BCAW tool

Release 1.0.0

May 28, 2020

Contents

CHAPTER 1

Guide

1.1 Introduction to Codon Usage Bias

During the translation process from mRNA to protein, information is transmitted in the form of nucleotide triplets named codons. Amino acids are degenerate, having more than one codon representing each except for methionine (Met) and tryptophan (Trp). Thus, codons encoding the same amino acid are known as synonymous codons. Many studies on different organisms show that synonymous codons are not used uniformly within and between genes of one genome. This phenomenon called synonymous codon usage bias (SCUB) or codon usage bias (CUB) [1–4]. Further, the degree of the unequal use of synonymous codons differs between species [5, 6]. Hence, each organism has its optimal codons, where an optimal codon is defined as a codon which is more frequently used in highly expressed genes than in the lowly expressed genes [7]. Two main factors shaping the codon usage of an organism are mutation and selection [8–10]. Other factors also known to influence the CUB of an organism are nucleotide composition [11], synonymous substitution rate [12], tRNA abundance [13], codon hydropathy and DNA replication initiation sites [14], gene length [15] and expression level [16]. Investigating the CUB of an organism can tell about genes' molecular evolution, expression, and host-pathogen coadaptation. Furthermore, in biotechnology applications, CUB and predicted optimal codons could help to design highly expressed genes and construct suitable cloning vectors [17, 18].

1.1.1 Equations used for codon usage bias analysis

Effective number of codons

The effective number of codons (ENc) measures the bias of using a smaller subset of codons apart from the equal use of synonymous codons. In addition, the ENc measures the codon usage unbalance among genes, where an amino acid is encoded by one codon in a gene would be biased and negatively correlated with the ENc value. This measure ranges from 20-61 with higher values indicating more codons being used for each amino acid, i.e less bias and vice versa. This measures codon bias irrespective of gene length, and can be an indicator of codon usage with reference to mutational bias [19]. ENc was calculated using the equation given by [20]:

$$F_{\text{CF}} = \sum_{i=1}^{m} \left(\frac{n_i + 1}{n + m} \right)^2$$

Then ENc could be calculated by:

$$N_{\text{c.CF}} = \frac{1}{F_{\text{CF}}}$$

Where n_i is the count of codon i in m amino acid family and m is the number of codons in an amino acid family

Codon adaptation index

Codon adaptation index (CAI) uses a reference set of highly expressed genes (e.g. ribosomal genes), this measure is an indicator of gene expression levels and natural selection; it ranges from 0 to 1 with higher values indicating stronger bias with respect to the reference set, therefore this method is an indicator of selection for a bias toward translational efficiency [21, 22]. Codon adaptation index (CAI) was calculated by the equation given by [22, 23]:

$$CAI = exp\frac{1}{L}\sum_{k=1}^{L}\ln w_{c(k)}$$

Where, L is the count of codons in the gene and *wc*(k) is the relative adaptiveness value for the *k*-th codon in the gene.

Relative synonymous codon usage

Relative synonymous codon usage (RSCU) calculate the observed count to the expected count of each codon to be analyzed, as RSCU less than one means codon observed less frequently than the average codon usage, while RSCU more than one means codon is observed more frequently than the average codon usage[21]. The equation to calculate the RSCU is [21]:

$$RSCU = \frac{O_{\text{ac}}}{\frac{1}{k_a}\sum_{c\in C_a} O_{\text{ac}}}$$

Where Oac is the count of codon c for amino acid and ka is the number of synonymous codons.

Translational selection index

Translational selection (P2) index measure the bias of anticodon-codon interactions, from which we can indicate the translation efficiency [24]. Generally P2-index ranges from 0 to 1. Its values have been noted to be high for highly expressed genes and low for lowly expressed genes [7,25]. i [24]By taking the averages of numbers resulted from this equation for each CDS:

$$P2 = \frac{WWC + SSU}{WWY + SSY}$$

Where, W = A or U(T), S = G or C, and Y = C or U(T).

Hydropathicity (Gravy) and Aromaticity (Aroma) indices

In analyzing the natural selection for shaping the codon usage bias, two indices, including Gravy and Aroma scores, were used in many studies [26–28]. Thus, the variation of the two indices reflects the amino acid usage. A higher Gravy or Aroma value suggests a more hydrophobic or aromatic amino acid product.

Parity Rule 2 -plot Analysis

All the nucleotides content at the third codon position (A3, T3, G3, and C3) were calculated, then for each gene AT-bias (A3/(A3 + T3)) and GC-bias (G3/(G3 + C3)) were estimated and used as the ordinate and the abscissa respectively in the plot, with both coordinates equal to 0.5, where A = T and G = C [29]. The genes positions on the plot along the ordinate and the abscissa tell about factors influence the CUB. If genes over the plot view are scattered equally, then the CUB is likely to be solely caused by the mutation [30].

Establish reference genes set

Genes with high expression level within the organims genomes are ussually named reference gene sets. Reference gene sets are used to calculate the CAI. In BCAWT two options may be used:

1. Reference genes set given by the users.
2. An auto option where BCAWT generates a genes reference set using 10% of genes have the lowest ENc values (highest biased genes).

Determination of putative optimal codons

BCAWT uses the correlation method described in [20] to determine the putative optimal codons. Where each synonymous codon RSCU in one amino acid family correlated with all genes ENc, an optimal codon for each amino acid family was defined as the codon which has the strongest negative correlation RSCU with ENc values, and with a significant p-value less than 0.05/n where n is equal to the number of synonymous codons in such amino acid family.

Correspondence analysis

Excluding Met and Trp codons, it is an advantage to perform multivariate statistical analysis on the rest of 59 codons to examine the variations in the codon usage bias among all the CDSs. One way to do that is correspondence analysis (COA)[31,32] by plotting groups of genes on continuous axes in multidimensional space according to the trends affecting the synonymous codon usage within the genes group.

1.1.2 References

1. Gu W, Zhou T, Ma J, Sun X, Lu Z. Analysis of synonymous codon usage in SARS Coronavirus and other viruses in the Nidovirales. Virus Res. 2004;101: 155–161. doi:10.1016/j.virusres.2004.01.006
2. Vicario S, Moriyama EN, Powell JR. Codon usage in twelve species of Drosophila. BMC Evol Biol. 2007;7: 1–17. doi:10.1186/1471-2148-7-226
3. Behura SK, Severson DW. Comparative analysis of Codon usage bias and Codon context patterns between dipteran and hymenopteran sequenced genomes. PLoS One. 2012;7. doi:10.1371/journal.pone.0043111
4. Boël G, Letso R, Neely H, Price WN, Su M, Luff J, et al. Codon influence on protein expression in E.coli. 2016;529: 358–363. doi:10.1038/nature16509.Codon
5. Dohra H, Fujishima M, Suzuki H. Analysis of amino acid and codon usage in Paramecium bursaria. FEBS Lett. Federation of European Biochemical Societies; 2015;589: 3113–3118. doi:10.1016/j.febslet.2015.08.033
6. Qiu S, Zeng K, Slotte T, Wright S, Charlesworth D. Reduced efficacy of natural selection on codon usage bias in selfing Arabidopsis and Capsella species. Genome Biol Evol. 2011;3: 868–880. doi:10.1093/gbe/evr085
7. Wang L, Xing H, Yuan Y, Wang X, Saeed M, Tao J, et al. Genome-wide analysis of codon usage bias in four sequenced cotton species. PLoS One. 2018; 1–17. doi:10.1371/journal.pone.0194372

8. Chen H, Sun S, Norenburg JL, Sundberg P. Mutation and selection cause codon usage and bias in mitochondrial genomes of ribbon worms (Nemertea). PLoS One. 2014;9. doi:10.1371/journal.pone.0085631

9. Zalucki YM, Power PM, Jennings MP. Selection for efficient translation initiation biases codon usage at second amino acid position in secretory proteins. Nucleic Acids Res. 2007;35: 5748–5754. doi:10.1093/nar/gkm577

10. Prabha R, Singh DP, Sinha S, Ahmad K, Rai A. Genome-wide comparative analysis of codon usage bias and codon context patterns among cyanobacterial genomes. Mar Genomics. Elsevier B.V.; 2017;32: 31–39. doi:10.1016/j.margen.2016.10.001

11. Palidwor GA, Perkins TJ, Xia X. A general model of Codon bias due to GC mutational bias. PLoS One. 2010;5. doi:10.1371/journal.pone.0013431

12. Marais G, Mouchiroud D, Duret L. Neutral effect of recombination on base composition in Drosophila. Genet Res. 2003;81: 79–87. doi:10.1017/S0016672302006079

13. Rocha EPC. Codon usage bias from tRNA's point of view: Redundancy, specialization, and efficient decoding for translation optimization. Genome Res. 2004; 2279–2286. doi:10.1101/gr.2896904

14. Huang Y, Koonin E V., Lipman DJ, Przytycka TM. Selection for minimization of translational frameshifting errors as a factor in the evolution of codon usage. Nucleic Acids Res. 2009;37: 6799–6810. doi:10.1093/nar/gkp712

15. Duret L, Mouchiroud D. Expression pattern and, surprisingly, gene length shape codon usage in Caenorhabditis, Drosophila, and Arabidopsis. Proc Natl Acad Sci U S A. 1999;96: 4482–7. Available: https://www.ncbi.nlm.nih.gov/pubmed/10200288. doi: 10.1073/pnas.96.8.4482

16. Hiraoka Y, Kawamata K, Haraguchi T, Chikashige Y. Codon usage bias is correlated with gene expression levels in the fission yeast Schizosaccharomyces pombe. Genes to Cells. 2009;14: 499–509. doi:10.1111/j.1365-2443.2009.01284.x

17. Pandit A, Sinha S. Differential trends in the codon usage patterns in HIV-1 genes. PLoS One. 2011;6: 1–10. doi:10.1371/journal.pone.0028889

18. Liu H, He R, Zhang H, Huang Y, Tian M, Zhang J. Analysis of synonymous codon usage in Zea mays. Mol Biol Rep. 2010;37: 677–684. doi:10.1007/s11033-009-9521-7

19. Wright F. The "effective number of codons" used in a gene. Gene. 1990;87: 23–29. doi:10.1016/0378-1119(90)90491-9

20. Sun X, Yang Q, Xia X. An improved implementation of effective number of codons (Nc). Mol Biol Evol. 2013;30: 191–196. doi:10.1093/molbev/mss201

21. Sharp PM, Li W. Codon Adaptation Index and its potential applications Nucleic Acids Research. 1987;15: 1281–1295.

22. Ran W, Higgs PG. Contributions of Speed and Accuracy to Translational Selection in Bacteria. PLoS One. 2012;7. doi:10.1371/journal.pone.0051652. doi:10.1093/nar/15.3.1281

23. Lee, B. D. (2018). Python Implementation of Codon Adaptation Index. Journal of Open Source Software, 3 (30), 905. doi:10.21105/joss.00905

24. Chakraborty S, Nag D, Mazumder TH, Uddin A. Codon usage pattern and prediction of gene expression level in Bungarus species. Gene. Elsevier B.V.; 2016; doi:10.1016/j.gene.2016.11.023

25. Gatherer D, Mcewan N. Small regions of preferential codon usage and their effect on overall codon bias - the case of the plp gene. biochem mol biol int. 1997;43: 107–114. doi:10.1080/15216549700203871

26. Choudhury MN, Uddin A, Chakraborty S. Nucleotide composition and codon usage bias of SRY gene. Andrologia. 2018;50: 1–11. doi:10.1111/and.12787

27. Rao Y, Wang Z, Chai X, Nie Q, Zhang X. Hydrophobicity and aromaticity are primary factors shaping variation in amino acid usage of chicken proteome. PLoS One. 2014;9. doi:10.1371/journal.pone.0110381

28. Chen Y, Li X, Chi X, Wang S, Ma Y, Chen J. Comprehensive analysis of the codon usage patterns in the envelope glycoprotein E2 gene of the classical swine fever virus. PLoS One. 2017; 1–14. doi:10.1371/journal.pone.0183646

29. Sueoka N. Intrastrand parity rules of DNA base composition and usage biases of synonymous codons. J Mol Evol. 1995;40: 318–325. doi:10.1007/BF00163236

30. Sueoka N. Near Homogeneity of PR2-Bias Fingerprints in the Human Genome and Their Implications in Phylogenetic Analyses. Mol Evol. 2001; 469–476. doi:10.1007/s002390010237

31. Emmanuelle Lerat, Christian Biémont, Pierre Capy, Codon Usage and the Origin of P Elements, Molecular Biology and Evolution, Volume 17, Issue 3, March 2000, Pages 467–468. doi:10.1093/oxfordjournals.molbev.a026326

32. Denis C.ShieldsPaul M.Sharp. Drosophila I. Evidence that Mutation Patterns Vary Among Drosophila Transposable Elements. J Mol Biol. 1989; 843–846. doi:10.1016/0022-2836(89)90252-0

1.2 Statement of Need

There are no tools available enable users to run a whole automated workflow for codon usage bias analysis. Using python 3.7 BCAW Tool (Bio Codon Analysis Workflow Tool) was developed to address this problem. BCAW Tool manages a complete automated workflow to analyze the codon usage bias for genes and genomes of any organism. With minimum coding skills.

For more details about CUB, and the equations used in BCAWT https://bcaw-tools-documentation.readthedocs.io/en/latest/intro.html

1.3 Citation

Anwar, (2019). BCAWT: Automated tool for codon usage bias analysis for molecular evolution. Journal of Open Source Software, 4(42), 1500, https://doi.org/10.21105/joss.01500

1.4 Usage

1.4.1 Automated testing

First download the **'example FASTA file <https://raw.githubusercontent.com/AliYoussef96/BCAW-Tool/master/tests/Ecoli.fasta >'_** containing a coding sequence then run:

```
from BCAWT import BCAWT_auto_test

BCAWT_auto_test.auto_test(["Ecoli.fasta"])

BCAWT_auto_test.auto_check_files()

>> test is completed 'successfully'
```

to automatically run a test on the resulting files.

1.4.2 Main Usage

from BCAWT import BCAWT

BCAWT.BCAW(['Ecoli.fasta'],'result_folder',genetic_code_=11,Auto=True)

Important Note: BCAW Tool expects coding sequences

1.4.3 Input

- `main_fasta_file` (list): list of string of the file's path or file-like object
- `save_folder_name` (str): folder name where the result will be saved
- `ref_fasta_file` (list): list of string of the file's path or file-like object, default = None
- `Auto` (bool): default = False, if `ref_fasta_file` is not None.
- `genetic_code` (int): default = 1, the genetic code ID from the NCBI table

1.4.4 To obtain FASTA file for a species of interest

Say that the species of interest is *Escherichia coli* str. K-12 substr. MG1655:

1. Go to the NCBI's database.
2. In the search bar write `Escherichia coli str. K-12 substr. MG1655, complete genome`.
3. Choose one of the results (depending on what you want in your analysis).
3. On the write of the page, you will find **send to** option. From **sent to** select **Coding Sequences** then **FASTA nucleotides** Finally, press on **Create File**

For NCBI Genomes Download questions, see their FAQ.

1.5 Outputs

1.5.1 Expected CSV output

CSV file name	Description
ATCG	Contains gene id, GC, GC1, GC2, GC3, GC12, AT, AT3 A3, T3, C3, G3, GRAVY, AROMO and gene length
CA_RSCU	Each RSCU result for each codon in each genes
CA_RSCUcodons	Correspondence analysis for the first 4 axes of each codon
CA_RSCUgenes	Correspondence analysis for the first 4 axis of gene
CAI	Gene id and CAI for each gene
ENc	Gene id and ENc for each gene
P2-index	Gene id and P2 index for each gene
optimal codons	Putative optimal codons

1.5.2 All output plots

1.6 Dependencies

- Biopython
- Pandas
- CAI
- SciPy
- Matplotlib
- NumPy

1.7 Installation

Using pip:

```
pip install BCAWT
```

1.8 More information

1. For more information about the codon usage bias (CUB) equations used to analyze CUB in the BCAW tool and the API BCAW tool's documentation.
2. For more information about the output plots.
3. For more information about the abbreviations used.

1.9 More About Plots

All output plots are described here

1.10 Contribution Guidelines

Contributions to the software are welcome

For bugs and suggestions, the most effective way is by raising an issue on the github issue tracker. Github allows you to classify your issues so that we know if it is a bug report, feature request or feedback to the authors.

If you wish to contribute some changes to the code then you should submit a [pull request](https://github.com/AliYoussef96/BCAW-Tool/pulls) How to create a Pull Request? [documentation on pull requests](https://help.github.com/en/articles/about-pull-requests)

1.11 License

1.12 Help

If you need more help please contact ali.mo.anwar@std.agr.cu.edu.eg.

1.13 Citation

Anwar, (2019). BCAWT: Automated tool for codon usage bias analysis for molecular evolution. Journal of Open Source Software, 4(42), 1500, https://doi.org/10.21105/joss.01500

CHAPTER 2

API

2.1 BCAWT package

2.1.1 BCAWT.BCAWT module

BCAWT.**BCAW**(*main_fasta_file*, *save_path=''*, *ref_fasta_file=None*, *genetic_code_=1*, *Auto=False*)

BCAWT (Bio Codon Analysis Workflow Tool), it manages a complete workflow to analysis the codon usage bias for genes and genomes of any organism..

Args:

main_fasta_file (list): list of string of the file's path or file-like object

save_path (str): absolute path to the directory to save the result in, default = the current directory

ref_fasta_file (list): list of string of the file's path or file-like object, default = None

genetic_code_(int): default = 1, The Genetic Codes number described by NCBI (https://www.ncbi.nlm.nih.gov/Taxonomy/Utils/wprintgc.cgi)

Auto (bool): default = False, if ref_fasta_file not None.

Notes:

- Auto (bool): should be = True to auto-generate a reference set, when arg (ref_fasta_file) not available (= None)

Returns:

for details see: https://bcaw-tools-documentation.readthedocs.io/en/latest/Table_output.html

2.1.2 BCAWT.BCAWT_auto_test module

BCAWT_auto_test.**auto_check_files**(*path*)

Check the expected outputs.

Args: path: absolute path to the directory to save the result in

Returns:

None

BCAWT_auto_test.**auto_test**(*path=''*, *test_file=''*)
Run a demo test with 23 results (see the documentation for more details about the expected output)

Args:

path: absolute path to the directory to save the result in test_file: absolute path to the fasta file that will be tested

Returns: None

2.1.3 BCAWT.ATCG3 module

ATCG3.**ACTG3**(*sequ*, *A=False*, *T=False*, *C=False*, *G=False*)
Calculate A, T, G, and C content at the third position.

Args:

sequ (str): DNA sequence A (bool): default = False T (bool): default = False C (bool): default = False G (bool): default = False

Returns:

- A3 content if arg(A) is True
- T3 content if arg(T) is True
- C3 content if arg(C) is True
- G3 content if arg(G) is True
- None if all args are False

2.1.4 BCAWT.CA module

CA.**CA**(*file*)
correspondence analysis.

Args:

file (directory): csv file contains genes' RSCU values

Returns:

- csv file contains genes' values for the first 4 axes of the correspondence analysis result
- csv file contains codons' values for the first 4 axes of the correspondence analysis result
- plot the genes first 2 axes values of the correspondence analysis result
- plot the codons first 2 axes values of the correspondence analysis result

2.1.5 BCAWT.CA_RSCU module

CA_RSCU.**CA_RSCU**(*allseq*, *allseq_name*, *The_Genetic_Codes_number=1*)
calculate RSCU values for correspondence analysis.

Args:

allseq (str): DNA sequence allseq_name (str) : gene name The_Genetic_Codes_number (int) : default = 1, The Genetic Codes number described by NCBI (https://www.ncbi.nlm.nih.gov/Taxonomy/Utils/wprintgc.cgi)

Returns: DataFrame: DataFrame contains [gene_name and RSCU values]

2.1.6 BCAWT.GC123 module

GC123.**GC1**(*sequ*)
Calculates the GC content an the first codon position.

Args:

sequ (str): DNA sequence

Returns: int: GC1 value

GC123.**GC12**(*sequ*)
Calculates the GC content average at the 1st and 2nd codon positions.

Args:

sequ (str): DNA sequence

Returns: int: GC12 value

GC123.**GC2**(*sequ*)
Calculates the GC content an the 2nd codon position..

Args:

sequ (str): DNA sequence

Returns: int: GC2 value

GC123.**GC3**(*sequ*)
Calculates the GC content an the 3rd codon position..

Args:

sequ (str): DNA sequence

Returns: int: GC3 value

2.1.7 BCAWT.GRAVY_AROMO module

GRAVY_AROMO.**GRAvy_ARomo**(*seq*, *genetic_code_=1*, *G=False*, *A=False*)
calculating Gravy and Aroma for DNA sequence.

Args: seq (str):DNA sequence genetic_code_(int): default = 1, The Genetic Codes number described by NCBI (https://www.ncbi.nlm.nih.gov/Taxonomy/Utils/wprintgc.cgi) G (bool): default = False A (bool): default = False

Returns:

- Gravy value if arg(G) is True
- Aroma value if arg(A) is True
- None if both args are False

2.1.8 BCAWT.Optimal_codon_corr_method module

`Optimal_codon_corr_method.`**`op_corr`**(*ENc_file_name*, *RSCU_file_name*)

determine the optimal codons using the correlation method described here: https://doi.org/10.1371/journal.pgen.1000556

Args:

ENc_file_name (file): file contains the ENc values for a set of genes RSCU_file_name (file): file contains the RSCU values for a set of genes

Returns: DataFrame contains the optimal codons

2.1.9 BCAWT.P2_index module

`P2_index.`**`P2_index`**(*sequ*, *wwc=False*, *sst=False*, *wwy=False*, *ssy=False*, *p2=False*)

calculate P2 index.

Args: sequ (str): DNA sequence wwc (bool): default = False sst (bool): default = False wwy (bool): default = False ssy (bool): default = False p2 (bool): default = False

Returns:

- wwc value if arg(wwc) is True
- sst value if arg(sst) is True
- wwy value if arg(wwy) is True
- ssy value if arg(ssy) is True
- p2 value if arg(p2) is True

2.1.10 BCAWT.PR2_plot_data module

`PR2_plot_data.`**`PR2_plot`**(*sequ*, *o=False*, *a=False*)

Generate data for PR2 plot.

Args: sequ (str): DNA sequence o (bool): default = False a (bool): default = False

Returns:

- ordinate for PR2 plot if arg (o) is True
- abscissa for PR2 plot if arg (a) is True

CHAPTER 3

Indices and tables

- genindex
- modindex
- search

Python Module Index

Index