RCDI/eRCDI (QUICK MANUAL)

INTRODUCTION

What is RCDI?

The RCDI server is a web-application that calculates the Relative Codon Deoptimization Index (RCDI) and an expected value of the RCDI for a set of query sequences by generating random sequences with similar G+C content and amino acid composition to the input. This expected RCDI therefore provides a direct threshold value for discerning whether the differences in the RCDI value are statistically significant and arise from the codon preferences or whether they are merely artifacts that arise from internal biases in the G+C composition and/or amino acid composition of the query sequences.

Availability

http://genomes.urv.cat/CAIcal/RCDI

INPUTS

Format of the sequences

DNA or RNA sequences are allowed and must be introduced in FASTA forma.

Example:

```
>SeqA
ATGCAGAACGACGCCGGCGAATTTGTGGACTTATACGTGCCTCGGAAGTGCTCAGCCTCTAATCGCATTATAGCC
GCTAAGGACCACGCTAGCATCCAGATGAATGTCGCTGAGGTAGACCGCACCACAGGGAGATTTAACGGCCAGTT
TAAGACTTATGGAATATGCGGGGCCATTCGGAGAATGGGGGAGAGCGATGACAGCATCCTCAGGCTGGCGAAG
GCCGACGGGATTGTTTCAAAAAACTTC
>SeqB
ATGCAGAATGACGCAGGCGAGTTTGTGGATCTATATGTGCCACGGAAGTGCTCGGCCTCCAATAGGATTATTGCC
GCAAAAGACCATGCCTCTATCCAGATGAATGTTGCAGAAGTGGACCGGTCTACCGGCAGGTTTAACGGGCAGTT
CAAAACGTATGGCATTTGCGGGGCTATCCGACGGATGGGTGAAAGCGATGATAGTATCCTGCGCTTAGCAAAGG
CTGACGGGATTGTATCGAAGAACTTC
>SeqC
ATGCAAAACGATGCCGGCGAGTTTGTGGACCTCTATGTGCCTAGAAAATGTTCGGCTTCCAATCGTATCATTGGT
GCCAAGGATCATGCCAGCATCCAGATGAACGTTGCCGAGGTTGATAAGGTGACAGGCAGGTTTAATGGCCAGTT
CAAGACTTACGCGATTTGCGGCGCGATCAGACGAATGGGCGAGAGCGACGACTCAATCTTGAGGTTAGCAAAGG
CCGACGGGATCGTTAGCAAAAACTTC
```

Format of the reference set

An easy way to introduce the codon usage reference table in RCDI/eRCDI is to copy and paste the codon usage tables from *Codon Usage Database* (Nakamura et al., 2000). We have therefore added a link to this database (and also codon usage tables from model organisms) in the left frame of the server.

The codon usage table from the '*Codon Usage Database*' format allowed in RCDI is as follows:

> **Fields: [triplet] [frequency: per thousand] ([number])...**

Example:

```
UUU 17.4(586747) UCU 15.0(507382) UAU 12.1(408578) UGU 10.5(352664)
UUC 20.4(687969) UCC 17.7(596425) UAC 15.3(516505) UGC 12.6(426761)
UUA 7.5(254407) UCA 12.1(409879) UAA 1.1( 35822) UGA 1.6( 55514)
UUG 12.8(432797) UCG 4.5(150335) UAG 0.8( 27554) UGG 13.3(447152)
CUU 13.1(440882) CCU 17.5(589809) CAU 10.8(363555) CGU 4.6(155426)
CUC 19.7(664417) CCC 20.0(675558) CAC 15.1(509431) CGC 10.6(357380)
CUA 7.1(240672) CCA 16.9(569871) CAA 12.1(408697) CGA 6.2(208816)
CUG 39.9(1347830) CCG 7.0(237033) CAG 34.3(1157220) CGG 11.6(390529)
AUU 15.8(532975) ACU 13.0(438753) AAU 16.7(563795) AGU 12.1(408481)
AUC 20.9(705646) ACC 19.0(641707) AAC 19.0(642797) AGC 19.5(656528)
AUA 7.4(249300) ACA 15.0(504527) AAA 24.1(812474) AGA 11.9(402225)
AUG 22.0(744022) ACG 6.1(205470) AAG 32.0(1079579) AGG 11.9(402146)
GUU 11.0(370035) GCU 18.5(624602) GAU 21.7(732533) GGU 10.8(364282)
GUC 14.6(491325) GCC 28.1(947810) GAC 25.2(850343) GGC 22.5(758251)
GUA 7.1(238697) GCA 15.9(537665) GAA 28.6(964323) GGA 16.4(553492)
GUG 28.3(956245) GCG 7.5(253270) GAG 39.7(1340672) GGG 16.6(558612)
```

We have also introduced another format as follows:

Fields: [triplet] ([number])...

Example:

```
TTT (171) TCT (147) TAT (124) TGT (99)
TTC (203) TCC (172) TAC (158) TGC (119)
TTA (73) TCA (118) TAA (0) TGA (0)
TTG (125) TCG (45) TAG (0) TGG (122)
CTT (127) CCT (175) CAT (104) CGT (47)
CTC (187) CCC (197) CAC (147) CGC (107)
CTA (69) CCA (170) CAA (121) CGA (63)
CTG (392) CCG (69) CAG (343) CGG (115)
ATT (165) ACT (131) AAT (174) AGT (121)
ATC (218) ACC (192) AAC (199) AGC (191)
ATA (71) ACA (150) AAA (248) AGA (113)
ATG (221) ACG (63) AAG (331) AGG (110)
GTT (111) GCT (185) GAT (230) GGT (112)
GTC (146) GCC (282) GAC (262) GGC (230)
GTA (72) GCA (160) GAA (301) GGA (168)
GTG (288) GCG (74) GAG (404) GGG (160)
```

%G+C content option

Users may define (**optional**) the %G+C of the random sequences. If this option is not select, the program uses the %G+C content of the input sequences.

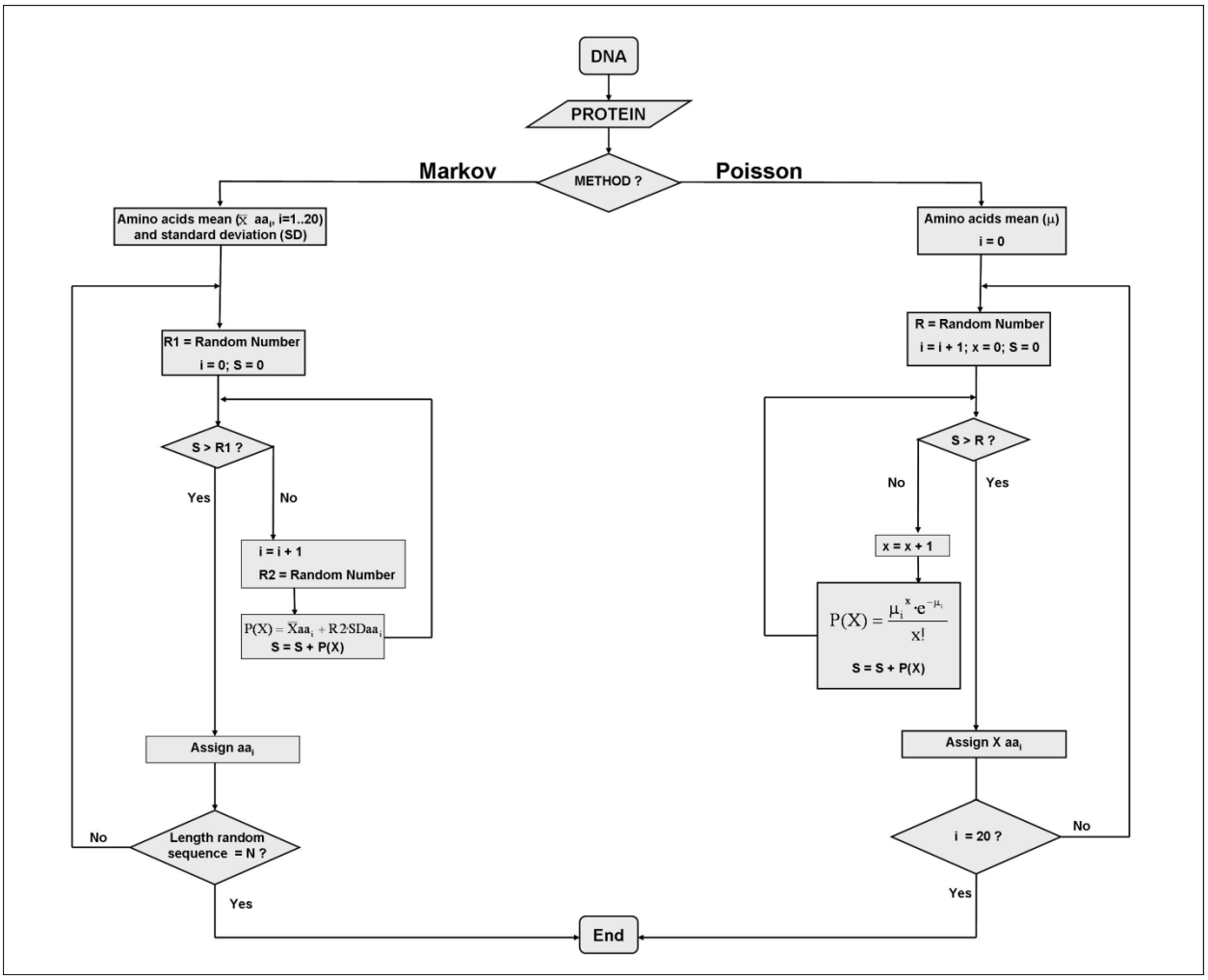
Upper (one-side) tolerance limit

Given a confidence limit and a percentage of the population (also called coverage) chosen by the user, an expected value of RCDI is estimated using an upper one-sided tolerance interval for a normal distribution. A tolerance interval is a way to determine a range that, with a given **confidence level (90%, 95% or 99%)**, will contain a certain **percentage of the population (90%, 95% or 99%)**. In our case, the upper limit represents the value that is not exceeded by the specified fraction of the randomly generated sequences with the chosen confidence limit.

Poisson or Markov method?

Both methods are similar in the sense that, using the amino acid composition of the query sequences and random generated numbers, they generate a series of amino acid sequences that are compositionally equivalent to the query. These sequences are then back-translated to DNA using the mean G+C content of the query. The RCDI of these random-generated sequences are then used to estimate an expected value of RCDI.

Both methods provide similar results. The differences between the two methods can be seen in the following diagram:



The Markov method is a Markov Model of order 0. This means that the probability of an amino acid to occupy a position in a protein is independent of the other sites. The Markov method generates the random sequences by adding one amino acid each time to a generated amino acid sequence, since this sequence has the desired length. The amino acid is chosen randomly using the amino acid frequencies of the query sequences and a random number.

The Poisson method, on the other hand, assumes that the number of times that each amino acid is used in a protein follows a Poisson distribution. The amino acid frequencies of the query sequences multiplied by the length of the sequence to be generated are the expected values (called h) of the number of times each amino acid is used in a protein. From these expected values, the probability of finding each amino acid 0, 1, 2, 3, 4, .., n times is calculated from the expression p(n) = e(-h) hn / n! From these probabilities and a random number, the number of each amino acid in a random generated sequence is calculated.

OUTPUTS

Genes' parameters

This output shows the RCDI values, the frequencies for all codons (calculated using the following formula: *[(CiFa/CiFh)Ni]*) and the %G+C of each gene. CiFa is the relative frequency of codon i for a specific amino acid in the test sequence; CiFh is the relative frequency of codon i for a specific amino acid in the reference sequence and Ni is the number of occurrences of codon i in the test sequence

Global parameters

This output is related to the parameters used to create 'random' sequences calculated, i.e. the mean %G+C and amino acid composition, and the number of 'random' sequences used to calculate the eRCDI.

Statistical tests

Chi-square goodness-of-fit: a chi-square test is conducted to compare the goodness-of-fit between the amino acid frequencies or G+C content of each sequence of the query and their mean values.

Kolmogorov-Smirnov test: the expected value of RCDI is estimated from the mean and standard deviation of the RCDI of the randomly generated sequences using a tolerance interval based on a normal distribution. To estimate this expected value, the user has to choose two parameters: the level of significance and the percentage of the population or coverage. To check whether the RCDI of the randomly generated sequences follow a normal distribution, a Kolmogorov-Smirnov test is made.

Expected RCDI

This output provides the mean RCDI value from 'random' sequences and the eRCDI value. Given a confidence limit (90%, 95% or 99%) and a percentage of the population (also called coverage: 90%, 95% or 99%) chosen by the user, an expected value of RCDI is estimated using an upper one-sided tolerance interval for a normal distribution. A tolerance interval is a way to determine a range that, with a given confidence level, will contain a certain percentage of the population. In our case, the upper limit represents the value that is not exceeded by the specified fraction of the randomly generated sequences with the chosen confidence limit.