# Package 'coil'

October 12, 2022

**Type** Package

**Title** Contextualization and Evaluation of COI-5P Barcode Data

**Version** 1.2.3

**Author** Cameron M. Nugent

**Maintainer** Cameron M. Nugent <nugentc@uoguelph.ca>

**Description** Designed for the cleaning, contextualization and assessment of cytochrome c oxidase I DNA barcode data (COI-5P, or the five prime portion of COI). It contains functions for placing COI-5P barcode sequences into a common reading frame, translating DNA sequences to amino acids and for assessing the likelihood that a given barcode sequence includes an insertion or deletion error. The error assessment relies on the comparison of input sequences against nucleotide and amino acid profile hidden Markov models (PHMMs) (for details see Durbin et al. 1998, ISBN: 9780521629713) trained on a taxonomically diverse set of reference sequences. The functions are provided as a complete pipeline and are also available individually for efficient and targeted analysis of barcode data.

**License** GPL-3

**Encoding** UTF-8

**LazyData** true

**Depends** R(>= 3.0.0)

**Imports** ape, aphid, seqinr

**Suggests** knitr, rmarkdown, testthat

**VignetteBuilder** knitr

**RoxygenNote** 7.1.1

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2021-04-20 18:20:03 UTC

## R topics documented:

---

`aa_coi_PHMM` *Amino acid profile hidden Markov model for coi5p.*

---

## Description

This model is stored in the coi5p package and was trained on a representative sample of the barcode of life database (http://www.boldsystems.org/index.php).

## Usage

```
aa_coi_PHMM
```

## Format

An object of class `PHMM` of length 14.

---

`censored_translation` *Censored Translation of a DNA string.*

---

## Description

Translate a DNA sequence using the censored translation table. This translates codons for which the amino acids are unambiguous across mitochondrial genetic codes across the animal kingdom and does not translate those for which the amino acid varies, but rather outputs a ? in the string.

## Usage

```
censored_translation(dna_str, reading_frame = 1)
```

### Arguments

| | |
|---|---|
| dna_str | The DNA string to be translated. |
| reading_frame | Set the reading frame of the sequence. reading_frame = 1 (default) means the first bp in the string is the start of the first codon, can pass: 1, 2, or 3. i.e. reading_frame = 2 indicates that the second bp in the string is the start of the first codon. |

### Examples

```
#translate a string of DNA:
censored_translation(example_nt_string)
#manually override the reading frame:
censored_translation(example_nt_string, reading_frame = 2)
```

---

## coi5p *Build a coi5p object from a DNA sequence string.*

---

### Description

Build a coi5p object from a DNA sequence string.

### Usage

```
coi5p(x = character(), name = character())
```

### Arguments

| | |
|---|---|
| x | A nucleotide string. Valid characters within the nucleotide string are: "a", "t", "g", "c", "-" and "n". coil treats both '-' and 'n' characters as placeholder nucleotides when comparing to the PHMM. The nucleotide string can be input as upper case, but will be automatically converted to lower case. |
| name | An optional character string that serves as the identifier for the sequence. |

### Value

An object of class "coi5p"

### Examples

```
#build an unnamed coi5p object
dat = coi5p(example_nt_string)
#build a named coi5p sequence
dat = coi5p(example_nt_string, name = "example_seq1")
#available components in the coi5p object:
dat$raw
dat$name
```

`coi5p_pipe` *Run the entire coi5p pipeline for an input sequence.*

## Description

This function will take a raw DNA sequence string and run each of the coi5p methods in turn (coi5p, frame, translate, indel_check). Note that if you are not interested in all components of the output (i.e. only want sequences set in frame reading or translated), then the coi5p analysis functions can be called individually to avoid unnecessary computation.

## Usage

```
coi5p_pipe(
  x,
  ...,
  name = character(),
  trans_table = 0,
  frame_offset = 0,
  triple_translate = FALSE,
  nt_PHMM = coil::nt_coi_PHMM,
  aa_PHMM = coil::aa_coi_PHMM,
  indel_threshold = -358.88
)
```

## Arguments

| | |
|---|---|
| `x` | A nucleotide string. Valid characters within the nucleotide string are: 'a', 't', 'g', 'c', '-', and 'n'. The nucleotide string can be input as upper case, but will be automatically converted to lower case. |
| `...` | Additional arguments to be passed between methods. |
| `name` | An optional character string. Identifier for the sequence. |
| `trans_table` | The translation table to use for translating from nucleotides to amino acids. Default is 0, which indicates that censored translation should be performed. If the taxonomy of the sample is known, use the function which_trans_table() to determine the translation table to use. |
| `frame_offset` | The offset to the reading frame to be applied for translation. By default the offset is zero, so the first character in the framed sequence is considered the first nucleotide of the first codon. Passing frame_offset = 1 would make the second character in the framed sequence the first nucleotide of the first codon. |
| `triple_translate` | Optional argument indicating if the translation of sequences should be tested in all three forward reading frames. The reading frame with the most likely amino acid PHMM score is returned. This will decrease the rate of sequencing framing errors, at the cost of increased processing time. Note this argument will overrule any passed frame_offset value (all options tried). Default is False. |

`nt_PHMM` The profile hidden Markov model against which the raw sequence should be compared in the framing step. Default is the full COI-5P nucleotide PHMM (nt_coi_PHMM).

`aa_PHMM` The profile hidden Markov model against which the translated amino acid sequence should be compared in the indel_check step. Default is the full COI-5P amino acid PHMM (aa_coi_PHMM).

`indel_threshold`
The log likelihood threshold used to assess whether or not sequences are likely to contain an indel. Default is -358.88. Values lower than this will be classified as likely to contain an indel and values higher will be classified as not likely to contain an indel. For recommendations on selecting a indel_threshold value, consult: Nugent et al. 2019 (doi: https://doi.org/10.1101/2019.12.12.865014).

## Value

An object of class `"coi5p"`

## See Also

`coi5p`

`frame`

`translate`

`indel_check`

`which_trans_table`

`subsetPHMM`

## Examples

```
dat = coi5p_pipe(example_nt_string)
#full coi5p object can then be printed
dat
#components of output coi5p object can be called individually:
dat$raw    #raw input sequence
dat$name   #name that was passed
dat$framed #sequence in common reading frame
dat$aaSeq  #sequence translated to amino acids (censored)
dat$indel_likely #whether an insertion or deletion likely exists in the sequence
dat$stop_codons #whether or not there are stop codons in the amino acid sequence
dat = coi5p_pipe(example_nt_string , trans_table = 5)
dat$aaSeq #sequence translated to amino acids using designated translation table
```

coil **coil**: *evaluation of COI-5P barcode data*

## Description

**coil** is an R package designed for the cleaning, contextualization and assessment of cytochrome c oxidase I DNA barcode data (COI-5P, or the five prime portion of COI). It contains functions for placing COI-5P barcode sequences into a common reading frame, translating DNA sequences to amino acids and for assessing the likelihood that a given barcode sequence includes an insertion or deletion error. These functions are provided as a single function analysis pipeline and are also available individually for efficient and targeted analysis of barcode data.

## Details

**coil** is built around the custom 'coi5p' object, which takes a COI-5P DNA barcode sequence as input. The package contains functions for: setting a sequence in reading frame, translating the sequence to amino acids and checking the sequence for evidence of insertion or deletion errors

## Functions

- `coi5p_pipe` Run the entire coi5p analysis pipeline for an input sequence.
- `coi5p` Builds a coi5p class object.
- `frame` Sets the sequence into a common reading frame
- `which_trans_table` Suggests a translation table for a taxonomic designation.
- `censored_translation` Conducts translation, but ambigious codons are translated to placeholders.
- `translate` Translate a DNA sequence to amino acids.
- `indel_check` Check to see if an insertion or deletion error is likely.

## Data

- `example_nt_string` String of DNA barcode data used in the package documentation's examples.
- `example_barcode_data` A dataframe of coi5p barcode data, demonstrating different example cases.

## Author(s)

Cameron M. Nugent

## example_barcode_data *Example barcode data.*

### Description

A nine line dataframe of coi5p barcode data with the following columns:

### Usage

```
example_barcode_data
```

### Format

An object of class `data.frame` with 9 rows and 5 columns.

### Details

id - the unique identifier for the sample

genetic_code - the genetic code for translation of the sample (features NA for unknowns)

taxa - a taxonomic designation associated with the sample

sequence - the DNA sequence associated with the sample

notes - notes on the sequence structure

## example_nt_string *Example coi5p DNA sequence string*

### Description

This string of barcode data is used in the package documentation's examples and within the vignette demonstrating how to use the package.

### Usage

```
example_nt_string
```

### Format

An object of class `character` of length 1.

`flatten_coi5p` *Flatten a list of coi5p output objects into a dataframe.*

## Description

This helper function is designed to act upon a list of coi5p objects and extract the object components that the user requires.

## Usage

```
flatten_coi5p(x, keep_cols = "all")
```

## Arguments

| | |
|---|---|
| `x` | A list of coi5p objects. |
| `keep_cols` | The name of a coi5p object component, or a vector of components that should be turned into dataframe columns. Available components are: name, raw, framed, aaSeq, aaScore, indel_likely, stop_codons. |

## Value

A dataframe with the coi5p object information flattened into columns.

## See Also

`coi5p_pipe`

## Examples

```
#create a list of coi5p objects
coi_output = lapply(example_barcode_data$sequence, function(x){
    coi5p_pipe(x)
  })
#flatten the list into a dataframe
coi_df = flatten_coi5p(coi_output)
#extract only a single column
coi_framed = flatten_coi5p(coi_output, keep_cols = "framed")
#or subset multiple columns
coi_framed = flatten_coi5p(coi_output, keep_cols = c("framed", "aaSeq"))
```

`frame` *Take a coi5p sequence and place it in reading frame.*

## Description

Take a coi5p sequence and place it in reading frame.

## Usage

```
frame(x, ...)

## S3 method for class 'coi5p'
frame(x, ..., nt_PHMM = nt_coi_PHMM)
```

## Arguments

| | |
|---|---|
| `x` | A coi5p class object. |
| `...` | Additional arguments to be passed between methods. |
| `nt_PHMM` | The profile hidden Markov model against which the raw sequence should be compared. Default is the full COI-5P nucleotide PHMM (nt_coi_PHMM). |

## Details

This function compares the raw sequence against the nucleotide PHMM using the Viterbi algorithm (for details see Durbin et al. 1998, ISBN: 9780521629713). The path of hidden states produced by the comparison is used to establish the reading frame of the sequence. If leading insert states are present, the front of the sequence is trimmed to the first continuous set of match states and the sequence is re-compared to the nucleotide PHMM. This is done because spurious or outlier matches early in the sequence can lead to incorrect establishment of the reading frame. Realigning only the truncated version of the sequence to the PHMM improves correct reading frame establishment, although this can also result in the loss of a few bp of true barcode sequence on the peripherals of the sequence.

## Value

An object of class `"coi5p"`

## See Also

`coi5p`

`subsetPHMM`

## Examples

```
#previously run function:
dat = coi5p(example_nt_string)

dat = frame(dat)

#additional components in output coi5p object:
dat$framed
```

---

`indel_check` *Check if a coi5p sequence likely contains an error.*

---

## Description

Check if a coi5p sequence likely contains an error.

## Usage

```
indel_check(x, ...)

## S3 method for class 'coi5p'
indel_check(x, ..., indel_threshold = -358.88, aa_PHMM = aa_coi_PHMM)
```

## Arguments

| | |
|---|---|
| `x` | A coi5p class object for which frame() and translate() have been run. |
| `...` | Additional arguments to be passed between methods. |
| `indel_threshold` | The log likelihood threshold used to assess whether or not sequences are likely to contain an indel. Default is -358.88. Values lower than this will be classified as likely to contain an indel and values higher will be classified as not likely to contain an indel. |
| `aa_PHMM` | The profile hidden Markov model against which the translated amino acid sequence should be compared. Default is the full COI-5P amino acid PHMM (aa_coi_PHMM). |

## Details

The indel check function analyzes the framed and translated DNA sequences in two ways in order to allow users to make an informed decision about whether or not a DNA sequence contains a frameshift error. This test is designed to detect insertion or deletion errors resulting from technical errors in DNA sequencing, but can in some instances identify biological contaminants (i.e. if the contaminant sequence uses a different genetic code than the target, or if the contaminants are things such as pseudogenes that possess sequences that are highly divergent from animal COI-5P sequences).

The two tests performed are: (1) a query for stop codons in the amino acid sequence and (2) an evaluation of the log likelihood value resulting from the comparison of the framed coi5p amino

acid sequence against the COI-5P amino acid PHMM. The default likelihood value for identifying a sequence is likely erroneous is -358.88. Sequences with likelihood values lower than this will receive an indel_likely value of TRUE. The threshold of -358.88 was experimentally determined to be the optimal likelihood threshold for separating of full-length sequences with and without errors when the censored translation option is used. Sequences will have higher likelihood values when a specific genetic code is used. Sequences will have lower likelihood values when they are not complete barcode sequences (i.e. <500bp in length). For these reasons the likelihood threshold is not a specific value but a parameter that can be altered based on the type of translation and length of the sequences. Below are experimentally determined suggested values for different size and translation table combinations.

Short barcode sequences, known genetic code: indel_threshold = -354.44

Short barcode sequences, unknown genetic code: indel_threshold = -440.24

Full length barcode sequences, known genetic code: indel_threshold = -246.20

Full length barcode sequences, unknown genetic code: indel_threshold = -358.88

Source: Nugent et al. 2019 (doi: https://doi.org/10.1101/2019.12.12.865014).

## Value

An object of class `"coi5p"`

## See Also

`coi5p`

`frame`

`translate`

`subsetPHMM`

## Examples

```
#previously run functions:
dat = coi5p(example_nt_string)
dat = frame(dat)
dat = translate(dat)
#current function
dat = indel_check(dat)
#with custom indel threshold
dat = indel_check(dat, indel_threshold = -400)
#additional components in output coi5p object:
dat$stop_codons #Boolean - Indicates if there are stop codons in the amino acid sequence.
dat$indel_likely #Boolean - Indicates if the likelihood score below the specified indel_threshold.
dat$aaScore #view the amino acid log likelihood score
```

## nt_coi_PHMM *Nucleotide profile hidden Markov model for coi5p.*

### Description

This model is stored in the coi5p package and was trained on a representative sample of the barcode of life database (http://www.boldsystems.org/index.php).

### Usage

```
nt_coi_PHMM
```

### Format

An object of class PHMM of length 14.

## subsetPHMM *Subset an existing PHMM.*

### Description

The subsetPHMM function allows an existing PHMM to be subset by profile position. This eliminates the need for the training of an additional, smaller model if query sequences should be compared to only a subsection of an existing PHMM. The nt_coi_PHMM and aa_coi_PHMM can therefore be subset using this function to constrain coil's framing and error evaluation to a subset of the COI-5P region

### Usage

```
subsetPHMM(x, start, end)
```

### Arguments

| | |
|---|---|
| x | an object of class "PHMM" to be subset. |
| start | The first PHMM position to be inclused in the output PHMM. |
| end | The last PHMM position to be included in the output PHMM. |

### Value

an object of class "PHMM"

### See Also

derivePHMM

## Examples

```
## subset positions 2-100 of the COI-5P PHMM
short_nt_PHMM <- subsetPHMM(nt_coi_PHMM, 2, 100)
```

---

`translate` *Translate a coi5p sequence.*

---

## Description

Translate a coi5p sequence.

## Usage

```
translate(x, ...)

## S3 method for class 'coi5p'
translate(x, ..., trans_table = 0, frame_offset = 0)
```

## Arguments

| | |
|---|---|
| `x` | A coi5p class object for which frame() has been run. |
| `...` | Additional arguments to be passed between methods. |
| `trans_table` | The translation table to use for translating from nucleotides to amino acids. Default is 0, which indicates that censored translation should be performed. If the taxonomy of the sample is known, use the function which_trans_table() to determine the translation table to use. |
| `frame_offset` | The offset to the reading frame to be applied for translation. By default the offset is zero, so the first character in the framed sequence is considered the first nucleotide of the first codon. Passing frame_offset = 1 would offset the sequence by one and therefore make the second character in the framed sequence the first nucleotide of the first codon. |

## Details

The translate function allows for the translation of framed sequences from nucleotides to amino acids, both in instances when the correct genetic code corresponding to a sequence is known, and in instances when taxonomic information is unavailable or unreliable.

## Value

An object of class `"coi5p"`

## See Also

`coi5p`

`frame`

`which_trans_table`

## Examples

```
#previously run functions:
dat = coi5p(example_nt_string )
dat = frame(dat)
#translate when the translation table is not known:
dat = translate(dat)
#translate when the translation table is known:
dat = translate(dat, trans_table = 5)
#additional components in output coi5p object:
dat$aaSeq
```

---

# trans_df *Data frame containing the translation table recommendation.*

---

## Description

Data frame containing the translation table recommendation.

## Usage

```
trans_df
```

## Format

An object of class `data.frame` with 4609 rows and 2 columns.

---

# which_trans_table *Determine the translation table to use for a given taxonomic group.*

---

## Description

Recommends which translation table to use if taxonomic data are available. The recommendations are based on the translation tables reported for different taxonomic classifications on the Barcode of Life Data Systems (BOLD - http://www.boldsystems.org/index.php).

## Usage

```
which_trans_table(x)
```

## Arguments

| | |
|---|---|
| x | A taxonomic designation (allowed ranks: family, order, class, phylum). |

## Details

If which_trans_table is unable to identify a translation table, more information on translation tables can be found here: https://www.ncbi.nlm.nih.gov/Taxonomy/Utils/wprintgc.cgi

## Value

An integer indicating the correct translation table.

## Examples

```
which_trans_table("Chordata") #phylum
which_trans_table("Actinopterygii") #class
which_trans_table("Akentrogonida")  #order
which_trans_table("Hydrobiidae") #family
```

# Index