# gtAI

*Release 1.0.0*

**Jul 04, 2023**

# Contents

CHAPTER 1

# Guide

## 1.1 Introduction

gtAI is a package implemented in python to measure the tRNA adaptation index (tAI) [1], based on a novel approach [2]. The main advantages of this approach:

1) It requires the tRNA gene copy number (or tRNA levels) and coding sequences, without the need for additional gene expression information (can be used If available).
2) It uses a genetic algorithm to reach the best set of Sij-values (equation 1).
3) It outperforms previously suggested methods (the original tAI [1] and stAI [3]) in tRNA adaptation index (tAI) computation by producing significantly better results.

**Note: The "g" in gtAI stands for genetic, as in the genetic algorithm used in the implementation, but it calculates the known tAI [1].**

## 1.2 gtAI Workflow

1) A reference set is obtained by taking 5% (or more) of coding sequences from a tested genome with the lowest ENc values (equation 3). (Or insert a reference set of interest)
2) Then, RSCU values for the reference set are generated (equation 4).
3) The genetic algorithm will search for the Sij weights that maximizes the correlation between (equation 1) and (equation 4).
4) The final Wi values are calculated based on the optimized Sij weights from step 3.
5) Calculated Wi values are normalized (wi), then the tAI value for a coding sequence (g) is calculated using equation 3.

## 1.3 Dependencies

1. Biopython
2. pandas
3. numpy
4. gaft
5. lxml

**Note : Python >=3.7 is required.**

## 1.4 Installation guide

**gtAI is available on:**

- GitHub: https://github.com/AliYoussef96/gtAI
- PYPI: https://pypi.org/project/gtAI/

**Installation using pip**

```
pip install gtAI
```

**Note:**

A graphical user interface (GUI) version is available (only for windows 10 users). For more information on how to install and how to use it, please see the documentation: GUI documentation

## 1.5 Example

1- Import gtAI functions.

```
from gtAI import Run_gtAI

from gtAI import gtAI
```

2- In this example, we will use Saccharomyces cerevisiae S288C coding sequences.

3- Prepare the tRNA gene copy number of the tested genome.

The user has two options; a) input the tRNA gene copy number as python dictionary or, b) using GtRNAdb() function, the user can get it automatically from the GtRNA database, using the link to the tested genome (In our case Saccharomyces cerevisiae S288C). Or by tRNADB_CE() function to get the tRNA gene copy number from tRNADB_CE database using also the link to the tested genome.

In this example, the second option (b) will be used.

```
url_GtRNAdb = "http://gtrnadb.ucsc.edu/genomes/eukaryota/Scere3/"

#### From GtRNAdb

GtRNA = gtAI.GtRNAdb(url_GtRNAdb)
```

**for more infromation about GtRNAdb() as well as tRNADB_CE(); go to API part.**

4- Parameter settings for gtai_analysis() function.

```
genetic_code_number = 1
ref_fasta = ""
bacteria = False
size_pop = 60
generation_number = 100
```

**for more infromation about gtai_analysis() and the parameters; go to API part.**

5- Run gtAI.

```
df_tai , final_dict_wi, rel_values = Run_gtAI.gtai_analysis(main_fasta,GtRNA,genetic_
↪code_number,bacteria=bacteria, size_pop=size_pop,generation_number=generation_
↪number)
```

Returns:

```
df_tai (dataframe): Contains each gene id and its gtAI value
final_dict_wi (dict): Contains each codon and its absolute adaptiveness value
rel_values (dict): Contains each codon and its relative adaptiveness values
```

6- To save the gtAI result as a CSV file.

```
import pandas as pd
df_tai.to_csv("test.csv", header=True)
```

Output example

## 1.6 Equations

- Equation 1 [1].

$$W_i = \sum_{j=1}^{n_i}(1 - s_{ij})tGCN_{ij}$$

- Equation 2 [1].

$$w_i = \begin{cases} W_i/W_{max} & if\ W_i \neq 0 \\ w_{mean} & else \end{cases}$$

- Equation 3 [1].

$$tAI = (\prod_{k=1}^{l_g} w_{i_{kg}})^{1/l_g}$$

- Equation 4 [4].

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

- Equation 5 [5].

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

## 1.7 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 How to create a Pull Request? documentation on pull requests

## 1.8 Citation

## 1.9 LICENSE

GNU GENERAL PUBLIC LICENSE Version 3, 29 June 2007

Copyright (C) 2007 Free Software Foundation, Inc. <https://fsf.org/> Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

For more information: https://github.com/AliYoussef96/gtAI/blob/master/LICENSE

## 1.10 Help

If you need more help please contact ali.mo.anwar@std.agr.cu.edu.eg or aliali.mostafa99@gmail.com or khodary-saif@gmail.com.

## 1.11 References

1. Reis, M. d. (2004). Solving the riddle of codon usage preferences: a test for translational selection. Nucleic Acids Research, 32(17), 5036–5044. https://doi.org/10.1093/nar/gkh834
2.
3. Sabi, R., & Tuller, T. (2014). Modelling the Efficiency of Codon–tRNA Interactions Based on Codon Usage Bias. DNA Research, 21(5), 511–526. https://doi.org/10.1093/dnares/dsu017
4. Sharp PM, Li W. Codon Adaptation Index and its potential applications Nucleic Acids Research. 1987;15: 1281–1295.
5. Sun X, Yang Q, Xia X. An improved implementation of effective number of codons (Nc). Mol Biol Evol. 2013;30: 191–196. https://doi.org/10.1093/molbev/mss201

CHAPTER 2

# API

## 2.1 gtAI package

### 2.1.1 Submodules

### 2.1.2 gtAI.CA_RSCU module

CA_RSCU.**RSCU**(*allseq*, *allseq_name*, *The_Genetic_Codes_number*)

calculate RSCU values

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 codons and RSCU values

### 2.1.3 gtAI.Run_gtAI module

### 2.1.4 gtAI.gtAI module

gtAI.**GtRNAdb**(*url*)

Get the tRNA genes count from GtRNAdb database

Args:

url (string): a url to anticodon table for organism from GtRNAdb database (http://gtrnadb.ucsc.edu/)

Returns:

A dictionary of each anticodon and its gene count

Raises:

ValueError if the URL, not a valid for GtRNAdb database

Example:

#example 1

> GtRNAdb("http://gtrnadb.ucsc.edu/genomes/eukaryota/Hsapi19/")

gtAI.**abs_Wi**(*dict_anticodon_number*, *Sug*, *Sci*, *Sai*, *Sgu*, *Sal*, *bacteria=False*)

Calculate the absolute adaptiveness values for each codon.

Args:

dict_anticodon_number (dict): a merged dictionary of anticodon-codon with each anticodon tRNA gene copy number returned from dict_codon_anticodon_count() function.

Sug (int): the S-value for codon with (U) in the third position and (G) in first anticodon position

Sci (int): the S-value for codon with (C) in the third position and (I) in first anticodon position

Sai (int): the S-value for codon with (A) in the third position and (I) in first anticodon position

Sgu (int): the S-value for codon with (G) in the third position and (U) in first anticodon position

Sal (int): the S-value for codon with (A) in the third position and (L) in first anticodon position ( if bacteria = True)

bacteria (bool): True If the tested organism is prokaryotic or archaea, else equal to False ( default = False )

Returns:

A dictionary of the absolute adaptiveness values for each codon

Note:

All Sij values (as Sug) should be a number from 0 to 1

Raises:

ValueError if the length of dict_anticodon_number equal to zero ValueError if any of Sij values are less than 0 or greater than 1

Example:

> anticodon_dict = tRNADB_CE("trna.ie.niigata-u.ac.jp/cgi-bin/trnadb/whole_anticodon.cgi?GID=|CP001631&DTYPE=C
# Return an anticodon table of Acidimicrobium ferrooxidans DSM 10331 > anticodon_codon = dict_codon_anticodon ( anticanticodon_dictodon ) > dict_codon_anticodon_count = dict_codon_anticodon_count(anticodon_codon,anticodon_dict,bacteria = True) > abs_Wi(dict_codon_anticodon_count, Sug=1, Sci=1, Sai=1, Sgu=1, Sal=1, bacteria=True)

gtAI.**calc_Tai**(*DNA*, *rel_dict_wi*, *genetic_code_number=1*)

Calculate the tRNA adaptation index of a gene.

Args:

DNA (str): a coding sequence of DNA ( should only contain A, C, T, and G )

rel_dict_wi (dict): dictionary of the relative adaptiveness values for each codon returned from rel_Wi() function

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

Returns:

the tRNA adaptation index of a gene

Raises:

ValueError if the length of dict_anticodon_number equal to zero

Example:

> anticodon_dict = tRNADB_CE("trna.ie.niigata-u.ac.jp/cgi-bin/trnadb/whole_anticodon.cgi?GID=|CP001631&DTYPE=C
# Return an anticodon table of Acidimicrobium ferrooxidans DSM 10331 > anticodon_codon = dict_codon_anticodon ( anticanticodon_dictodon ) > dict_codon_anticodon_count = dict_codon_anticodon_count(anticodon_codon,anticodon_dict,bacteria = True) > abs_Wi = abs_Wi(dict_codon_anticodon_count, Sug=1, Sci=1, Sai=1, Sgu=1, Sal=1, bacteria=True) > rel_Wi(abs_Wi, 11)

gtAI.**dict_codon_anticodon**(*anti_codon_dict*)

Identify all potential anticodons for each codon

Args:

anti_codon_dict (dict): a dictionary of all anticodons for an organism ( returned from ( tRNADB_CE() ) or ( GtRNAdb() ) )

Returns:

A dictionary of all potential anticodons for each codon

Raises:

ValueError if the length of anti_codon_dict equal to zero

Example:

> anticodon_dict = tRNADB_CE("trna.ie.niigata-u.ac.jp/cgi-bin/trnadb/whole_anticodon.cgi?GID=|CP001631&DTYPE=C
# Return an anticodon table of Acidimicrobium ferrooxidans DSM 10331 > dict_codon_anticodon ( anticanticodon_dictodon )

gtAI.**dict_codon_anticodon_count**(*dic_codon_anticodon*, *dict_tGCN_main*, *bacteria=False*)

Merge anticodon-codon dictionary with each anticodon tRNA gene copy number.

Args:

dic_codon_anticodon (dict): A dictionary of all potential anticodons for each codon returned from dict_codon_anticodon() function.

dict_tGCN_main (dict): a dictionary of all anticodons for an organism ( returned from ( tRNADB_CE() function ) or ( GtRNAdb() function ) )

bacteria (bool): True If the tested organism is prokaryotic or archaea, else equal to False ( default = False )

Returns:

a merged dictionary of anticodon-codon with each anticodon tRNA gene copy number.

Raises:

ValueError if the length of anti_codon_dict equal to zero ValueError if the length of dict_tGCN_main equal to zero

Example:

> anticodon_dict = tRNADB_CE("trna.ie.niigata-u.ac.jp/cgi-bin/trnadb/whole_anticodon.cgi?GID=|CP001631&DTYPE=C
# Return an anticodon table of Acidimicrobium ferrooxidans DSM 10331 > anticodon_codon = dict_codon_anticodon ( anticanticodon_dictodon ) > dict_codon_anticodon_count(anticodon_codon,anticodon_dict,bacteria = True)

gtAI.**rel_Wi**(*dict_abs_Wi*, *genetic_code_number=1*)

Calculate the relative adaptiveness values for each codon.

Args:

dict_abs_Wi (dict): dictionary of the absolute adaptiveness values for each codon returned from abs_Wi() function

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

Returns:

a dictionary of the relative adaptiveness values for each codon

Raises:

ValueError if the length of dict_abs_Wi equal to zero

Example:

> anticodon_dict = tRNADB_CE("trna.ie.niigata-u.ac.jp/cgi-bin/trnadb/whole_anticodon.cgi?GID=|CP001631&DTYPE=C
# Return an anticodon table of Acidimicrobium ferrooxidans DSM 10331 > anticodon_codon = dict_codon_anticodon ( anticanticodon_dictodon ) > dict_codon_anticodon_count = dict_codon_anticodon_count(anticodon_codon,anticodon_dict,bacteria = True) > abs_Wi = abs_Wi(dict_codon_anticodon_count, Sug=1, Sci=1, Sai=1, Sgu=1, Sal=1, bacteria=True) > rel_Wi(abs_Wi, 11)

gtAI.**tRNADB_CE**(*url*)

Get the tRNA genes count from tRNADB-CE database

Args:

url (string): a url to anticodon table for organism from tRNADB_CE database (http://trna.ie.niigata-u.ac.jp/cgi-bin/trnadb/index.cgi)

Returns:

A dictionary of each anticodon and its gene count

Raises:

ValueError if the URL, not a valid for tRNADB-CE database

Example:

> tRNADB_CE("trna.ie.niigata-u.ac.jp/cgi-bin/trnadb/whole_anticodon.cgi?GID=|CP001631&DTYPE=CMP&VTYPE=1"
# Return an anticodon table of Acidimicrobium ferrooxidans DSM 10331

CHAPTER 3

# Indices and tables

- genindex
- modindex
- search

# Python Module Index

## c



## g

# Index