For this exercise we will use the same commands as the previous exercise, but we will turn them into a bash script capable of downloading and scanning multiple SRA datasets, instead of just one. Before we can do that, we need to discuss how variables work in bash.

1. Start by declaring some variables. Open your terminal and try declaring the following variables – you can check their value with the command echo. (don’t forget to add a $ before the variable name!)
   1. x=10 b. x = 5 c. name=kyle d. name=john doe

Which of the above commands did and did not work? How could you fix them?

|  |  |
| --- | --- |
| x=3 | sets x equal to 3 |
| x = 3 | runs the command x with arguments = and 3 |
| $x=3 | runs the command x=3 where x becomes its value |
| x=$3 | sets x equal to whatever the variable 3 is set to |

Note: By default, all variables are empty (you can test this with echo $asdf). This means the last command will not generate an error and will instead set x equal to nothing – making it a possible source of bugs later!

1. Now we are going to combine two tools we have already covered in this class. Last week we used 2 different commands to download metagenomes from the SRA and scan them with bowtie2 for crassphage.

fastq-dump --outdir ~/ --skip-technical --readids --dumpbase --clip SRR3403834

bowtie2 -x crassphage\_index --no-unal -S SRR3403834.sam -q SRR3403834.fastq

These commands work great if you are interested in SRR3403834, but to download and scan other metagenomes, every “SRR3403834” will need to be replaced with the name of another metagenome. This is where we will use another piece of code from earlier in the semester.

while read my\_line; And the text file “SRA\_list.txt” contained:

do SRR3403834

echo $my\_line SRR3403835

done <SRA\_list.txt SRR3403836

Create a new folder for todays exercise, cd into it and add:

1. The crassphage\_index files from bowtie2-build that we used in our previous scan
2. A text file containing the 3 SRA run IDs from SRA\_list.txt
3. A new shell script, that will contain all the commands needed to run the search.
4. Munge – (verb) to change and mix code or data formats in a potentially destructive and irreversible way.
   1. Munge the while loop, fastq-dump, and bowtie2 commands from #1 in your shell script.
   2. For testing purposes, be sure to add -x 100 to your fastq-dump command
      1. What does -x do for fastq-dump?
   3. File not found? Check your paths! The fastq-dump command from #1 will download the fastq-files to your home directory, but you are doing all of this work inside a new folder you created.
5. **Homework** – Between now and next class, use magicblast or bowtie2 and fastq-dump to generate at least 3 (Non-empty) SAM files. We will begin analyzing these files at the start of the next class.