Relative tRNA concentrations data

Dmytro Strunin

3/24/2020

Input data

Data exported to .csv format for further analysis.

```
codon_tab <- read_csv('./dat/ftb/codons_rel_freq.csv')
```

```
## Parsed with column specification:
## cols(
##   probe = col_character(),
##   anticodon = col_character(),
##   codon = col_character(),
##   freq = col_double(),
##   sd = col_double()
## )
```

```
head(codon_tab) %>%
  kable(digits = 2, caption = 'Raw data (only first 6 rows are shown)') %>%
  kable_styling()
```

Data cleaning

- remove digits from `anticodon`/`codon` strings (those are still in the `probe` variable);
- split `anticodon` by `-` to separate it into *amino acid* (`aa`) and *codon/anticodon* variables;
- remove amino acid identifier from `codon` variable;
- work with `codon` variable, to make all alternative codons from records like `CGC/T/C/G`:
  - Extract first two letters from each codon string;
  - Extract the part from third letter up to the end;
    - Split it by `/` into three letters;
  - Combine first two letters and all the parts obtained in previous step one by one;
- unnest `codon` variable (it is a list right now);

Table 1: Raw data (only first 6 rows are shown)

probe	anticodon	codon	freq	sd
PaeARGACG	Arg-ACG	Arg-CGT/C/G	1.24	0.22
PaeARGACG2	Arg-ACG-2	Arg-CGT/C/G	2.08	0.01
PaeARGCCG	Arg-CCG	Arg-CGC/T	1.28	0.24
PaeARGCCG214	Arg-CCG-214	Arg-CGC/T	1.26	0.17
PaeARGCCT	Arg-CCU	Arg-AGG	1.10	0.13
PaeARGTCT	Arg-UCU	Arg-AGA	1.61	0.11

Table 2: Initial cleaning (only first 6 rows are shown)

probe	aa	anticodon	codon	freq	sd
PaeARGACG	Arg	ACG	CGT	1.24	0.22
PaeARGACG	Arg	ACG	CGC	1.24	0.22
PaeARGACG	Arg	ACG	CGG	1.24	0.22
PaeARGACG2	Arg	ACG	CGT	2.08	0.01
PaeARGACG2	Arg	ACG	CGC	2.08	0.01
PaeARGACG2	Arg	ACG	CGG	2.08	0.01

```
codon_tab <- codon_tab %>%
  mutate_at(vars(anticodon, codon), ~ gsub('-\\d+$', '', .)) %>%
  tidyr::separate(
    col  = anticodon,
    into = c('aa', 'anticodon'),
    sep  = '-'
  ) %>%
  dplyr::mutate(codon = gsub('^\\w+-(.+)$', '\\1', codon)) %>%
  dplyr::mutate(
    codon   = purrr::map(
      codon,
      ~ paste0(
        substr(., 1, 2),
        unlist(strsplit(substr(., 3, nchar(.)), split = '/'))
      )
    )
  ) %>%
  tidyr::unnest(codon)
```

```
## Warning: Expected 2 pieces. Additional pieces discarded in 1 rows [33].
```

```
head(codon_tab) %>%
  kable(digits = 2, caption = 'Initial cleaning (only first 6 rows are shown)') %>%
  kable_styling()
```

Amino acids count

First we can check how many aminoacids are there:

```
codon_tab %>%
  arrange(aa) %>%
  pull(aa) %>%
  table
```

```
## .
##  Ala  Arg  Asn  Asp  Cys fMet  Gln  Glu  Gly  His  Ile  Leu  Lys  Met  Phe  Pro
##    5   12    2    2    1    1    3    1    4    2    3    9    2    1    2    6
##  Sec  Ser  Thr  Trp  Tyr  Val
##    1    8    4    1    2    3
```

It seems fine, at least from the first glance, since there are 22amino acids out of usual 21, and it seems that all of them are there:

```
missing_aa <- AMINO_ACID_CODE[!(AMINO_ACID_CODE %in% codon_tab$aa)]
```

Except abbreviation for special cases: Pyl, Asx, Xle, Glx, Xaa, and with addition of *N-Formylmethionine*.

Codons count

The problem seems to be in codons list. In the table below we can see the count of codons per each amino acid, for standard genetic code:

```
aa_cdns <- data.frame(aa = AMINO_ACID_CODE[GENETIC_CODE], codon = names(GENETIC_CODE))

aa_cdns %>%
  na.omit() %>%            # remove stop codons
  pull(aa) %>%
  table()
```

```
## .
## Ala Arg Asn Asp Cys Gln Glu Gly His Ile Leu Lys Met Phe Pro Ser Thr Trp Tyr Val
##   4   6   2   2   2   2   2   4   2   3   6   2   1   2   4   6   4   1   2   4
```

And comparing it to our data:

```
codon_tab %>%
  arrange(aa) %>%
  pull(aa) %>%
  table()
```

```
## .
##  Ala  Arg  Asn  Asp  Cys fMet  Gln  Glu  Gly  His  Ile  Leu  Lys  Met  Phe  Pro
##    5   12    2    2    1    1    3    1    4    2    3    9    2    1    2    6
##  Sec  Ser  Thr  Trp  Tyr  Val
##    1    8    4    1    2    3
```