Programming with R
University of Potsdam SS 2020
Detlef Groth
Saturdays Test - Exam – October 3rd, 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 two variables. If the required arguments are not given, call the help function and exit the application. The two arguments should be the filename for a FASTA sequence file and a sequence pattern. Do a check if the given filename exists and warn the user if it does not exists on the uses harddisk. Provide an useful help message if the two arguments are not correctly given using the help function. The help message should contain, program name, author name and an usage line like :

```
Detlef's FastaParser1.R – search sequence patterns
Author: Detlef Groth, University of Potsdam, 2020
Usage: FastaParser1.R file.fasta searchpattern
```

R filename first task: ______________________________

2. Opening and searching in the FASTA file (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 *searchSequence* function to display the sequence IDs which contain the given sequence search pattern. Hint the *grepl* command might help in this function. The two arguments for this function should be the input filename and as second argument the search pattern. You should write the output directly to the terminal, giving the sequence ids, the pattern and if it hits or does not matches the pattern. You can ignore the line break problem in your search. Below is a possible sample run of your program with parts of its tabular output:

```
$ FastaParser2.R sars-cov1.fasta MAT
sp|P59637|VEMP_CVHSA      MAT   false
sp|P59596|VME1_CVHSA      MAT   false
sp|Q7TLC7|Y14_CVHSA       MAT   false
sp|P59595|NCAP_CVHSA      MAT   false
sp|P0C6X7|R1AB_CVHSA      MAT   true
...
sp|P59634|NS6_CVHSA       MAT   false
sp|P0C6U8|R1A_CVHSA       MAT   true
sp|Q7TFA1|NS7B_CVHSA      MAT   false
sp|P59632|AP3A_CVHSA      MAT   false
```

R filename second task: ____________________________________

Matrikel-Number: Firstname Lastname:

Theory 3 (2 points): Explain in your own words for what the shebang line on Linux systems is used and why you should add it as well on Windows R scripts.

Theory 4 (2 points): Explain in a few sentences what are the three different argument types R-functions can take and what are they used for. Write a declaration of a short function (no implementation) which has all three argument types in its parameter list.