CAI Documentation

Benjamin Lee

Oct 02, 2018

Contents

An implementation of Sharp and Li's 1987 formulation of the codon adaption index.

CHAPTER 1

Installation

This module is available from PyPi and can be downloaded with the following command:

```
$ pip install CAI
```

To install the latest development version:

```
$ pip install git+https://github.com/Benjamin-Lee/CodonAdaptationIndex.git
```

CHAPTER 2

Quickstart

Finding the CAI of a sequence is easy:

```
>>> from CAI import CAI
>>> CAI("ATG...", reference=["ATGTTT...", "ATGCGC...",...])
0.24948128951724224
```

Similarly, from the command line:

```
$ CAI -s sequence.fasta -r reference_sequences.fasta
0.24948128951724224
```

Determining which sequences to use as the reference set is left to the user, though the HEG-DB is a great resource of highly expressed genes.

CHAPTER 3

Contributing and Getting Support

If you encounter any issues using CAI, feel free to create an issue.

To contribute to the project, please create a pull request. For more information on how to do so, please look at GitHub's documentation on pull requests.

CHAPTER 4

Citation

Benjamin Lee. (2017). Python Implementation of Codon Adaptation Index. *Zenodo*. http://doi.org/10.5281/zenodo.843854

JOSS citation coming soon.

CHAPTER 5

Contact

I'm available for contact at benjamin_lee@college.harvard.edu.

CHAPTER 6

Reference

Sharp, P. M., & Li, W. H. (1987). The codon adaptation index–a measure of directional synonymous codon usage bias, and its potential applications. *Nucleic Acids Research*, 15(3), 1281–1295.

CHAPTER 7

Table of Contents

7.1 Usage

7.1.1 Basic Usage

As covered in *Quickstart*, the basic `CAI()` function is fast and easy. Simply import it and get to your science. Note that it also plays nicely with Biopython Seq objects:

```
>>> from CAI import CAI
>>> from Bio.Seq import Seq
>>> CAI(Seq("AAT"), reference=[Seq("AAC")])
0.5
```

The CLI is equally easy to use. For example, to find the CAI of the native GFP gene with respect to the highly expressed genes in *E. coli*, only one command is required:

```
$ CAI -r example_seqs/ecol.heg.fasta -s example_seqs/gfp.fasta
0.3753543123685772
```

Note: Both `CAI` and `cai` are valid commands.

More example sequences can be found in the `example_seqs` directory on GitHub.

7.1.2 Advanced Usage

If you have already computed the weights or RSCU values of the reference set, you can supply `CAI()` with one or the other as arguments. They must be formatted as a dictionary and contain values for every codon.

To calculate RSCU without calculating CAI, you can use `RSCU()`. `RSCU()`'s only required argument is a list of sequences.

Similarly, to calculate the weights of reference sequences, you can use *relative_adaptiveness()*. *relative_adaptiveness()* takes either a list of sequences as the `sequences` parameter or a dictionary of RSCUs as the `RSCUs` parameter.

> **Warning:** If you are computing large numbers of CAIs with the same reference sequences, first calculate their weights with *relative_adaptiveness()* and then pass that to *CAI()* to eliminate redundant computation.

So, to modify the example in *Quickstart*:

```
>>> from CAI import CAI, relative_adaptiveness
>>> sequences=["ATGTTT...", "ATGCGC...",...]
>>> weights = relative_adaptiveness(sequences=sequences)
>>> CAI("ATG...", weights=weights)
0.24948128951724224
```

These are exactly equivalent:

```
>>> assert CAI("ATG...", weights=weights) == CAI("ATG...", reference=sequences)
True
```

except the former will be faster if you're using the same weights repeatedly.

7.1.3 Other Genetic Codes

All functions in CAI support an optional `genetic_code` parameter, which is set by default to 11 (the standard genetic code).

In the CLI, there is an optional "-g" parameter that changes the genetic code:

```
$ CAI -s sequence.fasta -r reference_sequences.fasta -g 22
0.25135779681923687
```

7.2 API Reference

RSCU (*sequences*, *genetic_code=11*)

Calculates the relative synonymous codon usage (RSCU) for a set of sequences.

RSCU is 'the observed frequency of [a] codon divided by the frequency expected under the assumption of equal usage of the synonymous codons for an amino acid' (page 1283).

In math terms, it is

$$\frac{X_{ij}}{\frac{1}{n_i}\sum_{j=1}^{n_i} x_{ij}}$$

"where X is the number of occurrences of the j th codon for the i th amino acid, and n is the number (from one to six) of alternative codons for the i th amino acid" (page 1283).

Parameters

- **sequences** (*list*) – The reference set of sequences.
- **genetic_code** (*int, optional*) – The translation table to use. Defaults to 11, the standard genetic code.

Returns The relative synonymous codon usage.

Return type dict

Raises `ValueError` – When an invalid sequence is provided or a list is not provided.

relative_adaptiveness(*sequences=None*, *RSCUs=None*, *genetic_code=11*)

Calculates the relative adaptiveness/weight of codons.

The relative adaptiveness is "the frequency of use of that codon compared to the frequency of the optimal codon for that amino acid" (page 1283).

In math terms, w_{ij}, the weight for the j th codon for the i th amino acid is

$$w_{ij} = \frac{\text{RSCU}_{ij}}{\text{RSCU}_{imax}}$$

where "RSCU_{imax} [is] the RSCU… for the frequently used codon for the i th amino acid" (page 1283).

Parameters

- **sequences** (*list, optional*) – The reference set of sequences.
- **RSCUs** (*dict, optional*) – The RSCU of the reference set.
- **genentic_code** (*int, optional*) – The translation table to use. Defaults to 11, the standard genetic code.

Note: Either `sequences` or `RSCUs` is required.

Returns A mapping between each codon and its weight/relative adaptiveness.

Return type dict

Raises

- `ValueError` – When neither `sequences` nor `RSCUs` is provided.
- `ValueError` – See *RSCU()* for details.

CAI(*sequence*, *weights=None*, *RSCUs=None*, *reference=None*, *genetic_code=11*)

Calculates the codon adaptation index (CAI) of a DNA sequence.

CAI is "the geometric mean of the RSCU values… corresponding to each of the codons used in that gene, divided by the maximum possible CAI for a gene of the same amino acid composition" (page 1285).

In math terms, it is

$$\left(\prod_{k=1}^{L} w_k\right)^{\frac{1}{L}}$$

where w_k is the relative adaptiveness of the k th codon in the gene (page 1286).

Parameters

- **sequence** (*str*) – The DNA sequence to calculate the CAI for.
- **weights** (*dict, optional*) – The relative adaptiveness of the codons in the reference set.
- **RSCUs** (*dict, optional*) – The RSCU of the reference set.
- **reference** (*list*) – The reference set of sequences.

Note: One of `weights`, `reference` or `RSCUs` is required.

Returns The CAI of the sequence.

Return type float

Raises

- `TypeError` – When anything other than one of either reference sequences, or RSCU dictionary, or weights is provided.
- `ValueError` – See *`RSCU()`* for details.
- `KeyError` – When there is a missing weight for a codon.

Warning: Will return nan if the sequence only has codons without synonyms.

7.3 CLI Reference

```
$ CAI --help
Usage: CAI [OPTIONS]

Options:
  -s, --sequence FILE         The sequence to calculate the CAI for.
                              [required]
  -r, --reference FILE        The reference sequences to calculate CAI
                              against.  [required]
  -g, --genetic-code INTEGER  The genetic code to use. Defaults to 11.
  --help                      Show this message and exit.
```

7.4 License

This software is licensed under the MIT License. If you're unfamiliar with software licenses, here is a handy summary of the license.

For reference, the license is reproduced below:

```
MIT License

Copyright (c) 2017 Benjamin Lee

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.
```

(continues on next page)

(continued from previous page)

```
THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.
```

CHAPTER 8

Indices and tables

- genindex
- modindex
- search

Python Module Index

C

Index