
candis Documentation

Release 0.1.0

Achilles Rasquinha

Jun 14, 2017

Contents

1	Guide - User	3
1.1	Introduction	3
1.2	Installation	3
1.3	Quickstart	4
2	Guide - API	5
2.1	Developer Interface	5
2.2	Command-Line Interface	5
2.3	Rich Internet Application	6

“A data mining suite for DNA microarrays.”

Release: v0.1.0 (*Installation*)

candis is an Open Source data mining suite (released under the GNU General Public License v3) for DNA microarrays that consists of a wide collection of tools you require, right from Data Extraction to Model Deployment.

candis helps you with

Data Extraction

Extracting data from National Center for Biotechnology Information’s Entrez engine using the `candis.entrez` API.

```
>>> from candis import entrez
>>> api = entrez.API('<your_email_ID>')
>>> api.info() # List of available databases from NCBI.
['pubmed', 'protein', 'nuccore', ...]
```

candis is created and currently maintained by [Achilles Rasquinha](#).

candis officially supports Python 2.7+ and 3.5+.

Introduction

What's in the name?

Installation

Building from source

candis is actively developed on [GitHub](#) and is always available.

You can clone the base repository with `git` as follows:

```
$ git clone git@github.com:achillesrasquinha/candis.git
```

Optionally, you could download the [tarball](#) or [zipball](#) as follows:

For Linux Users

```
$ curl -OL https://github.com/achillesrasquinha/tarball/candis
```

For Windows Users

```
$ curl -OL https://github.com/achillesrasquinha/zipball/candis
```

Install necessary dependencies

```
$ pip install -r requirements.txt
```

Then, go ahead and install **candis** in your site-packages as follows:

```
$ python setup.py install
```

Check to see if you've installed **candis** correctly.

```
>>> import candis
```

Quickstart

Developer Interface

Configuration

class `candis.Config` (*schema=None*)

An n-ary tree-like configuration object. Each leaf node of the tree holds a configuration value. A leaf node is denoted by an uppercase attribute whereas each internal node is denoted by a capitalized attribute.

Parameters `schema` (dict (default - { })) – a dict-like object for configuration data.

Example

```
>>> import candis
>>> config = candis.Config({ 'name': 'candis', 'version': '0.1.0' })
>>> config.NAME
'candis'
```

Command-Line Interface

To launch **candis**'s Rich Internet Application (RIA), simply type

```
$ candis
```

At any given time, you can view the help as follows

```
$ candis --help
```

via Python

All of **candis**'s functionality can be executed through the `candis.main()` function provided by the `candis.cli` module.

`candis.main(argv=None)`

A starting execution point for `candis`

Parameters `argv` (`list(str)` or `None` (default)) – (optional) a list of valid arguments

Returns A valid exit code. Check out [Python Documentation](#) for a list of various exit codes.

Return type `int`

Example

```
>>> import candis
>>> candis.main() # Launch the Rich Internet Application (RIA)
```

Rich Internet Application

C

`Config` (class in `candis`), [5](#)

M

`main()` (in module `candis`), [6](#)