Tolkein

Release 0.4.0

Oct 13, 2021

Contents

CHAPTER 1

Tolkein

1.1 Tree of Life Kit of Evolutionary Informatics Novelties

1.2 Installation

```
conda install -c tolkit tolkein
```

or

pip install tolkein

You can also install the in-development version with:

```
pip install https://github.com/tolkit/tolkein/archive/main.zip
```

1.3 Documentation

https://tolkein.readthedocs.io/

1.4 Development

To run all tests run:

```
tox
```

CHAPTER 2

Installation

At the command line:

```
pip install tolkein
```

CHAPTER 3

Usage

To use Tolkein in a project:

```
import tolkein
```

CHAPTER 4

Reference

4.1 tobin

Binning functions.

tolkein.lib.tobin.**readable_bin**(*value*, *, *start_digits=None*)

Place values in human readable bins.

Parameters

- **value** (*float*) – Number to bin.
- **start_digits** (*list, optional*) – List of threshold values between 1 and 10. Defaults to *[1, 2, 3, 5]*

```
>>> readable_bin(123456789)
'200M'

>>> readable_bin(56789012234)
'100G'

>>> readable_bin(567.89)
'1k'

>>> readable_bin(21, start_digits=[1, 2, 4, 8])
'40'

>>> readable_bin(-1234567)
'-2M'
```

4.2 tofetch

Fetch methods.

class tolkein.lib.tofetch.**TqdmUpTo**(*iterable=None*, *desc=None*, *total=None*, *leave=True*, *file=None*, *ncols=None*, *mininterval=0.1*, *maxinterval=10.0*, *miniters=None*, *ascii=None*, *disable=False*, *unit='it'*, *unit_scale=False*, *dynamic_ncols=False*, *smoothing=0.3*, *bar_format=None*, *initial=0*, *position=None*, *postfix=None*, *unit_divisor=1000*, *write_bytes=None*, *lock_args=None*, *nrows=None*, *colour=None*, *delay=0*, *gui=False*, ***kwargs*)

Provides *update_to(n)* which uses *tqdm.update(delta_n)*.

From tqdm documentation.

update_to(*b=1*, *bsize=1*, *tsize=None*)

Tqdm update_to method.

Parameters

- **b** (*int, optional*) – Number of blocks transferred so far [default: 1].
- **bsize** (*int, optional*) – Size of each block (in tqdm units) [default: 1].
- **tsize** (*int, optional*) – Total size (in tqdm units).

tolkein.lib.tofetch.**extract_tar**(*filename*, *path*)

Extract tarred archive.

Parameters

- **filename** (*str*) – Name of tar file to extract.
- **path** (*str*) – Path to extract tar file.

Returns Count of file members extracted from tar archive.

Return type int

tolkein.lib.tofetch.**fetch_file**(*url*, *path*, *decode=True*)

Fetch a remote file.

Parameters

- **url** (*str*) – Remote URL to fetch.
- **path** (*str*) – Path to extract tar file.
- **decode** (*bool, optional*) – Determines whether to unzip content. Defaults to True.

tolkein.lib.tofetch.**fetch_ftp**(*url*, *filename*)

Fetch a file via ftp.

tolkein.lib.tofetch.**fetch_stream**(*url*, ***, *decode=True*, *show_progress=True*)

Stream download.

Parameters

- **decode** (*bool, optional*) – Determines whether to unzip content. Defaults to True.
- **show_progress** (*bool, optional*) – Show a progress bar to indicate file streaming progress. Defaults to True.

Yields *str* – 1024 byte chunk of remote URL.

tolkein.lib.tofetch.**fetch_tar**(*url*, *path*)

Fetch and extract tarred archives.

Parameters

- **url** (*str*) – Remote URL to fetch.
- **path** (*str*) – Path to extract tar file.

Returns Count of file members extracted from tar archive.

Return type int

tolkein.lib.tofetch.**fetch_tmp_file**(*url*)
Fetch a remote URL to a temporary file.

Parameters **url** (*str*) – Remote URL to fetch.

Returns Temporary filename.

Return type str

tolkein.lib.tofetch.**fetch_url**(*url*)
Fetch a URL.

Parameters **url** (*str*) – Remote URL to fetch.

Returns Content of file as a string. Will return None if response is not OK.

Return type str

4.3 tofile

Read, write and parse files.

tolkein.lib.tofile.**delete_file**(*filename*)
Delete a file if exists.

Parameters **filename** (*str*) – Name of file to delete.

tolkein.lib.tofile.**load_yaml**(*filename*)
Parse a JSON/YAML file.

Parameters **filename** (*str*) – Name of JSON/YAML file to parse.

Returns Dict or list of file content.

tolkein.lib.tofile.**open_file_handle**(*filename*)
Open a filehandle.

Automatically detect gzipped files based on suffix.

Parameters **filename** (*str*) – Name of file to read.

Returns An open filehandle. Will return None if the file cannot be opened.

tolkein.lib.tofile.**read_file**(*filename*)
Read a whole file into memory.

Automatically detect gzipped files based on suffix.

Parameters **filename** (*str*) – Name of file to read.

Returns Content of file as a string. Will return None if file cannot be read.

Return type str

tolkein.lib.tofile.**stream_fasta**(*filename*)
Stream a FASTA file, sequence by sequence.

Automatically detect gzipped files based on suffix.

Parameters **filename** (*str*) – Name of FASTA file to read.

Yields A tuple of:

```
(
    str: Sequence ID,
    str: Sequence string
)
```

tolkein.lib.tofile.**write_file**(*filename*, *data*, *, *plain=False*)
Write a file, use suffix to determine type and compression.

- types: '.json', '.yaml'
- compression: None, '.gz'

Parameters

- **filename** (*str*) – Name of FASTA file to read.
- **data** – data to write to file.
- **plain** (*bool, optional*) – Whether to treat data as plain text. Defaults to False.

Returns Whether file was written successfully.

Return type bool

4.4 toinsdc

INSDC methods.

tolkein.lib.toinsdc.**count_taxon_assembly_meta**(*root*)
Count INSDC assemblies descended from root taxon.

Parameters **root** (*int*) – Root taxon taxid.

Returns Count of assemblies for taxa descended from root. Will return None on error.

Return type int

tolkein.lib.toinsdc.**fetch_wgs_assembly_meta**(*root*, *, *count=-1*, *offset=0*, *page=10000*)
Query INSDC WGS assemblies descended from root taxon.

Parameters

- **root** (*int*) – Root taxon taxid.
- **count** (*int*) – Number of assemblies to return. Default value (-1) returns all assemblies.
- **offset** (*int*) – Offset of first assembly to return. Defaults to 0.
- **page** (*int*) – Number of assemblies to fetch per API request. Defaults to 10000.

Yields *dict* – A dict of INSDC WGS assembly metadata keyed on sample accession.

tolkein.lib.toinsdc.**stream_taxon_assembly_meta**(*root*, *, *count=-1*, *offset=0*, *page=10000*)
Query INSDC assemblies descended from root taxon.

Parameters

- **root** (*int*) – Root taxon taxid.

- **count** (*int*) – Number of assemblies to return. Default value (-1) returns all assemblies.
- **offset** (*int*) – Offset of first assembly to return. Defaults to 0.
- **page** (*int*) – Number of assemblies to fetch per API request. Defaults to 10000.

Yields *dict* – Normalised dict of INSDC metadata.

4.5 tolog

Log events.

class tolkein.lib.tolog.**DisableLogger**
Logger context management.

```
>>> my_logger.info('Print log messages')
>>> with DisableLogger():
...     my_logger.info('Disable log messages')
>>> my_logger.info('Print log messages again')
```

__enter__()
Set logging level to critical.

__exit__(*x*, *y*, *z*)
Set logging level back to default.

tolkein.lib.tolog.**logger**(*name='tolkein'*)
Create logger.

Parameters **name** (*str, optional*) – Logger name. Defaults to "tolkein".

Returns A logger instance.

Return type logging.Logger

4.6 totax

Taxonomy methods.

tolkein.lib.totax.**add_xrefs**(*names*, *xrefs*)
Add xrefs to a list of taxon names.

tolkein.lib.totax.**parse_ncbi_names_dmp**(*path*, *nodes*)
Parse names.dmp file and add to nodes dict.

tolkein.lib.totax.**parse_ncbi_nodes_dmp**(*path*)
Parse NCBI format nodes.dmp file.

tolkein.lib.totax.**parse_ncbi_taxdump**(*path*, *root=None*)
Expand lineages from nodes dict.

tolkein.lib.totax.**parse_ott_names_dmp**(*path*, *nodes*)
Parse synonyms.tsv file and add to nodes dict.

tolkein.lib.totax.**parse_ott_nodes_dmp**(*path*)
Parse Open tree of Life taxonomy.tsv file.

tolkein.lib.totax.**parse_ott_taxdump**(*path*, *root=None*)
Expand lineages from nodes dict.

tolkein.lib.totax.**parse_taxonomy**(*taxonomy_type*, *path*, *root=None*)
Parse taxonomy into list of dicts.

tolkein.lib.totax.**stream_nodes**(*nodes*, *roots*)
Add lineage info and stream taxonomy nodes.

CHAPTER 5

Contributing

5.1 Bug reports

When reporting a bug please include:

- Your operating system name and version.
- Any details about your local setup that might be helpful in troubleshooting.
- Detailed steps to reproduce the bug.

5.2 Documentation improvements

Contributions to the official Tolkein docs and internal docstrings are always welcome.

5.3 Feature requests and feedback

The best way to send feedback is to file an issue at https://github.com/tolkit/tolkein/issues.

If you are proposing a feature:

- Explain in detail how it would work.
- Keep the scope as narrow as possible, to make it easier to implement.
- Remember that this is a volunteer-driven project, and that code contributions are welcome :)

5.4 Development

To set up *tolkein* for local development:

1. Fork tolkein (look for the "Fork" button).
2. Clone your fork locally:

```
git clone git@github.com:USERNAME/tolkein.git
```

3. Create a branch for local development:

```
git checkout -b name-of-your-bugfix-or-feature
```

   Now you can make your changes locally.

4. When you're done making changes run all the checks and docs builder with tox one command:

```
tox
```

5. Commit your changes and push your branch to GitHub:

```
git add .
git commit -m "Your detailed description of your changes."
git push origin name-of-your-bugfix-or-feature
```

6. Submit a pull request through the GitHub website.

5.4.1 Pull Request Guidelines

If you need some code review or feedback while you're developing the code just make the pull request.

For merging, you should:

1. Include passing tests (run `tox`)[1].
2. Update documentation when there's new API, functionality etc.
3. Add a note to `CHANGELOG.rst` about the changes.
4. Add yourself to `AUTHORS.rst`.

5.4.2 Tips

To run a subset of tests:

```
tox -e envname -- pytest -k test_myfeature
```

To run all the test environments in *parallel*:

```
tox -p
```

[1] If you don't have all the necessary python versions available locally you can rely on Travis - it will run the tests for each change you add in the pull request.
It will be slower though ...

CHAPTER 6

Authors

- Richard Challis - https://twitter.com/rjchallis

CHAPTER 7

Changelog

7.1 0.2.0

Features: * Added fetch methods * Switched code style to black/flake8

Bug fixes: * Stopped log messages printing twice

7.2 0.0.1 (2020-07-02)

- First release on PyPI.

CHAPTER 8

Indices and tables

- genindex
- modindex
- search

Python Module Index

t

Index

Symbols

A

C

D

E

F

L

O

P

R

S

T

U

W