

Package: onekp (via r-universe)

September 28, 2024

Title Retrieve Data from the 1000 Plants Initiative (1KP)

Version 0.3.0

Description The 1000 Plants Initiative (www.onekp.com) has sequenced the transcriptomes of over 1000 plant species. This package allows these sequences and metadata to be retrieved and filtered by code, species or recursively by clade. Scientific names and NCBI taxonomy IDs are both supported.

Depends R (>= 3.4.0)

Imports rvest, magrittr, dplyr, xml2, methods, rlang, taxizedb (>= 0.1.6)

Remotes ropensci/taxizedb

License MIT + file LICENSE

URL <https://docs.ropensci.org/onekp>, <https://github.com/ropensci/onekp>

BugReports <https://github.com/ropensci/onekp/issues>

Encoding UTF-8

LazyData true

RoxygenNote 7.1.0

Suggests testthat, covr, knitr, rmarkdown

VignetteBuilder knitr

Repository <https://ropensci.r-universe.dev>

RemoteUrl <https://github.com/ropensci/onekp>

RemoteRef master

RemoteSha 6d9f0241194c0cb5d4e1c6422c0444d0d343e76e

Contents

| | |
|--------------------------|---|
| download | 2 |
| filter | 3 |
| onekp | 4 |
| print.OneKP | 5 |
| retrieve_onekp | 5 |

download*Download a dataset*

Description

These functions will return all files in the OneKP object of the given type (protein or DNA FASTA files for `download_peptides` and `download_nucleotides`, respectively). If you do not want to retrieve all these files (there are over a thousand), then you should filter the OneKP object first, using the `filter_by_*` functions.

Usage

```
download_peptides(x, dir = file.path(tempdir(), "peptides"), absolute = FALSE)

download_nucleotides(
  x,
  dir = file.path(tempdir(), "nucleotides"),
  absolute = FALSE
)
```

Arguments

| | |
|-----------------------|---|
| <code>x</code> | OneKP object |
| <code>dir</code> | Directory in which to store the downloaded data |
| <code>absolute</code> | If TRUE, return absolute paths (default=FALSE) |

Value

character vector of paths to the files that were downloaded

Examples

```
## Not run:
data(onekp)

# Filter by 1KP code (from `onekp@table$code` column)
seqs <- filter_by_code(onekp, c('URDJ', 'ROAP'))

# Download FASTA files to temporary directory
download_peptides(seqs)
download_nucleotides(seqs)

## End(Not run)
```

`filter`*Filter a OneKP object*

Description

Filter a OneKP object

Usage

```
filter_by_code(x, code)
```

```
filter_by_clade(x, clade)
```

```
filter_by_species(x, species)
```

Arguments

| | |
|----------------------|---|
| <code>x</code> | OneKP object |
| <code>code</code> | character vector of 1KP IDs (e.g. URDJ) |
| <code>clade</code> | vector of clade-level NCBI taxonomy IDs or scientific names |
| <code>species</code> | vector of species-level scientific names or NCBI taxonomy IDs |

Value

OneKP object

Examples

```
data(onekp)

# filter by 1KP ID
filter_by_code(onekp, c('URDJ', 'ROAP'))

# filter by species name
filter_by_species(onekp, 'Pinus radiata')

# filter by species NCBI taxon ID
filter_by_species(onekp, 3347)

# filter by clade name scientific name
filter_by_clade(onekp, 'Brassicaceae')

# filter by clade NCBI taxon ID
filter_by_clade(onekp, 3700)
```

onekp

OneKP metadata file

Description

The object stored here should be exactly the same as the object returned from `retrieve_onekp()`. It is stored here for convenience and to save time in examples (`retrieve_onekp` takes around 30 seconds to run).

The 1000 Plants Initiative (www.onekp.com) has sequenced the transcriptomes of over 1000 plant species. This package allows these sequences and metadata to be retrieved and filtered by code, species or recursively by clade. Scientific names and NCBI taxonomy IDs are both supported.

Usage

onekp

Format

OneKP object

Main Functions

`retrieve_onekp` - retrieve all 1KP metadata

`filter_by_code` - filter metadata by 1KP code

`filter_by_clade` - filter metadata by clade

`filter_by_species` - filter metadata by species

`download_peptides` - get protein sequences linked to metadata

`download_nucleotides` - get DNA sequences linked to metadata

Author(s)

Zebulun Arendsee <email: zbwrnz@gmail.com>

Bug Reports

Any bugs or issues can be reported at <<https://github.com/ropensci/onekp/issues>>

| | |
|-------------|-------------------------------------|
| print.OneKP | <i>OneKP print generic function</i> |
|-------------|-------------------------------------|

Description

OneKP print generic function

Usage

```
## S3 method for class 'OneKP'  
print(x, ...)
```

Arguments

| | |
|-----|-------------------------------|
| x | OneKP object |
| ... | Additional arguments (unused) |

| | |
|----------------|-------------------------------|
| retrieve_onekp | <i>Retrieve data from 1KP</i> |
|----------------|-------------------------------|

Description

Download the table of metadata for each transcriptome from the 1KP website (http://www.onekp.com/public_data.html). The metadata are wrapped into a OneKp S4 object. This object contains two data.frames: 1) @table, the main metadata table and 2) @links a map from resource to URL (mostly for internal use).

Usage

```
retrieve_onekp(add_taxids = TRUE, filter = TRUE)
```

Arguments

| | |
|------------|--|
| add_taxids | If TRUE, add NCBI taxon ids for each species. This requires downloading the NCBI taxonomy database, which will require a few extra minutes the first time you run the function. This step is necessary only if you wish to filter by NCBI taxon ids. |
| filter | If TRUE, filter out entries that are associated with a single species (for example crosses or datasets pooled across a genus). If set to TRUE, then add_taxids will also be set to TRUE. |

Details

This dataset is also saved as package data, you can access this with `data(onekp)`.

The metadata table contains the following columns:

- species - species scientific name
- code - 4-letter 1KP transcriptome unique identifier
- family - the taxonomic family
- tissue - the tissue(s) that where sequenced
- peptides - the filename for the transcript proteins
- nucleotides - the filename for the transcript DNA
- tax_id (optional) - the species NCBI taxonomy ID

Value

OneKP object

Examples

```
## Not run:  
# scrape data from the OneKP website  
kp <- retrieve_onekp()  
# print to see data summary  
kp  
# access the metadata table  
head(kp@table)  
  
## End(Not run)
```

Index

* **datasets**

onekp, [4](#)

download, [2](#)

download_nucleotides (download), [2](#)

download_peptides (download), [2](#)

filter, [3](#)

filter_by_clade (filter), [3](#)

filter_by_code (filter), [3](#)

filter_by_species (filter), [3](#)

onekp, [4](#)

print.OneKP, [5](#)

retrieve_onekp, [5](#)