Module 3R

[type your name here in between the quotation marks]

01 June, 2021

Contents

Fork in the Road **1**

Bioconductor **2**
Installation 2
Starting a Bioconductor Session 2

The Samples **3**
Task 1 3

BLOSUM62 **4**
Task 2 4

Pairwise Sequence Alignments **5**
Task 3 5
Task 4 6

Multiple Sequence Alignment **7**
Task 5 7

Wrap-Up **8**
References 8

Fork in the Road

In this bioinformatics adventure, we are going to continue to look amino acid sequences and compute similarity scores using the *BLOSUM62* matrix, but we will take a glimpse at some of the tools that are available in the world of *Bioconductor*. This set of tasks are set after Exercise 3 in the *Sequence Similarity* materials. Load the R markdown file (the file with the .rmd extension) in `RStudio`

In this activity, we will practice with `R` code. Some useful Windows keyboard shortcuts that `R` programmers and students like to use while coding include the following (and code blocks are the gray rectangles where the `R` code goes):

- create a new code block: CTRL-ALT-I
- run a line of code: CTRL-ENTER
- run an entire code block: CTRL-SHIFT-ENTER

(and replace `CTRL` with `CMD` for Mac)

These R markdown files are best viewed in the HTML output (but PDF and Word versions have been supplied for convenience).

Bioconductor

Bioconductor is a collection of `R` packages that has been built by several bioinformatics researchers to perform common calculations in their field. In their own words, "*Bioconductor* provides tools for the analysis and comprehension of high-throughput genomic data. Bioconductor uses the R statistical programming language, and is open source and open development. It has two releases each year, and an active user community.'' They keep the project and code documentation on their website at https://bioconductor.org/

Installation

(This activity assumes that you have already installed `R` and you are advised to also have installed `RStudio`.)

If you have not worked with Bioconductor before, the mechanisms for installing packages is different than the usual packages that are hosted by CRAN. In this module, we will be using the `Biostrings` package (documentation). Copy and paste the following line of code into the `console` (bottom left area of `RStudio`) and then press ENTER to run the installation process.

```
if (!requireNamespace("BiocManager", quietly = TRUE))
    install.packages("BiocManager")
```

Next, now that `BiocManager` (the package manager for Bioconductor) has been installed, we can use the double-colon operator (::) to specifically call the `install` function from the `BiocManager` package and install the `Biostrings` package from Bioconductor. Once again, copy and paste the following line of code into the `console` (bottom left area of `RStudio`) and then press ENTER to run the installation process.

```
BiocManager::install("Biostrings")
```

At this point, you can verify the presence of our new code packages in RStudio by looking in the `Packages` pane (bottom right area in `RStudio`).

Starting a Bioconductor Session

After installing a code package, the `library` command tells `RStudio` that we want to have the `Biostrings` package available for our use presently in the current coding *environment*. Run the following code (either by pressing the green triangle here in RStudio or using the keyboard shortcut).

```
library("Biostrings")
```

The Samples

We will once again use the HomoloGene database at NCBI and the following sample of amino acid sequences.

```
Homo_sapiens <- "MGDVEKGKKIFIMKCSQCHTVEKGGKHKTGPNLHGLFGRKTGQAPGYSYTAANKNKGIIWGEDTLMEYLENPKKYIPGTKMIFVGI
Macaca_mulatta <- "MGDVEKGKKIFVMKCSQCHTVEKGGKHKTGPNLHGLFGRKTGQAPGYSNTAANKNKGITWGEDTLMEYLENPKKYIPGTKMIFV
Bos_taurus <- "MGDVEKGKKIFVQKCAQCHTVEKGGKHKTGPNLHGLFGRKTGQAPGFSYTDANKNKGITWGEETLMEYLENPKKYIPGTKMIFAGIKK
Gallus_gallus <- "MGDIEKGKKIFVQKCSQCHTVEKGGKHKTGPNLHGLFGRKTGQAEGFSYTDANKNKGITWGEDTLMEYLENPKKYIPGTKMIFAG
Xenopus_Silurana_tropicalis <- "MGDAEKGKKIFVQKCSQCHTVEKGGKHKTGPNLHGLFGRKTGQAEGFSYTDANKNKGIVWDEGTLLEYLEN
Danio_rerio <- "MGDVEKGKKVFVQKCAQCHTVENGGKHKVGPNLWGLFGRKTGQAEGFSYTDANKSKGIVWGEDTLMEYLENPKKYIPGTKMIFAGIK
Drosophila_melanogaster <- "MGVPAGDVEKGKKLFVQRCAQCHTVEAGGKHKVGPNLHGLIGRKTGQAAGFAYTDANKAKGITWNEDTLFEYLEN
Zea_mays <- "MASFSEAPPGNPKAGEKIFKTKCAQCHTVDKGAGHKQGPNLNGLFGRQSGTTAGYSYSAGNKNKAVVWEEDTLYEYLLNPKKYIPGTKMV
Saccharomyces_cerevisiae_S288c <- "MTEFKAGSAKKGATLFKTRCLQCHTVEKGGPHKVGPNLHGIFGRHSGQAEGYSYTDANIKKNVLWDEN
```

Task 1

1. In previous modules, we talked about similarity and alignment. Upon doing a visual glimpse of the amino acid sequences above, what do you observe about the sequences? Type your answer into the space below.
2. Summarize why we sought out the *cytochrome c* genome in those species.

BLOSUM62

For convenience, the block substitution matrix `BLOSUM62` is available in the `Biostrings` package. First, we use the `data` function to bring the matrix into the current coding environment.

```
data("BLOSUM62")
```

Task 2

Copy, paste, and run the following code in the `console` of RStudio (will open the matrix in a new tab); then type your observations about the matrix below and outside the code block.

```
View(BLOSUM62)
```

Pairwise Sequence Alignments

In this section, we will get a sense of how the `Biostrings` package code can help us perform pairwise sequence alignments. In addition to taking a pair of sequences for inputs, the `pairwiseAlignment` function has a few parameters that bioinformatics researchers may use to modify their calculations. Copy, paste, and run the following code in the `console` to bring up the *manual* page for this function. Skim over the documentation and notice the variety of parameters (after the "pattern, subject" places for inputs).

```
?pairwiseAlignment
```

We will continue to use the block substitution matrix `BLOSUM62` for the calculation of similarity scores. We will also use the `AAString` function along the way (technical note: this function eases handling of much longer sequences and sequences/alignments with gaps). Otherwise, we will use the default parameter settings which include `gapOpening = 10` and `gapExtension = 4`.

For example, the following code will perform a pairwise sequence alignment analysis and save the results into a variable called `alignment_result`

```
alignment_result <- pairwiseAlignment(AAString(Bos_taurus),
                                      AAString(Danio_rerio),
                                      substitutionMatrix = BLOSUM62)
```

Let us look at the contents of `alignment_result`.

```
alignment_result
```

```
## Global PairwiseAlignmentsSingleSubject (1 of 1)
## pattern: MGDVEKGKKIFVQKCAQCHTVEKGGKHKTGPNLH...PKKYIPGTKMIFAGIKKKGEREDLIAYLKKATNE
## subject: MGDVEKGKKVFVQKCAQCHTVENGGKHKVGPNLW...PKKYIPGTKMIFAGIKKKGERADLIAYLKSATS-
## score: 491
```

Task 3

1. Run the following code, and then describe in your own words what the `compareStrings` function does. (Hint: you can use `?compareStrings` to bring up the manual page)

```
compareStrings(alignment_result)
```

```
## [1] "MGDVEKGKK?FVQKCAQCHTVE?GGKHK?GPNL?GLFGRKTGQA?GFSYTDANK?KGI?WGE?TLMEYLENPKKYIPGTKMIFAGIKKKGER?DL
```

2. Run the following code, and then describe in your own words what the `pid` function does.

```
pid(alignment_result)
```

```
## [1] 89.42308
```

Task 4

1. Perform a pairwise sequence alignment (with the `pairwiseAlignment`, `compareStrings`, and `pid` functions) on a different pair of sequences that you think are *similar* (in the evolutionary sense).
2. Perform a pairwise sequence alignment (with the `pairwiseAlignment`, `compareStrings`, and `pid` functions) on a different pair of sequences that you think are *dissimilar* (in the evolutionary sense).

Multiple Sequence Alignment

Finally, run the following code to perform a multiple-sequence alignment on our sample of amino acid sequences.

```
shortest_sequence_length <- min(nchar(Bos_taurus),
                                nchar(Danio_rerio),
                                nchar(Drosophila_melanogaster),
                                nchar(Gallus_gallus),
                                nchar(Homo_sapiens),
                                nchar(Macaca_mulatta),
                                nchar(Saccharomyces_cerevisiae_S288c),
                                nchar(Xenopus_Silurana_tropicalis),
                                nchar(Zea_mays))

aama_result <- AAMultipleAlignment(c(Bos_taurus, Danio_rerio, Drosophila_melanogaster,
                      Gallus_gallus, Homo_sapiens, Macaca_mulatta,
                      Saccharomyces_cerevisiae_S288c, Xenopus_Silurana_tropicalis,
                      Zea_mays),
                    end = shortest_sequence_length)

rownames(aama_result) <- c("Bos_taurus", "Danio_rerio", "Drosophila_melanogaster",
                      "Gallus_gallus", "Homo_sapiens", "Macaca_mulatta",
                      "Saccharomyces_cerevisiae_S288c", "Xenopus_Silurana_tropicalis",
                      "Zea_mays")
print(aama_result)
```

```
## AAMultipleAlignment with 9 rows and 104 columns
##      aln                                                    names
## [1] MGDVEKGKKIFVQKCAQCHTVEKGGK...KMIFAGIKKKGEREDLIAYLKKATN Bos_taurus
## [2] MGDVEKGKKVFVQKCAQCHTVENGGK...KMIFAGIKKKGERADLIAYLKSATS Danio_rerio
## [3] MGVPAGDVEKGKKLFVQRCAQCHTVE...IPGTKMIFAGLKKPNERGDLIAYLK Drosophila_melano...
## [4] MGDIEKGKKIFVQKCSQCHTVEKGGK...KMIFAGIKKKSERVDLIAYLKDATS Gallus_gallus
## [5] MGDVEKGKKIFIMKCSQCHTVEKGGK...KMIFVGIKKKEERADLIAYLKKATN Homo_sapiens
## [6] MGDVEKGKKIFVMKCSQCHTVEKGGK...KMIFVGIKKREERADLIAYLKKATN Macaca_mulatta
## [7] MTEFKAGSAKKGATLFKTRCLQCHTV...YIPGTKMAFGGLKKEKDRNDLITYL Saccharomyces_cer...
## [8] MGDAEKGKKIFVQKCSQCHTVEKGGK...KMIFAGIKKKGERQDLIAYLKQSTS Xenopus_Silurana_...
## [9] MASFSEAPPGNPKAGEKIFKTKCAQC...PKKYIPGTKMVFPGLKKPQERADLI Zea_mays
```

Task 5

1. Based on the results of the previous code block and your work with these materials, what complications do you think bioinformatics researchers encounter when wishing to perform a multiple-sequence alignment?

Wrap-Up

- Click the `knit` button (top left of this coding session's area, near the `save` button). This process will combine the `R markdown`, `HTML` code, and `R` code and create an `HTML` file (that can be viewed in any internet browser).
- This HTML file is what your instructor will request for the assignment submission.

References

This module of activity was adapted from the materials *Sequence Similarity: An inquiry based and "under the hood" approach for incorporating molecular sequence alignment in introductory undergraduate biology courses* by Kleinschmit, et al. (CourseSource link)

Further information about the pairwise- and multiple-sequence alignment `Biostrings` tools can be found in the vignettes (Bioconductor link)

```
sessionInfo()
```

```
## R version 4.1.0 (2021-05-18)
## Platform: x86_64-w64-mingw32/x64 (64-bit)
## Running under: Windows 10 x64 (build 19042)
##
## Matrix products: default
##
## locale:
## [1] LC_COLLATE=English_United States.1252
## [2] LC_CTYPE=English_United States.1252
## [3] LC_MONETARY=English_United States.1252
## [4] LC_NUMERIC=C
## [5] LC_TIME=English_United States.1252
##
## attached base packages:
## [1] stats4    parallel  stats     graphics  grDevices utils     datasets
## [8] methods   base
##
## other attached packages:
## [1] Biostrings_2.60.0   GenomeInfoDb_1.28.0 XVector_0.32.0
## [4] IRanges_2.26.0      S4Vectors_0.30.0    BiocGenerics_0.38.0
##
## loaded via a namespace (and not attached):
##  [1] crayon_1.4.1           digest_0.6.27          bitops_1.0-7
##  [4] magrittr_2.0.1         evaluate_0.14          zlibbioc_1.38.0
##  [7] rlang_0.4.11           stringi_1.6.2          rmarkdown_2.8
## [10] tools_4.1.0            stringr_1.4.0          RCurl_1.98-1.3
## [13] xfun_0.23              yaml_2.2.1             compiler_4.1.0
## [16] htmltools_0.5.1.1      knitr_1.33             GenomeInfoDbData_1.2.6
```