Programming with R
University of Potsdam SS 2020
Detlef Groth
Exam – October 6th, 2020

You have 90 minutes time for the implementation of the programming tasks. Make after the steps 1-2 intermediate versions: FastaParser1.R and FastaParser2.R. Submit at the end of the exam those single versions by USB stick or by E-Mail preferentially as zip archive (surname.zip) to me (E-Mail: dgroth@uni-potsdam.de). 75% of the tasks will be given on execute correctness and 25% of the tasks will be weighted by usefulness and clearness of the implementation.
The theory questions in 3 and 4 are to be answered first without any aids and the sheet with the answers is to be handed in after about 10-15 minutes. For the computer tasks 1-2 all aids are allowed during the exam. This does not include personal support from fellow students or other persons except from me.

Clarification

With this I state, that I will not take and give any not allowed support during the exam.

Name, Matrikel-Number.: Signature:

1. 4 points (layout console application)
2. 4 points (implementation of console application)
3. 2 points (theory)
4. 2 points (theory)

Sum: 12 points

Good luck !!

1. Layout of console application and utilizing command line arguments (4 points):
Create the basic outline of a console application with *main* and *help* function which explains the purpose of the application and the usage of command line arguments. Save the possible command line arguments in a single vector. If no argument is given or the user uses the argument '--help' or '-h' as the only argument, call the help function and exit the application. The arguments should be one or more filenames for FASTA sequence files. Do a check if the given filenames exist and warn the user if one of the files does not exist on the users hard disk. Provide in such a case thereafter useful help message indicating which file does not exists and all finally the help function. The help message should contain, program name, author name and an usage line like :

```
Detlef's FastaParser1.R – count amino acids
Author: Detlef Groth, University of Potsdam, 2020
Usage: FastaParser1.R file1.fasta [file2.fasta ...]
```

R filename first task: ______________________________

2. Opening and counting amino acids in the FASTA files (4 points): The input files we are working with are: sars-cov1.fasta and sars-cov2.fasta from the Moodle course. Our program should however work with any FASTA file. Implement a *countSequence* function which return the ids and the sequence lengths as a *data.frame. (Good programmers don't worry about the code, they worry about their data - Linus Torvalds)*. The argument of the function should be the fasta filename, the return value should be the data frame with the columns id and length. Hints: you can create data frames out of vectors using *df=data.frame(col1=vec1,col2=vec2)* where col1 and col2 are the column names and you can extend vectors using *vec1=c()* and *vec1=append(vec1,1)*. So it is best to create two empty vectors before you open the file and then in your file *while* loop extend those vectors. At the end, after the while loop, just create and return the data frame made out of those two vectors.
In the *main* function you should loop over your arguments and call for each file *countSequence*. You should then with the data frame do a printing to the terminal by looping over the rows of the data frame: *for (i in 1:nrow(df)) { }*.

```
$ FastaParser2.R sars-cov1.fasta  sars-cov2.fasta
sp|P59637|VEMP_CVHSA      76
sp|P59596|VME1_CVHSA      221
...
sp|Q7TFA0|NS8A_CVHSA      39
sp|P59594|SPIKE_CVHSA     1255
sp|P0DTC3|AP3A_SARS2      275
sp|P0DTD3|Y14_SARS2       73
sp|P0DTC2|SPIKE_SARS2     1273
sp|P0DTC4|VEMP_SARS2      75
...
```

R filename second task: ____________________________________

Matrikel-Number: Firstname Lastname:

Theory 3 (2 points): Explain in your own words advantages and disadvantages of command line applications (CLI) in comparison to graphical user interfaces (GUI).

Theory 4 (2 points): Explain in a few sentences the purpose of object oriented programming and name two systems which are available in the language core of R or as external libraries. Name a object oriented system which can be used to directly save the object with its methods and properties using the saveRDS function (+ 1 point).