The Journal of Open Source Software

BCAWT: Automated tool for codon usage bias analysis for molecular evolution

Ali Mostafa Anwar[1]

1 Department of Genetics, Faculty of Agriculture, Cairo University, 12613, Cairo, Egypt

DOI: 10.21105/joss.01500

Software

- Review
- Repository
- Archive

Submitted: 03 June 2019
Published: 20 October 2019

Summary

The redundancy in the genetic code means that apart from methionine and tryptophan, an amino acid is encoded by at least two codons. Different codons for the same amino acid are termed synonymous codons. Synonymous codon usage is strongly influenced by evolutionary forces namely, selection and mutation and may vary strongly within or among organisms The preference of specific codons over others contributes to this variation and this phenomenon is called codon usage bias (CUB).

Many measurements have been developed to analyze and study CUB; effective number of codons (ENc) (Wright, 1990), codon adaptation index (CAI), relative synonymous codon usage (RSCU) (Sharp & Li, 1987) and, translational selection index (P2-index) (Liyuan Wang & Sun, 2018). Also, statistical analysis has been used to investigate the effect of different factors as selection and mutation on shaping CUB such as; Correspondence analysis, Parity Rule 2-plot Analysis and, Neutrality Plot (Hui Song & Nan, 2017). BCAWT was developed to analyze such phenomena (Codon Usage Bias) by the aforementioned measurements.

Various tools are available to analyze and measure CUB, but they lack some important measurements and plots for CUB analysis. What BCAWT does is an automated workflow to study the CUB of any organism's genes by all the measurements and plots mentioned above. Furthermore, using the correlation method to determine the optimal codons described by Hershberg & Petrov (2009) is implemented for the first time in BCAWT. The tool also includes statistical analysis such as correspondence analysis, correlation analysis, and t-test.

Implementation

BCAWT was developed using python 3.7 with built-in and third-party packages (Lee, 2018). BCAWT API is easy to use. For example, the following code snippet shows how to analyze genes for a coding sequence within the file `Ecoli.fasta`, for a genetic code specified, and to save the results to a folder named save_path.

```
from BCAWT import BCAWT
BCAWT.BCAW(['Ecoli.fasta'],'save_path',genetic_code_=11,Auto=True)
# processing...
```

The expected outputs from BCAWT can be divided into three groups. The first group is data in the CSV format (see Table 1), the second group is plots (summarized in Fig 1), and the last group is text files, whereby each text file contains results for a different statistical test. The equations used for analyzing CUB in the tool, and the API are reported in BCAWT's documentation.

The advantages of BCAWT over existing tools are; 1) the automated workflow, 2) the ability to handle large numbers of genes, 3) the method used to determine optimal codons, named the correlation method, is only available in BCAWT, 4) visualization and plotting capability, including the creation of violin plots for nucleotide contents, removing the need for other plotting software.

Output summary

BCAWT returns CSV files containing the CUB indices output (Table 1).

Table 1: Explanation of the CSV output files from BCAWT. (Abbreviations)

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	contains ; each RSCU result for each codon in each genes
CA_RSCUcodons	contains ; correspondence analysis first 4 axis for each codon
CA_RSCUgenes	contains ; correspondence analysis first 4 axis for each gene
CAI	contains ; gene id and CAI index
ENc	contains ; gene id and ENc index.
P2-index	contains ; gene id and P2 index
optimal codons	contains; putative optimal codons detected

Furthermore, BCAWT returns 11 plots (Fig 1), enabling an easy interpretation of the results.

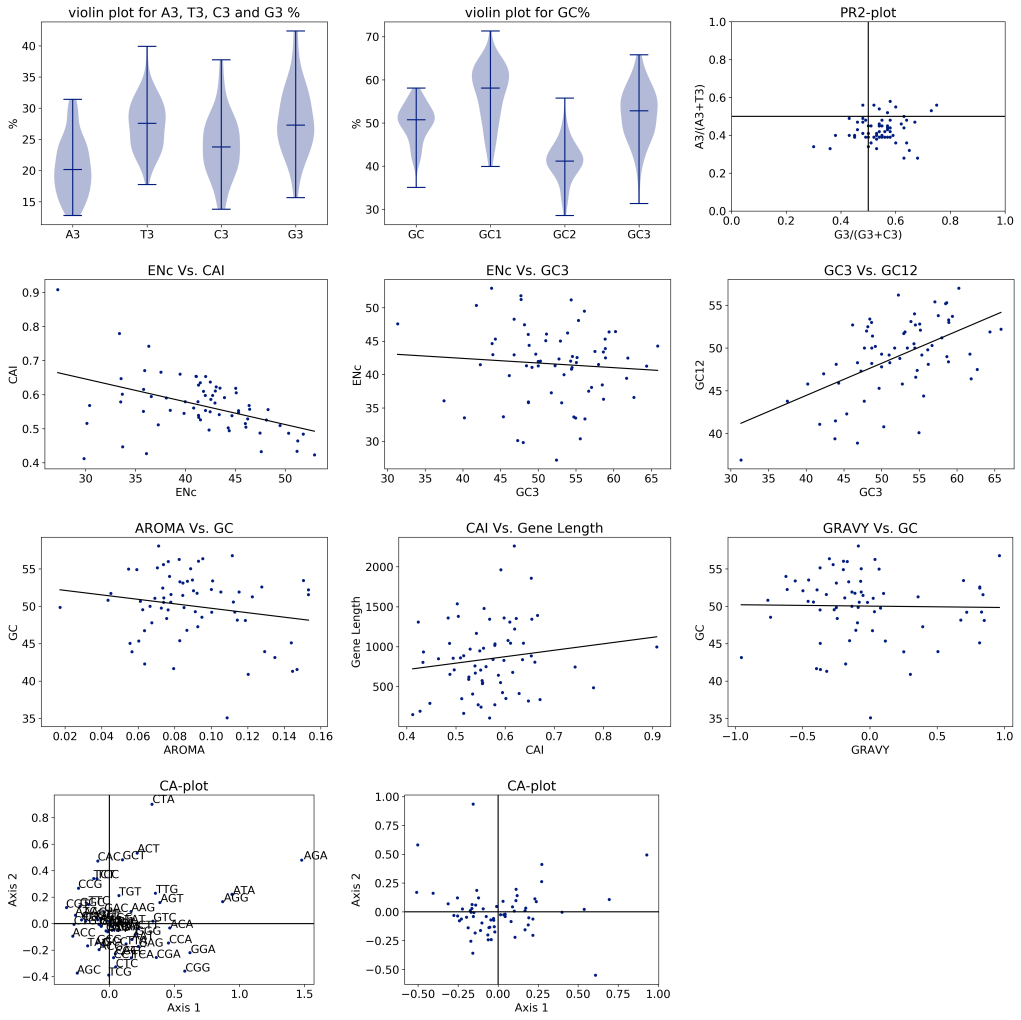

Figure 1: All output plots from BCAWT analysis for genes coding sequence from Escherichia coli

References

Hershberg, Ruth, & Petrov, D. A. (2009). General rules for optimal codon choice. *PLoS Genetics*, *5*(7), 1–10. doi:10.1371/journal.pgen.1000556

Hui Song, Q. S., Jing Liu, & Nan, Z. (2017). Comprehensive analysis of codon usage bias in seven epichloë species and their peramine-coding genes. *Frontiers in Microbiology*, *8*(6), 1–12. doi:10.3389/fmicb.2017.01419

Lee, B. D. (2018). Python implementation of codon adaptation index. *Journal of Open Source Software*, *3*(30). doi:10.21105/joss.00905

Liyuan Wang, Y. Y., Huixian Xing, & Sun, X. (2018). Genome-wide analysis of codon usage bias in four sequenced cotton species. *PLoS ONE*, *13*(3), 1–17. doi:10.1371/journal.pone.0194372

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. doi:10.1093/nar/15.3.1281

Wright, F. (1990). The "effective number of codons" used in a gene. *Gene*, *87*(1), 23–29. doi:10.1016/0378-1119(90)90491-9