

---

# **Bold Retriever Documentation**

***Release 1.0.0***

**Carlos Pena**

April 10, 2015



<b>1</b>	<b>Bold Retriever</b>	<b>3</b>
1.1	Run this way . . . . .	3
1.2	Speed . . . . .	4
1.3	Full documentation . . . . .	4
<b>2</b>	<b>Installation</b>	<b>5</b>
<b>3</b>	<b>Usage</b>	<b>7</b>
3.1	How to run <code>bold_retriever</code> . . . . .	7
3.2	As an alternative you can use <code>bold_retriever</code> as a Python module . . . . .	7
<b>4</b>	<b>Contributing</b>	<b>9</b>
4.1	Types of Contributions . . . . .	9
4.2	Get Started! . . . . .	10
4.3	Pull Request Guidelines . . . . .	10
4.4	Tips . . . . .	11
<b>5</b>	<b>Credits</b>	<b>13</b>
5.1	Development Lead . . . . .	13
5.2	Contributors . . . . .	13
<b>6</b>	<b>History</b>	<b>15</b>
<b>7</b>	<b>Indices and tables</b>	<b>17</b>



Contents:



---

## Bold Retriever

---

This script accepts FASTA files containing COI sequences. It queries the BOLD database <http://boldsystems.org/> in order to get the taxa identification based on the sequences.

### 1.1 Run this way

- clone repository:

```
cd $USERAPPL
git clone https://github.com/carlosp420/bold_retriever.git
```

- install dependencies (python2.7):

```
cd bold_retriever
module load biopython-env
pip install -r requirements.txt
```

- run software

You have to choose one of the databases available from BOLD <http://www.boldsystems.org/index.php/resources/api?type=idengine> and enter it as argument:

- COX1\_SPECIES
- COX1
- COX1\_SPECIES\_PUBLIC
- COX1\_L640bp

For example:

```
python bold_retriever.py -f ZA2013-0565.fasta -db COX1_SPECIES
```

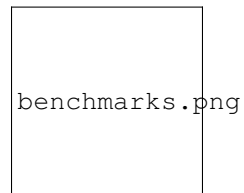
- output:

seq_id	bold_id	similarity	division	class	order	family	species
OTU_99	FBNE064-11	1	animal	Insecta	Neuroptera	Hemerobiidae	Hemerobius pin
OTU_99	NEUF1079-11	1	animal	Insecta	Neuroptera	Hemerobiidae	Hemerobius pin
OTU_99	FBNE172-13	0.9937	animal	Insecta	Neuroptera	Hemerobiidae	Hemerobius atr
OTU_99	FBNE162-13	0.9936	animal	Insecta	Neuroptera	Hemerobiidae	Hemerobius con
OTU_99	TTSOW138-09	0.9811	animal	Insecta	Neuroptera	Hemerobiidae	Hemerobius ova
OTU_99	CNPAH380-13	0.9811	animal	Insecta	Neuroptera	Hemerobiidae	Hemerobius
OTU_99	CNKOF1602-14	0.9811	animal	Insecta	Neuroptera	Hemerobiidae	Hemerobius pin
OTU_99	NRAS173-11	0.9748	animal	Insecta	Neuroptera	Hemerobiidae	Hemerobius con

OTU_99	SSBAE2911-13	0.9748	animal	Collembola	None	None	Collembola
OTU_99	CNPAQ117-13	0.9686	animal	Insecta	Neuroptera	Hemerobiidae	Hemerobius hum

## 1.2 Speed

**bold\_retriever** uses the library Twisted for performing asynchronous calls. This speeds up the total processing time:



## 1.3 Full documentation

See the full documentation at <http://bold-retriever.readthedocs.org>



---

# Installation

---

You can download the latest version of the software here: [https://github.com/carlosp420/bold\\_retriever/releases](https://github.com/carlosp420/bold_retriever/releases)

Or, at the command line:

```
$ # Clone repository
$ git clone https://github.com/carlosp420/bold_retriever.git
$ cd bold_retriever
$ # install dependencies
$ pip install -r requirements.txt
```

Run the software by specifying a FASTA file as input and a BOLD database for queries:

```
$ python bold_retriever.py -f ZA2013-0565.fasta -db COX1_SPECIES
```

As an alternative, if you have virtualenvwrapper installed:

```
$ # install software
$ mkvirtualenv bold_retriever
$ pip install bold_retriever
$ # install dependencies
$ pip install -r requirements.txt
```



---

## Usage

---

### 3.1 How to run `bold_retriever`

You have to choose one of the databases available from BOLD <http://www.boldsystems.org/index.php/resources/api?type=idengine> and enter it as argument:

- COX1\_SPECIES
- COX1
- COX1\_SPECIES\_PUBLIC
- COX1\_L640bp

For example:

```
python bold_retriever.py -f ZA2013-0565.fasta -db COX1_SPECIES
```

The output should look like this:

bold_id	seq_id	similarity	collection_country	division	taxon
FIDIP558-11	TE-14-27_FHYP_av	0.9884	Finland	animal	Diptera
GBDP6413-09	TE-14-27_FHYP_av	0.9242	None	animal	Ornithomya anchineura
GBDP2916-07	TE-14-27_FHYP_av	0.922	None	animal	Stenepteryx hirundinis
GBDP2919-07	TE-14-27_FHYP_av	0.9149	None	animal	Ornithomya biloba
GBDP2908-07	TE-14-27_FHYP_av	0.9078	None	animal	Ornithoctona sp. P-20
GBDP2918-07	TE-14-27_FHYP_av	0.9076	None	animal	Ornithomya chloropus
GBDP2935-07	TE-14-27_FHYP_av	0.8936	None	animal	Crataerina pallida
GBMIN26225-13	TE-14-27_FHYP_av	0.8889	None	animal	Lucilia sericata
GBDP5820-09	TE-14-27_FHYP_av	0.8833	None	animal	Coenosia tigrina
GBMIN26204-13	TE-14-27_FHYP_av	0.883	None	animal	Lucilia cuprina
GBMIN18768-13	TE-14-27_FHYP_av	0.8823	Brazil	animal	Ornithoctona erythrocephala

### 3.2 As an alternative you can use `bold_retriever` as a Python module

To use Bold Retriever in a project:

```
>>> from Bio import SeqIO
>>> from bold_retriever import bold_retriever as br

>>> # database from BOLD
```

```
>>> db = "COX1_SPECIES"

>>> all_ids = []
>>> for seq_record in SeqIO.parse("tests/ionx13.fas", "fasta"):
...     my_ids = br.request_id(seq_record.seq, seq_record.id, db)
Psocoptera 0.9796
Selenops mexicanus 0.8933
Austrophorocera Janzen03 0.8736
Austrophorocera Janzen04 0.8667
Lepidoptera 0.8667
Proechimys simonsi 0.8667
Diptera 0.8667
Scathophaga stercoraria 0.8667
Culex quinquefasciatus 0.8667
Folsomia fimetaria L1 0.8652
Lepidopsocidae sp. RS-2001 0.8639
lepidopsocid RS-2001 0.8639
Selenops micropalpus 0.859
Geocoris pallidipennis 0.8586
Selenops sp. 2 SCC-2009 0.8571
Mermessus trilobatus 0.8571
Drosophila neotestacea 0.8571
Hemiptera 0.8556
Miromantis mirandula 0.8537
Houghia gracilis 0.8533
Adoxophyes nr. marmarogodes 0.8533
Trichoptera 0.8533
Araneae 0.8533
Hydroporus morio 0.8533
Rodentia 0.8533
```

In that case the output will be contained in the variable `my_ids` and will look like this:

```
[{'bold_id': 'FIPSO166-14',
'collection_country': 'Finland',
'id': 'ionx13',
'seq': 'AATTTGAGCTGGTATACTTGGGACTAGTTTAAGAATCTTAATTCGACTTGAGTTAGGCCAACCAAGGTTTATTtttAGAAGATGACCAAACAT',
'similarity': '0.9796',
'tax_id': 'Psocoptera'},
{'bold_id': 'GBCH4611-10',
'collection_country': 'None',
'id': 'ionx13',
'seq': 'AATTTGAGCTGGTATACTTGGGACTAGTTTAAGAATCTTAATTCGACTTGAGTTAGGCCAACCAAGGTTTATTtttAGAAGATGACCAAACAT',
'similarity': '0.8933',
'tax_id': 'Selenops mexicanus'},
{'bold_id': 'ASTAQ477-06',
'collection_country': 'Costa Rica',
'id': 'ionx13',
'seq': 'AATTTGAGCTGGTATACTTGGGACTAGTTTAAGAATCTTAATTCGACTTGAGTTAGGCCAACCAAGGTTTATTtttAGAAGATGACCAAACAT',
'similarity': '0.8736',
'tax_id': 'Austrophorocera Janzen03'},
{'bold_id': 'ASTAR353-07',
'collection_country': 'Costa Rica',
'id': 'ionx13',
'seq': 'AATTTGAGCTGGTATACTTGGGACTAGTTTAAGAATCTTAATTCGACTTGAGTTAGGCCAACCAAGGTTTATTtttAGAAGATGACCAAACAT',
'similarity': '0.8667',
'tax_id': 'Austrophorocera Janzen04'}]
```

---

## Contributing

---

Contributions are welcome, and they are greatly appreciated! Every little bit helps, and credit will always be given. You can contribute in many ways:

### 4.1 Types of Contributions

#### 4.1.1 Report Bugs

Report bugs at [https://github.com/carlosp420/bold\\_retriever/issues](https://github.com/carlosp420/bold_retriever/issues).

If you are 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.

#### 4.1.2 Fix Bugs

Look through the GitHub issues for bugs. Anything tagged with “bug” is open to whoever wants to implement it.

#### 4.1.3 Implement Features

Look through the GitHub issues for features. Anything tagged with “feature” is open to whoever wants to implement it.

#### 4.1.4 Write Documentation

Bold Retriever could always use more documentation, whether as part of the official Bold Retriever docs, in docstrings, or even on the web in blog posts, articles, and such.

#### 4.1.5 Submit Feedback

The best way to send feedback is to file an issue at [https://github.com/carlosp420/bold\\_retriever/issues](https://github.com/carlosp420/bold_retriever/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 contributions are welcome :)

## 4.2 Get Started!

Ready to contribute? Here's how to set up *bold\_retriever* for local development.

1. Fork the *bold\_retriever* repo on GitHub.
2. Clone your fork locally:

```
$ git clone git@github.com:your_name_here/bold_retriever.git
```

3. Install your local copy into a virtualenv. Assuming you have virtualenvwrapper installed, this is how you set up your fork for local development:

```
$ mkvirtualenv bold_retriever
$ cd bold_retriever/
$ python setup.py develop
```

4. Create a branch for local development:

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

Now you can make your changes locally.

5. When you're done making changes, check that your changes pass flake8 and the tests, including testing other Python versions with tox:

```
$ flake8 bold_retriever tests
$ python setup.py test
$ tox
```

To get flake8 and tox, just pip install them into your virtualenv.

6. 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
```

7. Submit a pull request through the GitHub website.

## 4.3 Pull Request Guidelines

Before you submit a pull request, check that it meets these guidelines:

1. The pull request should include tests.
2. If the pull request adds functionality, the docs should be updated. Put your new functionality into a function with a docstring, and add the feature to the list in README.rst.
3. The pull request should work for Python 2.6, 2.7, 3.3, and 3.4, and for PyPy. Check [https://travis-ci.org/carlosp420/bold\\_retriever/pull\\_requests](https://travis-ci.org/carlosp420/bold_retriever/pull_requests) and make sure that the tests pass for all supported Python versions.

## 4.4 Tips

To run a subset of tests:

```
$ python -m unittest tests.test_bold_retriever
```





---

**Credits**

---

## 5.1 Development Lead

- Carlos Pena <[mycalesis@gmail.com](mailto:mycalesis@gmail.com)>

## 5.2 Contributors

None yet. Why not be the first?



---

## History

---

- v1.0.0: Using Twisted for asynchronous calls and increase in speed.
- **v0.2.4: Reorganizing columns in output file. Querying the API for family** name of taxa.
- v0.2.2: Killed bug taxon search.
- v0.2.1: Killed bug in scraping web `Public_BIN` for species ID.
- v0.2.0: Scraping web `Public_BIN` for species ID.
- v0.1.9: Added `request_id` test and option to run fuction in debug mode.
- v0.1.8: Fixed bug for exception when BOLD sends empty list of taxon names.
- v0.1.7: Fixed bug for exception when BOLD sends empty list of taxon names.
- v0.1.6: Append taxon identification results to file as we get them.
- v0.1.5: Additionat tests coverage 92%
- v0.1.4: Fixed bug in `taxon_search` function
- v0.1.3: Coverage 75%
- v0.1.2: Pep8 and test coverage 69%
- v0.1.1: Packaged as Python module.
- v0.1.0: You can specify which BOLD datase should be used for BLAST of FASTA sequences.
- v0.0.7: Catching exception for NULL, list and text returned instead of XML from BOLD.
- v0.0.6: Catching exception for malformed XML from BOLD.
- v0.0.5: Catch exception when BOLD sends funny data such as `{"481541": []}`.



---

## Indices and tables

---

- *genindex*
- *modindex*
- *search*