

---

**genomehubs**

*Release 2.5.17*

**genomehubs**

**Sep 02, 2022**



# CONTENTS

<b>1</b>	<b>GenomeHubs</b>	<b>1</b>
1.1	Installation . . . . .	1
1.2	Documentation . . . . .	1
1.3	Development . . . . .	1
<b>2</b>	<b>Installation</b>	<b>3</b>
<b>3</b>	<b>Usage</b>	<b>5</b>
<b>4</b>	<b>Reference</b>	<b>7</b>
4.1	init . . . . .	7
4.2	parse . . . . .	9
4.3	index . . . . .	10
4.4	fill . . . . .	12
<b>5</b>	<b>Contributing</b>	<b>15</b>
5.1	Bug reports . . . . .	15
5.2	Documentation improvements . . . . .	15
5.3	Feature requests and feedback . . . . .	15
5.4	Development . . . . .	15
<b>6</b>	<b>Authors</b>	<b>19</b>
<b>7</b>	<b>Changelog</b>	<b>21</b>
7.1	2.0.0 (2020-07-02) . . . . .	21
<b>8</b>	<b>Indices and tables</b>	<b>23</b>
	<b>Python Module Index</b>	<b>25</b>
	<b>Index</b>	<b>27</b>



## GENOMEHUBS

### 1.1 Installation

```
conda install -c tolkit genomehubs
```

or

```
pip install genomehubs
```

You can also install the in-development version with:

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

### 1.2 Documentation

<https://genomehubs.readthedocs.io/>

### 1.3 Development

To run all tests run:

```
tox
```



## INSTALLATION

At the command line:

```
pip install genomehubs
```





---

## CHAPTER THREE

---

### USAGE

To use genomehubs in a project:

```
import genomehubs
```



## REFERENCE

### 4.1 init

Initialise a GenomeHubs instance.

**Usage:**

```
genomehubs init [--hub-name STRING] [--hub-path PATH] [--hub-version PATH]
[--config-file PATH...] [--config-save PATH] [--es-host URL...] [--es-url URL] [--insdc-metadata] [--insdc-root INT...] [--restore-indices] [--taxonomy-path PATH] [--taxonomy-source STRING] [--taxonomy-ncbi-root INT] [--taxonomy-ncbi-url URL] [--taxonomy-ott-root INT] [--taxonomy-ott-url URL] [--taxonomy-jsonl PATH] [--taxonomy-format STRING] [--taxonomy-root STRING] [--taxonomy-url URL] [--taxonomy-file PATH...] [--taxon-preload] [--docker-contain STRING...] [--docker-network STRING] [--docker-timeout INT] [--docker-es-container STRING] [--docker-es-image URL] [--reset] [--force-reset] [-h]--help]
[-v]--version]
```

**Options:**

**--hub-name** STRING GenomeHubs instance name.

**--hub-path** PATH GenomeHubs instance root directory path.

**--hub-version** STR GenomeHubs instance version string.

**--config-file** PATH Path to YAML file containing configuration options.

**--config-save** PATH Path to write configuration options to YAML file.

**--es-host** URL ElasticSearch hostname/URL and port.

**--es-url** URL Remote URL to fetch ElasticSearch code.

**--insdc-metadata** Flag to index metadata for public INSDC assemblies.

**--insdc-root** INT Root taxid when indexing public INSDC assemblies.

**--restore-indices** Flag to restore taxon and assembly indices.

**--taxonomy-path** DIR Path to directory containing raw taxonomies.

**--taxonomy-source** STRING Name of taxonomy to use (ncbi or ott).

**--taxonomy-ncbi-root** INT Root taxid for NCBI taxonomy index.

**--taxonomy-ncbi-url** URL Remote URL to fetch NCBI taxonomy.

**--taxonomy-ott-root** INT Root taxid for Open Tree of Life taxonomy index.

**--taxonomy-ott-url** URL Remote URL to fetch Open Tree of Life taxonomy.

**--taxonomy-format STRING** Format of taxonomy (ncbi, ott). Newick support is planned.

**--taxonomy-root STRING** Root taxid.

**--taxonomy-file PATH** Taxonomy file names.

**--taxonomy-url URL** Remote URL to fetch taxonomy.

**--taxonomy-jsonl PATH** Path to JSON Lines format taxonomy file of additional taxa.

**--taxon-preload** Flag to preload all taxa in taxonomy into taxon index.

**--docker-contain STRING** GenomeHubs component to run in Docker.

**--docker-network STRING** Docker network name.

**--docker-timeout STRING** Time in seconds to wait for a component to start in Docker.

**--docker-es-container STRING** ElasticSearch Docker container name.

**--docker-es-image STRING** ElasticSearch Docker image name.

**--reset** Flag to reset GenomeHubs instance if already exists.

**--force-reset** Flag to force reset GenomeHubs instance if already exists.

**-h, --help** Show this

**-v, --version** Show version number

## Examples

# 1. New GenomeHub with default settings `./genomehubs init`

# 2. New GenomeHub in specified directory, populated with Lepidoptera assembly # metadata from INSDC `./genomehubs init --hub-path /path/to/GenomeHub --insdc-root 7088 --insdc-meta`

`genomehubs.lib.init.add_jsonl_to_taxonomy(stream, jsonl)`

Add entries from JSON Lines format file to taxonomy stream.

`genomehubs.lib.init.cli()`

Entry point.

`genomehubs.lib.init.extend_lineage(entry)`

Add current taxon to beginning of lineage.

`genomehubs.lib.init.main(args)`

Initialise genomehubs.

`genomehubs.lib.init.process_subspecies(data)`

Find species name from subspecies and add to lineage.

## 4.2 parse

Parse a local or remote data source.

### Usage:

```
genomehubs parse [-btk] [-btk-root STRING...]
                 [-wikidata PATH] [-wikidata-root STRING...] [-wikidata-xref STRING...] [-gbif] [-gbif-root
                 STRING...] [-gbif-xref STRING...] [-ncbi-datasets-genome PATH] [-ncbi-datasets-sample PATH]
                 [-refseq-mitochondria] [-refseq-organelles] [-refseq-plastids] [-refseq-root NAME] [-outfile PATH] [-
                 h|-help] [-v|-version]
```

### Options:

<b>--btk</b>	Parse assemblies in BlobToolKit
<b>--btk-root STRING</b>	Scientific name of root taxon
<b>--gbif</b>	Parse taxa in GBIF
<b>--gbif-root STRING</b>	GBIF taxon ID of root taxon
<b>--gbif-xref STRING</b>	Include link to external reference from GBIF (e.g. NBN, BOLD)
<b>--meta PATH</b>	YAML format file of analysis metadata
<b>--wikidata PATH</b>	Parse taxa in WikiData dump
<b>--wikidata-root STRING</b>	WikiData taxon ID of root taxon
<b>--wikidata-xref STRING</b>	Include link to external reference from WikiData (e.g. NBN, BOLD)
<b>--ncbi-datasets-genome PATH</b>	Parse NCBI Datasets genome directory
<b>--ncbi-datasets-sample PATH</b>	Parse sample data from NCBI Datasets genome directory
<b>--outfile PATH</b>	Save parsed output to file
<b>--refseq-mitochondria</b>	Parse mitochondrial genomes from the NCBI RefSeq organelle collection
<b>--refseq-organelles</b>	Parse all genomes from the NCBI RefSeq organelle collection
<b>--refseq-plastids</b>	Parse plastid genomes from the NCBI RefSeq organelle collection
<b>--refseq-root NAME</b>	Name (not taxId) of root taxon
<b>-h, --help</b>	Show this
<b>-v, --version</b>	Show version number

`genomehubs.lib.parse.cli()`

Entry point.

`genomehubs.lib.parse.main(args)`

Parse data sources.

`genomehubs.lib.parse.remove_temporary_types(types)`

Remove any keys labelled temporary from a types file.

## 4.3 index

Index a file, directory or repository.

### Usage:

```
genomehubs index [-hub-name STRING] [-hub-path PATH] [-hub-version PATH]
[-config-file PATH...] [-config-save PATH] [-es-host URL...] [-assembly-dir PATH] [-feature-dir
PATH] [-sample-dir PATH] [-taxon-dir PATH] [-taxon-repo URL] [-taxon-exception PATH] [-taxon-
lookup STRING] [-taxon-lookup-root STRING] [-taxon-lookup-in-memory] [-taxon-id-as-xref STRING]
[-taxon-matching-ranks INT] [-taxon-spellcheck] [-taxonomy-source STRING] [-file PATH...] [file-dir
PATH...] [-remote-file URL...] [-remote-file-dir URL...] [-taxon-id STRING] [-assembly-id STRING]
[-sample-id STRING] [-analysis-id STRING] [-file-title STRING] [-file-description STRING] [-file-
metadata PATH] [-dry-run] [-h|-help] [-v|-version]
```

### Options:

**--hub-name STRING** GenomeHubs instance name.

**--hub-path PATH** GenomeHubs instance root directory path.

**--hub-version STR** GenomeHubs instance version string.

**--config-file PATH** Path to YAML file containing configuration options.

**--config-save PATH** Path to write configuration options to YAML file.

**--es-host URL** Elasticsearch hostname/URL and port.

**--assembly-dir PATH** Path to directory containing assembly-level data.

**--sample-dir PATH** Path to directory containing sample-level data.

**--feature-dir PATH** Path to directory containing feature-level data.

**-taxon-lookup-root STRING** Root taxon Id for in-memory lookup. **-taxon-lookup STRING** Taxon name class to lookup (scientific|any). [Default: scientific] **-taxon-lookup-in-memory** Flag to use in-memory taxon name lookup. **-taxon-id-as-xref STRING** Set source DB name to treat taxon\_id in file as xref. **-taxon-matching-ranks INT** Number of ancestral ranks that must match to import a taxon based on name match. [Default: 2]

**--taxon-spellcheck** Flag to use fuzzy matching to match taxon names.

**--taxon-dir PATH** Path to directory containing taxon-level data.

**--taxon-repo URL** Remote git repository containing taxon-level data. Optionally include *~branch-name* suffix.

**--taxon-exception PATH** Path to directory to write taxon data that failed to import.

**--taxonomy-source STRING** Name of taxonomy to use (ncbi or ott).

**--file PATH** Path to file for generic file import.

**--file-dir PATH** Path to directory containing generic files to import.

**--remote-file URL** Location of remote file for generic file import.

**--remote-file-dir URL** Location of remote directory containing generic files to import.

**--taxon-id STRING** Taxon ID to index files against.

**--sample-id STRING** Sample ID to index files against.

**--assembly-id STRING** Assembly ID to index files against.  
**--analysis-id STRING** Analysis ID to index files against.  
**--file-title STRING** Default title for indexed files.  
**--file-description STRING** Default description for all indexed files.  
**--file-metadata PATH** CSV, TSV, YAML or JSON file metadata with one entry per file to be indexed.  
**--dry-run** Flag to run without loading data into the elasticsearch index.  
**-h, --help** Show this  
**-v, --version** Show version number

## Examples

# 1. Index all files in a remote repository `/genomehubs index --taxon-repo https://github.com/genomehubs/goat-data`

`genomehubs.lib.index.cli()`

Entry point.

`genomehubs.lib.index.convert_features_to_docs(with_ids)`

Convert features to docs.

`genomehubs.lib.index.group_rows(taxon_id, rows, with_ids, without_ids, taxon_asm_data, imported_rows, types, failed_rows, blanks)`

Group processed rows by available taxon info for import.

`genomehubs.lib.index.index_feature_records(es, opts, taxonomy_name, with_ids, blanks)`

Index a feature records.

`genomehubs.lib.index.index_features(es, opts, *, dry_run=False)`

Index assembly features.

`genomehubs.lib.index.index_file(es, types, names, data, opts, *, taxon_table=None, shared_values=None, exclusions=None)`

Index a file.

`genomehubs.lib.index.index_sample_records(es, taxonomy_name, opts, with_ids, blanks, taxon_types, taxon_asm_data, index_type='sample')`

Index sample records.

`genomehubs.lib.index.index_taxon_records(es, taxonomy_name, opts, with_ids, blanks, types)`

Index a taxon records.

`genomehubs.lib.index.index_taxon_sample(es, opts, index='taxon', *, dry_run=False, taxonomy_name)`

Call taxon- or sample-specific indexing functions.

`genomehubs.lib.index.main(args)`

Index files.

`genomehubs.lib.index.not_blank(key, obj, blanks)`

Test value is not blank.

```
genomehubs.lib.index.process_taxon_sample_records(es, taxonomy_name, opts, processed_rows,
                                                    with_ids, blanks, taxon_asm_data, imported_rows,
                                                    types, failed_rows, header, taxon_table,
                                                    taxon_types)
```

Process taxon and sample records.

```
genomehubs.lib.index.set_feature_types(types)
```

Set types for feature properties.

```
genomehubs.lib.index.summarise_imported_taxa(docs, imported_taxa)
```

Summarise taxon information from a stram of taxon docs.

## 4.4 fill

Fill attribute values.

**Usage:**

```
genomehubs fill [-hub-name STRING] [-hub-path PATH] [-hub-version PATH]
                [-config-file PATH...] [-config-save PATH] [-es-host URL...] [-taxonomy-source STRING] [-traverse-
                limit STRING] [-traverse-infer-ancestors] [-traverse-infer-descendants] [-traverse-infer-both] [-traverse-
                threads INT] [-traverse-depth INT] [-traverse-root STRING] [-traverse-weight STRING] [-h|-help] [-
                v|-version]
```

**Options:**

```
--hub-name STRING  GenomeHubs instance name.
--hub-path PATH    GenomeHubs instance root directory path.
--hub-version STR  GenomeHubs instance version string.
--config-file PATH Path to YAML file containing configuration options.
--config-save PATH Path to write configuration options to YAML file.
--es-host URL      Elasticsearch hostname/URL and port.
--taxonomy-source STRING  Name of taxonomy to use (ncbi or ott).
--traverse-depth INT  Maximum depth for tree traversal relative to root taxon.
--traverse-infer-ancestors  Flag to enable tree traversal from tips to root.
--traverse-infer-descendants  Flag to enable tree traversal from root to tips.
--traverse-infer-both  Flag to enable tree traversal from tips to root and back to tips.
--traverse-limit STRING  Maximum rank to ascend to during traversal. [Default: null]
--traverse-root ID     Root taxon id for tree traversal.
--traverse-threads INT  Number of threads to use for tree traversal. [Default: 1]
--traverse-weight STRING  Weighting scheme for setting values during tree traversal.
-h, --help          Show this
-v, --version        Show version number
```



## Examples

# 1. Traverse tree up to taxon\_id 7088 ./genomehubs fill --traverse-root 7088

```
genomehubs.lib.fill.apply_summary(summary, values, *, primary_values=None, summary_types=None,
                                   max_value=None, min_value=None, order=None)
```

Apply summary statistic functions.

```
genomehubs.lib.fill.cli()
```

Entry point.

```
genomehubs.lib.fill.copy_attribute_summary(source, meta)
```

Copy an attribute summary, removing values.

```
genomehubs.lib.fill.deduped_list(arr)
```

Remove duplicate values from a list.

```
genomehubs.lib.fill.deduped_list_length(arr)
```

Find number of unique values in a list.

```
genomehubs.lib.fill.earliest(arr, *args)
```

Select earliest date from a list.

```
genomehubs.lib.fill.enum(tup)
```

Use list index to prioritise values.

```
genomehubs.lib.fill.flatten_list(arr)
```

Flatten a list by expanding any nested lists.

```
genomehubs.lib.fill.get_max_depth(es, *, index)
```

Find max depth of root lineage.

```
genomehubs.lib.fill.get_max_depth_by_lineage(es, *, index, root)
```

Find max depth of specified root lineage.

```
genomehubs.lib.fill.latest(arr, *args)
```

Select earliest date from a list.

```
genomehubs.lib.fill.main(args)
```

Initialise genomehubs.

```
genomehubs.lib.fill.range(arr)
```

Calculate difference between max and min values.

```
genomehubs.lib.fill.set_aggregation_source(attribute, source=None)
```

Set attribute aggregation source.

```
genomehubs.lib.fill.set_attributes_to_descend(meta, traverse_limit)
```

Set which attributes should have values inferred from ancestral taxa.

```
genomehubs.lib.fill.set_traverse_values(summaries, values, primary_values, count, max_value,
                                         min_value, meta, attribute, value_type, traverse, source)
```

Set values use for tree traversal.

```
genomehubs.lib.fill.set_values_from_descendants(*, attributes, descendant_values, meta, taxon_id,
                                              parent, taxon_rank, traverse_limit, parents,
                                              descendant_ranks=None, attr_dict=None,
                                              limits=None)
```

Set attribute summary values from descendant values.

```
genomehubs.lib.fill.stream_descendant_nodes_missing_attributes(es, *, index, attributes, root,
                                                                size=10)
```

Get entries descended from root that lack one or more attributes.

```
genomehubs.lib.fill.stream_missing_attributes_at_level(es, *, nodes, attrs, template, level=1)
```

Stream all descendant nodes with missing attributes.

```
genomehubs.lib.fill.stream_nodes_by_root_depth(es, *, index, root, depth, size=10)
```

Get entries by depth of root taxon.

```
genomehubs.lib.fill.summarise_attribute_values(attribute, meta, *, values=None, count=0,
                                                max_value=None, min_value=None, source='direct')
```

Calculate a single summary value for an attribute.

```
genomehubs.lib.fill.summarise_attributes(*, attributes, attrs, meta, parent, parents)
```

Set attribute summary values.

```
genomehubs.lib.fill.track_descendant_ranks(node, descendant_ranks)
```

Keep track of descendant ranks.

```
genomehubs.lib.fill.track_missing_attribute_values(node, missing_attributes, attr_dict, desc_attrs,
                                                    desc_attr_limits)
```

Keep track of missing attribute values for in memory traversal.

```
genomehubs.lib.fill.traverse_from_root(es, opts, *, template, root=None, max_depth=None, log=True)
```

Traverse a tree, filling in values.

```
genomehubs.lib.fill.traverse_from_tips(es, opts, *, template, root=None, max_depth=None)
```

Traverse a tree, filling in values.

```
genomehubs.lib.fill.traverse_handler(es, opts, template)
```

Handle single or multi-threaded tree traversal.

```
genomehubs.lib.fill.traverse_helper(params)
```

Wrap traverse\_tree for multithreaded traversal.

```
genomehubs.lib.fill.traverse_tree(es, opts, template, root, max_depth)
```

Propagate values by tree traversal.

## 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 *genomehubs* 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/genomehubs/genomehubs/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 code contributions are welcome

### 5.4 Development

To install the development version of *genomehubs*:

1. Clone the *genomehubs* repository:

```
git clone https://github.com/genomehubs/genomehubs
```

2. Install the dependencies using pip:

```
cd genomehubs  
pip install -r requirements.txt
```

3. Build and install the *genomehubs* package:

```
python3 setup.py sdist bdist_wheel \  
&& echo y | pip uninstall genomehubs \  
&& pip install dist/genomehubs-2.0.0-py3-none-any.whl
```

To set up *genomehubs* for local development:

1. Fork *genomehubs* <<https://github.com/genomehubs/genomehubs>> - (look for the “Fork” button).
2. Clone your fork locally:

```
git clone git@github.com:USERNAME/genomehubs.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*)<sup>1</sup>.
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*.

---

<sup>1</sup> 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 ...

### 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
```



## **AUTHORS**

- Richard Challis - <https://twitter.com/rjchallis>
- Sujai Kumar - <https://twitter.com/sujaik>





## CHANGELOG

### 7.1 2.0.0 (2020-07-02)

- First release on PyPI.



## INDICES AND TABLES

- `genindex`
- `modindex`
- `search`



## PYTHON MODULE INDEX

### g

`genomehubs.lib.fill`, [12](#)  
`genomehubs.lib.index`, [10](#)  
`genomehubs.lib.init`, [7](#)  
`genomehubs.lib.parse`, [9](#)



## A

`add_jsonl_to_taxonomy()` (in module `genomehubs.lib.init`), 8  
`apply_summary()` (in module `genomehubs.lib.fill`), 13

## C

`cli()` (in module `genomehubs.lib.fill`), 13  
`cli()` (in module `genomehubs.lib.index`), 11  
`cli()` (in module `genomehubs.lib.init`), 8  
`cli()` (in module `genomehubs.lib.parse`), 9  
`convert_features_to_docs()` (in module `genomehubs.lib.index`), 11  
`copy_attribute_summary()` (in module `genomehubs.lib.fill`), 13

## D

`deduped_list()` (in module `genomehubs.lib.fill`), 13  
`deduped_list_length()` (in module `genomehubs.lib.fill`), 13

## E

`earliest()` (in module `genomehubs.lib.fill`), 13  
`enum()` (in module `genomehubs.lib.fill`), 13  
`extend_lineage()` (in module `genomehubs.lib.init`), 8

## F

`flatten_list()` (in module `genomehubs.lib.fill`), 13

## G

`genomehubs.lib.fill`  
 module, 12  
`genomehubs.lib.index`  
 module, 10  
`genomehubs.lib.init`  
 module, 7  
`genomehubs.lib.parse`  
 module, 9  
`get_max_depth()` (in module `genomehubs.lib.fill`), 13  
`get_max_depth_by_lineage()` (in module `genomehubs.lib.fill`), 13  
`group_rows()` (in module `genomehubs.lib.index`), 11

## I

`index_feature_records()` (in module `genomehubs.lib.index`), 11  
`index_features()` (in module `genomehubs.lib.index`), 11  
`index_file()` (in module `genomehubs.lib.index`), 11  
`index_sample_records()` (in module `genomehubs.lib.index`), 11  
`index_taxon_records()` (in module `genomehubs.lib.index`), 11  
`index_taxon_sample()` (in module `genomehubs.lib.index`), 11

## L

`latest()` (in module `genomehubs.lib.fill`), 13

## M

`main()` (in module `genomehubs.lib.fill`), 13  
`main()` (in module `genomehubs.lib.index`), 11  
`main()` (in module `genomehubs.lib.init`), 8  
`main()` (in module `genomehubs.lib.parse`), 9  
 module  
   `genomehubs.lib.fill`, 12  
   `genomehubs.lib.index`, 10  
   `genomehubs.lib.init`, 7  
   `genomehubs.lib.parse`, 9

## N

`not_blank()` (in module `genomehubs.lib.index`), 11

## P

`process_subspecies()` (in module `genomehubs.lib.init`), 8  
`process_taxon_sample_records()` (in module `genomehubs.lib.index`), 11

## R

`range()` (in module `genomehubs.lib.fill`), 13  
`remove_temporary_types()` (in module `genomehubs.lib.parse`), 9

## S

[set\\_aggregation\\_source\(\)](#) (in module *genomehubs.lib.fill*), [13](#)  
[set\\_attributes\\_to\\_descend\(\)](#) (in module *genomehubs.lib.fill*), [13](#)  
[set\\_feature\\_types\(\)](#) (in module *genomehubs.lib.index*), [12](#)  
[set\\_traverse\\_values\(\)](#) (in module *genomehubs.lib.fill*), [13](#)  
[set\\_values\\_from\\_descendants\(\)](#) (in module *genomehubs.lib.fill*), [13](#)  
[stream\\_descendant\\_nodes\\_missing\\_attributes\(\)](#) (in module *genomehubs.lib.fill*), [14](#)  
[stream\\_missing\\_attributes\\_at\\_level\(\)](#) (in module *genomehubs.lib.fill*), [14](#)  
[stream\\_nodes\\_by\\_root\\_depth\(\)](#) (in module *genomehubs.lib.fill*), [14](#)  
[summarise\\_attribute\\_values\(\)](#) (in module *genomehubs.lib.fill*), [14](#)  
[summarise\\_attributes\(\)](#) (in module *genomehubs.lib.fill*), [14](#)  
[summarise\\_imported\\_taxa\(\)](#) (in module *genomehubs.lib.index*), [12](#)

## T

[track\\_descendant\\_ranks\(\)](#) (in module *genomehubs.lib.fill*), [14](#)  
[track\\_missing\\_attribute\\_values\(\)](#) (in module *genomehubs.lib.fill*), [14](#)  
[traverse\\_from\\_root\(\)](#) (in module *genomehubs.lib.fill*), [14](#)  
[traverse\\_from\\_tips\(\)](#) (in module *genomehubs.lib.fill*), [14](#)  
[traverse\\_handler\(\)](#) (in module *genomehubs.lib.fill*), [14](#)  
[traverse\\_helper\(\)](#) (in module *genomehubs.lib.fill*), [14](#)  
[traverse\\_tree\(\)](#) (in module *genomehubs.lib.fill*), [14](#)