Custom array sequence generation

Tomas Bjorklund

Thu Oct 29 09:54:57 2020

This script generates all unique AA sequences for the CustomArray production

```
suppressPackageStartupMessages(library(knitr))
```

Loading source files

```
source(file.path("functions", "AAtoDNA.R"))
source(file.path("functions", "GeneCodon.R"))
# Override the GeneCodon function with local version containing human codons
unlockBinding("GeneCodon", as.environment("package:GeneGA"))
assign("GeneCodon", GeneCodon, as.environment("package:GeneGA"))

allSequences <- readFasta("input/DNA-lib_RetrogradeTransport.fasta")
AAlist <- data.frame(Class = character(), Family = character(), Strain = character(),
    Note = character(), Number = character(), Name = character(), AAfragment = character(),
    stringsAsFactors = FALSE)

# allSequences <- allSequences[124:129] #Debug row
```

Generation of AA table for the selected proteins

```
strt <- Sys.time()
for (i in 1:length(allSequences)) {
    thisID <- as.character(ShortRead::id(allSequences[i]))
    thisSeq <- sread(allSequences[i])
    thisAA <- Biostrings::translate(thisSeq, genetic.code = GENETIC_CODE, if.fuzzy.codon = "solve")
    AAlist[i, c("Class", "Family", "Strain", "Note", "Number", "Name", "AAfragment")] <- c(BBmisc::explode(t
        sep = ","), as.character(thisAA))
}
```

The generateFragments function

```
generateFragments <- function(minLength, maxLength, frequency = 1) {
    # Generate a table to store all AA sequences
    fragList <- data.frame(Class = character(), Family = character(), Strain = character(),
        Note = character(), Number = character(), Name = character(), AAstart = integer(),
        AAstop = integer(), AAfragment = character(), stringsAsFactors = FALSE)


    makeAllFrags <- function(k) {
        thisFullAA <- AAlist[k, "AAfragment"]
        count = length(fragList[, 1]) + 1
        for (l in seq(1, (width(thisFullAA) - minLength), frequency)) {
            for (m in (l + minLength - 1):(min(l + maxLength - 1, width(thisFullAA)))) {
```

```
                # Truncate string to the relevant fragment:
                thisFragment <- substr(thisFullAA, l, m)
                # Take away any sequence that starts with a start codon ATG:
                if (substr(thisFragment, 1, 1) != "M") {
                  fragList[count, c("Class", "Family", "Strain", "Note", "Number",
                    "Name", "AAstart", "AAstop", "AAfragment")] <- c(AAlist[k,
                    c("Class", "Family", "Strain", "Note", "Number", "Name")],
                    l, m, thisFragment)
                  ## Inserts the fragment with information into the new data frame
                  count <- count + 1
                }
            }
        }
        return(fragList)
    }

    fragList <- do.call(rbind, mclapply(1:length(AAlist[, 1]), makeAllFrags,
        mc.preschedule = TRUE, mc.cores = detectCores()))

    # Control if any sequences contain non-AA characters and save them into a
    # separate list
    discardList <- fragList[grep("[[:punct:]|X]", fragList[, "AAfragment"]),
        ]

    # Remove any sequence containing non AA characters
    fragList <- fragList[grep("[[:punct:]|X]", fragList[, "AAfragment"], invert = TRUE),
        ]


    # Sort the fragments, find unique strings and count number of duplicates
    sortedFragments <- rev(sort(table(fragList[, "AAfragment"])))

    # Run the AAtoDNA function to convert all AA sequences to human
    # codon-optimized DNA sequences
    row.names(sortedFragments) <- mclapply(row.names(sortedFragments), fullOPT = FALSE,
        species = "hsa", AAtoDNA, mc.preschedule = TRUE, mc.set.seed = TRUE,
        mc.silent = FALSE, mc.cores = detectCores(), mc.cleanup = TRUE)  #

    sortedFragments <- sortedFragments[order(row.names(sortedFragments))]

    return(sortedFragments)
}
```

Execution of the function

```
sortedFragments.14aa <- generateFragments(14, 14, 1)
```

Add the overhangs for amplication PCR and Gibson assembly into the AAV plasmid

```
fivePrime <- tolower("AACCTCCAGAGAGGCAACGCT")
threePrime <- tolower("GCCAGACAAGCAGCTACCGCA")
row.names(sortedFragments.14aa) <- paste(fivePrime, row.names(sortedFragments.14aa),
    threePrime, sep = "")

sortedFragments.14aa.G4S <- generateFragments(14, 14, 3)
sortedFragments.14aa.A5 <- sortedFragments.14aa.G4S
```

Add the overhangs including G4S spacers for amplication PCR and Gibson assembly into the AAV plasmid

```
fivePrime <- tolower("AACCTCCAGAGAGGCAACGGAGGCGGAGGAAGT")
threePrime <- tolower("GGAGGCGGCGGAAGCAGACAAGCAGCTACCGCA")
row.names(sortedFragments.14aa.G4S) <- paste(fivePrime, row.names(sortedFragments.14aa.G4S),
    threePrime, sep = "")
```

Add the overhangs including A5 spacers for amplication PCR and Gibson assembly into the AAV plasmid

```
fivePrime <- tolower("AACCTCCAGAGAGGCAACGCTGCTGCAGCAGCC")
threePrime <- tolower("GCAGCTGCAGCTGCCAGACAAGCAGCTACCGCA")
row.names(sortedFragments.14aa.A5) <- paste(fivePrime, row.names(sortedFragments.14aa.A5),
    threePrime, sep = "")
```

Generate 22aa fragments

```
sortedFragments.22aa <- generateFragments(22, 22, 3)
```

Add the overhangs for amplication PCR and Gibson assembly into the AAV plasmid

```
fivePrime <- tolower("AACCTCCAGAGAGGCAACGCT")
threePrime <- tolower("GCCAGACAAGCAGCTACCGCA")
row.names(sortedFragments.22aa) <- paste(fivePrime, row.names(sortedFragments.22aa),
    threePrime, sep = "")
```

Merge all separate fragment lists into one complete list

```
sortedFragments <- c(sortedFragments.22aa, sortedFragments.14aa, sortedFragments.14aa.A5,
    sortedFragments.14aa.G4S)

print(paste("Number of unique fragments:", length(unique(names(sortedFragments))),
    sep = " "))
```

```
[1] "Number of unique fragments: 92343"
```

```
write.table(c("Sequence", unique(names(sortedFragments))), "data/SortedFragments_all.txt",
    row.names = F, col.names = F, quote = F, sep = "\t")

print(Sys.time() - strt)
```

```
Time difference of 2.674503 mins
```

```
devtools::session_info()
```

```
Session info -------------------------------------------------------------
```

```
 setting  value
 version  R version 3.4.2 (2017-09-28)
 system   x86_64, linux-gnu
 ui       X11
 language (EN)
 collate  en_US.UTF-8
 tz       UTC
 date     2020-10-29
```

```
Packages -----------------------------------------------------------------
```

```
 package             * version  date       source
 acepack               1.4.1    2016-10-29 CRAN (R 3.4.2)
 ade4                  1.7-8    2017-08-09 CRAN (R 3.4.2)
 backports             1.1.1    2017-09-25 CRAN (R 3.4.2)
 base                * 3.4.2    2017-10-06 local
 base64enc             0.1-3    2015-07-28 CRAN (R 3.4.2)
```

```
BBmisc                1.11     2017-03-10 CRAN (R 3.4.2)
Biobase             * 2.36.2   2017-11-29 Bioconductor
BiocGenerics        * 0.22.1   2017-11-29 Bioconductor
BiocParallel        * 1.10.1   2017-11-29 Bioconductor
Biostrings          * 2.44.2   2017-11-29 Bioconductor
bitops                1.0-6    2013-08-17 CRAN (R 3.4.2)
checkmate             1.8.4    2017-09-25 CRAN (R 3.4.2)
cluster               2.0.6    2017-03-16 CRAN (R 3.4.2)
colorspace            1.3-2    2016-12-14 CRAN (R 3.4.2)
compiler              3.4.2    2017-10-06 local
data.table            1.10.4-2 2017-10-12 url
datasets            * 3.4.2    2017-10-06 local
DelayedArray        * 0.2.7    2017-11-29 Bioconductor
devtools            * 1.13.3   2017-08-02 CRAN (R 3.4.2)
digest                0.6.12   2017-01-27 CRAN (R 3.4.2)
evaluate              0.10.1   2017-06-24 CRAN (R 3.4.2)
foreign               0.8-69   2017-06-21 CRAN (R 3.4.2)
formatR               1.5      2017-04-25 CRAN (R 3.4.2)
Formula             * 1.2-2    2017-07-10 CRAN (R 3.4.2)
GeneGA              * 1.26.0   2017-11-29 Bioconductor
GenomeInfoDb        * 1.12.3   2017-11-29 Bioconductor
GenomeInfoDbData      0.99.0   2017-11-29 Bioconductor
GenomicAlignments   * 1.12.2   2017-11-29 Bioconductor
GenomicRanges       * 1.28.6   2017-11-29 Bioconductor
ggplot2             * 2.2.1    2016-12-30 CRAN (R 3.4.2)
graphics            * 3.4.2    2017-10-06 local
grDevices           * 3.4.2    2017-10-06 local
grid                  3.4.2    2017-10-06 local
gridExtra             2.3      2017-09-09 CRAN (R 3.4.2)
gtable                0.2.0    2016-02-26 CRAN (R 3.4.2)
hash                * 2.2.6    2013-02-21 CRAN (R 3.4.2)
Hmisc               * 4.0-3    2017-05-02 CRAN (R 3.4.2)
hms                   0.3      2016-11-22 CRAN (R 3.4.2)
htmlTable             1.9      2017-01-26 CRAN (R 3.4.2)
htmltools             0.3.6    2017-04-28 CRAN (R 3.4.2)
htmlwidgets           0.9      2017-07-10 CRAN (R 3.4.2)
httr                  1.3.1    2017-08-20 CRAN (R 3.4.2)
hwriter               1.3.2    2014-09-10 CRAN (R 3.4.2)
IRanges             * 2.10.5   2017-11-29 Bioconductor
kableExtra          * 0.5.2    2017-09-15 url
knitr               * 1.17     2017-08-10 CRAN (R 3.4.2)
lattice             * 0.20-35  2017-03-25 CRAN (R 3.4.2)
latticeExtra          0.6-28   2016-02-09 CRAN (R 3.4.2)
lazyeval              0.2.0    2016-06-12 CRAN (R 3.4.2)
magrittr              1.5      2014-11-22 CRAN (R 3.4.2)
Matrix                1.2-11   2017-08-21 url
matrixStats         * 0.52.2   2017-04-14 CRAN (R 3.4.2)
memoise               1.1.0    2017-04-21 CRAN (R 3.4.2)
methods             * 3.4.2    2017-10-06 local
munsell               0.4.3    2016-02-13 CRAN (R 3.4.2)
nnet                  7.3-12   2016-02-02 CRAN (R 3.4.2)
parallel            * 3.4.2    2017-10-06 local
plyr                  1.8.4    2016-06-08 CRAN (R 3.4.2)
R6                    2.2.2    2017-06-17 CRAN (R 3.4.2)
RColorBrewer          1.1-2    2014-12-07 CRAN (R 3.4.2)
Rcpp                  0.12.13  2017-09-28 url
RCurl                 1.95-4.8 2016-03-01 CRAN (R 3.4.2)
```

```
readr                   1.1.1    2017-05-16 CRAN (R 3.4.2)
rlang                   0.1.2    2017-08-09 CRAN (R 3.4.2)
rmarkdown               1.6      2017-06-15 url
rpart                   4.1-11   2017-04-21 CRAN (R 3.4.2)
rprojroot               1.2      2017-01-16 CRAN (R 3.4.2)
Rsamtools             * 1.28.0   2017-11-29 Bioconductor
rvest                   0.3.2    2016-06-17 CRAN (R 3.4.2)
S4Vectors             * 0.14.7   2017-11-29 Bioconductor
scales                  0.5.0    2017-08-24 CRAN (R 3.4.2)
seqinr                * 3.4-5    2017-08-01 CRAN (R 3.4.2)
ShortRead             * 1.34.2   2017-11-29 Bioconductor
splines                 3.4.2    2017-10-06 local
stats                 * 3.4.2    2017-10-06 local
stats4                * 3.4.2    2017-10-06 local
stringi                 1.1.5    2017-04-07 url
stringr                 1.2.0    2017-02-18 CRAN (R 3.4.2)
SummarizedExperiment * 1.6.5     2017-11-29 Bioconductor
survival              * 2.41-3   2017-04-04 CRAN (R 3.4.2)
tibble                  1.3.4    2017-08-22 CRAN (R 3.4.2)
tools                   3.4.2    2017-10-06 local
utils                 * 3.4.2    2017-10-06 local
withr                   2.0.0    2017-07-28 url
xml2                    1.1.1    2017-01-24 CRAN (R 3.4.2)
XVector               * 0.16.0   2017-11-29 Bioconductor
yaml                    2.1.14   2016-11-12 CRAN (R 3.4.2)
zlibbioc                1.22.0   2017-11-29 Bioconductor
```