Introduction to Bioinformatics
BIOL-3951 / COMP-3550 / BIOL-7941

Lab 2: Introduction to Nextflow

If you are using your own laptop, you will need to install Nextflow as explained in https://www.nextflow.io/index.html. Basically you need to have Java 8 or later installed, and then install Nextflow by entering the following command in the terminal:

```
curl -fsSL get.nextflow.io | bash
```

This command creates a `nextflow` launcher in the current directory.
Complete the installation by moving the launcher into a directory on your `PATH`.

Question 1.

Assume you have a directory with several fasta files. Each of these fasta files have many sequences on it. You need to get a plot showing the distribution of the sequence lengths in all of these files.

In this lab, you will write a Nextflow pipeline to achieve this task.

Resources: Nextflow, shell commands.

Instructions:

1. Download the file lengthPlot.nf from D2L and use a text editor to open it.
2. Download the file lab2Files.tar.gz from D2L and extract its contents in the same directory containing the file lengthPlot.nf
3. Add an input parameter on line 8 of the file by writing the following:

```
params.inDir = "fastaFiles"
```

4. Get all the files in the directory in a channel called fastaFiles by writing the following:

```
Channel
.fromPath( "${params.inDir}/*.fa" )
.ifEmpty { error "Cannot find any input sequence files matching:
*.fa" }
.set { fastaFiles }
```

What is going on in these lines? We are creating a channel that has all the fasta files in the directory given as parameter to the program. ${params.inDir} is a placeholder which will be replaced by the actual value provided when the script is executed. If the directory is empty there will be an error message. The channel is

given the name fastaFiles.

5. For each fasta file in the directory, we are going to get a tab-delimited text file with the length of the sequences. The process getLengths will do this. The input to this process are the files in the channel fastaFiles. In the input definition section of the process write:

```
file aFile from fastaFiles
```

In the output definition section of the process write:

```
file 'lengths.txt' into lengths
```

In the script section, we are going to use `awk` to get the length of the sequences. Thus in the script definition section of the process write the following command:

```
awk -F "\\n" '!/^>/ {print length (\$0) } /^>/ { printf "%s\\t", \$0}'
${aFile} > lengths.txt
```

Note that all special characters that should not be interpolated or have a special meaning for Nextflow have a **** preceding them.

What is awk doing?

- `!/^>/` - means if there is not a > at the beginning of the line then
- `{print length (\$0) }` - means print the number of characters in the line
- `/^>/` - means if there is a > at the beginning of the line then
- `{ printf "%s\\t", \$0}` – means print the line followed by a tab.

6. Lines 35 to 37 have the instructions to concatenate all the files created by the process getLengths into a single file called seqLengths.txt and to link this file to a channel called lengths2.

7. The process plotlength will receive as input the file from channel lengths2 and will generate as output a file called lenDist.pdf which will be linked to channel called plot. Modify the script to add in line 46 the input definition and in line 49 the output definition. Note that the script section is already filled in with R code.

8. Finally let's copy the file generated by the process plotLength into the current directory as a file named lenDistribution.pdf. To do this, write at the end of the script the following code:

```
plot
.collectFile(name: file("lenDistribution.pdf"))
```

9. Save your script (it should look like the script shown in the file lengthPlot_key.nf.pdf available in D2L) and run it by typing in the terminal:

```
nextflow run lengthPlot.nf
```

In the terminal you should see an output similar to:

```
N E X T F L O W  ~  version 0.31.1
Launching `lengthPlot.nf` [curious_mcnulty] - revision: 3c42c8fe93
[warm up] executor > local
[49/1fa8e7] Submitted process > getLengths (2)
[65/208a52] Submitted process > getLengths (1)
[ce/405548] Submitted process > getLengths (3)
[52/fe5bd8] Submitted process > plotLength (1)
```

10. Take a look at the files generated by the pipeline; namely, seqLength.txt and lenDistribution.pdf.

Question 2.

Instructions:

1. Download the Nextflow script lengthPlot_perFile.nf from D2L. Study this script.
2. Do you think this script does something different from the script you just wrote (lengthPlot.nf)? If so, what?
3. Run the script by typing in the terminal:

   ```
   nextflow run lengthPlot_perFile.nf --inDir fastaFiles
   ```

   Note that now we are giving explicitly a value to the parameter inDir.

4. Look at the output and the files generated by the script. Can you explain why do you get this output and these files?

Question 3.

Instructions:

1. Write a Nextflow script called myFirstScript.nf that receives one parameter called **inFile** and has a single process called **getColumn**. This single process will extract the second column of the file using the shell command cut and store this column in a file called **"SecondCol.txt"**. Your script should create this file in your current directory.

   Note: To assign the file given as an input parameter to a channel you can use as

input for a process, you need to do the following:

```
aFile = file("${params.inFile}")
```

2. Test your script by running it using as input the file seqLengths.txt. To run it, type in the terminal the following:

```
nextflow run myFirstScript.nf --inFile seqLengths.txt
```

When you are finished:

1. Answer the quiz available in D2L