Scientific Programming – Bioconductor data types and packages in R

Exercises:

1) Download the set of SARS-Cov2 genome sequences provided on the course's WeBeep-website ("materials" → "exercises" → "data" → "SARS2_NCBIVirus_complete_genomes_20200329.fasta").
Note: although this file is only from March 29, 2020, it is more than outdated. By now, many more SARS-Cov2 genomes are available, for example at: https://www.ncbi.nlm.nih.gov/datasets/coronavirus/genomes/ (921,212 as of March 6, 2022). For our purpose, the 146 sequences from March 2020 are enough.
Use the `Biostrings` package to load the SARS-Cov2 genome sequences.

- Whats the average length of the genome sequences of the novel coronavirus? What's its median length? (It's sad how much trouble such few basepairs can cause …)
- Get the list of sequence names from the `DNAStringSet` object
- Extract the sequences (character strings) of the first three genomes in the set (using one single command, no loop)

2) Use the `ShortRead` package to load the NGS reads from the file `SP1.fq` (provided in BeeP).

- Extract the sequences (character strings) of the first three reads (using one single command!)
- Extract the base qualities (as character strings) for the first three reads (one command!). Hint: you will have to access the `quality` data field of the object returned by the `quality`-function …
- Produce the HTML quality assessment report for this read set and have a look at it. (Please be aware that many of the plots do not look like what you would expect from a larger and more modern data set!).
- Have a look at the ShortRead vignette at http://bioconductor.org/packages/release/bioc/vignettes/ShortRead/inst/doc/Overview.pdf
- The Vignette describes a larger test data set (20,000 reads, 72 bases each): E-MTAB-1147, file "ERR127302_1_subset.fastq.gz". Load the data and produce the HTML quality assessment report.
- Verify in a single command that all reads do have precisely 72 bases length. Hint: get the corresponding DNAStringSet and use functions `width` and `unique`. Another possible approach, based on logical comparison, can be implemented with the function `all`.
- Read Section 3.2 (Filtering and Trimming) of the vignette. Do you have an idea of what is happening here? Apply this to the 20,000 reads from the E-MTAB-1147 example data. If you have already loaded it, you can apply the single filtering and trimming steps directly to the loaded dataset and do not need to re-load it in chunks as described in Section 3.2. Questions:
  - How many reads are left after filtering and trimming?
  - Do they all have the same length (72 bases)? If not, what range of lengths/widths do they have?
  - If you want to visualize the distribution of lengths, you can use the function `hist` on the width/length vector.

3) Read the BAM file `ex1.bam` provided as an example with the `Rsamtools` package.

- Transform the read data to a `DataFrame`. Check whether the `DataFrame` has row names.
- Transform the read data to a `data.frame` and check for row names, too.

4) Read the vignette of `VariantAnnotation`:

http://bioconductor.org/packages/release/bioc/vignettes/VariantAnnotation/inst/doc/VariantAnnotation.pdf

- Execute the shown examples to get familiar with the data. Explore the data objects beyond what is directly shown in the vignette, e.g., check what classes they are, what structures they have, how you can access individual data items, whether standard ranges (e.g., vcf[1:5]) work and what the output is, etc.

5) Read and explore also the vignette on how to filter variants:
http://bioconductor.org/packages/release/bioc/vignettes/VariantAnnotation/inst/doc/filterVcf.pdf

6) Load the human reference genome and arbitrarily extract genomic regions from it. "Play" with the `GRanges` object you have to define for this purpose to extract different sequences from both strands of the genome. Use the two different ways of creating the `GRanges` object we have seen in the lecture.

7) Instead using arbitrary genomic regions defined through an own Granges object, use known gene annotation to get the sequences of a subset of genes, coding regions, transcripts, etc.

8) Taking the gene annotation, group exons by genes and then …

- Get the DNA sequence for the third exon of the 10,000th gene in the list. How long is that exon and what is its sequence?
- Get the DNA sequence of the third exon of the gene AKT3 (NCBI gene ID 10000). Is it longer than the sequence of the 10,000th gene in the list?
  (Be aware of this problem whenever you work with numeric gene IDs!!!!)