




---

## NCBI Genome Downloading Scripts

   [10.5281/zenodo.8192486](https://doi.org/10.5281/zenodo.8192486)

Some script to download bacterial and fungal genomes from NCBI after they restructured their FTP a while ago.

Idea shamelessly stolen from Mick Watson's Kraken downloader scripts that can also be found in Mick's GitHub repo. However, Mick's scripts are written in Perl specific to actually building a Kraken database (as advertised).

So this is a set of scripts that focuses on the actual genome downloading.

### Installation

```
1 pip install ncbi-genome-download
```

Alternatively, clone this repository from GitHub, then run (in a python virtual environment)

```
1 pip install .
```

If this fails on older versions of Python, try updating your `pip` tool first:

```
1 pip install --upgrade pip
```

and then rerun the `ncbi-genome-download` install.

Alternatively, `ncbi-genome-download` is packaged in `conda`. Refer the the Anaconda/miniconda site to install a distribution (highly recommended). With that installed one can do:

```
1 conda install -c bioconda ncbi-genome-download
```

`ncbi-genome-download` is only developed and tested on Python releases still under active support by the Python project. At the moment, this means versions 3.7, 3.8, 3.9, 3.10 and 3.11. Specifically, no attempt at testing under Python versions older than 3.7 is being made.

If your system is stuck on an older version of Python, consider using a tool like Homebrew to obtain a more up-to-date version.

`ncbi-genome-download` 0.2.12 was the last version to support Python 2.

### Usage

To download all bacterial RefSeq genomes in GenBank format from NCBI, run the following:

---

```
1 ncbi-genome-download bacteria
```

Downloading multiple groups is also possible:

```
1 ncbi-genome-download bacteria,viral
```

**Note:** To see all available groups, see `ncbi-genome-download --help`, or simply use `all` to check all groups. Naming a more specific group will reduce the download size and the time needed to find the sequences to download.

If you're on a reasonably fast connection, you might want to try running multiple downloads in parallel:

```
1 ncbi-genome-download bacteria --parallel 4
```

To download all fungal GenBank genomes from NCBI in GenBank format, run:

```
1 ncbi-genome-download --section genbank fungi
```

To download all viral RefSeq genomes in FASTA format, run:

```
1 ncbi-genome-download --formats fasta viral
```

It is possible to download multiple formats by supplying a list of formats or simply downloading all formats:

```
1 ncbi-genome-download --formats fasta,assembly-report viral
2 ncbi-genome-download --formats all viral
```

To download only completed bacterial RefSeq genomes in GenBank format, run:

```
1 ncbi-genome-download --assembly-levels complete bacteria
```

It is possible to download multiple assembly levels at once by supplying a list:

```
1 ncbi-genome-download --assembly-levels complete,chromosome bacteria
```

To download only bacterial reference genomes from RefSeq in GenBank format, run:

```
1 ncbi-genome-download --refseq-categories reference bacteria
```

To download bacterial RefSeq genomes of the genus *Streptomyces*, run:

```
1 ncbi-genome-download --genera Streptomyces bacteria
```

**Note:** This is a simple string match on the organism name provided by NCBI only.

You can also use this with a slight trick to download genomes of a certain species as well:

---

```
1 ncbi-genome-download --genera "Streptomyces coelicolor" bacteria
```

**Note:** The quotes are important. Again, this is a simple string match on the organism name provided by the NCBI.

Multiple genera is also possible:

```
1 ncbi-genome-download --genera "Streptomyces coelicolor,Escherichia coli" bacteria
```

You can also put genus names into a file, one organism per line, e.g.:

```
1 Streptomyces
2 Amycolatopsis
```

Then, pass the path to that file (e.g. `my_genera.txt`) to the `--genera` option, like so:

```
1 ncbi-genome-download --genera my_genera.txt bacteria
```

**Note:** The above command will download all *Streptomyces* and *Amycolatopsis* genomes from RefSeq.

You can make the string match fuzzy using the `--fuzzy-genus` option. This can be handy if you need to match a value in the middle of the NCBI organism name, like so:

```
1 ncbi-genome-download --genera coelicolor --fuzzy-genus bacteria
```

**Note:** The above command will download all bacterial genomes containing “coelicolor” anywhere in their organism name from RefSeq.

To download bacterial RefSeq genomes based on their NCBI species taxonomy ID, run:

```
1 ncbi-genome-download --species-taxids 562 bacteria
```

**Note:** The above command will download all RefSeq genomes belonging to *Escherichia coli*.

To download a specific bacterial RefSeq genomes based on its NCBI taxonomy ID, run:

```
1 ncbi-genome-download --taxids 511145 bacteria
```

**Note:** The above command will download the RefSeq genome belonging to *Escherichia coli str. K-12 substr. MG1655*.

It is also possible to download multiple species taxids or taxids by supplying the numbers in a comma-separated list:

```
1 ncbi-genome-download --taxids 9606,9685 --assembly-level chromosome vertebrate_mammalian
```

---

**Note:** The above command will download the reference genomes for cat and human.

In addition, you can put multiple species taxids or taxids into a file, one per line and pass that filename to the `--species-taxids` or `--taxids` parameters, respectively.

Assuming you had a file `my_taxids.txt` with the following contents:

```
1 9606
2 9685
```

You could download the reference genomes for cat and human like this:

```
1 ncbi-genome-download --taxids my_taxids.txt --assembly-levels
  chromosome vertebrate_mammalian
```

It is possible to also create a human-readable directory structure in parallel to mirroring the layout used by NCBI:

```
1 ncbi-genome-download --human-readable bacteria
```

This will use links to point to the appropriate files in the NCBI directory structure, so it saves file space. Note that links are not supported on some Windows file systems and some older versions of Windows.

It is also possible to re-run a previous download with the `--human-readable` option. In this case, `ncbi-genome-download` will not download any new genome files, and just create human-readable directory structure. Note that if any files have been changed on the NCBI side, a file download will be triggered.

There is a “dry-run” option to show which accessions would be downloaded, given your filters:

```
1 ncbi-genome-download --dry-run bacteria
```

If you want to filter for the “relation to type material” column of the assembly summary file, you can use the `--type-materials` option. Possible values are “any”, “all”, “type”, “reference”, “synonym”, “proxytype”, and/or “neotype”. “any” will include assemblies with no relation to type material value defined, “all” will download only assemblies with a defined value. Multiple values can be given, separated by comma:

```
1 ncbi-genome-download --type-materials type,reference
```

By default, `ncbi-genome-download` caches the assembly summary files for the respective taxonomic groups for one day. You can skip using the cache file by using the `--no-cache` option. The output of `--help` also shows the cache directory, should you want to remove any of the cached files.

To get an overview of all options, run

---

```
1 ncbi-genome-download --help
```

### As a method

You can also use it as a method call. Pass the pythonised keyword arguments (`_` instead of `-`) as described above or in the `--help`:

```
1 import ncbi_genome_download as ngd
2 ngd.download()
```

**Note:** To specify a taxonomic group, like *bacteria*, use the `group` keyword.

### Contributed Scripts: `gimme_taxa.py`

This script lets you find out what TaxIDs to pass to `ngd`, and will write a simple one-item-per-line file to pass in to it. It utilises the `ete3` toolkit, so refer to their site to install the dependency if it's not already satisfied.

You can query the database using a particular TaxID, or a scientific name. The primary function of the script is to return all the child taxa of the specified parent taxa. The script has various options for what information is written in the output.

A basic invocation may look like:

```
1 # Fetch all descendent taxa for Escherichia (taxid 561):
2 python gimme_taxa.py -o ~/mytaxafile.txt 561
3
4 # Alternatively, just provide the taxon name
5 python gimme_taxa.py -o all_descendent_taxids.txt Escherichia
6
7 # You can provide multiple taxids and/or names
8 python gimme_taxa.py -o all_descendent_taxids.txt 561,
   Methanobrevibacter
```

On first use, a small sqlite database will be created in your home directory by default (change the location with the `--database` flag). You can update this database by using the `--update` flag. Note that if the database is not in your home directory, you must specify it with `--database` or a new database will be created in your home directory.

To see all help:

```
1 python gimme_taxa.py
2 python gimme_taxa.py -h
3 python gimme_taxa.py --help
```

---

## **Citing `ncbi-genome-download`**

You can cite `ncbi-genome-download` via the Zenodo deposit under DOI: 10.5281/zenodo.8192432 or the specific DOI for the version you used.

## **License**

All code is available under the Apache License version 2, see the [LICENSE](#) file for details.