PACKEIS

version 1.0

- Documentation -

1. Scope and prerequisites

PACKEIS is a software that allows us to assess whether or not a coding sequence represents an extreme solution in terms of backfolding, considering the alternative coding sequences that could have been realized by evolution in order to encode the given peptide sequence based on usage of synonymous codons.
PACKEIS calculates average probabilities for base pairing in predicted local structures using RNAplfold and RNAfold from the ViennaRNA Package (https://www.tbi.univie.ac.at/RNA/). To run PACKEIS you need to install the ViennaRNA Package first. Windows user can use a local copy of `RNAplfold.exe` and `RNAfold.exe` located in the PACKEIS directory. By default, PACKEIS uses RNAfold instead of RNAplfold for input sequences below 100 bp in length. Next, it predicts the degree of backfolding (DBF) as measured in percent of paired bases within the original ORF (oORF) as the sum of the base pairing probabilities for each position. In a second step, PACKEIS generates a set of alternative ORFs (aORFs), each of which still codes for the same amino acid sequence, and calculates the DBF for each aORF as described above. By comparing the DBF of the oORF with those of the aORFs, PACKEIS outputs a measurand (DBF-score) that allows to assess the probability that the DBF of the oORF is a product of chance, where a DBF-score of 0 means that none of the aORFs exhibits a lower DBF, while a DBF-score of 1 means that none of the aORFs exhibits a higher DBF.
Running PACKEIS on your local machine requires the installation of a Perl interpreter. Perl is pre-installed on common Linux and Mac systems. For Windows you can download and install either StrawberryPerl (www.strawberryperl.com) or ActivePerl (www.activestate.com/activeperl/downloads). In addition, you

2. Getting started

Usage

Start PACKEIS from the terminal (command line) with the following command:

```
perl PACKEIS.pl -i [input] (-option [value])
```

Value for -i can be a FASTA file or a plain sequence. Use -i multiple times for multiple input files and/or sequences. You should not use multiple files with identical base names (e.g. input.fas together with input.fasta). When using FASTA input files make sure to use non-identical headers within one FASTA file. All output files will be copied to the folder `PACKEIS.[date of execution]` (e.g. `PACKEIS.Tue-Jan-15-14h19m05s-2019`).

Models for reconstruction of aORFs

For the reconstruction of aORFs, PACKEIS implements three different models. Model0 (free) uses equal frequencies for all synonymous codons of a specific amino acid. Model1 (strict) uses codon frequencies that reflect the codon frequencies of the corresponding input file (or alternatively codon frequencies as defined by the user). Model2 (shuffle) reconstructs aORFs by shuffling those codons that are already present in a given oORF. In contrast to model0 and model1, codon frequencies and GC-content are perfectly preserved in each of the aORFs compared to the oORF when using model2 and aberrant codon usage or GC-content of a oORF can be excluded as causative for extreme DBF-scores (Figure 1).
If you want to apply model1 for generation of alternative coding sequences and use your own codon frequencies you can edit one of the example codon usage files (`codon_usage.STD.table` for the standard genetic code or `codon_usage.MIT.table` for the mitochondrial genetic code) and use option `-codon_usage [file]`. Please note that the sum of codon frequencies for any amino acid should always be 1. If you apply model1 without providing your own codon frequencies, codon frequencies will be taken from each input file separately and applied to input sequences of the corresponding file. This way, when providing a complete set of coding sequences of a species in one input file, PACKEIS will use the species specific codon usage. Codon usage deduced from codon frequencies in input files will be saved to the output folder.

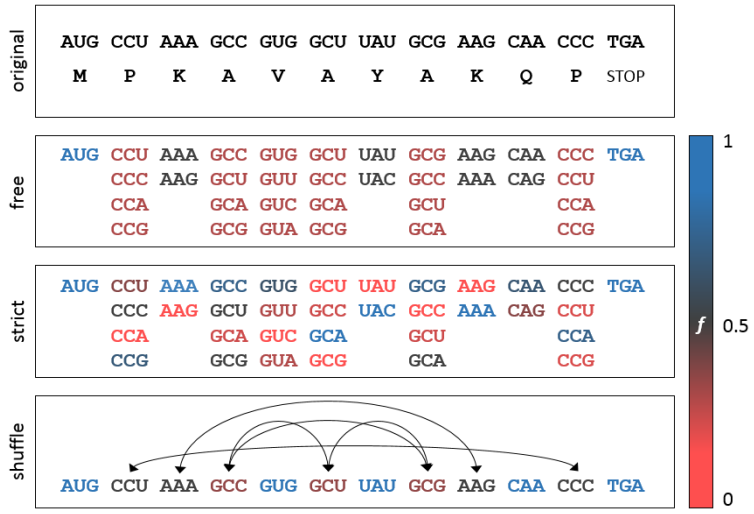

Figure 1. PACKEIS uses different models to generate alternative ORFs based on the original ORF. Colors refer to the probability for a specific codon to be placed at a given position. In model0 (free) codons for a specific amino acid have equal probabilities. In model1 (strict) the probabilities are derived from the global codon usage of the species in question. In model2 (shuffle), present codons are shuffled. Note that in the above example V can only be encoded by GUG when applying model2 since no alternative V-codons are present in the peptide.

3. Output files

Log file

All output files will be copied to the folder `PACKEIS.[date of execution]` (e.g. `PACKEIS.Tue-Jan-15-14h19m05s-2019`). PACKEIS creates a log file that will look like this:

```
Time: Tue Jan 15 14:56:37 2019
Model: 1
Genetic code: Standard
Alternative sequences: 100
Threads: 1
Min. length of a input seq. to run RNAplfold instead of RNAfold: 100nt

SPEC: input_file_1 -> 1
TR: FASTA_header_1 -> 1
TR: FASTA_header_2 -> 2
TR: FASTA_header_3 -> 3
SPEC: input_file_2 -> 2
TR: FASTA_header_1 -> 1
TR: FASTA_header_2 -> 2
TR: FASTA_header_3 -> 3
[...]
```

PACKEIS assigns a numerical ID to each transcript of a given input file. Result files in the subdirectory `/raw` will be named according to the input file base name and transcript ID (FASTA headers may contain characters that are not permitted for file names), e.g. `raw/input_file_1.1.results`. Any errors that occur during the execution of PACKEIS will appear at the bottom of this file.

Codon usage tables

When applying model1 without providing specified codon frequencies, PACKEIS will deduce codon frequencies from input files(s) and output a codon usage table for each input file that will look like this:

```
A->GCU:0.43
A->GCC:0.2
A->GCA:0.29
A->GCG:0.08
C->UGU:0.59375
C->UGC:0.40625
[...]
```

The codon usage tables will be named according to the input file name, e.g. `codon_usage.input_file_1.table`. These files can be used (and edited manually) to specify codon frequencies in later PACKEIS runs.

DBF-scores

DBF-scores for all input sequences will be saved in DBFscores.txt. The file will look like this:

```
input_file       transcript         DBF-score
input_file_1     FASTA_header_1     0.04
input_file_1     FASTA_header_2     0.09
input_file_1     FASTA_header_3     0.49
input_file_2     FASTA_header_1     0.87
input_file_2     FASTA_header_2     0.39
input_file_2     FASTA_header_3     0.21
[...]
```

Raw result files

Foreach input sequence PACKEIS generates a raw result file within the `/raw` subdirectory that lists the number of paired bases in the oORF and all aORFs. These files will be named according to the input file base name together with a numerical sequence ID, e.g. `raw/input_file_1.1.results`. Check the log file to find the sequence ID which corresponds to a transcript name (FASTA header in input sequence) of interest. When using the option `-keep_aCDS`, PACKEIS will additionally save the sequence of each aORF. Note that this might require a lot of disk space, roughly [size of all input sequences] x [number of aORFs (default=100)]. The raw result file will look like this:

```
orig_CDS   586   AUGUUUGUAUACAUUUAUAUCAUCCUU [...]
altCDS_1   547   AUGUUUGUGUACAUCUACAUCAUUCUA [...]
altCDS_1   547   AUGUUCGUCUACAUAUAUAUUAUUCUC [...]
altCDS_1   547   AUGUUUGUCUACAUUUAUAUUAUCCUU [...]
[...]
```

If you have a gene of interest and you are looking for an alternative coding sequence which has an extraordinary high/low amount of paired bases, you can run PACKEIS with one input sequence (gene of interest) and use a high number of aORFs to be generated (e.g. 10000) together with the option `-keep_aCDS` like this:

```
perl PACKEIS.pl –i AUGTCCAGTTAACGATTTACGTAGCCCTAGTTAGCTAGCTAC –keep_aCDS –a 10000
```

Use model2 (shuffle) if you do not want to alter the internal codon usage or GC-content. You can then copy the data within the corresponding raw result file into a spread sheet and sort it according to the number of paired bases in order to get alternative coding sequences with extremely high/low number of paired bases. Think of using all available CPUs with the option `-threads [number of CPUs]`.

Distribution of DBF-scores

In the absence of selection that act on the secondary structure (structural selection), all DBF-scores are equally likely. Therefore, presence of structural selection can only be seriously attested using a large number of input sequences and subsequent testing for an enrichment of extreme DBF-scores. To facilitate these tests, PACKEIS outputs a file named `DBFscores_distribution.txt` that shows the distribution of DBF-scores from sequences of a given input file.

Traits of sequences with specific DBF-score

PACKEIS will output one table per input file (e.g. `traits-per-DBFscore.input_file_1.table`) that lists codon frequencies, amino acid frequencies and GC-content of sequences with a specific DBF-score. This allows to e.g. check for over-/under-representation of codons or amino acids in transcripts with high/low DBF-scores. Codon frequencies refer to frequencies of codons per amino acid, i.e. frequencies for UGU and UGC (both coding for cysteine) will sum to 1. Amino acid frequencies refer to total amino acid frequencies along all input sequences within the given input file, i.e. all 20 amino acid frequencies for a given DBF-score will sum to 1. GC-content is calculated i) based on the average GC-content of all input sequences within the given input file and ii) based on concatenated input sequences of a given input file.

4. Command line options

```
-input OR -i [file or string]   Name of input file(s) in FASTA format.
                                Alternatively type sequence as plain text.
-model OR -m [0,1,2]            Model to be used for reconstruction of
                                alternative coding sequences. 0=free,
                                1=strict, 2=shuffle. Default=2.
-codon_usage OR -c [file]       File that contains user-defined codon usage.
                                Only relevant (but optionally) in combination
                                with -m 1.
-mt                             Use mitochondrial genetic code. Default=off.
-altseqs OR -a [int]            Number of alternative coding sequences to be
                                created per input sequence. Default=100.
-threshold OR -l [int]          Minimum length of a input sequence [nt] to
                                run RNAplfold (local folding). Otherwise use
                                RNAfold (global folding). Default=100.
-keep_aCDS OR -k                Save alternative coding sequences for each
                                input sequence. Default=off.
-threads OR -t [int]            Number of parallel threads. Default=1.
-silent OR -s                   Less output to STDOUT.
-help OR -h                     Prints this screen.
```

5. Citation

A publication of the PACKEIS software in a peer-reviewed journal is pending. A preprint of the article is available at bioRxiv.

Daniel Gebert, Julia Jehn and David Rosenkranz. 2019. Widespread selection for high and low secondary structure in coding sequences across all domains of life. *Unpublished* [preprint available on www.biorxiv.org, doi: https://doi.org/10.1101/524538].