[Re] Local alignment statistics - Alignment random sequences

Nathan Brouwer

11/6/2019

Basic simulation experiment workflow

Below I outline the basic steps needed to replicate Table 1 of Altchul and Gish (1996). It assumes that you have produced the table of amino acid frequences from Robinson and Robinson. Be sure to run the associated script prior to runn this.

Preliminaries: Packages / libraries

Two packages are needed for this simulation experiment:

1. Biostrings: for pairwiseAlignment() function
2. extRemes: For calculating parameters related to the Gumbel extreme value distribution.

```
library(Biostrings)
```

Load Robinson AA frequencies

```
robinson.aafreq <- read.csv(file = "robinson_aafreq.csv")
```

Example simulation experiment workflow

The following code walks through the basic workflow of the simulation for a random sequence of length n = 191. THis is a single iteration of a single subexperiment. THis would need to be repeated 10000 time to produce the first row of Table 1.

Random sequence 1

Create first sequence

```
seq1 <- sample(x = robinson.aafreq$aa1,
                  size = 191,
                  replace = TRUE,
                  prob = robinson.aafreq$aa.freq)
```

Turn sequence into string

```
seq1 <- paste(seq1, sep = "",
                  collapse = "")
```

Random sequence 2

```
# create sequence
seq2 <- sample(x = robinson.aafreq$aa1,
                  size = 2981,
                  replace = TRUE,
```

```
                  prob = robinson.aafreq$aa.freq)

# turn into a string
seq2 <- paste(seq2, sep = "",collapse = "")
```

Align simulated sequences

pairwiseAlignment() from Biostrings is used to do a local alignment of the two random sequences.

```
alignment.i <- pairwiseAlignment(pattern = seq1,
                                 subject = seq2,
                    type               = "local",
                    substitutionMatrix = "BLOSUM62",
                    gapOpening         = 12,
                    gapExtension       = 1)
```

Look at the alignment; note that for long alignments this can be abridged with a "..." in the middle

```
alignment.i
```

```
## Local PairwiseAlignmentsSingleSubject (1 of 1)
## pattern:  [131] GDPFVTRK-KALANLNRLDLTLSKTVYDAQGF
## subject: [2347] GEPGVRLQYAALTKINTMSIQLSWTL-EAAGF
## score: 33
```

Process alignment

Once and alignment has been generated it needs to be processed

Alignment score (s): Score based on specified scoring matrix and indel parameters. This is accessed using the function score().

```
score.i         <- score(alignment.i)
score.i
```

```
## [1] 33
```

Alignment length: Extract full alignment sequence as a string with Biostrings::toString()

```
alignment.seq.i <- toString(alignment.i)
alignment.seq.i
```

```
## [1] "GDPFVTRK-KALANLNRLDLTLSKTVYDAQGF"
```

Determine the actual length of the alignment with base R nchar() function

```
length.i        <- nchar(alignment.seq.i)
length.i
```

```
## [1] 32
```

The score and length then need to be stored in a dataframe and the above process repeated several thousand times.