Function to simulate random macromolecular sequence

Nathan Brouwer

10/29/2019

A function to simulate a random sequence

Version 1

```
r_molec_seq_vs1 <- function(units,
                        prob,
                        length){
  # create sequnce
  ## "units" = molecular subunits to sample from
  ### can be DNA (ATCG), mRNA, amino acids, etc
  ## "length" = length of squence to generate
  ## "prob" = probablity of sampling an element of "units"
  seq.n.i <- sample(x = units,
                    size = length,
                    replace = TRUE,
                    prob = prob)

  # convert to charater string
  seq.n.i <- paste(seq.n.i,sep = "",collapse = "")

  # return result
  return(seq.n.i)

}
```

Test the function. There's no defaults so it

```
r_molec_seq_vs1()
```

```
r_molec_seq_vs1(units =c("A","T","C","G"),
                prob = c(0.25,0.25,0.25,0.25),
                length = 10)
```

```
## [1] "AAAACCGACC"
```

Test with real data, the Robinson and Robinson amino acid frequencies

```
robinson.aafreq <- read.csv("robinson_aafreq.csv")
r_molec_seq_vs1(units = robinson.aafreq$aa1,
                prob = robinson.aafreq$aa.freq,
                length = 10)
```

```
## [1] "FWNSSPTGVP"
```

Version 2

Add defaults for units and prob

```
r_molec_seq_vs2 <- function(units = c("A","T","C","G"),
                            prob  = c(0.25,0.25,0.25,0.25),
                            length){

  seq.n.i <- sample(x = units,
                    size = length,
                    replace = TRUE,
                    prob = prob)

  seq.n.i <- paste(seq.n.i,sep = "",collapse = "")

  return(seq.n.i)

}
```

Now it works even if all we give it is a length

```
r_molec_seq_vs2(length = 100)
```

```
## [1] "GGGGAGAGGACCCTCTGCATCGAAGCGTATACTTACTCTCCAGAAATGACTGATAGCATACGCTGGTTAACGACCCACTAGTGGCTTCTAGTTATA
```

Version 3

Now we'll give it a deafult for length

```
r_molec_seq_vs3 <- function(units = c("A","T","C","G"),
                            prob  = c(0.25,0.25,0.25,0.25),
                            length = 100){

  seq.n.i <- sample(x = units,
                    size = length,
                    replace = TRUE,
                    prob = prob)

  seq.n.i <- paste(seq.n.i,sep = "",collapse = "")

  return(seq.n.i)

}
```

Now it works even if the parentheses are empty

```
r_molec_seq_vs3()
```

```
## [1] "CCGTCAAACGGGGCAAGGGAATGATCACAAAGTCATGAGTGGAAATAGGTGGGAGGATTCAACTGGAAATACCCGCGCCGACTAGGTTTCGGGAAT
```

Version 4: Adding conditions and warnigns

This version tests whether the function in being run with the defaults, and throws a warning if that's true. The defaults are meant just for testing the function, and if someone runs the function with the defaults it might be that they forgot to change them.

```
r_molec_seq_vs4 <- function(units = c("A","T","C","G"),
                            prob  = c(0.25,0.25,0.25,0.25),
                            length = 100){
```

```
  if(all(units  == c("A","T","C","G")) == TRUE &
     all(prob   == c(0.25,0.25,0.25,0.25)) == TRUE &
     length == 100){
    warning("Note: all parameters set to defaults.  Did you want to change something?")
     }

  seq.n.i <- sample(x = units,
                    size = length,
                    replace = TRUE,
                    prob = prob)

  seq.n.i <- paste(seq.n.i,sep = "",collapse = "")

  return(seq.n.i)

}
```

This throws a warning

```
r_molec_seq_vs4()
```

```
## Warning in r_molec_seq_vs4(): Note: all parameters set to defaults. Did you
## want to change something?
```

```
## [1] "CGGGATGCAGAGCGAGGGAGTGGACGACCGGGAGCTATCGTGGACGAAGTGCAGTCTACACCTCATTGCGACCCATGGTGAACAGATCTGCTGCC
```

This doesnt

```
r_molec_seq_vs4(prob = c(0.3,0.3,0.2,0.2))
```

```
## [1] "CGTAGTACAATGTAATGTTCGGTAAACTATTGGAGTGGCGCGCCTGGTTCAACCCCATCATTGAACGTAGACGAACAAAATCAAAATAATCCTAC
```

Version 5

Here I've added an additional condition to ask whether the user wants to return a string or a vetor.

```
r_molec_seq_vs5 <- function(units = c("A","T","C","G"),
                            prob  = c(0.25,0.25,0.25,0.25),
                            length = 100,
                             as.string = TRUE){

  if(all(units  == c("A","T","C","G")) == TRUE &
     all(prob   == c(0.25,0.25,0.25,0.25)) == TRUE &
     length == 100){
    warning("Note: all parameters set to defaults.  Did you want to change something?")
     }

  seq.n.i <- sample(x = units,
                    size = length,
                    replace = TRUE,
                    prob = prob)

  if(as.string == TRUE){
    seq.n.i <- paste(seq.n.i,sep = "",collapse = "")
```

```
  }

  return(seq.n.i)

}
```

Return as a vectro

```
r_molec_seq_vs5(prob = c(0.3,0.3,0.2,0.2),
                as.string = F,
                length = 10)
```

```
##  [1] "G" "G" "T" "A" "T" "A" "A" "T" "G" "T"
```

Return as a string

```
r_molec_seq_vs5(prob = c(0.3,0.3,0.2,0.2),
                as.string = T,
                length = 10)
```

```
## [1] "TGATCAGATT"
```

The default is to return a string, so as.string = T is option

```
r_molec_seq_vs5(prob = c(0.3,0.3,0.2,0.2),
                length = 10)
```

```
## [1] "TCAATGAGGG"
```