, sids_load, sids_save, sqs_count, sqs_load, sqs_save, stage_args_check, stages_run, tax_download, taxdict_gen, taxtree_gen, txids_get, txnds_count, warn

---

blastdb_gen *Generate a BLAST database*

Description

Generate BLAST database in wd for given sequences.

Usage

```
blastdb_gen(sqs, dbfl, ps)
```

Arguments

sqs	Sequences
dbfl	Outfile for database
ps	Parameters list, generated with parameters()

See Also

Other run-private: batcher, blast_clstr, blast_filter, blast_setup, blast_sqs, blastcache_load, blastcache_save, blastn_run, cache_rm, cache_setup, clade_select, clstr2_calc, clstr_all, clstr_direct, clstr_sqs, clstr_subtree, clstrarc_gen, clstrarc_join, clstrrec_gen, clstrs_calc, clstrs_join, clstrs_merge, clstrs_renumber, clstrs_save, cmdln, descendants_get, download_obj_check, error, hierarchic_download, info, ncbicache_load, ncbicache_save, obj_check, obj_load, obj_save, parameters_load, parameters_setup, parent_get, progress_init, progress_read, progress_reset, progress_save, rank_get, safely_connect, search_and_cache, searchterm_gen, seeds_blast, seq_download, seqarc_gen, seqrec_augment, seqrec_convert, seqrec_gen, seqrec_get, sids_check, sids_get, sids_load, sids_save, sqs_count, sqs_load, sqs_save, stage_args_check, stages_run, tax_download, taxdict_gen, taxtree_gen, txids_get, txnds_count, warn

blastn_run *Launch blastn*

Description

Use `blastn` to BLAST all-vs-all using a BLAST database.

Usage

```
blastn_run(dbfl, outfl, ps)
```

Arguments

`dbfl`	Database file
`outfl`	Output file
`ps`	Parameters list, generated with parameters()

See Also

Other run-private: `batcher`, `blast_clstr`, `blast_filter`, `blast_setup`, `blast_sqs`, `blastcache_load`, `blastcache_save`, `blastdb_gen`, `cache_rm`, `cache_setup`, `clade_select`, `clstr2_calc`, `clstr_all`, `clstr_direct`, `clstr_sqs`, `clstr_subtree`, `clstrarc_gen`, `clstrarc_join`, `clstrrec_gen`, `clstrs_calc`, `clstrs_join`, `clstrs_merge`, `clstrs_renumber`, `clstrs_save`, `cmdln`, `descendants_get`, `download_obj_check`, `error`, `hierarchic_download`, `info`, `ncbicache_load`, `ncbicache_save`, `obj_check`, `obj_load`, `obj_save`, `parameters_load`, `parameters_setup`, `parent_get`, `progress_init`, `progress_read`, `progress_reset`, `progress_save`, `rank_get`, `safely_connect`, `search_and_cache`, `searchterm_gen`, `seeds_blast`, `seq_download`, `seqarc_gen`, `seqrec_augment`, `seqrec_convert`, `seqrec_gen`, `seqrec_get`, `sids_check`, `sids_get`, `sids_load`, `sids_save`, `sqs_count`, `sqs_load`, `sqs_save`, `stage_args_check`, `stages_run`, `tax_download`, `taxdict_gen`, `taxtree_gen`, `txids_get`, `txnds_count`, `warn`

blast_clstr *Cluster BLAST Results*

Description

Find single-linkage clusters from BLAST results. Identifies seed sequence.

Usage

```
blast_clstr(blast_res)
```

Arguments

`blast_res`	BLAST results

Value

List of list

list of cluster descriptions

See Also

Other run-private: batcher, blast_filter, blast_setup, blast_sqs, blastcache_load, blastcache_save, blastdb_gen, blastn_run, cache_rm, cache_setup, clade_select, clstr2_calc, clstr_all, clstr_direct, clstr_sqs, clstr_subtree, clstrarc_gen, clstrarc_join, clstrrec_gen, clstrs_calc, clstrs_join, clstrs_merge, clstrs_renumber, clstrs_save, cmdln, descendants_get, download_obj_check, error, hierarchic_download, info, ncbicache_load, ncbicache_save, obj_check, obj_load, obj_save, parameters_load, parameters_setup, parent_get, progress_init, progress_read, progress_reset, progress_save, rank_get, safely_connect, search_and_cache, searchterm_gen, seeds_blast, seq_download, seqarc_gen, seqrec_augment, seqrec_convert, seqrec_gen, seqrec_get, sids_check, sids_get, sids_load, sids_save, sqs_count, sqs_load, sqs_save, stage_args_check, stages_run, tax_download, taxdict_gen, taxtree_gen, txids_get, txnds_count, warn

---

`blast_filter` *Filter BLAST results*

Description

Given a BLAST output, filters query-subject pairs such that only HSPs with a coverage greater than `mncvrg` (specified in the pipeline parameters) remain. Filters both: query-subject and subject-query pairs, if one of the coverages is insufficient. HSP coverage is obtained from the BLAST column `qcovs`.

Usage

```
blast_filter(blast_res, ps)
```

Arguments

`blast_res`	BLAST results
`ps`	Parameters list, generated with parameters()

Value

data.frame blast res

See Also

Other run-private: batcher, blast_clstr, blast_setup, blast_sqs, blastcache_load, blastcache_save, blastdb_gen, blastn_run, cache_rm, cache_setup, clade_select, clstr2_calc, clstr_all, clstr_direct, clstr_sqs, clstr_subtree, clstrarc_gen, clstrarc_join, clstrrec_gen, clstrs_calc, clstrs_join, clstrs_merge, clstrs_renumber, clstrs_save, cmdln, descendants_get, download_obj_check, error, hierarchic_download, info, ncbicache_load, ncbicache_save, obj_check, obj_load,

obj_save, parameters_load, parameters_setup, parent_get, progress_init, progress_read, progress_reset, progress_save, rank_get, safely_connect, search_and_cache, searchterm_gen, seeds_blast, seq_download, seqarc_gen, seqrec_augment, seqrec_convert, seqrec_gen, seqrec_get, sids_check, sids_get, sids_load, sids_save, sqs_count, sqs_load, sqs_save, stage_args_check, stages_run, tax_download, taxdict_gen, taxtree_gen, txids_get, txnds_count, warn

---

`blast_setup` *Ensures NCBI BLAST tools are installed*

Description

Ensures NCBI BLAST executables are installed on the system. Tests version number of BLAST tools.

Usage

```
blast_setup(d, v, wd)
```

Arguments

d	Directory to NCBI BLAST tools
v	v, T/F
wd	Working directory

Details

BLAST tools must be version >= 2.0

Value

list

See Also

Other run-private: batcher, blast_clstr, blast_filter, blast_sqs, blastcache_load, blastcache_save, blastdb_gen, blastn_run, cache_rm, cache_setup, clade_select, clstr2_calc, clstr_all, clstr_direct, clstr_sqs, clstr_subtree, clstrarc_gen, clstrarc_join, clstrrec_gen, clstrs_calc, clstrs_join, clstrs_merge, clstrs_renumber, clstrs_save, cmdln, descendants_get, download_obj_check, error, hierarchic_download, info, ncbicache_load, ncbicache_save, obj_check, obj_load, obj_save, parameters_load, parameters_setup, parent_get, progress_init, progress_read, progress_reset, progress_save, rank_get, safely_connect, search_and_cache, searchterm_gen, seeds_blast, seq_download, seqarc_gen, seqrec_augment, seqrec_convert, seqrec_gen, seqrec_get, sids_check, sids_get, sids_load, sids_save, sqs_count, sqs_load, sqs_save, stage_args_check, stages_run, tax_download, taxdict_gen, taxtree_gen, txids_get, txnds_count, warn

blast_sqs *BLAST All vs All*

Description

Return BLAST results from BLASTing all vs all for given sequences. Returns NULL if no BLAST results generated.

Usage

```
blast_sqs(txid, typ, sqs, ps, lvl)
```

Arguments

txid	Taxonomic node ID, numeric
typ	Cluster type, 'direct' or 'subtree'
sqs	Sequences
ps	Parameters list, generated with parameters()
lvl	Integer, number of message indentations indicating code depth.

Value

blast_res data.frame or NULL

See Also

Other run-private: batcher, blast_clstr, blast_filter, blast_setup, blastcache_load, blastcache_save, blastdb_gen, blastn_run, cache_rm, cache_setup, clade_select, clstr2_calc, clstr_all, clstr_direct, clstr_sqs, clstr_subtree, clstrarc_gen, clstrarc_join, clstrrec_gen, clstrs_calc, clstrs_join, clstrs_merge, clstrs_renumber, clstrs_save, cmdln, descendants_get, download_obj_check, error, hierarchic_download, info, ncbicache_load, ncbicache_save, obj_check, obj_load, obj_save, parameters_load, parameters_setup, parent_get, progress_init, progress_read, progress_reset, progress_save, rank_get, safely_connect, search_and_cache, searchterm_gen, seeds_blast, seq_download, seqarc_gen, seqrec_augment, seqrec_convert, seqrec_gen, seqrec_get, sids_check, sids_get, sids_load, sids_save, sqs_count, sqs_load, sqs_save, stage_args_check, stages_run, tax_download, taxdict_gen, taxtree_gen, txids_get, txnds_count, warn

bromeliads *Example phylota object (bromeliads)*

Description

Final phylota object produced by running pipeline for the bromeliads, Bromeliaceae (4613).

Usage

```
data(bromeliads)
```

Format

bromeliads is generated from read_phylota().

Examples

```
data(bromeliads) # load object
str(bromeliads)
```

cache_rm *Delete a cache*

Description

Deletes a cache from a wd.

Usage

```
cache_rm(wd)
```

Arguments

wd	Working directory

See Also

Other run-private: batcher, blast_clstr, blast_filter, blast_setup, blast_sqs, blastcache_load, blastcache_save, blastdb_gen, blastn_run, cache_setup, clade_select, clstr2_calc, clstr_all, clstr_direct, clstr_sqs, clstr_subtree, clstrarc_gen, clstrarc_join, clstrrec_gen, clstrs_calc, clstrs_join, clstrs_merge, clstrs_renumber, clstrs_save, cmdln, descendants_get, download_obj_check, error, hierarchic_download, info, ncbicache_load, ncbicache_save, obj_check, obj_load, obj_save, parameters_load, parameters_setup, parent_get, progress_init, progress_read, progress_reset, progress_save, rank_get, safely_connect, search_and_cache, searchterm_gen, seeds_blast, seq_download, seqarc_gen, seqrec_augment, seqrec_convert, seqrec_gen, seqrec_get, sids_check, sids_get, sids_load, sids_save, sqs_count, sqs_load, sqs_save, stage_args_check, stages_run, tax_download, taxdict_gen, taxtree_gen, txids_get, txnds_count, warn

cache_setup *Set-up a cache*

Description

Creates a cache of parameters in the wd.

Usage

```
cache_setup(ps, ovrwrt = FALSE)
```

Arguments

ps	Parameters list, generated with parameters()
ovrwrt	Overwrite existing cache? Default FALSE.

Details

Warning: overwriting with this function will delete the existing cache.

See Also

Other run-private: batcher, blast_clstr, blast_filter, blast_setup, blast_sqs, blastcache_load, blastcache_save, blastdb_gen, blastn_run, cache_rm, clade_select, clstr2_calc, clstr_all, clstr_direct, clstr_sqs, clstr_subtree, clstrarc_gen, clstrarc_join, clstrrec_gen, clstrs_calc, clstrs_join, clstrs_merge, clstrs_renumber, clstrs_save, cmdln, descendants_get, download_obj_check, error, hierarchic_download, info, ncbicache_load, ncbicache_save, obj_check, obj_load, obj_save, parameters_load, parameters_setup, parent_get, progress_init, progress_read, progress_reset, progress_save, rank_get, safely_connect, search_and_cache, searchterm_gen, seeds_blast, seq_download, seqarc_gen, seqrec_augment, seqrec_convert, seqrec_gen, seqrec_get, sids_check, sids_get, sids_load, sids_save, sqs_count, sqs_load, sqs_save, stage_args_check, stages_run, tax_download, taxdict_gen, taxtree_gen, txids_get, txnds_count, warn

calc_mad *Calculate MAD score*

Description

For all sequences in a cluster(s) the MAD score.

Usage

```
calc_mad(phylota, cid)
```

Arguments

phylota	Phylota object
cid	Cluster ID(s)

Details

MAD is a measure of the deviation in sequence length of a cluster. Values range from 0 to 1. Clusters with values close to 1 have sequences with similar lengths.

Value

vector

See Also

Other tools-public: calc_wrdfrq, drop_by_rank, drop_clstrs, drop_sqs, get_clstr_slot, get_nsqs, get_ntaxa, get_sq_slot, get_stage_times, get_tx_slot, get_txids, is_txid_in_clstr, is_txid_in_sq, list_clstrrec_slots, list_ncbi_ranks, list_seqrec_slots, list_taxrec_slots, plot_phylota_pa, plot_phylota_treemap, read_phylota, write_sqs

Examples

```
data("bromeliads")
random_cids <- sample(bromeliads@cids, 10)
(calc_mad(phylota = bromeliads, cid = random_cids))
```

---

calc_wrdfrq *Calculate word frequencies*

Description

For all sequences in a cluster(s) calculate the frequency of separate words in either the sequence definitions or the reported feature name.

Usage

```
calc_wrdfrq(phylota, cid, min_frq = 0.1, min_nchar = 1, type = c("dfln",
  "nm"), ignr_pttrn = "[^a-z0-9]")
```

Arguments

phylota	Phylota object
cid	Cluster ID(s)
min_frq	Minimum frequency
min_nchar	Minimum number of characters for a word
type	Definitions (dfln) or features (nm)
ignr_pttrn	Ignore pattern, REGEX for text to ignore.

Details

By default, anything that is not alphanumeric is ignored. 'dfln' and 'nm' match the slot names in a SeqRec, see list_seqrec_slots().

Value

list

See Also

Other tools-public: calc_mad, drop_by_rank, drop_clstrs, drop_sqs, get_clstr_slot, get_nsqs, get_ntaxa, get_sq_slot, get_stage_times, get_tx_slot, get_txids, is_txid_in_clstr, is_txid_in_sq, list_clstrrec_slots, list_ncbi_ranks, list_seqrec_slots, list_taxrec_slots, plot_phylota_pa, plot_phylota_treemap, read_phylota, write_sqs

Examples

```
data('dragonflies')
# work out what gene region the cluster is likely representing with word freqs.
random_cids <- sample(dragonflies@cids, 10)
# most frequent words in definition line
(calc_wrdfrq(phylota = dragonflies, cid = random_cids, type = 'dfln'))
# most frequent words in feature name
(calc_wrdfrq(phylota = dragonflies, cid = random_cids, type = 'nm'))
```

---

clade_select *Get all node IDs that will be processed*

Description

All nodes with less than maximum number of nodes and sequences.

Usage

```
clade_select(txdct, ps)
```

Arguments

txdct	TxDct
ps	Parameters list, generated with parameters()

Value

vector of txids

See Also

Other run-private: batcher, blast_clstr, blast_filter, blast_setup, blast_sqs, blastcache_load, blastcache_save, blastdb_gen, blastn_run, cache_rm, cache_setup, clstr2_calc, clstr_all, clstr_direct, clstr_sqs, clstr_subtree, clstrarc_gen, clstrarc_join, clstrrec_gen, clstrs_calc, clstrs_join, clstrs_merge, clstrs_renumber, clstrs_save, cmdln, descendants_get, download_obj_check, error, hierarchic_download, info, ncbicache_load, ncbicache_save, obj_check, obj_load, obj_save, parameters_load, parameters_setup, parent_get, progress_init, progress_read, progress_reset, progress_save, rank_get, safely_connect, search_and_cache, searchterm_gen, seeds_blast, seq_download, seqarc_gen, seqrec_augment, seqrec_convert, seqrec_gen, seqrec_get, sids_check, sids_get, sids_load, sids_save, sqs_count, sqs_load, sqs_save, stage_args_check, stages_run, tax_download, taxdict_gen, taxtree_gen, txids_get, txnds_count, warn

---

`clstr2_calc` *Cluster sets of clusters identified in cluster stage*

Description

Loads cluster sets from cache. Extracts seed sequences and runs all-v-all BLAST of seeds to identify sister clusters. Sisters are then merged. An object of all sequences and clusters is then saved in cache.

Usage

```
clstr2_calc(ps)
```

Arguments

`ps`	Parameters list, generated with parameters()

See Also

Other run-private: batcher, blast_clstr, blast_filter, blast_setup, blast_sqs, blastcache_load, blastcache_save, blastdb_gen, blastn_run, cache_rm, cache_setup, clade_select, clstr_all, clstr_direct, clstr_sqs, clstr_subtree, clstrarc_gen, clstrarc_join, clstrrec_gen, clstrs_calc, clstrs_join, clstrs_merge, clstrs_renumber, clstrs_save, cmdln, descendants_get, download_obj_check, error, hierarchic_download, info, ncbicache_load, ncbicache_save, obj_check, obj_load, obj_save, parameters_load, parameters_setup, parent_get, progress_init, progress_read, progress_reset, progress_save, rank_get, safely_connect, search_and_cache, searchterm_gen, seeds_blast, seq_download, seqarc_gen, seqrec_augment, seqrec_convert, seqrec_gen, seqrec_get, sids_check, sids_get, sids_load, sids_save, sqs_count, sqs_load, sqs_save, stage_args_check, stages_run, tax_download, taxdict_gen, taxtree_gen, txids_get, txnds_count, warn

ClstrArc-class *Cluster record archive*

Description

Multiple cluster records.

Usage

```
## S4 method for signature 'ClstrArc'
as.character(x)

## S4 method for signature 'ClstrArc'
show(object)

## S4 method for signature 'ClstrArc'
print(x)

## S4 method for signature 'ClstrArc'
str(object, max.level = 2L, ...)

## S4 method for signature 'ClstrArc'
summary(object)

## S4 method for signature 'ClstrArc,character'
x[[i]]

## S4 method for signature 'ClstrArc,character,missing,missing'
x[i, j, ..., drop = TRUE]
```

Arguments

x	ClstrArc object
object	ClstrArc object
max.level	Maximum level of nesting for str()
...	Further arguments for str()
i	cid(s)
j	Unused
drop	Unused

Slots

ids Vector of cluster record IDs

clstrs List of ClstrArc named by ID

See Also

Other run-public: ClstrRec-class, Phylota-class, SeqArc-class, SeqRec-class, TaxDict-class, TaxRec-class, clusters2_run, clusters_run, parameters_reset, reset, restart, run, setup, taxise_run

Examples

```
data('aotus')
clstrarc <- aotus@clstrs
# this is a ClstrArc object
# it contains cluster records
show(clstrarc)
# you can access its different data slots with @
clstrarc@ids     # unique cluster ID
clstrarc@clstrs  # list of cluster records
# access cluster records [[
(clstrarc[[clstrarc@ids[[1]]]])  # first cluster record
# generate new cluster archives with [
(clstrarc[clstrarc@ids[1:10]])  # first 10 clusters
```

---

clstrarc_gen *Generate cluster archive container class*

Description

Takes a list of ClstrRecs, returns a ClstrArc.

Usage

```
clstrarc_gen(clstrrecs)
```

Arguments

clstrrecs	list of ClstrRecs

Value

ClstrArc

See Also

Other run-private: batcher, blast_clstr, blast_filter, blast_setup, blast_sqs, blastcache_load, blastcache_save, blastdb_gen, blastn_run, cache_rm, cache_setup, clade_select, clstr2_calc, clstr_all, clstr_direct, clstr_sqs, clstr_subtree, clstrarc_join, clstrrec_gen, clstrs_calc, clstrs_join, clstrs_merge, clstrs_renumber, clstrs_save, cmdln, descendants_get, download_obj_check, error, hierarchic_download, info, ncbicache_load, ncbicache_save, obj_check, obj_load, obj_save, parameters_load, parameters_setup, parent_get, progress_init, progress_read, progress_reset, progress_save, rank_get, safely_connect, search_and_cache, searchterm_gen,

seeds_blast, seq_download, seqarc_gen, seqrec_augment, seqrec_convert, seqrec_gen, seqrec_get, sids_check, sids_get, sids_load, sids_save, sqs_count, sqs_load, sqs_save, stage_args_check, stages_run, tax_download, taxdict_gen, taxtree_gen, txids_get, txnds_count, warn

---

clstrarc_join *Join two cluster archive*

Description

Take two ClstrArc classes and join them into a single ClstrArc.

Usage

```
clstrarc_join(clstrarc_1, clstrarc_2)
```

Arguments

clstrarc_1	ClstrArc
clstrarc_2	ClstrArc

Value

ClstrArc

See Also

Other run-private: batcher, blast_clstr, blast_filter, blast_setup, blast_sqs, blastcache_load, blastcache_save, blastdb_gen, blastn_run, cache_rm, cache_setup, clade_select, clstr2_calc, clstr_all, clstr_direct, clstr_sqs, clstr_subtree, clstrarc_gen, clstrrec_gen, clstrs_calc, clstrs_join, clstrs_merge, clstrs_renumber, clstrs_save, cmdln, descendants_get, download_obj_check, error, hierarchic_download, info, ncbicache_load, ncbicache_save, obj_check, obj_load, obj_save, parameters_load, parameters_setup, parent_get, progress_init, progress_read, progress_reset, progress_save, rank_get, safely_connect, search_and_cache, searchterm_gen, seeds_blast, seq_download, seqarc_gen, seqrec_augment, seqrec_convert, seqrec_gen, seqrec_get, sids_check, sids_get, sids_load, sids_save, sqs_count, sqs_load, sqs_save, stage_args_check, stages_run, tax_download, taxdict_gen, taxtree_gen, txids_get, txnds_count, warn

`ClstrRec-class` *Cluster record*

Description

Cluster record contains all information on a cluster.

Usage

```
## S4 method for signature 'ClstrRec'
as.character(x)

## S4 method for signature 'ClstrRec'
show(object)

## S4 method for signature 'ClstrRec'
print(x)

## S4 method for signature 'ClstrRec'
str(object, max.level = 2L, ...)

## S4 method for signature 'ClstrRec'
summary(object)
```

Arguments

`x`	`ClstrRec` object
`object`	`ClstrRec` object
`max.level`	Maximum level of nesting for str()
`...`	Further arguments for str()

Slots

`id` Cluster ID, integer

`sids` Sequence IDs

`nsqs` Number of sequences

`txids` Source txids for sequences

`ntx` Number of taxa

`typ` Cluster type: direct, subtree or merged

`seed` Seed sequence ID

`prnt` Parent taxonomic ID

See Also

Other run-public: ClstrArc-class, Phylota-class, SeqArc-class, SeqRec-class, TaxDict-class, TaxRec-class, clusters2_run, clusters_run, parameters_reset, reset, restart, run, setup, taxise_run

Examples

```
data('aotus')
clstrrec <- aotus@clstrs@clstrs[[1]]
# this is a ClstrRec object
# it contains cluster information
show(clstrrec)
# you can access its different data slots with @
clstrrec@id     # cluster id
clstrrec@sids   # sequence IDs
clstrrec@nsqs   # number of sequences
clstrrec@txids  # taxonomic IDs of sequences
clstrrec@ntx    # number unique taxonomic IDs
clstrrec@typ    # cluster type: merged, subtree, direct or paraphyly
clstrrec@prnt   # MRCA of all taxa
clstrrec@seed   # most inter-connected sequence
```

---

clstrrec_gen *Generate list of clusters*

Description

Takes a list of lists of cluster descriptions and generates ClstrRecs.

Usage

```
clstrrec_gen(clstr_list, txid, sqs, typ)
```

Arguments

clstr_list	List of list of cluster descriptions
txid	Taxonomic node ID
sqs	Sequence records
typ	Cluster type

Value

list of ClstrRecs

See Also

Other run-private: batcher, blast_clstr, blast_filter, blast_setup, blast_sqs, blastcache_load, blastcache_save, blastdb_gen, blastn_run, cache_rm, cache_setup, clade_select, clstr2_calc, clstr_all, clstr_direct, clstr_sqs, clstr_subtree, clstrarc_gen, clstrarc_join, clstrs_calc, clstrs_join, clstrs_merge, clstrs_renumber, clstrs_save, cmdln, descendants_get, download_obj_check, error, hierarchic_download, info, ncbicache_load, ncbicache_save, obj_check, obj_load, obj_save, parameters_load, parameters_setup, parent_get, progress_init, progress_read, progress_reset, progress_save, rank_get, safely_connect, search_and_cache, searchterm_gen, seeds_blast, seq_download, seqarc_gen, seqrec_augment, seqrec_convert, seqrec_gen, seqrec_get, sids_check, sids_get, sids_load, sids_save, sqs_count, sqs_load, sqs_save, stage_args_check, stages_run, tax_download, taxdict_gen, taxtree_gen, txids_get, txnds_count, warn

---

`clstrs_calc` *Calculate clusters for all sequences in wd*

Description

Loop through downloaded sequences for each clade and hierarchically find clusters using BLAST.

Usage

```
clstrs_calc(txdct, ps)
```

Arguments

`txdct`	Taxonomic dictionary
`ps`	Parameters list, generated with parameters()

See Also

Other run-private: batcher, blast_clstr, blast_filter, blast_setup, blast_sqs, blastcache_load, blastcache_save, blastdb_gen, blastn_run, cache_rm, cache_setup, clade_select, clstr2_calc, clstr_all, clstr_direct, clstr_sqs, clstr_subtree, clstrarc_gen, clstrarc_join, clstrrec_gen, clstrs_join, clstrs_merge, clstrs_renumber, clstrs_save, cmdln, descendants_get, download_obj_check, error, hierarchic_download, info, ncbicache_load, ncbicache_save, obj_check, obj_load, obj_save, parameters_load, parameters_setup, parent_get, progress_init, progress_read, progress_reset, progress_save, rank_get, safely_connect, search_and_cache, searchterm_gen, seeds_blast, seq_download, seqarc_gen, seqrec_augment, seqrec_convert, seqrec_gen, seqrec_get, sids_check, sids_get, sids_load, sids_save, sqs_count, sqs_load, sqs_save, stage_args_check, stages_run, tax_download, taxdict_gen, taxtree_gen, txids_get, txnds_count, warn

clstrs_join *Join clusters for merging*

Description

Uses seed sequence BLAST results and IDs to join clusters identified as sisters into single clusters. Resulting object is of joined clusters, merging is required to reformat the clusters for subsequent analysis.

Usage

```
clstrs_join(blast_res, seed_ids, all_clstrs, ps)
```

Arguments

blast_res	Seed sequence BLAST results
seed_ids	Seed sequence IDs
all_clstrs	List of all clusters
ps	Parameters list, generated with parameters()

Value

list of joined clusters

See Also

Other run-private: batcher, blast_clstr, blast_filter, blast_setup, blast_sqs, blastcache_load, blastcache_save, blastdb_gen, blastn_run, cache_rm, cache_setup, clade_select, clstr2_calc, clstr_all, clstr_direct, clstr_sqs, clstr_subtree, clstrarc_gen, clstrarc_join, clstrrec_gen, clstrs_calc, clstrs_merge, clstrs_renumber, clstrs_save, cmdln, descendants_get, download_obj_check, error, hierarchic_download, info, ncbicache_load, ncbicache_save, obj_check, obj_load, obj_save, parameters_load, parameters_setup, parent_get, progress_init, progress_read, progress_reset, progress_save, rank_get, safely_connect, search_and_cache, searchterm_gen, seeds_blast, seq_download, seqarc_gen, seqrec_augment, seqrec_convert, seqrec_gen, seqrec_get, sids_check, sids_get, sids_load, sids_save, sqs_count, sqs_load, sqs_save, stage_args_check, stages_run, tax_download, taxdict_gen, taxtree_gen, txids_get, txnds_count, warn

clstrs_merge *Merge joined clusters*

Description

Takes a list of joined clusters and computes each data slot to create a single merged cluster. txdct is required for parent look-up.

Usage

```
clstrs_merge(jnd_clstrs, txdct)
```

Arguments

jnd_clstrs	List of joined clusters
txdct	Taxonomic dictionary

Value

list of ClstrRecs

See Also

Other run-private: batcher, blast_clstr, blast_filter, blast_setup, blast_sqs, blastcache_load, blastcache_save, blastdb_gen, blastn_run, cache_rm, cache_setup, clade_select, clstr2_calc, clstr_all, clstr_direct, clstr_sqs, clstr_subtree, clstrarc_gen, clstrarc_join, clstrrec_gen, clstrs_calc, clstrs_join, clstrs_renumber, clstrs_save, cmdln, descendants_get, download_obj_check, error, hierarchic_download, info, ncbicache_load, ncbicache_save, obj_check, obj_load, obj_save, parameters_load, parameters_setup, parent_get, progress_init, progress_read, progress_reset, progress_save, rank_get, safely_connect, search_and_cache, searchterm_gen, seeds_blast, seq_download, seqarc_gen, seqrec_augment, seqrec_convert, seqrec_gen, seqrec_get, sids_check, sids_get, sids_load, sids_save, sqs_count, sqs_load, sqs_save, stage_args_check, stages_run, tax_download, taxdict_gen, taxtree_gen, txids_get, txnds_count, warn

clstrs_renumber *Renumber cluster IDs*

Description

Returns a ClstrArc with ID determined by the number of sequences in each cluster.

Usage

```
clstrs_renumber(clstrrecs)
```

Arguments

clstrrecs	List of clusters

Value

ClstrArc

See Also

Other run-private: batcher, blast_clstr, blast_filter, blast_setup, blast_sqs, blastcache_load, blastcache_save, blastdb_gen, blastn_run, cache_rm, cache_setup, clade_select, clstr2_calc, clstr_all, clstr_direct, clstr_sqs, clstr_subtree, clstrarc_gen, clstrarc_join, clstrrec_gen, clstrs_calc, clstrs_join, clstrs_merge, clstrs_save, cmdln, descendants_get, download_obj_check, error, hierarchic_download, info, ncbicache_load, ncbicache_save, obj_check, obj_load, obj_save, parameters_load, parameters_setup, parent_get, progress_init, progress_read, progress_reset, progress_save, rank_get, safely_connect, search_and_cache, searchterm_gen, seeds_blast, seq_download, seqarc_gen, seqrec_augment, seqrec_convert, seqrec_gen, seqrec_get, sids_check, sids_get, sids_load, sids_save, sqs_count, sqs_load, sqs_save, stage_args_check, stages_run, tax_download, taxdict_gen, taxtree_gen, txids_get, txnds_count, warn

---

clstrs_save *Save clusters to cache*

Description

Saves clusters generated by clstr_all to cache.

Usage

```
clstrs_save(wd, txid, clstrs)
```

Arguments

wd	Working directory
txid	Taxonomic ID, numeric
clstrs	cluster list

See Also

Other run-private: batcher, blast_clstr, blast_filter, blast_setup, blast_sqs, blastcache_load, blastcache_save, blastdb_gen, blastn_run, cache_rm, cache_setup, clade_select, clstr2_calc, clstr_all, clstr_direct, clstr_sqs, clstr_subtree, clstrarc_gen, clstrarc_join, clstrrec_gen, clstrs_calc, clstrs_join, clstrs_merge, clstrs_renumber, cmdln, descendants_get, download_obj_check, error, hierarchic_download, info, ncbicache_load, ncbicache_save, obj_check, obj_load, obj_save, parameters_load, parameters_setup, parent_get, progress_init, progress_read, progress_reset, progress_save, rank_get, safely_connect, search_and_cache, searchterm_gen, seeds_blast, seq_download, seqarc_gen, seqrec_augment, seqrec_convert, seqrec_gen, seqrec_get,

sids_check, sids_get, sids_load, sids_save, sqs_count, sqs_load, sqs_save, stage_args_check, stages_run, tax_download, taxdict_gen, taxtree_gen, txids_get, txnds_count, warn

---

clstr_all *Hierarchically cluster all sequences of a txid*

Description

Identifies all direct and subtree clusters for a taxonomic ID.

Usage

```
clstr_all(txid, sqs, txdct, ps, lvl = 0)
```

Arguments

txid	Taxonomic ID
sqs	Sequence object of all downloaded sequences
txdct	Taxonomic dictionary
ps	Parameters list, generated with parameters()
lvl	Integer, number of message indentations indicating code depth.

Value

ClstrArc

See Also

Other run-private: batcher, blast_clstr, blast_filter, blast_setup, blast_sqs, blastcache_load, blastcache_save, blastdb_gen, blastn_run, cache_rm, cache_setup, clade_select, clstr2_calc, clstr_direct, clstr_sqs, clstr_subtree, clstrarc_gen, clstrarc_join, clstrrec_gen, clstrs_calc, clstrs_join, clstrs_merge, clstrs_renumber, clstrs_save, cmdln, descendants_get, download_obj_check, error, hierarchic_download, info, ncbicache_load, ncbicache_save, obj_check, obj_load, obj_save, parameters_load, parameters_setup, parent_get, progress_init, progress_read, progress_reset, progress_save, rank_get, safely_connect, search_and_cache, searchterm_gen, seeds_blast, seq_download, seqarc_gen, seqrec_augment, seqrec_convert, seqrec_gen, seqrec_get, sids_check, sids_get, sids_load, sids_save, sqs_count, sqs_load, sqs_save, stage_args_check, stages_run, tax_download, taxdict_gen, taxtree_gen, txids_get, txnds_count, warn

clstr_direct *Cluster sequences directly associated with txid*

Description

In GenBank certain sequences may only be associated with a higher level taxon (e.g. genus, family ...). This function generates clusters from these sequences, alone. This function identifies such sequences in the sequence object and generates a list of clusters of cl_type 'direct'.

Usage

```
clstr_direct(txid, sqs, txdct, ps, lvl)
```

Arguments

txid	Taxonomic ID
sqs	Sequence object of all downloaded sequences
txdct	Taxonomic dictionary
ps	Parameters list, generated with parameters()
lvl	Integer, number of message indentations indicating code depth.

Value

ClstrArc

See Also

Other run-private: batcher, blast_clstr, blast_filter, blast_setup, blast_sqs, blastcache_load, blastcache_save, blastdb_gen, blastn_run, cache_rm, cache_setup, clade_select, clstr2_calc, clstr_all, clstr_sqs, clstr_subtree, clstrarc_gen, clstrarc_join, clstrrec_gen, clstrs_calc, clstrs_join, clstrs_merge, clstrs_renumber, clstrs_save, cmdln, descendants_get, download_obj_check, error, hierarchic_download, info, ncbicache_load, ncbicache_save, obj_check, obj_load, obj_save, parameters_load, parameters_setup, parent_get, progress_init, progress_read, progress_reset, progress_save, rank_get, safely_connect, search_and_cache, searchterm_gen, seeds_blast, seq_download, seqarc_gen, seqrec_augment, seqrec_convert, seqrec_gen, seqrec_get, sids_check, sids_get, sids_load, sids_save, sqs_count, sqs_load, sqs_save, stage_args_check, stages_run, tax_download, taxdict_gen, taxtree_gen, txids_get, txnds_count, warn

clstr_sqs *Identify clusters from sequences*

Description

Given a sequence object, this function will generate a list of cluster objects using BLAST

Usage

```
clstr_sqs(txid, sqs, ps, lvl, typ = c("direct", "subtree", "paraphyly"))
```

Arguments

txid	Taxonomic ID
sqs	Sequence object of sequences to be BLASTed
ps	Parameters list, generated with parameters()
lvl	Integer, number of message indentations indicating code depth.
typ	Direct, subtree or paraphyly?

See Also

Other run-private: batcher, blast_clstr, blast_filter, blast_setup, blast_sqs, blastcache_load, blastcache_save, blastdb_gen, blastn_run, cache_rm, cache_setup, clade_select, clstr2_calc, clstr_all, clstr_direct, clstr_subtree, clstrarc_gen, clstrarc_join, clstrrec_gen, clstrs_calc, clstrs_join, clstrs_merge, clstrs_renumber, clstrs_save, cmdln, descendants_get, download_obj_check, error, hierarchic_download, info, ncbicache_load, ncbicache_save, obj_check, obj_load, obj_save, parameters_load, parameters_setup, parent_get, progress_init, progress_read, progress_reset, progress_save, rank_get, safely_connect, search_and_cache, searchterm_gen, seeds_blast, seq_download, seqarc_gen, seqrec_augment, seqrec_convert, seqrec_gen, seqrec_get, sids_check, sids_get, sids_load, sids_save, sqs_count, sqs_load, sqs_save, stage_args_check, stages_run, tax_download, taxdict_gen, taxtree_gen, txids_get, txnds_count, warn

clstr_subtree *Cluster all sequences descending from a txid*

Description

Identifies clusters from sequences associated with a txid and all its descendants. Clusters returned by this function will thus be of cl_type 'subtree'.

Usage

```
clstr_subtree(txid, sqs, txdct, dds, ps, lvl)
```

Arguments

txid	Taxonomic ID
sqs	Sequence object of all downloaded sequences
txdct	Taxonomic dictionary
dds	Vector of direct descendants
ps	Parameters list, generated with parameters()
lvl	Integer, number of message indentations indicating code depth.

Value

ClstrArc

See Also

Other run-private: batcher, blast_clstr, blast_filter, blast_setup, blast_sqs, blastcache_load, blastcache_save, blastdb_gen, blastn_run, cache_rm, cache_setup, clade_select, clstr2_calc, clstr_all, clstr_direct, clstr_sqs, clstrarc_gen, clstrarc_join, clstrrec_gen, clstrs_calc, clstrs_join, clstrs_merge, clstrs_renumber, clstrs_save, cmdln, descendants_get, download_obj_check, error, hierarchic_download, info, ncbicache_load, ncbicache_save, obj_check, obj_load, obj_save, parameters_load, parameters_setup, parent_get, progress_init, progress_read, progress_reset, progress_save, rank_get, safely_connect, search_and_cache, searchterm_gen, seeds_blast, seq_download, seqarc_gen, seqrec_augment, seqrec_convert, seqrec_gen, seqrec_get, sids_check, sids_get, sids_load, sids_save, sqs_count, sqs_load, sqs_save, stage_args_check, stages_run, tax_download, taxdict_gen, taxtree_gen, txids_get, txnds_count, warn

---

clusters2_run *Run the cluster2 stage*

Description

Run the fourth stage of the phylotaR pipeline, cluster2. Identify clusters at higher taxonomic levels by merging sister clusters.

Usage

```
clusters2_run(wd)
```

Arguments

wd	Working directory

See Also

Other run-public: ClstrArc-class, ClstrRec-class, Phylota-class, SeqArc-class, SeqRec-class, TaxDict-class, TaxRec-class, clusters_run, parameters_reset, reset, restart, run, setup, taxise_run

Examples

```
## Not run:

  # Note: this example requires BLAST and internet to run.

  # example with temp folder
  wd <- file.path(tempdir(), 'aotus')
  # setup for aotus, make sure aotus/ folder already exists
  if (!dir.exists(wd)) {
    dir.create(wd)
  }
  ncbi_dr <- '[SET BLAST+ BIN PATH HERE]'
  setup(wd = wd, txid = 9504, ncbi_dr = ncbi_dr)  # txid for Aotus primate genus
  # individually run stages
  taxise_run(wd = wd)
  download_run(wd = wd)
  clusters_run(wd = wd)
  clusters2_run(wd = wd)

## End(Not run)
```

`clusters_run` *Run the cluster stage*

Description

Run the third stage of the phylotaR pipeline, cluster. This stage hierarchically traverses the taxonomy identifying all direct and subtree clusters from downloaded sequences. Any taxonomic nodes too small for cluster identification are placed into paraphyletic clusters.

Usage

```
clusters_run(wd)
```

Arguments

wd	Working directory

See Also

Other run-public: `ClstrArc-class`, `ClstrRec-class`, `Phylota-class`, `SeqArc-class`, `SeqRec-class`, `TaxDict-class`, `TaxRec-class`, `clusters2_run`, `parameters_reset`, `reset`, `restart`, `run`, `setup`, `taxise_run`

Examples

```
## Not run:

  # Note: this example requires BLAST and internet to run.

  # example with temp folder
  wd <- file.path(tempdir(), 'aotus')
  # setup for aotus, make sure aotus/ folder already exists
  if (!dir.exists(wd)) {
    dir.create(wd)
  }
  ncbi_dr <- '[SET BLAST+ BIN PATH HERE]'
  setup(wd = wd, txid = 9504, ncbi_dr = ncbi_dr)  # txid for Aotus primate genus
  # individually run stages
  taxise_run(wd = wd)
  download_run(wd = wd)
  clusters_run(wd = wd)

## End(Not run)
```

---

cmdln *Run a command via terminal/command prompt*

Description

Provide the command and arguments as a vector. Also can take a lgfl to which all stdout and stderr is written. If lgfl is not provided, a list is returned of 'status', 'stdout' and 'stderr'. Else only the status is returned - 1 success, 0 failed.

Usage

```
cmdln(cmd, args, lgfl = NULL)
```

Arguments

cmd	Command to be run
args	Vector of command arguments, each parameter and value must be a separate element
lgfl	File to which stdout/err will be written

Details

Note, stdout/err are returned as 'raw'. Use rawToChar() to convert to characters.

Value

status, integer or character

See Also

Other run-private: batcher, blast_clstr, blast_filter, blast_setup, blast_sqs, blastcache_load, blastcache_save, blastdb_gen, blastn_run, cache_rm, cache_setup, clade_select, clstr2_calc, clstr_all, clstr_direct, clstr_sqs, clstr_subtree, clstrarc_gen, clstrarc_join, clstrrec_gen, clstrs_calc, clstrs_join, clstrs_merge, clstrs_renumber, clstrs_save, descendants_get, download_obj_check, error, hierarchic_download, info, ncbicache_load, ncbicache_save, obj_check, obj_load, obj_save, parameters_load, parameters_setup, parent_get, progress_init, progress_read, progress_reset, progress_save, rank_get, safely_connect, search_and_cache, searchterm_gen, seeds_blast, seq_download, seqarc_gen, seqrec_augment, seqrec_convert, seqrec_gen, seqrec_get, sids_check, sids_get, sids_load, sids_save, sqs_count, sqs_load, sqs_save, stage_args_check, stages_run, tax_download, taxdict_gen, taxtree_gen, txids_get, txnds_count, warn

---

cycads *Example phylota object (cycads)*

Description

Final phylota object produced by running pipeline for the cycads, Cycadidae (1445963).

Usage

```
data(cycads)
```

Format

cycads is generated from read_phylota().

Examples

```
data(cycads) # load object
str(cycads)
```

---

descendants_get *Get descendants*

Description

Look-up either direct or all taxonomic descendants of a node from taxonomic dictionary.

Usage

```
descendants_get(id, txdct, direct = FALSE)
```

Arguments

id	txid
txdct	TaxDict
direct	T/F, return only direct descendants?

Value

vector

See Also

Other run-private: batcher, blast_clstr, blast_filter, blast_setup, blast_sqs, blastcache_load, blastcache_save, blastdb_gen, blastn_run, cache_rm, cache_setup, clade_select, clstr2_calc, clstr_all, clstr_direct, clstr_sqs, clstr_subtree, clstrarc_gen, clstrarc_join, clstrrec_gen, clstrs_calc, clstrs_join, clstrs_merge, clstrs_renumber, clstrs_save, cmdln, download_obj_check, error, hierarchic_download, info, ncbicache_load, ncbicache_save, obj_check, obj_load, obj_save, parameters_load, parameters_setup, parent_get, progress_init, progress_read, progress_reset, progress_save, rank_get, safely_connect, search_and_cache, searchterm_gen, seeds_blast, seq_download, seqarc_gen, seqrec_augment, seqrec_convert, seqrec_gen, seqrec_get, sids_check, sids_get, sids_load, sids_save, sqs_count, sqs_load, sqs_save, stage_args_check, stages_run, tax_download, taxdict_gen, taxtree_gen, txids_get, txnds_count, warn

---

download_obj_check *Check an object returned from rentrez function*

Description

Returns T/F. Checks if object returned from rentrez function is as expected.

Usage

```
download_obj_check(obj)
```

Arguments

obj	Object returned from rentrez function

Value

T/F

See Also

Other run-private: batcher, blast_clstr, blast_filter, blast_setup, blast_sqs, blastcache_load, blastcache_save, blastdb_gen, blastn_run, cache_rm, cache_setup, clade_select, clstr2_calc, clstr_all, clstr_direct, clstr_sqs, clstr_subtree, clstrarc_gen, clstrarc_join, clstrrec_gen, clstrs_calc, clstrs_join, clstrs_merge, clstrs_renumber, clstrs_save, cmdln, descendants_get, error, hierarchic_download, info, ncbicache_load, ncbicache_save, obj_check, obj_load, obj_save, parameters_load, parameters_setup, parent_get, progress_init, progress_read, progress_reset, progress_save, rank_get, safely_connect, search_and_cache, searchterm_gen, seeds_blast, seq_download, seqarc_gen, seqrec_augment, seqrec_convert, seqrec_gen, seqrec_get, sids_check, sids_get, sids_load, sids_save, sqs_count, sqs_load, sqs_save, stage_args_check, stages_run, tax_download, taxdict_gen, taxtree_gen, txids_get, txnds_count, warn

---

download_run *Run download stage*

Description

Run the second stage of phylotaR, download. This stage downloads sequences for all nodes with sequence numbers less than mxsqs. It hierarchically traverses the taxonomy for each node and downloads direct and subtree sequences for all descendants.

Usage

```
download_run(wd)
```

Arguments

wd	Working directory

Examples

```
## Not run:

  # Note: this example requires BLAST and internet to run.

  # example with temp folder
  wd <- file.path(tempdir(), 'aotus')
  # setup for aotus, make sure aotus/ folder already exists
  if (!dir.exists(wd)) {
    dir.create(wd)
  }
  ncbi_dr <- '[SET BLAST+ BIN PATH HERE]'
  setup(wd = wd, txid = 9504, ncbi_dr = ncbi_dr)  # txid for Aotus primate genus
  # individually run stages
  taxise_run(wd = wd)
  download_run(wd = wd)

## End(Not run)
```

dragonflies *Example phylota object (dragonflies)*

Description

Final phylota object produced by running pipeline for the dragonflies, Anisoptera (6962).

Usage

```
data(dragonflies)
```

Format

`dragonflies` is generated from read_phylota().

Examples

```
data(dragonflies) # load object
str(dragonflies)
```

drop_by_rank *Reduce clusters to specific rank*

Description

Identifies higher level taxa for each sequence in clusters for given rank. Selects representative sequences for each unique taxon using the choose_by functions. By default, the function will choose the top ten sequences by first sorting by those with fewest number of ambiguous sequences, then by youngest, then by sequence length.

Usage

```
drop_by_rank(phylota, rnk = "species", keep_higher = FALSE, n = 10,
  choose_by = c("pambgs", "age", "nncltds"), greatest = c(FALSE, FALSE,
  TRUE))
```

Arguments

phylota	Phylota object
rnk	Taxonomic rank
keep_higher	Keep higher taxonomic ranks?
n	Number of sequences per taxon
choose_by	Vector of selection functions
greatest	Greatest of lowest for each choose_by function

Value

phylota

See Also

Other tools-public: calc_mad, calc_wrdfrq, drop_clstrs, drop_sqs, get_clstr_slot, get_nsqs, get_ntaxa, get_sq_slot, get_stage_times, get_tx_slot, get_txids, is_txid_in_clstr, is_txid_in_sq, list_clstrrec_slots, list_ncbi_ranks, list_seqrec_slots, list_taxrec_slots, plot_phylota_pa, plot_phylota_treemap, read_phylota, write_sqs

Examples

```
data("dragonflies")
# For faster computations, let's only work with the 5 clusters.
dragonflies <- drop_clstrs(phylota = dragonflies, cid = dragonflies@cids[10:15])


# We can use drop_by_rank() to reduce to 10 sequences per genus for each cluster
(reduced_1 <- drop_by_rank(phylota = dragonflies, rnk = 'genus', n = 10,
                           choose_by = c('pambgs', 'age', 'nncltds'),
                           greatest = c(FALSE, FALSE, TRUE)))

# We can specify what aspects of the sequences we would like to select per genus
# By default we select the sequences with fewest ambiguous nucleotides (e.g.
# we avoid Ns), the youngest age and then longest sequence.
# We can reverse the 'greatest' to get the opposite.
(reduced_2 <- drop_by_rank(phylota = dragonflies, rnk = 'genus', n = 10,
                           choose_by = c('pambgs', 'age', 'nncltds'),
                           greatest = c(TRUE, TRUE, FALSE)))


# Leading to smaller sequnces ...
r1_sqlngth <- mean(get_sq_slot(phylota = reduced_1,
                               sid = reduced_1@sids, slt_nm = 'nncltds'))
r2_sqlngth <- mean(get_sq_slot(phylota = reduced_2,
                               sid = reduced_2@sids, slt_nm = 'nncltds'))
(r1_sqlngth > r2_sqlngth)
# ... with more ambigous characters ....
r1_pambgs <- mean(get_sq_slot(phylota = reduced_1, sid = reduced_1@sids,
                              slt_nm = 'pambgs'))
r2_pambgs <- mean(get_sq_slot(phylota = reduced_2, sid = reduced_2@sids,
                              slt_nm = 'pambgs'))
(r1_pambgs < r2_pambgs)
# .... and older ages (measured in days since being added to GenBank).
r1_age <- mean(get_sq_slot(phylota = reduced_1, sid = reduced_1@sids,
                           slt_nm = 'age'))
r2_age <- mean(get_sq_slot(phylota = reduced_2, sid = reduced_2@sids,
                           slt_nm = 'age'))
(r1_age < r2_age)


# Or... we can simply reduce the clusters to just one sequence per genus
```

```
(dragonflies <- drop_by_rank(phylota = dragonflies, rnk = 'genus', n = 1))
```

`drop_clstrs` *Drop cluster records from phylota object*

Description

Drops all clusters except those identified by user.

Usage

```
drop_clstrs(phylota, cid)
```

Arguments

`phylota`	Phylota object
`cid`	Cluster ID(s) to be kept

Value

phylota

See Also

Other tools-public: `calc_mad`, `calc_wrdfrq`, `drop_by_rank`, `drop_sqs`, `get_clstr_slot`, `get_nsqs`, `get_ntaxa`, `get_sq_slot`, `get_stage_times`, `get_tx_slot`, `get_txids`, `is_txid_in_clstr`, `is_txid_in_sq`, `list_clstrrec_slots`, `list_ncbi_ranks`, `list_seqrec_slots`, `list_taxrec_slots`, `plot_phylota_pa`, `plot_phylota_treemap`, `read_phylota`, `write_sqs`

Examples

```
data("dragonflies")
# specify cids to *keep*
random_cids <- sample(dragonflies@cids, 100)
# drop an entire cluster
nbefore <- length(dragonflies@cids)
dragonflies <- drop_clstrs(phylota = dragonflies, cid = random_cids)
nafter <- length(dragonflies@cids)
# now there are only 100 clusters
(nafter < nbefore)
```

drop_sqs *Drop sequences in a cluster*

Description

Drop all sequences in a cluster except those identified by user.

Usage

```
drop_sqs(phylota, cid, sid)
```

Arguments

phylota	Phylota object
cid	Cluster ID
sid	Sequence ID(s) to be kept

Value

phylota

See Also

Other tools-public: calc_mad, calc_wrdfrq, drop_by_rank, drop_clstrs, get_clstr_slot, get_nsqs, get_ntaxa, get_sq_slot, get_stage_times, get_tx_slot, get_txids, is_txid_in_clstr, is_txid_in_sq, list_clstrrec_slots, list_ncbi_ranks, list_seqrec_slots, list_taxrec_slots, plot_phylota_pa, plot_phylota_treemap, read_phylota, write_sqs

Examples

```
data("dragonflies")
# drop random sequences from cluster 0
clstr <- dragonflies[['0']]
# specify the sids to *keep*
sids <- sample(clstr@sids, 100)
(dragonflies <- drop_sqs(phylota = dragonflies, cid = '0', sid = sids))
# Note, sequences dropped may be represented in other clusters
```

error *Write error message to log*

Description

Inform a user if an error has occurred in log.txt, halt pipeline.

Usage

```
error(ps, ...)
```

Arguments

ps	Parameters list, generated with parameters()
...	Message elements for concatenating

See Also

Other run-private: batcher, blast_clstr, blast_filter, blast_setup, blast_sqs, blastcache_load, blastcache_save, blastdb_gen, blastn_run, cache_rm, cache_setup, clade_select, clstr2_calc, clstr_all, clstr_direct, clstr_sqs, clstr_subtree, clstrarc_gen, clstrarc_join, clstrrec_gen, clstrs_calc, clstrs_join, clstrs_merge, clstrs_renumber, clstrs_save, cmdln, descendants_get, download_obj_check, hierarchic_download, info, ncbicache_load, ncbicache_save, obj_check, obj_load, obj_save, parameters_load, parameters_setup, parent_get, progress_init, progress_read, progress_reset, progress_save, rank_get, safely_connect, search_and_cache, searchterm_gen, seeds_blast, seq_download, seqarc_gen, seqrec_augment, seqrec_convert, seqrec_gen, seqrec_get, sids_check, sids_get, sids_load, sids_save, sqs_count, sqs_load, sqs_save, stage_args_check, stages_run, tax_download, taxdict_gen, taxtree_gen, txids_get, txnds_count, warn

get_clstr_slot *Get slot data for each cluster record*

Description

Get slot data for cluster(s)

Usage

```
get_clstr_slot(phylota, cid, slt_nm = list_clstrrec_slots())
```

Arguments

phylota	Phylota object
cid	Cluster ID
slt_nm	Slot name

Value

vector

See Also

Other tools-public: calc_mad, calc_wrdfrq, drop_by_rank, drop_clstrs, drop_sqs, get_nsqs, get_ntaxa, get_sq_slot, get_stage_times, get_tx_slot, get_txids, is_txid_in_clstr, is_txid_in_sq, list_clstrrec_slots, list_ncbi_ranks, list_seqrec_slots, list_taxrec_slots, plot_phylota_pa, plot_phylota_treemap, read_phylota, write_sqs

Examples

```
data('aotus')
random_cid <- sample(aotus@cids, 1)
(get_clstr_slot(phylota = aotus, cid = random_cid, slt_nm = 'seed'))
# see list_clstrrec_slots() for available slots
(list_clstrrec_slots())
```

---

get_nsqs *Count number of sequences*

Description

Count the number of sequences in a cluster(s).

Usage

```
get_nsqs(phylota, cid)
```

Arguments

phylota	Phylota object
cid	Cluster ID(s)

Value

vector

See Also

Other tools-public: calc_mad, calc_wrdfrq, drop_by_rank, drop_clstrs, drop_sqs, get_clstr_slot, get_ntaxa, get_sq_slot, get_stage_times, get_tx_slot, get_txids, is_txid_in_clstr, is_txid_in_sq, list_clstrrec_slots, list_ncbi_ranks, list_seqrec_slots, list_taxrec_slots, plot_phylota_pa, plot_phylota_treemap, read_phylota, write_sqs

Examples

```
data("cycads")
# count seqs for a random 10 clusters
random_cids <- sample(cycads@cids, 10)
nsqs <- get_nsqs(phylota = cycads, cid = random_cids)
```

`get_ntaxa` *Count number of unique taxa*

Description

Count the number of unique taxa represented by cluster(s) or sequences in phylota table Use rnk to specify a taxonomic level to count. If NULL counts will be made to the lowest level reported on NCBI.

Usage

```
get_ntaxa(phylota, cid = NULL, sid = NULL, rnk = NULL,
  keep_higher = FALSE)
```

Arguments

`phylota`	Phylota object
`cid`	Cluster ID(s)
`sid`	Sequence ID(s)
`rnk`	Taxonomic rank
`keep_higher`	Keep higher taxonomic ranks?

Value

vector

See Also

Other tools-public: `calc_mad`, `calc_wrdfrq`, `drop_by_rank`, `drop_clstrs`, `drop_sqs`, `get_clstr_slot`, `get_nsqs`, `get_sq_slot`, `get_stage_times`, `get_tx_slot`, `get_txids`, `is_txid_in_clstr`, `is_txid_in_sq`, `list_clstrrec_slots`, `list_ncbi_ranks`, `list_seqrec_slots`, `list_taxrec_slots`, `plot_phylota_pa`, `plot_phylota_treemap`, `read_phylota`, `write_sqs`

Examples

```
data('bromeliads')
# how many species are there?
(get_ntaxa(phylota = bromeliads, cid = '0', rnk = 'species'))
# how many genera are there?
(get_ntaxa(phylota = bromeliads, cid = '0', rnk = 'genus'))
# how many families are there?
(get_ntaxa(phylota = bromeliads, cid = '0', rnk = 'family'))
# use list_ncbi_ranks() to see available rank names
(list_ncbi_ranks())
```

---

get_sq_slot *Get slot data for each sequence*

Description

Get slot data for either or sequences in a cluster of a vector of sequence IDs. Use list_seqrec_slots() for a list of available slots.

Usage

```
get_sq_slot(phylota, cid = NULL, sid = NULL, slt_nm = list_seqrec_slots())
```

Arguments

phylota	Phylota object
cid	Cluster ID
sid	Sequence ID(s)
slt_nm	Slot name

Value

vector

See Also

Other tools-public: calc_mad, calc_wrdfrq, drop_by_rank, drop_clstrs, drop_sqs, get_clstr_slot, get_nsqs, get_ntaxa, get_stage_times, get_tx_slot, get_txids, is_txid_in_clstr, is_txid_in_sq, list_clstrrec_slots, list_ncbi_ranks, list_seqrec_slots, list_taxrec_slots, plot_phylota_pa, plot_phylota_treemap, read_phylota, write_sqs

Examples

```
data('aotus')
random_sid <- sample(aotus@sids, 1)
(get_sq_slot(phylota = aotus, sid = random_sid, slt_nm = 'dfln'))
# see list_seqrec_slots() for available slots
(list_seqrec_slots())
```

get_stage_times *Get run times for different stages*

Description

Get slot data for taxa(s)

Usage

```
get_stage_times(wd)
```

Arguments

wd	Working directory

Value

list of runtimes in minutes

See Also

Other tools-public: calc_mad, calc_wrdfrq, drop_by_rank, drop_clstrs, drop_sqs, get_clstr_slot, get_nsqs, get_ntaxa, get_sq_slot, get_tx_slot, get_txids, is_txid_in_clstr, is_txid_in_sq, list_clstrrec_slots, list_ncbi_ranks, list_seqrec_slots, list_taxrec_slots, plot_phylota_pa, plot_phylota_treemap, read_phylota, write_sqs

Examples

```
## Not run:

  # Note, this example requires a wd with a completed phylotaR run
  # return a named list of the time take in minutes for each stage
  get_stage_times(wd = wd)

## End(Not run)
```

get_txids *Get taxonomic IDs by rank*

Description

Return taxonomic IDs for a vector of sequence IDs or all sequences in a cluster. User can specify what rank the IDs should be returned. If NULL, the lowest level is returned.

Usage

```
get_txids(phylota, cid = NULL, sid = NULL, txids = NULL, rnk = NULL,
  keep_higher = FALSE)
```

Arguments

phylota	Phylota object
cid	Cluster ID
sid	Sequence ID(s)
txids	Vector of txids
rnk	Taxonomic rank
keep_higher	Keep higher taxonomic IDs?

Details

txids can either be provided by user or they can be determined for a vector of sids or for a cid. If keep_higher is TRUE, any sequence that has a identity that is higher than the given rank will be returned. If FALSE, these sequences will return ".

Value

vector

See Also

Other tools-public: calc_mad, calc_wrdfrq, drop_by_rank, drop_clstrs, drop_sqs, get_clstr_slot, get_nsqs, get_ntaxa, get_sq_slot, get_stage_times, get_tx_slot, is_txid_in_clstr, is_txid_in_sq, list_clstrrec_slots, list_ncbi_ranks, list_seqrec_slots, list_taxrec_slots, plot_phylota_pa, plot_phylota_treemap, read_phylota, write_sqs

Examples

```
data('bromeliads')
# get all the genus IDs and names
genus_ids <- get_txids(phylota = bromeliads, txids = bromeliads@txids,
                       rnk = 'genus')
genus_ids <- unique(genus_ids)
# drop empty IDs -- this happens if a given lineage has no ID for specified rank
genus_ids <- genus_ids[genus_ids != '']
# get names
(get_tx_slot(phylota = bromeliads, txid = genus_ids, slt_nm = 'scnm'))
```

get_tx_slot *Get slot data for each taxon record*

Description

Get slot data for taxa(s)

Usage

```
get_tx_slot(phylota, txid, slt_nm = list_taxrec_slots())
```

Arguments

phylota	Phylota object
txid	Taxonomic ID
slt_nm	Slot name

Value

vector or list

See Also

Other tools-public: calc_mad, calc_wrdfrq, drop_by_rank, drop_clstrs, drop_sqs, get_clstr_slot, get_nsqs, get_ntaxa, get_sq_slot, get_stage_times, get_txids, is_txid_in_clstr, is_txid_in_sq, list_clstrrec_slots, list_ncbi_ranks, list_seqrec_slots, list_taxrec_slots, plot_phylota_pa, plot_phylota_treemap, read_phylota, write_sqs

Examples

```
data('aotus')
random_txid <- sample(aotus@txids, 1)
(get_tx_slot(phylota = aotus, txid = random_txid, slt_nm = 'scnm'))
# see list_taxrec_slots() for available slots
(list_taxrec_slots())
```

hierarchic_download *Hierarchically get sequences for a txid*

Description

Looks up and downloads sequences for a taxonomic ID.

Usage

```
hierarchic_download(txid, txdct, ps, lvl = 0)
```

Arguments

txid	Taxonomic node ID, numeric
txdct	Taxonomic dictionary
ps	Parameters list, generated with parameters()
lvl	Integer, number of message indentations indicating code depth.

Value

Vector of SeqRecs

See Also

Other run-private: batcher, blast_clstr, blast_filter, blast_setup, blast_sqs, blastcache_load, blastcache_save, blastdb_gen, blastn_run, cache_rm, cache_setup, clade_select, clstr2_calc, clstr_all, clstr_direct, clstr_sqs, clstr_subtree, clstrarc_gen, clstrarc_join, clstrrec_gen, clstrs_calc, clstrs_join, clstrs_merge, clstrs_renumber, clstrs_save, cmdln, descendants_get, download_obj_check, error, info, ncbicache_load, ncbicache_save, obj_check, obj_load, obj_save, parameters_load, parameters_setup, parent_get, progress_init, progress_read, progress_reset, progress_save, rank_get, safely_connect, search_and_cache, searchterm_gen, seeds_blast, seq_download, seqarc_gen, seqrec_augment, seqrec_convert, seqrec_gen, seqrec_get, sids_check, sids_get, sids_load, sids_save, sqs_count, sqs_load, sqs_save, stage_args_check, stages_run, tax_download, taxdict_gen, taxtree_gen, txids_get, txnds_count, warn

---

info *Write info message to log*

Description

Inform a user via log.txt of pipeline progress.

Usage

```
info(lvl, ps, ...)
```

Arguments

lvl	Integer, number of message indentations indicating code depth.
ps	Parameters list, generated with parameters()
...	Message elements for concatenating

See Also

Other run-private: batcher, blast_clstr, blast_filter, blast_setup, blast_sqs, blastcache_load, blastcache_save, blastdb_gen, blastn_run, cache_rm, cache_setup, clade_select, clstr2_calc, clstr_all, clstr_direct, clstr_sqs, clstr_subtree, clstrarc_gen, clstrarc_join, clstrrec_gen, clstrs_calc, clstrs_join, clstrs_merge, clstrs_renumber, clstrs_save, cmdln, descendants_get, download_obj_check, error, hierarchic_download, ncbicache_load, ncbicache_save, obj_check, obj_load, obj_save, parameters_load, parameters_setup, parent_get, progress_init, progress_read, progress_reset, progress_save, rank_get, safely_connect, search_and_cache, searchterm_gen, seeds_blast, seq_download, seqarc_gen, seqrec_augment, seqrec_convert, seqrec_gen, seqrec_get, sids_check, sids_get, sids_load, sids_save, sqs_count, sqs_load, sqs_save, stage_args_check, stages_run, tax_download, taxdict_gen, taxtree_gen, txids_get, txnds_count, warn

---

is_txid_in_clstr *Is txid in cluster?*

Description

Checks if given txid is represented by any of the sequences of a cluster by searching through all the sequence search organism lineages.

Usage

```
is_txid_in_clstr(phylota, txid, cid)
```

Arguments

phylota	Phylota
txid	Taxonomic ID
cid	Cluster ID

Value

boolean

See Also

Other tools-public: calc_mad, calc_wrdfrq, drop_by_rank, drop_clstrs, drop_sqs, get_clstr_slot, get_nsqs, get_ntaxa, get_sq_slot, get_stage_times, get_tx_slot, get_txids, is_txid_in_sq, list_clstrrec_slots, list_ncbi_ranks, list_seqrec_slots, list_taxrec_slots, plot_phylota_pa, plot_phylota_treemap, read_phylota, write_sqs

Examples

```
data(tinamous)
cid <- tinamous@cids[[1]]
clstr <- tinamous[[cid]]
sq <- tinamous[[clstr@sids[[1]]]]
txid <- sq@txid
# expect true
is_txid_in_clstr(phylota = tinamous, txid = txid, cid = cid)
```

---

is_txid_in_sq *Is txid in sequence?*

Description

Checks if given txid is represented by sequence by looking at sequence source organism's lineage.

Usage

```
is_txid_in_sq(phylota, txid, sid)
```

Arguments

phylota	Phylota
txid	Taxonomic ID
sid	Sequence ID

Value

boolean

See Also

Other tools-public: calc_mad, calc_wrdfrq, drop_by_rank, drop_clstrs, drop_sqs, get_clstr_slot, get_nsqs, get_ntaxa, get_sq_slot, get_stage_times, get_tx_slot, get_txids, is_txid_in_clstr, list_clstrrec_slots, list_ncbi_ranks, list_seqrec_slots, list_taxrec_slots, plot_phylota_pa, plot_phylota_treemap, read_phylota, write_sqs

Examples

```
data(tinamous)
sid <- tinamous@sids[[1]]
sq <- tinamous[[sid]]
txid <- sq@txid
# expect true
is_txid_in_sq(phylota = tinamous, txid = txid, sid = sid)
```

list_clstrrec_slots *List all ClstrRec slots*

Description

Returns a vector of all available ClstrRec slots of type character, integer and numeric.

Usage

```
list_clstrrec_slots()
```

Value

vector

See Also

Other tools-public: calc_mad, calc_wrdfrq, drop_by_rank, drop_clstrs, drop_sqs, get_clstr_slot, get_nsqs, get_ntaxa, get_sq_slot, get_stage_times, get_tx_slot, get_txids, is_txid_in_clstr, is_txid_in_sq, list_ncbi_ranks, list_seqrec_slots, list_taxrec_slots, plot_phylota_pa, plot_phylota_treemap, read_phylota, write_sqs

list_ncbi_ranks *List all NCBI Ranks*

Description

Returns a vector of all NCBI taxonomic ranks in descending order.

Usage

```
list_ncbi_ranks()
```

Value

vector

See Also

Other tools-public: calc_mad, calc_wrdfrq, drop_by_rank, drop_clstrs, drop_sqs, get_clstr_slot, get_nsqs, get_ntaxa, get_sq_slot, get_stage_times, get_tx_slot, get_txids, is_txid_in_clstr, is_txid_in_sq, list_clstrrec_slots, list_seqrec_slots, list_taxrec_slots, plot_phylota_pa, plot_phylota_treemap, read_phylota, write_sqs

list_seqrec_slots *List all SeqRec slots*

Description

Returns a vector of all available SeqRec slots of type character, integer and numeric.

Usage

```
list_seqrec_slots()
```

Value

vector

See Also

Other tools-public: calc_mad, calc_wrdfrq, drop_by_rank, drop_clstrs, drop_sqs, get_clstr_slot, get_nsqs, get_ntaxa, get_sq_slot, get_stage_times, get_tx_slot, get_txids, is_txid_in_clstr, is_txid_in_sq, list_clstrrec_slots, list_ncbi_ranks, list_taxrec_slots, plot_phylota_pa, plot_phylota_treemap, read_phylota, write_sqs

list_taxrec_slots *List all TaxRec slots*

Description

Returns a vector of all available TaxRec slots of type character, integer and numeric.

Usage

```
list_taxrec_slots()
```

Value

vector

See Also

Other tools-public: calc_mad, calc_wrdfrq, drop_by_rank, drop_clstrs, drop_sqs, get_clstr_slot, get_nsqs, get_ntaxa, get_sq_slot, get_stage_times, get_tx_slot, get_txids, is_txid_in_clstr, is_txid_in_sq, list_clstrrec_slots, list_ncbi_ranks, list_seqrec_slots, plot_phylota_pa, plot_phylota_treemap, read_phylota, write_sqs

mk_txid_in_sq_mtrx *Return matrix of txid in sequence*

Description

Searches through lineages of sequences' source organisms to determine whether each txid is represented by the sequence.

Usage

```
mk_txid_in_sq_mtrx(phylota, txids, sids = phylota@sids)
```

Arguments

phylota	Phylota
txids	Taxonomic IDs
sids	Sequence IDs

Value

matrix

See Also

Other tools-private: summary_phylota, update_phylota

ncbicache_load *Retrieve cached NCBI query*

Description

Run this function to load cached NCBI queries.

Usage

```
ncbicache_load(fnm, args, wd)
```

Arguments

fnm	NCBI Entrez function name
args	Args used for function
wd	Working directory

Value

rentrez result

See Also

Other run-private: batcher, blast_clstr, blast_filter, blast_setup, blast_sqs, blastcache_load, blastcache_save, blastdb_gen, blastn_run, cache_rm, cache_setup, clade_select, clstr2_calc, clstr_all, clstr_direct, clstr_sqs, clstr_subtree, clstrarc_gen, clstrarc_join, clstrrec_gen, clstrs_calc, clstrs_join, clstrs_merge, clstrs_renumber, clstrs_save, cmdln, descendants_get, download_obj_check, error, hierarchic_download, info, ncbicache_save, obj_check, obj_load, obj_save, parameters_load, parameters_setup, parent_get, progress_init, progress_read, progress_reset, progress_save, rank_get, safely_connect, search_and_cache, searchterm_gen, seeds_blast, seq_download, seqarc_gen, seqrec_augment, seqrec_convert, seqrec_gen, seqrec_get, sids_check, sids_get, sids_load, sids_save, sqs_count, sqs_load, sqs_save, stage_args_check, stages_run, tax_download, taxdict_gen, taxtree_gen, txids_get, txnds_count, warn

---

`ncbicache_save` *Save NCBI query result to cache*

Description

Run whenever NCBI queries are made to save results in cache in case the pipeline is run again.

Usage

```
ncbicache_save(fnm, args, wd, obj)
```

Arguments

`fnm`	NCBI Entrez function name
`args`	Args used for function
`wd`	Working directory
`obj`	NCBI query result

See Also

Other run-private: batcher, blast_clstr, blast_filter, blast_setup, blast_sqs, blastcache_load, blastcache_save, blastdb_gen, blastn_run, cache_rm, cache_setup, clade_select, clstr2_calc, clstr_all, clstr_direct, clstr_sqs, clstr_subtree, clstrarc_gen, clstrarc_join, clstrrec_gen, clstrs_calc, clstrs_join, clstrs_merge, clstrs_renumber, clstrs_save, cmdln, descendants_get, download_obj_check, error, hierarchic_download, info, ncbicache_load, obj_check, obj_load, obj_save, parameters_load, parameters_setup, parent_get, progress_init, progress_read, progress_reset, progress_save, rank_get, safely_connect, search_and_cache, searchterm_gen, seeds_blast, seq_download, seqarc_gen, seqrec_augment, seqrec_convert, seqrec_gen, seqrec_get, sids_check, sids_get, sids_load, sids_save, sqs_count, sqs_load, sqs_save, stage_args_check, stages_run, tax_download, taxdict_gen, taxtree_gen, txids_get, txnds_count, warn

obj_check *Check if an object exists*

Description

Check if an object exists in the cache.

Usage

```
obj_check(wd, nm)
```

Arguments

wd	Working directory
nm	Object name

Value

T/F

See Also

Other run-private: batcher, blast_clstr, blast_filter, blast_setup, blast_sqs, blastcache_load, blastcache_save, blastdb_gen, blastn_run, cache_rm, cache_setup, clade_select, clstr2_calc, clstr_all, clstr_direct, clstr_sqs, clstr_subtree, clstrarc_gen, clstrarc_join, clstrrec_gen, clstrs_calc, clstrs_join, clstrs_merge, clstrs_renumber, clstrs_save, cmdln, descendants_get, download_obj_check, error, hierarchic_download, info, ncbicache_load, ncbicache_save, obj_load, obj_save, parameters_load, parameters_setup, parent_get, progress_init, progress_read, progress_reset, progress_save, rank_get, safely_connect, search_and_cache, searchterm_gen, seeds_blast, seq_download, seqarc_gen, seqrec_augment, seqrec_convert, seqrec_gen, seqrec_get, sids_check, sids_get, sids_load, sids_save, sqs_count, sqs_load, sqs_save, stage_args_check, stages_run, tax_download, taxdict_gen, taxtree_gen, txids_get, txnds_count, warn

obj_load *Load a named object from the cache*

Description

Loads an object from the cache as stored by obj_save.

Usage

```
obj_load(wd, nm)
```

Arguments

wd	Working directory
nm	Object name

Value

object, multiple formats possible

See Also

Other run-private: batcher, blast_clstr, blast_filter, blast_setup, blast_sqs, blastcache_load, blastcache_save, blastdb_gen, blastn_run, cache_rm, cache_setup, clade_select, clstr2_calc, clstr_all, clstr_direct, clstr_sqs, clstr_subtree, clstrarc_gen, clstrarc_join, clstrrec_gen, clstrs_calc, clstrs_join, clstrs_merge, clstrs_renumber, clstrs_save, cmdln, descendants_get, download_obj_check, error, hierarchic_download, info, ncbicache_load, ncbicache_save, obj_check, obj_save, parameters_load, parameters_setup, parent_get, progress_init, progress_read, progress_reset, progress_save, rank_get, safely_connect, search_and_cache, searchterm_gen, seeds_blast, seq_download, seqarc_gen, seqrec_augment, seqrec_convert, seqrec_gen, seqrec_get, sids_check, sids_get, sids_load, sids_save, sqs_count, sqs_load, sqs_save, stage_args_check, stages_run, tax_download, taxdict_gen, taxtree_gen, txids_get, txnds_count, warn

---

obj_save *Save a named object in the cache*

Description

Save an object in the cache that can be loaded by obj_load.

Usage

```
obj_save(wd, obj, nm)
```

Arguments

wd	Working directory
obj	Object
nm	Object name

See Also

Other run-private: batcher, blast_clstr, blast_filter, blast_setup, blast_sqs, blastcache_load, blastcache_save, blastdb_gen, blastn_run, cache_rm, cache_setup, clade_select, clstr2_calc, clstr_all, clstr_direct, clstr_sqs, clstr_subtree, clstrarc_gen, clstrarc_join, clstrrec_gen, clstrs_calc, clstrs_join, clstrs_merge, clstrs_renumber, clstrs_save, cmdln, descendants_get, download_obj_check, error, hierarchic_download, info, ncbicache_load, ncbicache_save, obj_check, obj_load, parameters_load, parameters_setup, parent_get, progress_init, progress_read,

progress_reset, progress_save, rank_get, safely_connect, search_and_cache, searchterm_gen, seeds_blast, seq_download, seqarc_gen, seqrec_augment, seqrec_convert, seqrec_gen, seqrec_get, sids_check, sids_get, sids_load, sids_save, sqs_count, sqs_load, sqs_save, stage_args_check, stages_run, tax_download, taxdict_gen, taxtree_gen, txids_get, txnds_count, warn

---

`parameters` *Default parameters*

Description

Returns a parameter list with default parameter values.

Usage

```
parameters(wd = ".", txid = character(), mkblstdb = "", blstn = "",
  v = FALSE, ncps = 1, mxnds = 1e+05, mdlthrs = 3000, mnsql = 250,
  mxsql = 2000, mxrtry = 100, mxsqs = 50000, mxevl = 1e-10,
  mncvrg = 51, btchsz = 100, date = Sys.Date())
```

Arguments

`wd`	The working directory where all output files are saved.
`txid`	Taxonomic group of interest, allows vectors.
`mkblstdb`	File path to makeblastdb
`blstn`	File path to blastn
`v`	Print progress statements to console? Statements will always be printed to log.txt.
`ncps`	The number of threads to use in the local-alignment search tool.
`mxnds`	The maximum number of nodes descending from a taxonomic group. If there are more than this number, nodes at the lower taxonomic level are analysed.
`mdlthrs`	'Model organism threshold'. Taxa with more sequences than this number will be considered model organisms and a random mdlthrs subset of their sequences will be downloaded.
`mnsql`	The minimum length of sequence in nucleotide base pairs to download.
`mxsql`	The maximum length of sequence in nucleotide base pairs to download. Any longer sequences will be ignored.
`mxrtry`	The maximum number of attempts to make when downloading.
`mxsqs`	The maximum number of sequences to BLAST in all-vs-all searches. If there are more sequences for a node, BLAST is performed at the lower taxonomic level.
`mxevl`	The maximum E-value for a successful BLAST.
`mncvrg`	The maximum percentile coverage defining an overlapping BLAST hit. Sequences with BLAST matches with lower values are not considered orthologous.
`btchsz`	Batch size when querying NCBI
`date`	Date when pipeline was initiated

Details

This function is NOT used to change the parameters in a folder. Use parameters_reset() instead. The purpose of this function is to describe the paramaters and present their default values.

Value

list

`parameters_load` *Load parameters from cache*

Description

Parameters are held in cache, use this function to load parameters set for a wd.

Usage

```
parameters_load(wd)
```

Arguments

`wd`	Working directory

Value

Parameters list

See Also

Other run-private: `batcher`, `blast_clstr`, `blast_filter`, `blast_setup`, `blast_sqs`, `blastcache_load`, `blastcache_save`, `blastdb_gen`, `blastn_run`, `cache_rm`, `cache_setup`, `clade_select`, `clstr2_calc`, `clstr_all`, `clstr_direct`, `clstr_sqs`, `clstr_subtree`, `clstrarc_gen`, `clstrarc_join`, `clstrrec_gen`, `clstrs_calc`, `clstrs_join`, `clstrs_merge`, `clstrs_renumber`, `clstrs_save`, `cmdln`, `descendants_get`, `download_obj_check`, `error`, `hierarchic_download`, `info`, `ncbicache_load`, `ncbicache_save`, `obj_check`, `obj_load`, `obj_save`, `parameters_setup`, `parent_get`, `progress_init`, `progress_read`, `progress_reset`, `progress_save`, `rank_get`, `safely_connect`, `search_and_cache`, `searchterm_gen`, `seeds_blast`, `seq_download`, `seqarc_gen`, `seqrec_augment`, `seqrec_convert`, `seqrec_gen`, `seqrec_get`, `sids_check`, `sids_get`, `sids_load`, `sids_save`, `sqs_count`, `sqs_load`, `sqs_save`, `stage_args_check`, `stages_run`, `tax_download`, `taxdict_gen`, `taxtree_gen`, `txids_get`, `txnds_count`, `warn`

parameters_reset *Change parameters in a working directory*

Description

Reset parameters after running `setup()`.

Usage

```
parameters_reset(wd, parameters, values)
```

Arguments

wd	Working directory
parameters	Parameters to be changed, vector.
values	New values for each parameter, vector.

See Also

Other run-public: ClstrArc-class, ClstrRec-class, Phylota-class, SeqArc-class, SeqRec-class, TaxDict-class, TaxRec-class, clusters2_run, clusters_run, reset, restart, run, setup, taxise_run

Examples

```
## Not run:

  # Note: this example requires BLAST and internet to run.

  # example with temp folder
  wd <- file.path(tempdir(), 'aotus')
  # setup for aotus, make sure aotus/ folder already exists
  if (!dir.exists(wd)) {
    dir.create(wd)
  }
  ncbi_dr <- '[SET BLAST+ BIN PATH HERE]'
  setup(wd = wd, txid = 9504, ncbi_dr = ncbi_dr)  # txid for Aotus primate genus
  # run
  # run(wd = wd) # not running in test
  # use ctrl+c or Esc to kill
  # change parameters, e.g. min and max sequence lengths
  parameters_reset(wd = 'aotus', parameters = c('mnsql', 'mxsql'),
                   values = c(300, 1500))
  # see ?parameters
  # restart
  restart(wd = wd)

## End(Not run)
```

parameters_setup *Set Up Parameters*

Description

Initiates cache of parameters.

Usage

```
parameters_setup(wd, ncbi_execs, ...)
```

Arguments

wd	Working directory
ncbi_execs	File directories for NCBI tools, see blast_setup()
...	Set parameters, see parameters()

See Also

Other run-private: batcher, blast_clstr, blast_filter, blast_setup, blast_sqs, blastcache_load, blastcache_save, blastdb_gen, blastn_run, cache_rm, cache_setup, clade_select, clstr2_calc, clstr_all, clstr_direct, clstr_sqs, clstr_subtree, clstrarc_gen, clstrarc_join, clstrrec_gen, clstrs_calc, clstrs_join, clstrs_merge, clstrs_renumber, clstrs_save, cmdln, descendants_get, download_obj_check, error, hierarchic_download, info, ncbicache_load, ncbicache_save, obj_check, obj_load, obj_save, parameters_load, parent_get, progress_init, progress_read, progress_reset, progress_save, rank_get, safely_connect, search_and_cache, searchterm_gen, seeds_blast, seq_download, seqarc_gen, seqrec_augment, seqrec_convert, seqrec_gen, seqrec_get, sids_check, sids_get, sids_load, sids_save, sqs_count, sqs_load, sqs_save, stage_args_check, stages_run, tax_download, taxdict_gen, taxtree_gen, txids_get, txnds_count, warn

parent_get *Get taxonomic parent*

Description

Look-up MRCA of taxonomic id(s) from taxonomic dictionary

Usage

```
parent_get(id, txdct)
```

Arguments

id	txid(s)
txdct	TaxDict

Value

Character

See Also

Other run-private: batcher, blast_clstr, blast_filter, blast_setup, blast_sqs, blastcache_load, blastcache_save, blastdb_gen, blastn_run, cache_rm, cache_setup, clade_select, clstr2_calc, clstr_all, clstr_direct, clstr_sqs, clstr_subtree, clstrarc_gen, clstrarc_join, clstrrec_gen, clstrs_calc, clstrs_join, clstrs_merge, clstrs_renumber, clstrs_save, cmdln, descendants_get, download_obj_check, error, hierarchic_download, info, ncbicache_load, ncbicache_save, obj_check, obj_load, obj_save, parameters_load, parameters_setup, progress_init, progress_read, progress_reset, progress_save, rank_get, safely_connect, search_and_cache, searchterm_gen, seeds_blast, seq_download, seqarc_gen, seqrec_augment, seqrec_convert, seqrec_gen, seqrec_get, sids_check, sids_get, sids_load, sids_save, sqs_count, sqs_load, sqs_save, stage_args_check, stages_run, tax_download, taxdict_gen, taxtree_gen, txids_get, txnds_count, warn

---

Phylota-class *Phylota object*

Description

Phylota table contains all sequence, cluster and taxonomic information from a phylotaR pipeline run.

Usage

```
## S4 method for signature 'Phylota'
as.character(x)

## S4 method for signature 'Phylota'
show(object)

## S4 method for signature 'Phylota'
print(x)

## S4 method for signature 'Phylota'
str(object, max.level = 2L, ...)

## S4 method for signature 'Phylota'
summary(object)

## S4 method for signature 'Phylota,character'
x[[i]]
```

Arguments

x	Phylota object
object	Phylota object
max.level	Maximum level of nesting for str()
...	Further arguments for str()
i	Either sid or cid

Slots

cids IDs of all clusters

sids IDs of all sequences

txids IDs of all taxa

sqs All sequence records as SeqArc

clstrs All cluster records as ClstrArc

txdct Taxonomic dictionary as TaxDict

prnt_id Parent taxonomic ID

prnt_nm Parent taxonomic name

See Also

Other run-public: ClstrArc-class, ClstrRec-class, SeqArc-class, SeqRec-class, TaxDict-class, TaxRec-class, clusters2_run, clusters_run, parameters_reset, reset, restart, run, setup, taxise_run

Examples

```
data('aotus')
# this is a Phylota object
# it contains cluster, sequence and taxonomic information from a phylotaR run
show(aotus)
# you can access its different data slots with @
aotus@cids   # cluster IDs
aotus@sids   # sequence IDs
aotus@txids  # taxonomic IDs
aotus@clstrs # clusters archive
aotus@sqs    # sequence archive
aotus@txdct  # taxonomic dictionary
# see all of the available slots
(slotNames(aotus))
# access different sequences and clusters with [[
(aotus[['0']])              # cluster record 0
(aotus[[aotus@sids[[1]]]])  # first sequence record
# get a summary of the whole object
(summary(aotus))
# the above generates a data.frame with information on each cluster:
# ID - unique id in the object
# Type - cluster type
```

```
# Seed - most connected sequence
# Parent - MRCA of all represented taxa
# N_taxa - number of NCBI recognised taxa
# N_seqs - number of sequences
# Med_sql - median sequence length
# MAD - Maximum alignment density, values close to 1 indicate all sequences in
#  the cluster have a similar length.
# Definition - most common words (and frequency) in sequence definitions
# Feature - most common feature name (and frequency)
```

`plot_phylota_pa` *Plot presence/absence matrix*

Description

Plot presence/absence of taxa by each cluster in phylota object.

Usage

```
plot_phylota_pa(phylota, cids, txids, cnms = cids, txnms = txids)
```

Arguments

`phylota`	Phylota object
`cids`	Vector of cluster IDs
`txids`	Vector of taxonomic IDs
`cnms`	Cluster names
`txnms`	Taxonomic names

Details

Cluster names and taxonomic names can be given to the function, by default IDs are used.

Value

geom_object

See Also

Other tools-public: `calc_mad`, `calc_wrdfrq`, `drop_by_rank`, `drop_clstrs`, `drop_sqs`, `get_clstr_slot`, `get_nsqs`, `get_ntaxa`, `get_sq_slot`, `get_stage_times`, `get_tx_slot`, `get_txids`, `is_txid_in_clstr`, `is_txid_in_sq`, `list_clstrrec_slots`, `list_ncbi_ranks`, `list_seqrec_slots`, `list_taxrec_slots`, `plot_phylota_treemap`, `read_phylota`, `write_sqs`

Examples

```
library(phylotaR)
data(cycads)
# drop all but first ten
cycads <- drop_clstrs(cycads, cycads@cids[1:10])
# plot all
p <- plot_phylota_pa(phylota = cycads, cids = cycads@cids, txids = cycads@txids)
print(p)  # lots of information, difficult to interpret
# get genus-level taxonomic names
genus_txids <- get_txids(cycads, txids = cycads@txids, rnk = 'genus')
genus_txids <- unique(genus_txids)
# dropping missing
genus_txids <- genus_txids[genus_txids !=  '']
genus_nms <- get_tx_slot(cycads, genus_txids, slt_nm = 'scnm')
# make alphabetical for plotting
genus_nms <- sort(genus_nms, decreasing = TRUE)
# generate geom_object
p <- plot_phylota_pa(phylota = cycads, cids = cycads@cids, txids = genus_txids,
                     txnms = genus_nms)
# plot
print(p)  # easier to interpret
```

---

plot_phylota_treemap *Plot treemap of Phylota object*

Description

Treemaps show relative size with boxes. The user can explore which taxa or clusters are most represented either by sequence or cluster number. If cluster IDs are provided, the plot is made for clusters. If taxonomic IDs are provided, the plot is made for taxa.

Usage

```
plot_phylota_treemap(phylota, cids = NULL, txids = NULL, cnms = cids,
  txnms = txids, with_labels = TRUE, area = c("ntx", "nsq", "ncl"),
  fill = c("NULL", "typ", "ntx", "nsq", "ncl"))
```

Arguments

phylota	Phylota object
cids	Cluster IDs
txids	Taxonomic IDs
cnms	Cluster names
txnms	Taxonomic names
with_labels	Show names per box?
area	What determines the size per box?
fill	What determines the coloured fill per box?

Details

The function can take a long time to run for large Phylota objects over many taxonomic IDs because searches are made across lineages. The idea of the function is to assess the data dominance of specific clusters and taxa.

Value

geom_object

See Also

Other tools-public: calc_mad, calc_wrdfrq, drop_by_rank, drop_clstrs, drop_sqs, get_clstr_slot, get_nsqs, get_ntaxa, get_sq_slot, get_stage_times, get_tx_slot, get_txids, is_txid_in_clstr, is_txid_in_sq, list_clstrrec_slots, list_ncbi_ranks, list_seqrec_slots, list_taxrec_slots, plot_phylota_pa, read_phylota, write_sqs

Examples

```
data("tinamous")
# Plot clusters, size by n. sq, fill by n. tx
p <- plot_phylota_treemap(phylota = tinamous, cids = tinamous@cids,
                          area = 'nsq', fill = 'ntx')
print(p)
# Plot taxa, size by n. sq, fill by ncl
txids <- get_txids(tinamous, txids = tinamous@txids, rnk = 'genus')
txids <- txids[txids !=  '']
txids <- unique(txids)
txnms <- get_tx_slot(tinamous, txids, slt_nm = 'scnm')
p <- plot_phylota_treemap(phylota = tinamous, txids = txids, txnms = txnms,
                          area = 'nsq', fill = 'ncl')
print(p)
```

---

progress_init *Initialise progress list in cache*

Description

Creates a progress list recording each stage run in cache.

Usage

```
progress_init(wd)
```

Arguments

wd	Working directory

See Also

Other run-private: batcher, blast_clstr, blast_filter, blast_setup, blast_sqs, blastcache_load, blastcache_save, blastdb_gen, blastn_run, cache_rm, cache_setup, clade_select, clstr2_calc, clstr_all, clstr_direct, clstr_sqs, clstr_subtree, clstrarc_gen, clstrarc_join, clstrrec_gen, clstrs_calc, clstrs_join, clstrs_merge, clstrs_renumber, clstrs_save, cmdln, descendants_get, download_obj_check, error, hierarchic_download, info, ncbicache_load, ncbicache_save, obj_check, obj_load, obj_save, parameters_load, parameters_setup, parent_get, progress_read, progress_reset, progress_save, rank_get, safely_connect, search_and_cache, searchterm_gen, seeds_blast, seq_download, seqarc_gen, seqrec_augment, seqrec_convert, seqrec_gen, seqrec_get, sids_check, sids_get, sids_load, sids_save, sqs_count, sqs_load, sqs_save, stage_args_check, stages_run, tax_download, taxdict_gen, taxtree_gen, txids_get, txnds_count, warn

---

progress_read *Read the progress from cache*

Description

Return the last completed stage using the cache.

Usage

```
progress_read(wd)
```

Arguments

wd	Working directory

Value

stage name, character, or NA is complete

See Also

Other run-private: batcher, blast_clstr, blast_filter, blast_setup, blast_sqs, blastcache_load, blastcache_save, blastdb_gen, blastn_run, cache_rm, cache_setup, clade_select, clstr2_calc, clstr_all, clstr_direct, clstr_sqs, clstr_subtree, clstrarc_gen, clstrarc_join, clstrrec_gen, clstrs_calc, clstrs_join, clstrs_merge, clstrs_renumber, clstrs_save, cmdln, descendants_get, download_obj_check, error, hierarchic_download, info, ncbicache_load, ncbicache_save, obj_check, obj_load, obj_save, parameters_load, parameters_setup, parent_get, progress_init, progress_reset, progress_save, rank_get, safely_connect, search_and_cache, searchterm_gen, seeds_blast, seq_download, seqarc_gen, seqrec_augment, seqrec_convert, seqrec_gen, seqrec_get, sids_check, sids_get, sids_load, sids_save, sqs_count, sqs_load, sqs_save, stage_args_check, stages_run, tax_download, taxdict_gen, taxtree_gen, txids_get, txnds_count, warn

progress_reset *Reset progress*

Description

Reset progress to an earlier completed stage.

Usage

```
progress_reset(wd, stg)
```

Arguments

wd	Working directory
stg	Stage to which the pipeline will be reset

Details

For example, resetting the progress to 'download' mark stages 'download', 'cluster' and 'cluster2' as un-run.

See Also

Other run-private: batcher, blast_clstr, blast_filter, blast_setup, blast_sqs, blastcache_load, blastcache_save, blastdb_gen, blastn_run, cache_rm, cache_setup, clade_select, clstr2_calc, clstr_all, clstr_direct, clstr_sqs, clstr_subtree, clstrarc_gen, clstrarc_join, clstrrec_gen, clstrs_calc, clstrs_join, clstrs_merge, clstrs_renumber, clstrs_save, cmdln, descendants_get, download_obj_check, error, hierarchic_download, info, ncbicache_load, ncbicache_save, obj_check, obj_load, obj_save, parameters_load, parameters_setup, parent_get, progress_init, progress_read, progress_save, rank_get, safely_connect, search_and_cache, searchterm_gen, seeds_blast, seq_download, seqarc_gen, seqrec_augment, seqrec_convert, seqrec_gen, seqrec_get, sids_check, sids_get, sids_load, sids_save, sqs_count, sqs_load, sqs_save, stage_args_check, stages_run, tax_download, taxdict_gen, taxtree_gen, txids_get, txnds_count, warn

progress_save *Save current progress*

Description

Stores the pipeline progress in the cache.

Usage

```
progress_save(wd, stg)
```

Arguments

wd	Working directory
stg	Stage

See Also

Other run-private: batcher, blast_clstr, blast_filter, blast_setup, blast_sqs, blastcache_load, blastcache_save, blastdb_gen, blastn_run, cache_rm, cache_setup, clade_select, clstr2_calc, clstr_all, clstr_direct, clstr_sqs, clstr_subtree, clstrarc_gen, clstrarc_join, clstrrec_gen, clstrs_calc, clstrs_join, clstrs_merge, clstrs_renumber, clstrs_save, cmdln, descendants_get, download_obj_check, error, hierarchic_download, info, ncbicache_load, ncbicache_save, obj_check, obj_load, obj_save, parameters_load, parameters_setup, parent_get, progress_init, progress_read, progress_reset, rank_get, safely_connect, search_and_cache, searchterm_gen, seeds_blast, seq_download, seqarc_gen, seqrec_augment, seqrec_convert, seqrec_gen, seqrec_get, sids_check, sids_get, sids_load, sids_save, sqs_count, sqs_load, sqs_save, stage_args_check, stages_run, tax_download, taxdict_gen, taxtree_gen, txids_get, txnds_count, warn

---

rank_get *Get rank*

Description

Look-up taxonomic rank from dictionary.

Usage

```
rank_get(txid, txdct)
```

Arguments

txid	txid
txdct	TaxDict

Value

character

See Also

Other run-private: batcher, blast_clstr, blast_filter, blast_setup, blast_sqs, blastcache_load, blastcache_save, blastdb_gen, blastn_run, cache_rm, cache_setup, clade_select, clstr2_calc, clstr_all, clstr_direct, clstr_sqs, clstr_subtree, clstrarc_gen, clstrarc_join, clstrrec_gen, clstrs_calc, clstrs_join, clstrs_merge, clstrs_renumber, clstrs_save, cmdln, descendants_get, download_obj_check, error, hierarchic_download, info, ncbicache_load, ncbicache_save, obj_check, obj_load, obj_save, parameters_load, parameters_setup, parent_get, progress_init, progress_read, progress_reset, progress_save, safely_connect, search_and_cache, searchterm_gen, seeds_blast, seq_download, seqarc_gen, seqrec_augment, seqrec_convert, seqrec_gen, seqrec_get,

sids_check, sids_get, sids_load, sids_save, sqs_count, sqs_load, sqs_save, stage_args_check, stages_run, tax_download, taxdict_gen, taxtree_gen, txids_get, txnds_count, warn

---

read_phylota *Generate a Phylota object in R*

Description

Creates a Phylota object containing information on clusters, sequences and taxonomy from the working directory of a completed pipeline.

Usage

```
read_phylota(wd)
```

Arguments

wd	Working directory

Value

Phylota

See Also

Other tools-public: calc_mad, calc_wrdfrq, drop_by_rank, drop_clstrs, drop_sqs, get_clstr_slot, get_nsqs, get_ntaxa, get_sq_slot, get_stage_times, get_tx_slot, get_txids, is_txid_in_clstr, is_txid_in_sq, list_clstrrec_slots, list_ncbi_ranks, list_seqrec_slots, list_taxrec_slots, plot_phylota_pa, plot_phylota_treemap, write_sqs

Examples

```
## Not run:

  # Note, this example requires a wd with a completed phylotaR run
  phylota <- read_phylota(wd)

## End(Not run)
```

`reset` *Reset a phylotaR pipeline run*

Description

Resets the pipeline to a specified stage.

Usage

```
reset(wd, stage, hard = FALSE)
```

Arguments

`wd`	Working directory
`stage`	Name of stage to which the pipeline will be reset
`hard`	T/F, delete all cached data?

See Also

Other run-public: `ClstrArc-class`, `ClstrRec-class`, `Phylota-class`, `SeqArc-class`, `SeqRec-class`, `TaxDict-class`, `TaxRec-class`, `clusters2_run`, `clusters_run`, `parameters_reset`, `restart`, `run`, `setup`, `taxise_run`

Examples

```
## Not run:

  # Note: this example requires BLAST and internet to run.

  # example with temp folder
  wd <- file.path(tempdir(), 'aotus')
  # setup for aotus, make sure aotus/ folder already exists
  if (!dir.exists(wd)) {
    dir.create(wd)
  }
  ncbi_dr <- '[SET BLAST+ BIN PATH HERE]'
  setup(wd = wd, txid = 9504, ncbi_dr = ncbi_dr)  # txid for Aotus primate genus
  # individually run taxise
  taxise_run(wd = wd)
  # reset back to taxise as if it has not been run
  reset(wd = 'aotus', stage = 'taxise')
  # run taxise again ....
  taxise_run(wd = wd)

## End(Not run)
```

restart *Restart a phylotaR pipeline run*

Description

Restarts the running of a pipeline as started with `run`.

Usage

```
restart(wd, nstages = 4)
```

Arguments

`wd`	Working directory
`nstages`	Number of total stages to run, max 4.

See Also

Other run-public: `ClstrArc-class`, `ClstrRec-class`, `Phylota-class`, `SeqArc-class`, `SeqRec-class`, `TaxDict-class`, `TaxRec-class`, `clusters2_run`, `clusters_run`, `parameters_reset`, `reset`, `run`, `setup`, `taxise_run`

Examples

```
## Not run:

  # Note: this example requires BLAST and internet to run.

  # example with temp folder
  wd <- file.path(tempdir(), 'aotus')
  # setup for aotus, make sure aotus/ folder already exists
  if (!dir.exists(wd)) {
    dir.create(wd)
  }
  ncbi_dr <- '[SET BLAST+ BIN PATH HERE]'
  setup(wd = wd, txid = 9504, ncbi_dr = ncbi_dr)  # txid for Aotus primate genus
  # run and stop after 10 seconds
  R.utils::withTimeout(expr = {
    run(wd = wd)
  }, timeout = 10)
  # use ctrl+c or Esc to kill without a timelimit
  # and restart with ....
  restart(wd = wd)

## End(Not run)
```

run *Run phylotaR pipeline*

Description

Run the entire phylotaR pipeline. All generated files will be stored in the wd. The process can be stopped at anytime and restarted with `restart`. `nstages` must be a numeric value representing the number of stages that will be run. Stages are run in the following order: 1 - taxise, 2 - download, 3 - cluster and 4 - cluster2.

For example, specifying `nstages` = 3, will run taxise, download and cluster. Stages can also be run individually, see linked functions below.

Usage

```
run(wd, nstages = 4)
```

Arguments

`wd`	Working directory
`nstages`	Number of total stages to run, max 4.

See Also

Other run-public: `ClstrArc-class`, `ClstrRec-class`, `Phylota-class`, `SeqArc-class`, `SeqRec-class`, `TaxDict-class`, `TaxRec-class`, `clusters2_run`, `clusters_run`, `parameters_reset`, `reset`, `restart`, `setup`, `taxise_run`

Examples

```
## Not run:

  # Note: this example requires BLAST and internet to run.

  # example with temp folder
  wd <- file.path(tempdir(), 'aotus')
  # setup for aotus, make sure aotus/ folder already exists
  if (!dir.exists(wd)) {
    dir.create(wd)
  }
  ncbi_dr <- '[SET BLAST+ BIN PATH HERE]'
  # e.g. "/usr/local/ncbi/blast/bin/"
  setup(wd = wd, txid = 9504, ncbi_dr = ncbi_dr)  # txid for Aotus primate genus
  run(wd = wd)

## End(Not run)
```

`safely_connect` *Safely run rentrez function*

Description

Safely run a rentrez function. If the query fails, the function will retry.

Usage

```
safely_connect(func, args, fnm, ps)
```

Arguments

`func`	rentrez function
`args`	rentrez function arguments, list
`fnm`	rentrez function name
`ps`	Parameters list, generated with parameters()

Value

rentrez function results

See Also

Other run-private: `batcher`, `blast_clstr`, `blast_filter`, `blast_setup`, `blast_sqs`, `blastcache_load`, `blastcache_save`, `blastdb_gen`, `blastn_run`, `cache_rm`, `cache_setup`, `clade_select`, `clstr2_calc`, `clstr_all`, `clstr_direct`, `clstr_sqs`, `clstr_subtree`, `clstrarc_gen`, `clstrarc_join`, `clstrrec_gen`, `clstrs_calc`, `clstrs_join`, `clstrs_merge`, `clstrs_renumber`, `clstrs_save`, `cmdln`, `descendants_get`, `download_obj_check`, `error`, `hierarchic_download`, `info`, `ncbicache_load`, `ncbicache_save`, `obj_check`, `obj_load`, `obj_save`, `parameters_load`, `parameters_setup`, `parent_get`, `progress_init`, `progress_read`, `progress_reset`, `progress_save`, `rank_get`, `search_and_cache`, `searchterm_gen`, `seeds_blast`, `seq_download`, `seqarc_gen`, `seqrec_augment`, `seqrec_convert`, `seqrec_gen`, `seqrec_get`, `sids_check`, `sids_get`, `sids_load`, `sids_save`, `sqs_count`, `sqs_load`, `sqs_save`, `stage_args_check`, `stages_run`, `tax_download`, `taxdict_gen`, `taxtree_gen`, `txids_get`, `txnds_count`, `warn`

`searchterm_gen` *Construct GenBank Search Term*

Description

Construct search term for searching GenBank's nucleotide database. Limits the maximum size of sequences, avoids whole genome shotguns, predicted, unverified and synthetic sequences.

Usage

```
searchterm_gen(txid, ps, direct = FALSE)
```

Arguments

txid	Taxonomic ID
ps	Parameters list, generated with parameters()
direct	Node-level only or subtree as well? Default FALSE.

Value

character, search term

See Also

Other run-private: batcher, blast_clstr, blast_filter, blast_setup, blast_sqs, blastcache_load, blastcache_save, blastdb_gen, blastn_run, cache_rm, cache_setup, clade_select, clstr2_calc, clstr_all, clstr_direct, clstr_sqs, clstr_subtree, clstrarc_gen, clstrarc_join, clstrrec_gen, clstrs_calc, clstrs_join, clstrs_merge, clstrs_renumber, clstrs_save, cmdln, descendants_get, download_obj_check, error, hierarchic_download, info, ncbicache_load, ncbicache_save, obj_check, obj_load, obj_save, parameters_load, parameters_setup, parent_get, progress_init, progress_read, progress_reset, progress_save, rank_get, safely_connect, search_and_cache, seeds_blast, seq_download, seqarc_gen, seqrec_augment, seqrec_convert, seqrec_gen, seqrec_get, sids_check, sids_get, sids_load, sids_save, sqs_count, sqs_load, sqs_save, stage_args_check, stages_run, tax_download, taxdict_gen, taxtree_gen, txids_get, txnds_count, warn

---

search_and_cache *Run rentrez function and cache results*

Description

Safely run a rentrez function. If the query fails, the function will retry. All query results are cached. To remove cached data use hard reset.

Usage

```
search_and_cache(func, args, fnm, ps)
```

Arguments

func	rentrez function
args	rentrez function arguments, list
fnm	rentrez function name
ps	Parameters list, generated with parameters()

Value

rentrez function results

See Also

Other run-private: batcher, blast_clstr, blast_filter, blast_setup, blast_sqs, blastcache_load, blastcache_save, blastdb_gen, blastn_run, cache_rm, cache_setup, clade_select, clstr2_calc, clstr_all, clstr_direct, clstr_sqs, clstr_subtree, clstrarc_gen, clstrarc_join, clstrrec_gen, clstrs_calc, clstrs_join, clstrs_merge, clstrs_renumber, clstrs_save, cmdln, descendants_get, download_obj_check, error, hierarchic_download, info, ncbicache_load, ncbicache_save, obj_check, obj_load, obj_save, parameters_load, parameters_setup, parent_get, progress_init, progress_read, progress_reset, progress_save, rank_get, safely_connect, searchterm_gen, seeds_blast, seq_download, seqarc_gen, seqrec_augment, seqrec_convert, seqrec_gen, seqrec_get, sids_check, sids_get, sids_load, sids_save, sqs_count, sqs_load, sqs_save, stage_args_check, stages_run, tax_download, taxdict_gen, taxtree_gen, txids_get, txnds_count, warn

---

seeds_blast *BLAST seed sequences*

Description

Runs all-v-all blast for seed sequences.

Usage

```
seeds_blast(sqs, ps)
```

Arguments

sqs	All seed sequences to be BLASTed
ps	Parameters list, generated with parameters()

Value

blast res data.frame

See Also

Other run-private: batcher, blast_clstr, blast_filter, blast_setup, blast_sqs, blastcache_load, blastcache_save, blastdb_gen, blastn_run, cache_rm, cache_setup, clade_select, clstr2_calc, clstr_all, clstr_direct, clstr_sqs, clstr_subtree, clstrarc_gen, clstrarc_join, clstrrec_gen, clstrs_calc, clstrs_join, clstrs_merge, clstrs_renumber, clstrs_save, cmdln, descendants_get, download_obj_check, error, hierarchic_download, info, ncbicache_load, ncbicache_save, obj_check, obj_load, obj_save, parameters_load, parameters_setup, parent_get, progress_init, progress_read, progress_reset, progress_save, rank_get, safely_connect, search_and_cache, searchterm_gen, seq_download, seqarc_gen, seqrec_augment, seqrec_convert, seqrec_gen, seqrec_get, sids_check, sids_get, sids_load, sids_save, sqs_count, sqs_load, sqs_save,

stage_args_check, stages_run, tax_download, taxdict_gen, taxtree_gen, txids_get, txnds_count, warn

---

SeqArc-class *Sequence record archive*

Description

Multiple sequence records containing sequence data.

Usage

```
## S4 method for signature 'SeqArc'
as.character(x)

## S4 method for signature 'SeqArc'
show(object)

## S4 method for signature 'SeqArc'
print(x)

## S4 method for signature 'SeqArc'
str(object, max.level = 2L, ...)

## S4 method for signature 'SeqArc'
summary(object)

## S4 method for signature 'SeqArc,character'
x[[i]]

## S4 method for signature 'SeqArc,character,missing,missing'
x[i, j, ..., drop = TRUE]
```

Arguments

x	SeqArc object
object	SeqArc object
max.level	Maximum level of nesting for str()
...	Further arguments for str()
i	sid(s)
j	Unused
drop	Unused

Details

Sequences are stored as raw. Use rawToChar().

Slots

`ids` Vector of Sequence Record IDs

`nncltds` Vector of sequence lengths

`nambgs` Vector of number of ambiguous nucleotides

`txids` Vector source txid associated with each sequence

`sqs` List of SeqRecs named by ID

See Also

Other run-public: `ClstrArc-class`, `ClstrRec-class`, `Phylota-class`, `SeqRec-class`, `TaxDict-class`, `TaxRec-class`, `clusters2_run`, `clusters_run`, `parameters_reset`, `reset`, `restart`, `run`, `setup`, `taxise_run`

Examples

```
data('aotus')
seqarc <- aotus@sqs
# this is a SeqArc object
# it contains sequence records
show(seqarc)
# you can access its different data slots with @
seqarc@ids     # sequence IDs defined as accession + feature position
seqarc@nncltds # number of nucleotides of all sequences
seqarc@nambgs  # number of ambiguous nucleotides of all sequences
seqarc@txids   # all the taxonomic IDs for all sequences
seqarc@sqs     # list of all SeqRecs
# access sequence records [[
(seqarc[[seqarc@ids[[1]]]])  # first sequence record
# generate new sequence archives with [
(seqarc[seqarc@ids[1:10]])  # first 10 sequences
```

---

seqarc_gen *Generate sequence archive*

Description

Creates an S4 SeqArc from list of SeqRecs

Usage

```
seqarc_gen(seqrecs)
```

Arguments

`seqrecs`	List of SeqRecs

Value

SeqArc

See Also

Other run-private: batcher, blast_clstr, blast_filter, blast_setup, blast_sqs, blastcache_load, blastcache_save, blastdb_gen, blastn_run, cache_rm, cache_setup, clade_select, clstr2_calc, clstr_all, clstr_direct, clstr_sqs, clstr_subtree, clstrarc_gen, clstrarc_join, clstrrec_gen, clstrs_calc, clstrs_join, clstrs_merge, clstrs_renumber, clstrs_save, cmdln, descendants_get, download_obj_check, error, hierarchic_download, info, ncbicache_load, ncbicache_save, obj_check, obj_load, obj_save, parameters_load, parameters_setup, parent_get, progress_init, progress_read, progress_reset, progress_save, rank_get, safely_connect, search_and_cache, searchterm_gen, seeds_blast, seq_download, seqrec_augment, seqrec_convert, seqrec_gen, seqrec_get, sids_check, sids_get, sids_load, sids_save, sqs_count, sqs_load, sqs_save, stage_args_check, stages_run, tax_download, taxdict_gen, taxtree_gen, txids_get, txnds_count, warn

---

SeqRec-class *Sequence record*

Description

Sequence record contains sequence data.

Usage

```
## S4 method for signature 'SeqRec'
as.character(x)

## S4 method for signature 'SeqRec'
show(object)

## S4 method for signature 'SeqRec'
print(x)

## S4 method for signature 'SeqRec'
str(object, max.level = 2L, ...)

## S4 method for signature 'SeqRec'
summary(object)
```

Arguments

x	SeqRec object
object	SeqRec object
max.level	Maximum level of nesting for str()
...	Further arguments for str()

Details

Sequence is stored as raw. Use rawToChar().

Slots

`id` Unique ID

`nm` Best-guess sequence name

`accssn` Accession

`vrsn` Accession version

`url` URL

`txid` Taxonomic ID of source taxon

`orgnsm` Scientific name of source taxon

`sq` Sequence

`dfln` Definition line

`ml_typ` Molecule type, e.g. DNA

`rec_typ` Record type: Whole or feature

`nncltds` Number of nucleotides

`nambgs` Number of ambiguous nucleotides

`pambgs` Proportion of ambiguous nucleotides

`gcr` GC ratio

`age` Number of days between sequence upload and running pipeline

See Also

Other run-public: `ClstrArc-class`, `ClstrRec-class`, `Phylota-class`, `SeqArc-class`, `TaxDict-class`, `TaxRec-class`, `clusters2_run`, `clusters_run`, `parameters_reset`, `reset`, `restart`, `run`, `setup`, `taxise_run`

Examples

```
data('aotus')
seqrec <- aotus@sqs@sqs[[1]]
# this is a SeqRec object
# it contains sequence records
show(seqrec)
# you can access its different data slots with @
seqrec@id       # sequence ID, accession + feature location
seqrec@nm       # feature name, '' if none
seqrec@accssn   # accession
seqrec@vrsn     # accession version
seqrec@url      # NCBI GenBank URL
seqrec@txid     # Taxonomic ID
seqrec@orgnsm   # free-text organism name
seqrec@sq       # sequence, in raw format
seqrec@dfln     # sequence definition
```

```
seqrec@ml_typ   # molecule type
seqrec@rec_typ  # whole record or feature
seqrec@nncltds  # sequence length
seqrec@nambgs   # number of non-ATCGs
seqrec@pambgs   # proportion of non-ATCGs
seqrec@gcr      # GC-ratio
seqrec@age      # days since being added to GenBank
# get the sequence like so....
(rawToChar(seqrec@sq))
```

`seqrec_augment` *Augment sequence records list*

Description

Add taxids to records and convert to archive.

Usage

```
seqrec_augment(sqs, txdct)
```

Arguments

`sqs`	List of SeqRecs
`txdct`	Taxonomic Dictionary

Value

SeqArc

See Also

Other run-private: `batcher`, `blast_clstr`, `blast_filter`, `blast_setup`, `blast_sqs`, `blastcache_load`, `blastcache_save`, `blastdb_gen`, `blastn_run`, `cache_rm`, `cache_setup`, `clade_select`, `clstr2_calc`, `clstr_all`, `clstr_direct`, `clstr_sqs`, `clstr_subtree`, `clstrarc_gen`, `clstrarc_join`, `clstrrec_gen`, `clstrs_calc`, `clstrs_join`, `clstrs_merge`, `clstrs_renumber`, `clstrs_save`, `cmdln`, `descendants_get`, `download_obj_check`, `error`, `hierarchic_download`, `info`, `ncbicache_load`, `ncbicache_save`, `obj_check`, `obj_load`, `obj_save`, `parameters_load`, `parameters_setup`, `parent_get`, `progress_init`, `progress_read`, `progress_reset`, `progress_save`, `rank_get`, `safely_connect`, `search_and_cache`, `searchterm_gen`, `seeds_blast`, `seq_download`, `seqarc_gen`, `seqrec_convert`, `seqrec_gen`, `seqrec_get`, `sids_check`, `sids_get`, `sids_load`, `sids_save`, `sqs_count`, `sqs_load`, `sqs_save`, `stage_args_check`, `stages_run`, `tax_download`, `taxdict_gen`, `taxtree_gen`, `txids_get`, `txnds_count`, `warn`

seqrec_convert *Convert raw Entrez gbwithparts record to SeqRecs*

Description

Parses returned sequences features with Entrez, returns a SeqRec for each raw record.

Usage

```
seqrec_convert(raw_recs, ps)
```

Arguments

raw_recs	Raw records returned from Entrez fetch
ps	Parameters list, generated with parameters()

Value

SeqRecs

See Also

Other run-private: batcher, blast_clstr, blast_filter, blast_setup, blast_sqs, blastcache_load, blastcache_save, blastdb_gen, blastn_run, cache_rm, cache_setup, clade_select, clstr2_calc, clstr_all, clstr_direct, clstr_sqs, clstr_subtree, clstrarc_gen, clstrarc_join, clstrrec_gen, clstrs_calc, clstrs_join, clstrs_merge, clstrs_renumber, clstrs_save, cmdln, descendants_get, download_obj_check, error, hierarchic_download, info, ncbicache_load, ncbicache_save, obj_check, obj_load, obj_save, parameters_load, parameters_setup, parent_get, progress_init, progress_read, progress_reset, progress_save, rank_get, safely_connect, search_and_cache, searchterm_gen, seeds_blast, seq_download, seqarc_gen, seqrec_augment, seqrec_gen, seqrec_get, sids_check, sids_get, sids_load, sids_save, sqs_count, sqs_load, sqs_save, stage_args_check, stages_run, tax_download, taxdict_gen, taxtree_gen, txids_get, txnds_count, warn

seqrec_gen *Generate sequence record*

Description

Creates an S4 SeqRec

Usage

```
seqrec_gen(accssn, nm, txid, sq, dfln, orgnsm, ml_typ, rec_typ, vrsn, age,
  lctn = NULL)
```

Arguments

accssn	Accession ID
nm	Sequence name
txid	Taxonomic ID of source organism
sq	Sequence
dfln	Definition line
orgnsm	Source organism name
ml_typ	Molecule type
rec_typ	Sequence record type
vrsn	Accession version
age	Number of days since upload
lctn	Location numbers for features, e.g. '1..200'

Value

SeqRec

See Also

Other run-private: batcher, blast_clstr, blast_filter, blast_setup, blast_sqs, blastcache_load, blastcache_save, blastdb_gen, blastn_run, cache_rm, cache_setup, clade_select, clstr2_calc, clstr_all, clstr_direct, clstr_sqs, clstr_subtree, clstrarc_gen, clstrarc_join, clstrrec_gen, clstrs_calc, clstrs_join, clstrs_merge, clstrs_renumber, clstrs_save, cmdln, descendants_get, download_obj_check, error, hierarchic_download, info, ncbicache_load, ncbicache_save, obj_check, obj_load, obj_save, parameters_load, parameters_setup, parent_get, progress_init, progress_read, progress_reset, progress_save, rank_get, safely_connect, search_and_cache, searchterm_gen, seeds_blast, seq_download, seqarc_gen, seqrec_augment, seqrec_convert, seqrec_get, sids_check, sids_get, sids_load, sids_save, sqs_count, sqs_load, sqs_save, stage_args_check, stages_run, tax_download, taxdict_gen, taxtree_gen, txids_get, txnds_count, warn

---

seqrec_get *seqrec_get*

Description

Downloads sequences from GenBank in batches.

Usage

```
seqrec_get(txid, ps, direct = FALSE, lvl = 0)
```

Arguments

txid	NCBI taxonomic ID
ps	Parameters list, generated with parameters()
direct	Node-level only or subtree as well? Default FALSE.
lvl	Integer, number of message indentations indicating code depth.

Value

Vector of sequence records

See Also

Other run-private: batcher, blast_clstr, blast_filter, blast_setup, blast_sqs, blastcache_load, blastcache_save, blastdb_gen, blastn_run, cache_rm, cache_setup, clade_select, clstr2_calc, clstr_all, clstr_direct, clstr_sqs, clstr_subtree, clstrarc_gen, clstrarc_join, clstrrec_gen, clstrs_calc, clstrs_join, clstrs_merge, clstrs_renumber, clstrs_save, cmdln, descendants_get, download_obj_check, error, hierarchic_download, info, ncbicache_load, ncbicache_save, obj_check, obj_load, obj_save, parameters_load, parameters_setup, parent_get, progress_init, progress_read, progress_reset, progress_save, rank_get, safely_connect, search_and_cache, searchterm_gen, seeds_blast, seq_download, seqarc_gen, seqrec_augment, seqrec_convert, seqrec_gen, sids_check, sids_get, sids_load, sids_save, sqs_count, sqs_load, sqs_save, stage_args_check, stages_run, tax_download, taxdict_gen, taxtree_gen, txids_get, txnds_count, warn

---

seq_download *Download sequences for txids*

Description

Look up and download all sequences for given taxonomic IDs.

Usage

```
seq_download(txids, txdct, ps)
```

Arguments

txids	Taxonomic node IDs, numeric vector
txdct	Taxonomic dictionary
ps	Parameters list, generated with parameters()

Details

Sequence downloads are cached.

See Also

Other run-private: batcher, blast_clstr, blast_filter, blast_setup, blast_sqs, blastcache_load, blastcache_save, blastdb_gen, blastn_run, cache_rm, cache_setup, clade_select, clstr2_calc, clstr_all, clstr_direct, clstr_sqs, clstr_subtree, clstrarc_gen, clstrarc_join, clstrrec_gen, clstrs_calc, clstrs_join, clstrs_merge, clstrs_renumber, clstrs_save, cmdln, descendants_get, download_obj_check, error, hierarchic_download, info, ncbicache_load, ncbicache_save, obj_check, obj_load, obj_save, parameters_load, parameters_setup, parent_get, progress_init, progress_read, progress_reset, progress_save, rank_get, safely_connect, search_and_cache, searchterm_gen, seeds_blast, seqarc_gen, seqrec_augment, seqrec_convert, seqrec_gen, seqrec_get, sids_check, sids_get, sids_load, sids_save, sqs_count, sqs_load, sqs_save, stage_args_check, stages_run, tax_download, taxdict_gen, taxtree_gen, txids_get, txnds_count, warn

---

`setup` *Set-up parameters*

Description

Set up working directory with parameters.

Usage

```
setup(wd, txid, ncbi_dr = ".", v = FALSE, ...)
```

Arguments

`wd`	Working directory
`txid`	Root taxonomic ID(s), vector or numeric
`ncbi_dr`	Directory to NCBI BLAST tools, default '.'
`v`	Verbose, T/F
`...`	Additional parameters

Details

See `parameters`() for a description of all parameters and their defaults. You can change parameters after a folder has been set up with `parameters_reset`().

See Also

Other run-public: `ClstrArc-class`, `ClstrRec-class`, `Phylota-class`, `SeqArc-class`, `SeqRec-class`, `TaxDict-class`, `TaxRec-class`, `clusters2_run`, `clusters_run`, `parameters_reset`, `reset`, `restart`, `run`, `taxise_run`

Examples

```
## Not run:

  # Note: this example requires BLAST to run.

  # example with temp folder
  wd <- file.path(tempdir(), 'aotus')
  # setup for aotus, make sure aotus/ folder already exists
  if (!dir.exists(wd)) {
    dir.create(wd)
  }
  ncbi_dr <- '[SET BLAST+ BIN PATH HERE]'
  # e.g. "/usr/local/ncbi/blast/bin/"
  setup(wd = wd, txid = 9504, ncbi_dr = ncbi_dr)  # txid for Aotus primate genus
  # see ?parameters for all available parameter options

## End(Not run)
```

---

sids_check *Check if sids exist*

Description

Check if sids are already downloaded for a txid.

Usage

```
sids_check(wd, txid)
```

Arguments

wd	Working directory
txid	Taxonomic ID, numeric

Value

T/F

See Also

Other run-private: batcher, blast_clstr, blast_filter, blast_setup, blast_sqs, blastcache_load, blastcache_save, blastdb_gen, blastn_run, cache_rm, cache_setup, clade_select, clstr2_calc, clstr_all, clstr_direct, clstr_sqs, clstr_subtree, clstrarc_gen, clstrarc_join, clstrrec_gen, clstrs_calc, clstrs_join, clstrs_merge, clstrs_renumber, clstrs_save, cmdln, descendants_get, download_obj_check, error, hierarchic_download, info, ncbicache_load, ncbicache_save, obj_check, obj_load, obj_save, parameters_load, parameters_setup, parent_get, progress_init, progress_read, progress_reset, progress_save, rank_get, safely_connect, search_and_cache, searchterm_gen, seeds_blast, seq_download, seqarc_gen, seqrec_augment, seqrec_convert,

seqrec_gen, seqrec_get, sids_get, sids_load, sids_save, sqs_count, sqs_load, sqs_save, stage_args_check, stages_run, tax_download, taxdict_gen, taxtree_gen, txids_get, txnds_count, warn

---

sids_get *Return random set of sequence IDs*

Description

For a given txid return a random set of sequences associated.

Usage

```
sids_get(txid, direct, ps, retmax = 100, hrdmx = 1e+05)
```

Arguments

txid	NCBI taxon identifier
direct	Node-level only or subtree as well? Default FALSE.
ps	Parameters list, generated with parameters()
retmax	Maximum number of sequences when querying model organisms. The smaller the more random, the larger the faster.
hrdmx	Absolute maximum number of sequence IDs to download in a single query.

Details

For model organisms downloading all IDs can a take long time or even cause an xml parsing error. For any search with more than hrdmx sequences, this function we will run multiple small searches downloading retmax seq IDs at a time with different retstart values to generate a semi-random vector of sequence IDs. For all other searches, all IDs will be retrieved. Note, it makes no sense for mdlthrs in parameters to be greater than hrdmx in this function.

Value

vector of IDs

See Also

Other run-private: batcher, blast_clstr, blast_filter, blast_setup, blast_sqs, blastcache_load, blastcache_save, blastdb_gen, blastn_run, cache_rm, cache_setup, clade_select, clstr2_calc, clstr_all, clstr_direct, clstr_sqs, clstr_subtree, clstrarc_gen, clstrarc_join, clstrrec_gen, clstrs_calc, clstrs_join, clstrs_merge, clstrs_renumber, clstrs_save, cmdln, descendants_get, download_obj_check, error, hierarchic_download, info, ncbicache_load, ncbicache_save, obj_check, obj_load, obj_save, parameters_load, parameters_setup, parent_get, progress_init, progress_read, progress_reset, progress_save, rank_get, safely_connect, search_and_cache, searchterm_gen, seeds_blast, seq_download, seqarc_gen, seqrec_augment, seqrec_convert, seqrec_gen, seqrec_get, sids_check, sids_load, sids_save, sqs_count, sqs_load, sqs_save,

stage_args_check, stages_run, tax_download, taxdict_gen, taxtree_gen, txids_get, txnds_count, warn

---

sids_load *Load sids from cache*

Description

Load sids downloaded by sids_get function.

Usage

```
sids_load(wd, txid)
```

Arguments

wd	Working directory
txid	Taxonomic ID, numeric

Value

vector of sids

See Also

Other run-private: batcher, blast_clstr, blast_filter, blast_setup, blast_sqs, blastcache_load, blastcache_save, blastdb_gen, blastn_run, cache_rm, cache_setup, clade_select, clstr2_calc, clstr_all, clstr_direct, clstr_sqs, clstr_subtree, clstrarc_gen, clstrarc_join, clstrrec_gen, clstrs_calc, clstrs_join, clstrs_merge, clstrs_renumber, clstrs_save, cmdln, descendants_get, download_obj_check, error, hierarchic_download, info, ncbicache_load, ncbicache_save, obj_check, obj_load, obj_save, parameters_load, parameters_setup, parent_get, progress_init, progress_read, progress_reset, progress_save, rank_get, safely_connect, search_and_cache, searchterm_gen, seeds_blast, seq_download, seqarc_gen, seqrec_augment, seqrec_convert, seqrec_gen, seqrec_get, sids_check, sids_get, sids_save, sqs_count, sqs_load, sqs_save, stage_args_check, stages_run, tax_download, taxdict_gen, taxtree_gen, txids_get, txnds_count, warn

sids_save *Save sids to cache*

Description

Saves sids downloaded

Usage

```
sids_save(wd, txid, sids)
```

Arguments

wd	Working directory
txid	Taxonomic ID, numeric
sids	sids

See Also

Other run-private: batcher, blast_clstr, blast_filter, blast_setup, blast_sqs, blastcache_load, blastcache_save, blastdb_gen, blastn_run, cache_rm, cache_setup, clade_select, clstr2_calc, clstr_all, clstr_direct, clstr_sqs, clstr_subtree, clstrarc_gen, clstrarc_join, clstrrec_gen, clstrs_calc, clstrs_join, clstrs_merge, clstrs_renumber, clstrs_save, cmdln, descendants_get, download_obj_check, error, hierarchic_download, info, ncbicache_load, ncbicache_save, obj_check, obj_load, obj_save, parameters_load, parameters_setup, parent_get, progress_init, progress_read, progress_reset, progress_save, rank_get, safely_connect, search_and_cache, searchterm_gen, seeds_blast, seq_download, seqarc_gen, seqrec_augment, seqrec_convert, seqrec_gen, seqrec_get, sids_check, sids_get, sids_load, sqs_count, sqs_load, sqs_save, stage_args_check, stages_run, tax_download, taxdict_gen, taxtree_gen, txids_get, txnds_count, warn

sqs_count *Count number of sequences for txid*

Description

Return the number of sequences associated with a taxonomic ID on NCBI GenBank.

Usage

```
sqs_count(txid, ps, direct = FALSE)
```

Arguments

txid	Taxonomic ID
ps	Parameters list, generated with parameters()
direct	Node-level only or subtree as well? Default FALSE.

Value

integer

See Also

Other run-private: batcher, blast_clstr, blast_filter, blast_setup, blast_sqs, blastcache_load, blastcache_save, blastdb_gen, blastn_run, cache_rm, cache_setup, clade_select, clstr2_calc, clstr_all, clstr_direct, clstr_sqs, clstr_subtree, clstrarc_gen, clstrarc_join, clstrrec_gen, clstrs_calc, clstrs_join, clstrs_merge, clstrs_renumber, clstrs_save, cmdln, descendants_get, download_obj_check, error, hierarchic_download, info, ncbicache_load, ncbicache_save, obj_check, obj_load, obj_save, parameters_load, parameters_setup, parent_get, progress_init, progress_read, progress_reset, progress_save, rank_get, safely_connect, search_and_cache, searchterm_gen, seeds_blast, seq_download, seqarc_gen, seqrec_augment, seqrec_convert, seqrec_gen, seqrec_get, sids_check, sids_get, sids_load, sids_save, sqs_load, sqs_save, stage_args_check, stages_run, tax_download, taxdict_gen, taxtree_gen, txids_get, txnds_count, warn

---

sqs_load *Load sequences from cache*

Description

Load sequences downloaded by dwnld function.

Usage

```
sqs_load(wd, txid)
```

Arguments

wd	Working directory
txid	Taxonomic ID, numeric

Value

SeqArc

See Also

Other run-private: batcher, blast_clstr, blast_filter, blast_setup, blast_sqs, blastcache_load, blastcache_save, blastdb_gen, blastn_run, cache_rm, cache_setup, clade_select, clstr2_calc, clstr_all, clstr_direct, clstr_sqs, clstr_subtree, clstrarc_gen, clstrarc_join, clstrrec_gen, clstrs_calc, clstrs_join, clstrs_merge, clstrs_renumber, clstrs_save, cmdln, descendants_get, download_obj_check, error, hierarchic_download, info, ncbicache_load, ncbicache_save, obj_check, obj_load, obj_save, parameters_load, parameters_setup, parent_get, progress_init, progress_read, progress_reset, progress_save, rank_get, safely_connect, search_and_cache, searchterm_gen, seeds_blast, seq_download, seqarc_gen, seqrec_augment, seqrec_convert, seqrec_gen, seqrec_get, sids_check, sids_get, sids_load, sids_save, sqs_count, sqs_save, stage_args_check, stages_run, tax_download, taxdict_gen, taxtree_gen, txids_get, txnds_count, warn

---

sqs_save *Save sequences to cache*

Description

Saves sequences downloaded

Usage

```
sqs_save(wd, txid, sqs)
```

Arguments

wd	Working directory
txid	Taxonomic ID, numeric
sqs	Sequences

Details

Used within the dwnld function. Saves sequence data by txid in cache.

See Also

Other run-private: batcher, blast_clstr, blast_filter, blast_setup, blast_sqs, blastcache_load, blastcache_save, blastdb_gen, blastn_run, cache_rm, cache_setup, clade_select, clstr2_calc, clstr_all, clstr_direct, clstr_sqs, clstr_subtree, clstrarc_gen, clstrarc_join, clstrrec_gen, clstrs_calc, clstrs_join, clstrs_merge, clstrs_renumber, clstrs_save, cmdln, descendants_get, download_obj_check, error, hierarchic_download, info, ncbicache_load, ncbicache_save, obj_check, obj_load, obj_save, parameters_load, parameters_setup, parent_get, progress_init, progress_read, progress_reset, progress_save, rank_get, safely_connect, search_and_cache, searchterm_gen, seeds_blast, seq_download, seqarc_gen, seqrec_augment, seqrec_convert, seqrec_gen, seqrec_get, sids_check, sids_get, sids_load, sids_save, sqs_count, sqs_load, stage_args_check, stages_run, tax_download, taxdict_gen, taxtree_gen, txids_get, txnds_count, warn

stages_run *Sequentially run each stage*

Description

Runs stages from `frm` to `to`. Records stage progress in cache.

Usage

```
stages_run(wd, to, frm, stgs_msg, rstrt = FALSE)
```

Arguments

`wd`	Working directory
`to`	Total number of stages to run
`frm`	Starting stage to run from
`stgs_msg`	Printout stage message for log
`rstrt`	Restarting, T/F

See Also

Other run-private: `batcher`, `blast_clstr`, `blast_filter`, `blast_setup`, `blast_sqs`, `blastcache_load`, `blastcache_save`, `blastdb_gen`, `blastn_run`, `cache_rm`, `cache_setup`, `clade_select`, `clstr2_calc`, `clstr_all`, `clstr_direct`, `clstr_sqs`, `clstr_subtree`, `clstrarc_gen`, `clstrarc_join`, `clstrrec_gen`, `clstrs_calc`, `clstrs_join`, `clstrs_merge`, `clstrs_renumber`, `clstrs_save`, `cmdln`, `descendants_get`, `download_obj_check`, `error`, `hierarchic_download`, `info`, `ncbicache_load`, `ncbicache_save`, `obj_check`, `obj_load`, `obj_save`, `parameters_load`, `parameters_setup`, `parent_get`, `progress_init`, `progress_read`, `progress_reset`, `progress_save`, `rank_get`, `safely_connect`, `search_and_cache`, `searchterm_gen`, `seeds_blast`, `seq_download`, `seqarc_gen`, `seqrec_augment`, `seqrec_convert`, `seqrec_gen`, `seqrec_get`, `sids_check`, `sids_get`, `sids_load`, `sids_save`, `sqs_count`, `sqs_load`, `sqs_save`, `stage_args_check`, `tax_download`, `taxdict_gen`, `taxtree_gen`, `txids_get`, `txnds_count`, `warn`

stage_args_check *Check stage arguments*

Description

Ensures stage arguments are valid, raises an error if not.

Usage

```
stage_args_check(to, frm)
```

Arguments

to	ending stage
frm	starting stage

Value

character, stage message

See Also

Other run-private: batcher, blast_clstr, blast_filter, blast_setup, blast_sqs, blastcache_load, blastcache_save, blastdb_gen, blastn_run, cache_rm, cache_setup, clade_select, clstr2_calc, clstr_all, clstr_direct, clstr_sqs, clstr_subtree, clstrarc_gen, clstrarc_join, clstrrec_gen, clstrs_calc, clstrs_join, clstrs_merge, clstrs_renumber, clstrs_save, cmdln, descendants_get, download_obj_check, error, hierarchic_download, info, ncbicache_load, ncbicache_save, obj_check, obj_load, obj_save, parameters_load, parameters_setup, parent_get, progress_init, progress_read, progress_reset, progress_save, rank_get, safely_connect, search_and_cache, searchterm_gen, seeds_blast, seq_download, seqarc_gen, seqrec_augment, seqrec_convert, seqrec_gen, seqrec_get, sids_check, sids_get, sids_load, sids_save, sqs_count, sqs_load, sqs_save, stages_run, tax_download, taxdict_gen, taxtree_gen, txids_get, txnds_count, warn

---

sturgeons *Example phylota object (sturgeons)*

Description

Final phylota object produced by running pipeline for the sturgeons, Acipenseridae (7900).

Usage

```
data(sturgeons)
```

Format

sturgeons is generated from read_phylota().

Examples

```
data(sturgeons) # load object
str(sturgeons)
```

summary_phylota *Summarise clusters in Phylota Table*

Description

Generates a summary data.frame from all clusters in Phylota object.

Usage

```
summary_phylota(phylota)
```

Arguments

phylota	Phylota object

See Also

Other tools-private: mk_txid_in_sq_mtrx, update_phylota

tardigrades *Example phylota object (tardigrades)*

Description

Final phylota object produced by running pipeline for the tardigrades, Eutardigrada (42242).

Usage

```
data(tardigrades)
```

Format

tardigrades is generated from read_phylota().

Examples

```
data(tardigrades) # load object
str(tardigrades)
```

TaxDict-class *Taxonomic record dictionary*

Description

Taxonomic dictionary contains a taxonomic tree and NCBI taxonomy data for all taxonomic IDs.

Usage

```
## S4 method for signature 'TaxDict'
as.character(x)

## S4 method for signature 'TaxDict'
show(object)

## S4 method for signature 'TaxDict'
print(x)

## S4 method for signature 'TaxDict'
str(object, max.level = 2L, ...)

## S4 method for signature 'TaxDict'
summary(object)
```

Arguments

x	TaxDict object
object	TaxDict object
max.level	Maximum level of nesting for str()
...	Further arguments for str()

Slots

`txids` Taxonomic IDs of taxon records

`recs` Environment of records

`prnt` Parent taxonomic ID

`txtr` Taxonomic tree

See Also

Other run-public: ClstrArc-class, ClstrRec-class, Phylota-class, SeqArc-class, SeqRec-class, TaxRec-class, clusters2_run, clusters_run, parameters_reset, reset, restart, run, setup, taxise_run

Examples

```
data('aotus')
txdct <- aotus@txdct
# this is a TaxDict object
# it contains taxonomic information, including records and tree
show(txdct)
# you can access its different data slots with @
txdct@txids  # taxonomic IDs
txdct@recs   # taxonomic records environment
txdct@txtr   # taxonomic tree
txdct@prnt   # MRCA
# access any record through the records environment
txdct@recs[[txdct@txids[[1]]]]
# for interacting with the taxonomic tree, see the treeman package
summary(txdct@txtr)
```

---

taxdict_gen *Generate taxonomic dictionary*

Description

Takes a vector of txids and a list of taxonomic records and returns a taxonomic dictionary.

Usage

```
taxdict_gen(txids, recs, ps)
```

Arguments

txids	Vector of taxonomic IDs
recs	List of taxonomic records
ps	Parameters list, generated with parameters()

Value

TaxDict

See Also

Other run-private: batcher, blast_clstr, blast_filter, blast_setup, blast_sqs, blastcache_load, blastcache_save, blastdb_gen, blastn_run, cache_rm, cache_setup, clade_select, clstr2_calc, clstr_all, clstr_direct, clstr_sqs, clstr_subtree, clstrarc_gen, clstrarc_join, clstrrec_gen, clstrs_calc, clstrs_join, clstrs_merge, clstrs_renumber, clstrs_save, cmdln, descendants_get, download_obj_check, error, hierarchic_download, info, ncbicache_load, ncbicache_save, obj_check, obj_load, obj_save, parameters_load, parameters_setup, parent_get, progress_init, progress_read, progress_reset, progress_save, rank_get, safely_connect, search_and_cache, searchterm_gen, seeds_blast, seq_download, seqarc_gen, seqrec_augment, seqrec_convert, seqrec_gen, seqrec_get, sids_check, sids_get, sids_load, sids_save, sqs_count, sqs_load,

sqs_save, stage_args_check, stages_run, tax_download, taxtree_gen, txids_get, txnds_count, warn

---

taxise_run *Run taxise stage*

Description

Run the first stage of phylotaR, taxise. This looks up all descendant taxonomic nodes for a given taxonomic ID. It then looks up relevant taxonomic information and generates a taxonomic dictionary for user interaction after phylotaR has completed.

Usage

```
taxise_run(wd)
```

Arguments

wd	Working directory

Details

Objects will be cached.

See Also

Other run-public: ClstrArc-class, ClstrRec-class, Phylota-class, SeqArc-class, SeqRec-class, TaxDict-class, TaxRec-class, clusters2_run, clusters_run, parameters_reset, reset, restart, run, setup

Examples

```
## Not run:

  # Note: this example requires BLAST and internet to run.

  # example with temp folder
  wd <- file.path(tempdir(), 'aotus')
  # setup for aotus, make sure aotus/ folder already exists
  if (!dir.exists(wd)) {
    dir.create(wd)
  }
  ncbi_dr <- '[SET BLAST+ BIN PATH HERE]'
  setup(wd = wd, txid = 9504, ncbi_dr = ncbi_dr)  # txid for Aotus primate genus
  # individually run stages
  taxise_run(wd = wd)

## End(Not run)
```

TaxRec-class *Taxonomic record*

Description

Taxonomic dictionary contains a taxonomic tree and NCBI taxonomy data for all taxonomic IDs.

Usage

```
## S4 method for signature 'TaxRec'
as.character(x)

## S4 method for signature 'TaxRec'
show(object)

## S4 method for signature 'TaxRec'
print(x)

## S4 method for signature 'TaxRec'
str(object, max.level = 2L, ...)

## S4 method for signature 'TaxRec'
summary(object)
```

Arguments

x	TaxRec object
object	TaxRec object
max.level	Maximum level of nesting for str()
...	Further arguments for str()

Slots

id Taxonomic ID

scnm Scientific name

cmnm Common name

rnk Rank

lng Lineage

prnt Parent

See Also

Other run-public: ClstrArc-class, ClstrRec-class, Phylota-class, SeqArc-class, SeqRec-class, TaxDict-class, clusters2_run, clusters_run, parameters_reset, reset, restart, run, setup, taxise_run

Examples

```
data('aotus')
taxrec <- aotus@txdct@recs[[aotus@txdct@txids[[1]]]]
# this is a TaxRec object
# it contains NCBI's taxonomic information for a single node
show(taxrec)
# you can access its different data slots with @
taxrec@id    # taxonomic ID
taxrec@scnm  # scientific name
taxrec@cmnm  # common name, '' if none
taxrec@rnk   # rank
taxrec@lng   # lineage information: list of IDs and ranks
taxrec@prnt  # parent ID
```

---

taxtree_gen *Generate taxonomic tree*

Description

Generate a taxonomic tree for easy look up of taxonomic parents and descendants.

Usage

```
taxtree_gen(prinds, ids, root, ps)
```

Arguments

prinds	Vector of integers indicating preceding node.
ids	Vector of taxonomic IDs
root	ID of root taxon
ps	Parameters list, generated with parameters()

Value

TreeMan

TreeMan class

See Also

Other run-private: batcher, blast_clstr, blast_filter, blast_setup, blast_sqs, blastcache_load, blastcache_save, blastdb_gen, blastn_run, cache_rm, cache_setup, clade_select, clstr2_calc, clstr_all, clstr_direct, clstr_sqs, clstr_subtree, clstrarc_gen, clstrarc_join, clstrrec_gen, clstrs_calc, clstrs_join, clstrs_merge, clstrs_renumber, clstrs_save, cmdln, descendants_get, download_obj_check, error, hierarchic_download, info, ncbicache_load, ncbicache_save, obj_check, obj_load, obj_save, parameters_load, parameters_setup, parent_get, progress_init, progress_read, progress_reset, progress_save, rank_get, safely_connect, search_and_cache, searchterm_gen, seeds_blast, seq_download, seqarc_gen, seqrec_augment, seqrec_convert,

seqrec_gen, seqrec_get, sids_check, sids_get, sids_load, sids_save, sqs_count, sqs_load, sqs_save, stage_args_check, stages_run, tax_download, taxdict_gen, txids_get, txnds_count, warn

---

`tax_download` *Download taxonomic records*

Description

Downloads one batch of taxonomic records.

Usage

```
tax_download(ids, ps)
```

Arguments

ids	Vector of taxonomic IDs
ps	Parameters list, generated with parameters()

Value

list of list

See Also

Other run-private: batcher, blast_clstr, blast_filter, blast_setup, blast_sqs, blastcache_load, blastcache_save, blastdb_gen, blastn_run, cache_rm, cache_setup, clade_select, clstr2_calc, clstr_all, clstr_direct, clstr_sqs, clstr_subtree, clstrarc_gen, clstrarc_join, clstrrec_gen, clstrs_calc, clstrs_join, clstrs_merge, clstrs_renumber, clstrs_save, cmdln, descendants_get, download_obj_check, error, hierarchic_download, info, ncbicache_load, ncbicache_save, obj_check, obj_load, obj_save, parameters_load, parameters_setup, parent_get, progress_init, progress_read, progress_reset, progress_save, rank_get, safely_connect, search_and_cache, searchterm_gen, seeds_blast, seq_download, seqarc_gen, seqrec_augment, seqrec_convert, seqrec_gen, seqrec_get, sids_check, sids_get, sids_load, sids_save, sqs_count, sqs_load, sqs_save, stage_args_check, stages_run, taxdict_gen, taxtree_gen, txids_get, txnds_count, warn

tinamous *Example phylota object (tinamous)*

Description

Final phylota object produced by running pipeline for the tinamous, Tinamiformes (8802).

Usage

```
data(tinamous)
```

Format

`tinamous` is generated from read_phylota().

Examples

```
data(tinamous) # load object
str(tinamous)
```

txids_get *Searches for descendant taxonomic IDs*

Description

Searches NCBI taxonomy for all descendant taxonomic nodes.

Usage

```
txids_get(ps, retmax = 10000)
```

Arguments

`ps`	Parameters list, generated with parameters()
`retmax`	integer, maximum number of IDs to return per query

Value

Vector of txids

vector of ids

See Also

Other run-private: batcher, blast_clstr, blast_filter, blast_setup, blast_sqs, blastcache_load, blastcache_save, blastdb_gen, blastn_run, cache_rm, cache_setup, clade_select, clstr2_calc, clstr_all, clstr_direct, clstr_sqs, clstr_subtree, clstrarc_gen, clstrarc_join, clstrrec_gen, clstrs_calc, clstrs_join, clstrs_merge, clstrs_renumber, clstrs_save, cmdln, descendants_get, download_obj_check, error, hierarchic_download, info, ncbicache_load, ncbicache_save, obj_check, obj_load, obj_save, parameters_load, parameters_setup, parent_get, progress_init, progress_read, progress_reset, progress_save, rank_get, safely_connect, search_and_cache, searchterm_gen, seeds_blast, seq_download, seqarc_gen, seqrec_augment, seqrec_convert, seqrec_gen, seqrec_get, sids_check, sids_get, sids_load, sids_save, sqs_count, sqs_load, sqs_save, stage_args_check, stages_run, tax_download, taxdict_gen, taxtree_gen, txnds_count, warn

---

txnds_count *Count number of descending taxonomic nodes*

Description

Searches NCBI taxonomy and returns number of descendants taxonomic nodes (species, genera ...) of ID.

Usage

```
txnds_count(txid, ps)
```

Arguments

txid	Taxonomic ID
ps	Parameters list, generated with parameters()

Value

integer

See Also

Other run-private: batcher, blast_clstr, blast_filter, blast_setup, blast_sqs, blastcache_load, blastcache_save, blastdb_gen, blastn_run, cache_rm, cache_setup, clade_select, clstr2_calc, clstr_all, clstr_direct, clstr_sqs, clstr_subtree, clstrarc_gen, clstrarc_join, clstrrec_gen, clstrs_calc, clstrs_join, clstrs_merge, clstrs_renumber, clstrs_save, cmdln, descendants_get, download_obj_check, error, hierarchic_download, info, ncbicache_load, ncbicache_save, obj_check, obj_load, obj_save, parameters_load, parameters_setup, parent_get, progress_init, progress_read, progress_reset, progress_save, rank_get, safely_connect, search_and_cache, searchterm_gen, seeds_blast, seq_download, seqarc_gen, seqrec_augment, seqrec_convert, seqrec_gen, seqrec_get, sids_check, sids_get, sids_load, sids_save, sqs_count, sqs_load, sqs_save, stage_args_check, stages_run, tax_download, taxdict_gen, taxtree_gen, txids_get, warn

update_phylota *Update slots*

Description

After change, run to update slots.

Usage

```
update_phylota(phylota)
```

Arguments

phylota	Phylota

Value

Phylota

See Also

Other tools-private: mk_txid_in_sq_mtrx, summary_phylota

warn *Write warning message to log*

Description

Inform a user if a potential error has occurred in log.txt.

Usage

```
warn(ps, ...)
```

Arguments

ps	Parameters list, generated with parameters()
...	Message elements for concatenating

See Also

Other run-private: batcher, blast_clstr, blast_filter, blast_setup, blast_sqs, blastcache_load, blastcache_save, blastdb_gen, blastn_run, cache_rm, cache_setup, clade_select, clstr2_calc, clstr_all, clstr_direct, clstr_sqs, clstr_subtree, clstrarc_gen, clstrarc_join, clstrrec_gen, clstrs_calc, clstrs_join, clstrs_merge, clstrs_renumber, clstrs_save, cmdln, descendants_get, download_obj_check, error, hierarchic_download, info, ncbicache_load, ncbicache_save, obj_check, obj_load, obj_save, parameters_load, parameters_setup, parent_get, progress_init, progress_read, progress_reset, progress_save, rank_get, safely_connect, search_and_cache, searchterm_gen, seeds_blast, seq_download, seqarc_gen, seqrec_augment, seqrec_convert, seqrec_gen, seqrec_get, sids_check, sids_get, sids_load, sids_save, sqs_count, sqs_load, sqs_save, stage_args_check, stages_run, tax_download, taxdict_gen, taxtree_gen, txids_get, txnds_count

---

`write_sqs` *Write out sequences*

Description

Write out sequences, as .fasta, for a given vector of IDs.

Usage

```
write_sqs(phylota, outfile, sid, sq_nm = sid, width = 80)
```

Arguments

`phylota`	Phylota
`outfile`	Output file
`sid`	Sequence ID(s)
`sq_nm`	Sequence name(s)
`width`	Maximum number of characters in a line, integer

Details

The user can control the output definition lines of the sequences using the sq_nm. By default sequences IDs are used. Note, ensure the sq_nm are in the same order as sid.

See Also

Other tools-public: calc_mad, calc_wrdfrq, drop_by_rank, drop_clstrs, drop_sqs, get_clstr_slot, get_nsqs, get_ntaxa, get_sq_slot, get_stage_times, get_tx_slot, get_txids, is_txid_in_clstr, is_txid_in_sq, list_clstrrec_slots, list_ncbi_ranks, list_seqrec_slots, list_taxrec_slots, plot_phylota_pa, plot_phylota_treemap, read_phylota

Examples

```
data('aotus')
# get sequences for a cluster and write out
random_cid <- sample(aotus@cids, 1)
sids <- aotus[[random_cid]]@sids
write_sqs(phylota = aotus, outfile = file.path(tempdir(), 'test.fasta'),
          sq_nm = 'my_gene', sid = sids)
```

`yeasts` *Example phylota object (yeasts)*

Description

Final phylota object produced by running pipeline for the yeasts, Kazachstania (71245).

Usage

```
data(yeasts)
```

Format

`yeasts` is generated from read_phylota().

Examples

```
data(yeasts) # load object
str(yeasts)
```

Index