A More elaborative way to check codon quality: an open source program

Rakesh Kumar Shardiwal[1] and Dr. Sohrab Sartaj Sayed[2]

[1] Genseq Sdn Bhd, Cyberjaya, Malaysia
[2] JK Agri Genetics, Hyderabad, India

Introduction

Protein-coding genes are translated into amino acid polypeptides following the genetic code. The sequence of a gene directly determines the sequence of amino acids in the protein it produces [1]. In a reading frame of protein, each group of three consecutive nucleotides in the DNA (or RNA) sequence corresponds to an amino acid residue that will be incorporated into the protein sequence. These nucleotide triplets are called "codons". The correspondence between the codons and their coded amino acids constitutes the genetic code [2]. Genetic code elements have large number of redundancy. A direct result of the redundancy is the observation of codons that codes for the same amino acid (synonymous codons). These codons are very rarely used with equal frequency.

According to the study of Sharp and Li [3] in *Escherichia coli* and yeast *Saccharomyces cerevisiae*, there is a clear positive correlation between degree of codon bias and level of gene expression and it is desirable to quantify the degree of bias in each gene in such a way that comparisons can be made both within and between species. Codon bias is correlated with a corresponding bias of tRNA, which is a wide arrangement for optimizing the gene expression. On the other side, it is suggested that heterologous gene expression is not as sensitive to codon bias as previously thought, but that it is quite sensitive to other characteristics of the heterologous gene [4-5,9].

An optimal codon will get you more expression with good translation rate. On the other side non-optimal codons has been postulated to reduce translation rate, probably due to a relative scarcity of cognate tRNA species. Non-optimal codons have bit advantage in to maintain a low cellular concentration of the proteins that they encode [6,7-9].

Relative Synonymous Codon Usage (RSCU) measures the relative frequency that each codon suits to encode a particular amino acid.

Methodology

We have implemented an algorithm to optimize the codon, which is based on a simple effective measure of synonymous codon usage bias. The Relative Synonymous Codons Uses (RSCU) value for a codon is simply the observed frequency of that codon divided by the frequency expected under the assumption of equal usage of synonymous codons for an amino acid [5].

Thus,

$$RSCU_{ij} = \frac{X_{ij}}{\frac{1}{n_i}\sum_{j=1}^{n_i} X_{ij}} \quad [1]$$

Where Xij is the number of occurrences of the jth codon for the ith amino acid, and ni is the number (from one to six) of alternative codons for the ith amino acid. In the absence of any codon usage bias, the RSCU value would be 1.00. A codon that is used less frequently than expected will have a value of less than 1.00 and vice versa for a codon that is used more frequently than expected. The RAC (Relative Adaptiveness of a Codon) is calculated based on RSCU value, the frequency of use of that codon compared to the frequency of the optimal codon for that amino acid:

$$w_{ij} = RSCU_{ij} / RSCU_{imax} = X_{ij}/X_{imax} \quad [2]$$

Where RSCUimax and Ximax are the RSCU and X values for the most frequently used codon for the ith amino acid. Codon usage data have been complied for trpR gene lowly expressed regulatory gene and dnaK gene of Escherichia coli to obtain reference RSCU and RAC value.

Process Flow

The codon quality of coding sequences can be depicted in two different ways. The simplest way of depiction is to plot the codon usage frequency that can be found in common codon usage tables [5,10]. A more elaborate way to depict the codon quality is to convert the codon usage frequency into relative adaptiveness values. In contrast to the codon usage frequency the relative adaptiveness takes into account the number of codons which code for the respective amino acid. Selection of appropriate codon plays a major role in the determination of codon usage in all organisms; this program is implemented as Object Oriented way to get more efficient and accurate result to select most preferable codons. Our translation for each coding sequence (CDS) is based on genetic codes [2] and RSCU values. The basic principle for deriving relative adaptiveness values out of codon usage frequency values is the following. The codon usage table (Table 1) for trpR gene of Escherichia coli, [6,8-9] lists the following values for Glycine and Glutamate codons:

Table 1. The codon usage table for trpR gene of *Escherichia coli*.

AmAcid	Codon	Number	RSCU	RAC
Glu	GAG	1	0.18	0.10
Glu	GAA	10	1.81	1.00
Gly	GGA	7	0.55	0.30
Gly	GGT	23	1.82	1.00

The frequency of each codon is listed in the 3rd column named "Number". Comparing the RAC values for the best glutamic codon GAA (1.81) with the best glycine codon GGT (1.82) shows clearly that codon frequency values for one amino acid cannot be compared to those of other amino acids even within the same codon usage table. The codon quality of the following sequence stretch is analysed by plotting codon usage frequencies (Figure 1) and relative adaptiveness values (Figure 2).

Inputs

Our Program support three kinds of Input

1. If you have your customize sequence then you can use like this

```
my $seqobj = Bio::Tools::CodonOptTable
->new(
seq =>
'ATGGGGTGGGCACCATGCTGCTGTCGTGAATTTGG
GCACGATGGTGTACGTGCTCGTAGCTAGGGTGGGT
GGTTTG',
Id    => 'GeneFragment12',
accession_number => 'Myseq1',
alphabet => 'dna',
is_circular => 1,
genetic_code => 1,
);
```

2. If you want to read from file

```
my $seqobj = Bio::Tools::CodonOptTable
-> new(
file      => "contig.fasta",
format => 'Fasta',
genetic_code => 1,
);
```

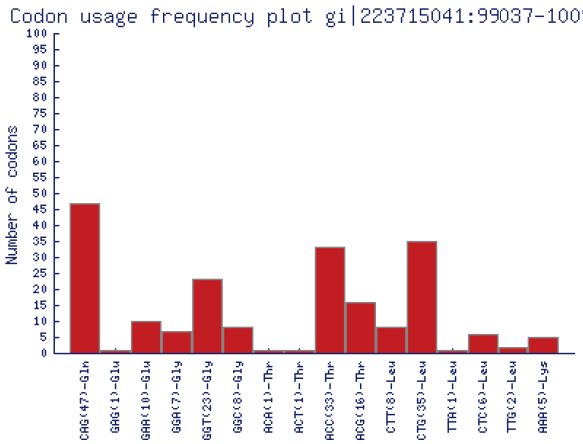

Figure 1. Plotting codon usage frequencies.

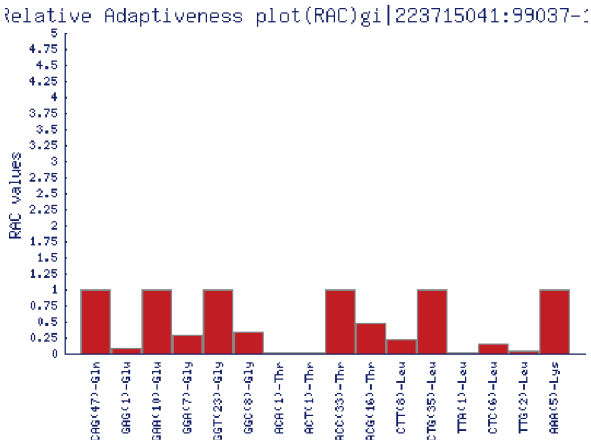

Figure 2. Relative adaptiveness usage frequencies.

3. If you have accession number only, so the program will download that sequence from NCBI and get you the optimization frequency.

```
my $seqobj = Bio::Tools::CodonOptTable
-> new(
ncbi_id => "J00522",
genetic_code => 1
);
```

3.1.2 Input Parameters

Seq	=> sequence string
display _ id	=> display id of the sequence (locus name)
accession _ number	=> accession number
primary _ id	=> primary id (Genbank id)
desc	=> description text
alphabet	=> molecule type (dna,rna,protein)
id	=> alias for display id
file	=> file location
format	=> file format
ncbi _ id	=> NCBI accession number
genetic _ code	=> 1 (Default)

Output

The program will produce three kinds of Output.

1.1 RSCU and RAC Table along with amino acid name of the codons

```
my $myCodons = $seqobj -> rscu_rac_
table();
if($myCodons)
{
for my $each_aa (@$myCodons)
{
print "Codon : ",$each_aa->{'codon'},"\
t";
print "Frequency : ",$each_aa-
>{'frequency'},"\t";
print "AminoAcid : ",$each_aa->{'aa_
name'},"\t";
print "RSCU Value : ",$each_aa-
>{'rscu'},"\t";
print "RAC Value : ",$each_aa-
>{'rac'},"\t";
print "\n";
}
}
```

2.1 Graph between RSCU and RAC for more statistical analysis

```
$seqobj -> generate_graph($myCodons,"my
output.gif");
```

3.1 Most preferred codon for the sequence

```
my $prefered_codons = $seqobj ->
prefered_codon ($myCodons);
while ( my ($amino_acid, $codon) =
each(%$prefered_codons ) )
{
print "AminoAcid : $amino_acid \t Codon
: $codon\n";
}
```

Web Interface

The current version of Bio::Tools::CodonOptTable is a 0.07 is a open source pure perl and bioperl program and users can use it with common gateway interface (CGI) perl and make good tool for codons optimizations.

Here is an example tool created with CodonOptimizer[1] [11]

Availability

http://search.cpan.org/~shardiwal/Bio-Tools-CodonOptTable-0.07/lib/Bio/Tools/CodonOptTable.pm

Results and discussion

In this study we have explored the potential of the RSCU and RAC bias table in gene expression. These RSCU and RAC is being used to optimize the codons to get higher expression of desired protein. Our program is based on Sharp and Li [3] study in *Escherichia coli* and yeast *Saccharomyces cerevisiae*.

In order to improve this situation, we have developed a Perl module that relies on the BioPerl bundle and implements the algorithm to optimize the codons for better gene expression. Furthermore, this module let the user to perform simple experiments with codons without having to develop a program or Perl script. We have used Object Oriented approach to solve this problem and provided a simple API (Application Programming Interface).

Our program has the ability to handle complete genome and draw graph of codons based on frequencies and RSCU. In future work, we will develop more comprehensive interface methods to annotate sequence to give more informative results.

Conclusion

This Perl Module is available in CPAN (Comprehensive Perl Archive Network), and can

1 http://bioinformatics.chhotikhatu.com/main.html

also be downloaded. A web-based application is also available (see availability).

References

1. Lewin R (1996). Patterns in Evolution - The New Molecular View. Scientific American Library, New York.
2. The Genetic Codes [http://www.ncbi.nlm.nih.gov/Taxonomy/Utils/wprintgc.cgi]
3. Sharp PM, Li WH (1987) The codon Adaptation Index--a measure of directional synonymous codon usage bias, and its potential applications. Nucleic Acids Res. 15(3):1281-95.
4. Ikemura T (1985) Codon usage and tRNA content in unicellular and multicellular organisms. Mol Biol Evol. 2(1):13–34.Holm L (1986) Codon usage and gene expression. Nucleic Acids Res., 14(7):3075–3087.
5. Ikemura T. (1985) Codon usage and tRNA content in unicellular and multicellular organisms. Mol Biol Evol. 2(1):13–34.
6. Grantham R., Gautier C., Gouy M., Jacobzone M., Mercier R. (1981) Codon catalog usage is a genome strategy modulated for gene expressivity. Nucleic Acids Res. 9(1):r43–r74.Gouy M., Gautier C. (1982) Codon usage in bacteria: correlation with gene expressivity. Nucleic Acids Res. 10(22):7055–7074.
7. Holm L. (1986) Codon usage and gene expression. Nucleic Acids Res. 14(7):3075–3087.
8. Sharp P.M. (1985) Does the 'noncoding' strand code? Nucleic Acids Res.13(4):1389–1397.
9. CodonOptimizer [http://bioinformatics.chhotikhatu.com/main.html]