BIO609: Bash scripting. Bash is a unix shell and command language. It is mostly the default login shell (the shell you work with when you login to a Unix machine).

Exercise 1: writing and executing a shell script

Try to copy/paste the below simple bash program into a file and execute it. The hello world bash script:

```
#!/bin/bash
echo "Hello World"
```

Copy the above 2 lines and save them to the file "`hello.sh`". The first line tells the Unix shell to interpret the program. In order to run the program you can either start it with:

Make your file executable by typing:

```
$ chmod +x hello.sh
```

And now you can simply type:

```
$ ./hello.sh
```

Note the "`./`" at the start of the command. This is because the directory where we stored `hello.sh` is not in the system variable `$PATH`.

Exercise 2: write a simple bash script

Often, you would like to run the same command with different parameters. As an exercise, write a simple bash script that will output numbers from 1 to 100. Use a for loop.

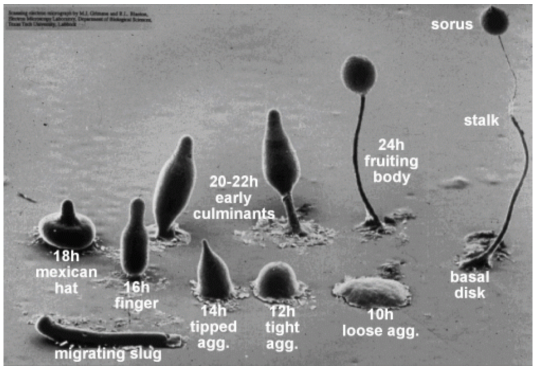
```
#!/bin/bash
for i in {1..100}
do
   echo $i
done
```

Save the above code to a file (e.g. `script.sh`), make the file executable (+x flag) and run it.

What is the output?

Exercise 3: download the Dictyostelium discoideum (dd) genome in FASTA format

Dictyostelium discoideum (www.dictybase.org) is an interesting social amoeba and a well studied model organism. The amoeba lives in the soil, and when nutrients run out, 100K amoebas aggregate into a fruiting body. The stalk amoebas "sacrifice" themselves and the sorus is blown away by the wind to a new location with nutrients (hopefully).


Scanning electron micrograph by M.J. Grimson and R.L. Blanton, Electron Microscopy Laboratory, Department of Biological Sciences, Texas Tech University, Lubbock

You can download the **dd genome FASTA** file directly from our server:

```
wget https://bioinfo.evolution.uzh.ch/share/bio609/dicty/dd.fasta
```

The FASTA format is widely used in sequence distribution, see the description at http://en.wikipedia.org/wiki/FASTA_format.

Use **grep** to find out how many chromosomes are present in the FASTA file. Use **grep** (💡 -v) to only print out the genomic sequence. How large is the genome?

📝 Exercise 4: searching for short sequences in the *Dictyostelium discoideum* genome

❓ Is the sequence “AAAAAGAGATACAT” present in the DD genome (`dd.fasta`)?

💡 You can simply use **grep** to find out.

📝 Exercise 5: handling RNA-seq data

Next-generation (short-read) sequencing data is usually stored in FASTQ files: https://en.wikipedia.org/wiki/FASTQ_format

FASTQ files are similar to FASTA files, but they also contain sequence quality.
We will work with 10 FASTQ files (10 samples) containing sequences from Dictyostelium discoideum RNA (RNA-seq).

Write a short script to download *.fastq.gz files from:

```
https://bioinfo.evolution.uzh.ch/share/bio609/dicty
```

Open the link in a browser and you should see 10 FASTQ files. Instead of clicking and downloading each file separately, write a script to download the files.

```
for run_id in {1..10}
```

In the for loop, you would need to adjust the name of the file and use command **wget** to download the files.

Exercise 6: download and install a local version of STAR

Download the short-read aligner STAR and compile it. This is a good exercise to download and install software on your own on a Unix system.

```
git clone https://github.com/alexdobin/STAR
```

Change to folder "source" and type "make". The software will compile. So you downloaded and compiled software from GitHub :)

Exercise 7: add the STAR binary to the path

If you type STAR in the command line, you will see that the program is not found. However if you type "./STAR", the program will run, but only if you are in the source directory (where the file is located).

So that we can run STAR from any folder on the system, we will add the folder containing the binary file STAR to the profile PATH variable.

Edit the file $HOME/.profile, by typing:

```
vi $HOME/.profile
```

And edit the last line to include the path to the **STAR/source folder**. If you now login and logout from your console and type STAR, the programm should start.

Exercise 8: map the RNA-seq data to the reference genome

First, you need to create an "index" for the dd.fasta reference genome. You can do this with:

```
mkdir istar # folder for genome index
STAR --runMode genomeGenerate --genomeDir istar --genomeFastaFiles dd.fasta
```

The command should take around 1-2 minutes to finish.

Then you can write a script to map the RNA-seq data to the reference genome using the newly created index above.

An example command would be:

```
STAR --genomeDir istar --readFilesIn sample1.fastq.gz --readFilesCommand zcat
```

After finishing, this creates a SAM file (alignments) named Aligned.out.sam. For each of the 10 samples, rename this file into sample_n.sam and convert the SAM file to a BAM file.

```
mv Aligned.out.sam sample1.sam
samtools view -bS sample1.sam > sample1.bam # convert sam to bam
samtools sort sample1.bam -o sample1.bam # sort bam file
samtools index sample1.bam # index bam file
rm sample1.sam # remove sample1.sam, since we don't need it anymore
```

Write a short script to repeat the above steps for all the 10 samples (sample1, sample2, ... sample10).

Exercise 9: download genome annotation in GFF format and count reads aligned to genes

Download the GTF file (genome annotation) for *Dictyosrtelium discoideum* from:

```
wget https://bioinfo.evolution.uzh.ch/share/bio609/dicty/dd.gtf
htseq-count -f bam sample1.bam dd.gtf > sample1.tab
```

Repeat this for all the samples, so in the end you should get 10 tab files with gene counts.

* Exercise 10: combine gene expression tables into one single table

This one is tricky, but you can use awk:

```
awk 'NF > 1{ a[$1] = a[$1]"\t"$2} END {for( i in a ) print i a[i]}' *.tab >
samples.tab
```

Stores the big table into the file **samples.tab**.

* Exercise 11: try to download and compile bowtie2, make an index of the genome and map the reads to the reference genome, count the reads aligning to genes and create the samples.tab table. Compare the STAR to the bowtie2 mappings.

System information, processes and other useful commands

uname -a	display system information
man *command*	display manual page of command
df -h	list mounted disks with available space
du -h *path*	show space usage
top	display running processes
kill -9 *pid*	kill process

File and folder manipulation, compression

pwd	display current folder
ls -l *path*	list files and folders
cd *path*	change folder to path
cd ~	change folder to home folder
mkdir *name*	make folder
rmdir *name*	remove folder
cp *source dest*	copy file/folder and all its contents
less *filename*	display file content
wc *filename*	count number of lines in file
head *filename*	shows first few lines of file
tail *filename*	shows last few lines of file
gzip *filename*	compress file with gzip (adds .gz extension)
gunzip *filename*	uncompress and remove .gz extension
tar xfz *filename.tar.gz*	uncompress files from tar.gz archive
tar zcvf *archive.tar.gz folder_to_compress*	creates archive.tar.gz
unzip filename.zip	unzip archive

Network and file transfer

wget URL	download file (html page) and save to current folder
ssh *username@host*	remote login to host with username
sftp username@host	remote login to host with username and transfer files

“vi” editor

$ vi *filename*	start editing file with vi
i	switch to “insert” mode
ESC	switch to “command” mode
:w	save
:q	quit
:x	save and quit
/<pattern>	search for pattern, <n> gives you the next match
:q!	quit without saving changes