

Week 2 Exercise BIOL 7200

1. **Environments** (total 10 points)

For the following questions, provide a short written answer. No screenshots are required.

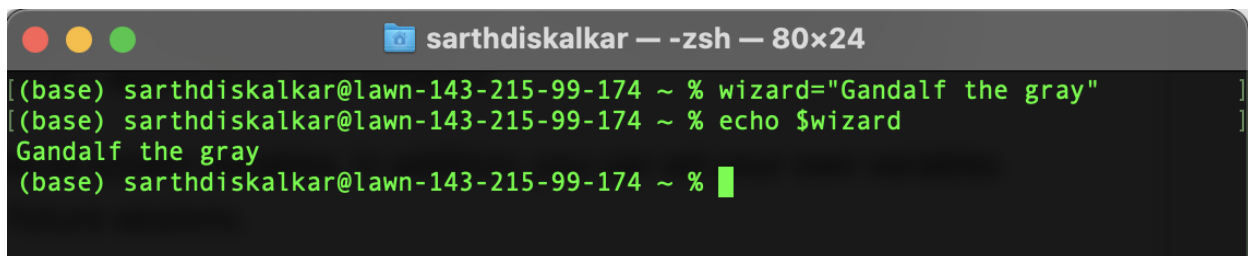
1. What is an environment in a Unix system? (1)

An environment in a Unix system is the array of strings (also called name-value pairs) present whenever a program is invoked. Basically, the environment is composed of variables and certain configurations that allows the shell to behave in a certain way.

2. How does the function or behavior of variables and aliases differ? (state 1 difference) (1)

Variables and aliases differ in the way that they have different uses. Variables store data and aliases serve as command shortcuts.

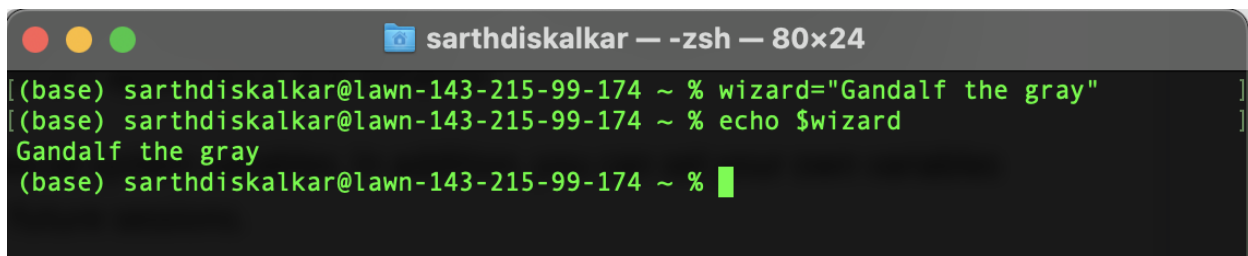
3. State the command you would use to create a variable called "wizard" which contains the string "Gandalf the gray". (1)

A terminal window titled 'sarthdiskalkar — -zsh — 80x24' with a dark background and green text. It shows the following commands and output:

```
[(base) sarthdiskalkar@lawn-143-215-99-174 ~ % wizard="Gandalf the gray" ]
[(base) sarthdiskalkar@lawn-143-215-99-174 ~ % echo $wizard ]
Gandalf the gray
(base) sarthdiskalkar@lawn-143-215-99-174 ~ % █
```

Wizard="Gandalf the gray"

4. State the command you would use to print the contents of the "wizard" variable. (1)

A terminal window titled 'sarthdiskalkar — -zsh — 80x24' with a dark background and green text. It shows the following commands and output:

```
[(base) sarthdiskalkar@lawn-143-215-99-174 ~ % wizard="Gandalf the gray" ]
[(base) sarthdiskalkar@lawn-143-215-99-174 ~ % echo $wizard ]
Gandalf the gray
(base) sarthdiskalkar@lawn-143-215-99-174 ~ % █
```

echo \$wizard

5. State the command you would use to make the variable "wizard" available to any subshells of your current shell. (1)

```

sarthdiskalkar — bash — 80x24
Last login: Tue Sep  5 15:46:53 on ttys000
(base) sarthdiskalkar@lawn-143-215-99-174 ~ % export wizard="Gandalf the gray"
(base) sarthdiskalkar@lawn-143-215-99-174 ~ % bash

The default interactive shell is now zsh.
To update your account to use zsh, please run `chsh -s /bin/zsh`.
For more details, please visit https://support.apple.com/kb/HT208050.
bash-3.2$ echo $wizard
Gandalf the gray
bash-3.2$

```

export wizard="Gandalf the gray"

- State the command you would use to create an alias, called "view_wizard", for the command you gave for question 1.4 (a command to print the contents of the "wizard" variable). (1)

alias view_wizard="echo \$wizard"

- State the command you would use to view the alias created in question 1.6. (1)

alias view_wizard

- Consider a situation in which the value stored in "wizard" may change (e.g., "Radagast the Brown"). Describe the difference in the behavior of the alias created in 1.6 if you create the alias using single quotes or double quotes. Explain the difference. (3)

```

sarthdiskalkar — zsh — 80x24
(base) sarthdiskalkar@lawn-143-215-99-174 ~ % wizard="Gandalf the gray"
(base) sarthdiskalkar@lawn-143-215-99-174 ~ % export wizard="Gandalf the gray"
(base) sarthdiskalkar@lawn-143-215-99-174 ~ % alias view_wizard="echo $wizard"
(base) sarthdiskalkar@lawn-143-215-99-174 ~ % view_wizard
Gandalf the gray
(base) sarthdiskalkar@lawn-143-215-99-174 ~ % alias view_diff='echo $wizard'
(base) sarthdiskalkar@lawn-143-215-99-174 ~ % view_diff
Gandalf the gray
(base) sarthdiskalkar@lawn-143-215-99-174 ~ % wizard="Radagast"
(base) sarthdiskalkar@lawn-143-215-99-174 ~ % view_wizard
Gandalf the gray
(base) sarthdiskalkar@lawn-143-215-99-174 ~ % view_diff
Radagast
(base) sarthdiskalkar@lawn-143-215-99-174 ~ %

```

It seems that when you use double quotes to create the alias, it captures the value of the variable in the command at the time the alias was created. When you use single quotes, it updates whenever the wizard variable is

changed and the command in the alias returns the new value variable string. Refer to this screenshot for proof of what happens when you use single quotes vs double quotes when creating an alias.

2. **PATH** (total 10 points)

For the following questions, provide a short written answer. No screenshots are required.

1. What does your `$PATH` variable determine about the behaviour of your shell? (1)

The `$PATH` variable determines where your shell looks for programs (executables) as it contains a colon-delimited list of directories. Whenever a command is issued, your shell checks the directories in the order of the `$PATH` variable and the first executable found is run.

2. Is your `$PATH` available to subshells or do you need to make it available to subshells (as in question 1.5)? (1)

The `$PATH` variable is available to subshells as seen in the screenshot below. The variable remains to be the same even when the bash subshell is created. The export command is not required here.

```
[(base) sarthdiskalkar@lawn-143-215-99-174 ~ % bash]
The default interactive shell is now zsh.
To update your account to use zsh, please run `chsh -s /bin/zsh`.
For more details, please visit https://support.apple.com/kb/HT208050.
[bash-3.2$ echo $PATH]
/Users/sarthdiskalkar/mambaforge/bin:/Users/sarthdiskalkar/mambaforge/condabin:/Library/Frameworks/Python.framework/Versions/3.11/bin:/usr/local/bin:/usr/bin:/bin:/usr/sbin:/sbin
[bash-3.2$ exit]
exit
[(base) sarthdiskalkar@lawn-143-215-99-174 ~ % echo $PATH]
/Users/sarthdiskalkar/mambaforge/bin:/Users/sarthdiskalkar/mambaforge/condabin:/Library/Frameworks/Python.framework/Versions/3.11/bin:/usr/local/bin:/usr/bin:/bin:/usr/sbin:/sbin
(base) sarthdiskalkar@lawn-143-215-99-174 ~ %
```

3. Consider a `$PATH` with the following value `"/usr/bin:/usr/local/bin:/bin"`. If I put an executable named `foo` in each of the directories in that `$PATH`, which directory contains the copy of `foo` which will run if I invoke it as follows (`$` here denotes a command prompt)? (2)
`$ foo`

Because `foo` is present in the first directory, the `'foo'` executable in the first directory `/usr/bin` will be invoked.

4. State the command you could use in question 2.3 to determine the location of the copy of `foo` being used.(1)

which foo

5. State a command you could use to add the directory `"/home/username/bin"` to the *beginning* of your `$PATH`. (1)

```
Last login: Tue Sep  5 17:27:00 on ttys000
(base) sarthdiskalkar@lawn-143-215-99-174 ~ % echo $PATH
/Users/sarthdiskalkar/mambaforge/bin:/Users/sarthdiskalkar/mambaforge/condabin:/Library/Frameworks/Python.framework/Versions/3.11/bin:/usr/local/bin:/usr/bin:/bin:/usr/sbin:/sbin
(base) sarthdiskalkar@lawn-143-215-99-174 ~ % export PATH="/home/sarthdiskalkar/bin:$PATH"
(base) sarthdiskalkar@lawn-143-215-99-174 ~ % echo $PATH
/home/sarthdiskalkar/bin:/Users/sarthdiskalkar/mambaforge/bin:/Users/sarthdiskalkar/mambaforge/condabin:/Library/Frameworks/Python.framework/Versions/3.11/bin:/usr/local/bin:/usr/bin:/bin:/usr/sbin:/sbin
(base) sarthdiskalkar@lawn-143-215-99-174 ~ %
```

`export PATH="/home/sarthdiskalkar/bin:$PATH"`

6. State a command you could use to add the directory `"/home/username/bin"` to the *end* of your `$PATH`. (1)

`export PATH="$PATH:/home/sarthdiskalkar/bin"`

7. Which file would you modify to change your `$PATH` for all future shell sessions? (1)

Since I have a mac, I would modify the `"~/.bash_profile"` file

8. Would changing the `$PATH` using the file you named in question 1.7 impact the `$PATH` variable of other users on the same computer? Why or why not? (2)

It does not impact the `$PATH` variable for other users on the same computer. This is because each user's shell environment is independent from each other and the `~/.bash_profile` file in a certain user is meant for that certain user only and not the other users.

3. **mamba basic usage** (total 10 points)

For the following questions, provide a short written answer. No screenshots are required. All answers which include commands should use **mamba** commands instead of **conda** where relevant.

1. What is the "base" environment in a **conda** or **mamba** installation? (1)

The base environment is indicated by the (base) before the prompt in the command line terminal. Upon downloading either mamba or conda, it will load into the default "base" environment. The base environment comes with a vital package manager and dependencies for Mamba or conda to function properly. It is best to load separate packages into a certain environment rather than all of them on the base to avoid version conflicts.

2. When should you install packages into the "base" environment and when should you install packages into another environment? (1)

Install packages into the base environment when installing system-wide tools that are all updated (version is up to date) and are going to be used by multiple projects. Also, if you want the packages to apply to all the users on the computer.

Installing packages into a specific environment is recommended when working on more than one project. Basically, you will create a specific environment for a specific project. This helps prevent version conflict and makes project management easier.

3. State the command to create a new environment named "week2ex" into which you install python version 3.11, seqtk, and blast.(1)

```
mamba create -n week2ex python=3.11 seqtk blast
```

4. State the command you would use to activate the environment created in question 3.4. (1)

```
mamba activate week2ex
```

5. State the command you would use to leave the environment created in question 3.4 if you had activated it. (1)

```
mamba deactivate
```

6. State the command you would use to export the environment created in question 3.4 to a file named "week2ex_env.yaml" in a cross-platform compatible way (i.e., so you could give it to someone running OSX or Ubuntu and they could recreate your environment). (1)

```
mamba env export --name week2ex --file week2ex_env.yaml
```

7. State the command you could use to create an environment named "new_week2_env" imported from the "week2ex_env.yaml" described in question 3.6. (1)

```
mamba env create --name new_week2_env --file week2ex_env.yaml
```

8. State the command you could use to delete the "new_week2_env" described in question 3.7. (1)

```
mamba env remove -n new_week2_env
```

9. State the command you could use to *install* the package "samtools" from the channel "bioconda" into your current environment. (1)

```
mamba install -c bioconda samtools
```

10. State the command you could use to *uninstall* the package "samtools" from your current environment. (1)

```
mamba remove samtools
```

4. Scripting in Bash (total 70 points)

For the following questions, you will be asked to write a Bash script that performs the described steps or function. Each answer should consist of three components:

- Copy and paste the contents of your script (ensure that any indentation renders correctly before submitting)
- Paste a screenshot your terminal executing the script as described and additional screenshots displaying requested outputs (Note: we won't ask you to show outputs that span more than one terminal window)
- Write any additional explanation requested

1. "dir_script.sh" (10 points)

Write a script named "dir_script" that performs the following steps. The usage of the script should be `./dir_script.sh <path>`

- Makes a directory as well as any missing parent directories
- enters the new directory (i.e., changes its working directory to be the newly created directory)
- prints its working directory to stdout

Run the script in your home directory, providing as input the path "~/abcxyz/demo_week2/dirsript/". In the screenshot, show that you do not have the dir "abcxyz" in your home dir before running the script, but that the dirs "~/abcxyz/demo_week2/dirsript/" do exist after you execute the script.

After running the script did your shell move its working directory to "~/abcxyz/demo_week2/dirsript/"? Why or why not?

Contents of script:

```
#!/bin/bash
```

```
mkdir -p "$1"
```

```
cd "$1"
```

```
pwd
```

```
(base) sarthdiskalkar@lawn-143-215-99-174 ~ % ls
2
Applications      Downloads          Pictures           bin                mambaforge        stderr.txt
BIO1 7200         Library           Public            dir_script.sh      mu_code           stdout.txt
Desktop           Mambaforge-Darwin-arm64.sh  PycharmProjects  directory_fix.txt  noicen            vcftools
Documents         Movies            a                 file1_copy.txt     noicen_noice
Music             anaconda3
(base) sarthdiskalkar@lawn-143-215-99-174 ~ % chmod +x dir_script.sh
(base) sarthdiskalkar@lawn-143-215-99-174 ~ % ./dir_script.sh ~/abcxyz/demo_week2/dirsript/
/Users/sarthdiskalkar/abcxyz/demo_week2/dirsript/
(base) sarthdiskalkar@lawn-143-215-99-174 ~ % pwd
/Users/sarthdiskalkar
(base) sarthdiskalkar@lawn-143-215-99-174 ~ % ./dir_script.sh ~/abcxyz/demo_week2/dirsript/
/Users/sarthdiskalkar/abcxyz/demo_week2/dirsript/
(base) sarthdiskalkar@lawn-143-215-99-174 ~ % ls
2
Applications      Downloads          Pictures           anaconda3          file2              numberpart2.txt
BIO1 7200         Library           Public            bin                mambaforge        stderr.txt
Desktop           Mambaforge-Darwin-arm64.sh  PycharmProjects  dir_script.sh      mu_code           stdout.txt
Documents         Movies            a                 directory_fix.txt  noicen            vcftools
Music             abcxyz
(base) sarthdiskalkar@lawn-143-215-99-174 ~ % rm -r abcxyz
(base) sarthdiskalkar@lawn-143-215-99-174 ~ % ls
2
Applications      Downloads          Pictures           bin                mambaforge        stderr.txt
BIO1 7200         Library           Public            dir_script.sh      mu_code           stdout.txt
Desktop           Mambaforge-Darwin-arm64.sh  PycharmProjects  directory_fix.txt  noicen            vcftools
Documents         Movies            a                 file1_copy.txt     noicen_noice
Music             anaconda3
(base) sarthdiskalkar@lawn-143-215-99-174 ~ % source ./dir_script.sh ~/abcxyz/demo_week2/dirsript/
/Users/sarthdiskalkar/abcxyz/demo_week2/dirsript/
(base) sarthdiskalkar@lawn-143-215-99-174 dirsript %
```

When source wasn't called, the working directory did not move to "~/abcxyz/demo_week2/dirsript/". This is because the shell session works independently from the script's subprocess. When a script is run, the changes that occur to the working directory only affect the script subprocess. However, when source is called, the script commands are executed in the current shell. So basically, if source is not called, the script and current shell session act independently.

2. "change_headers.sh" (30 points)

The next two questions will be linked as you will be using the same data to answer them both. Each question will begin with a description of the data you will be manipulating and an explanation of why you are performing the manipulation. These scripts are genuinely useful in a bioinformatic analysis and we shall develop them over the coming weeks using the concepts we cover in class.

The following section is a description of FASTA format data. If you are already familiar with FASTA format sequence data, all the information required to complete this task is given in the "Your task" section below.

Your task

Write a script that adds the accession number (i.e., the filename without the extension) to the beginning of each header line in the file and outputs the modified FASTA sequence to a new file. The output file must be a valid FASTA format (i.e., follow the format rules described above).

The usage of your script should be `./change_headers.sh <input file> <output file>` Hint: `sed` can use environment variables in expressions when the expression is in *double* quotes.

In your submission sheet include the following:

The content of your script

A screenshot of a terminal in which you show the following using issued commands:

The first 3 headers in one of the assembly files

The execution of your script on the file you showed headers from

The first 3 headers in the output file showing the expected modifications

Content of script:

```
#!/bin/bash
```

```
inputFile="$1"
```

```
outputFile="$2"
```

```
awk '/^>/{header=$0; sub(/^>/', '', header); print ">"FILENAME"_header; next} {print}' "$inputFile" > "$outputFile"
```

```
echo "The top 3 headers in the output file: "
```

```
#grep '^>' "$inputFile"|head -n 3 "$outputFile"
```

```
grep -m 3 '^>' "$outputFile"
```

```
(base) sarthdiskalkar@lawn-143-215-99-174 Exercise 2 % cat ERR430992.fna | grep '^>' ERR430992.fna | head -n 3
>NODE_1_length_922990_cov_42.400140
>NODE_2_length_692635_cov_34.522667
>NODE_3_length_661814_cov_39.894131
(base) sarthdiskalkar@lawn-143-215-99-174 Exercise 2 % ./change_headers.sh ERR430992.fna output3.fasta
The top 3 headers in the output file:
>ERR430992.fna_NODE_1_length_922990_cov_42.400140
>ERR430992.fna_NODE_2_length_692635_cov_34.522667
>ERR430992.fna_NODE_3_length_661814_cov_39.894131
(base) sarthdiskalkar@lawn-143-215-99-174 Exercise 2 %
```

```
(base) sarthdiskalkar@lawn-143-215-99-174 Exercise 2 % cat ERR430992.fna | grep '^>' ERR430992.fna | head -n 3
>NODE_1_length_922990_cov_42.400140
>NODE_2_length_692635_cov_34.522667
>NODE_3_length_661814_cov_39.894131
(base) sarthdiskalkar@lawn-143-215-99-174 Exercise 2 % ./change_headers.sh ERR430992.fna output3.fasta
The top 3 headers in the output file:
>ERR430992.fna_NODE_1_length_922990_cov_42.400140
>ERR430992.fna_NODE_2_length_692635_cov_34.522667
>ERR430992.fna_NODE_3_length_661814_cov_39.894131
(base) sarthdiskalkar@lawn-143-215-99-174 Exercise 2 % ls
CRISPR_1f.fna      ERR431227.fna      change_headers.sh  dir_script.sh      output1.fasta      sdiskalkar3.docx
ERR430992.fna     Exercises-Week2.pdf dir_script          output.fasta       output3.fasta      ~$iskalkar3.docx
(base) sarthdiskalkar@lawn-143-215-99-174 Exercise 2 %
```



```
#!/bin/bash
```

```
query_file="$1"  
subject_file="$2"  
output_file="$3"
```

```
blastn -query "$query_file" -subject "$subject_file" -task blastn-short -outfmt '6 qseqid sseqid  
length nident' | awk '$3 == $4' > "$output_file"  
num=$(wc -l < "$output_file")
```

```
echo "Number of perfect matches: $num"
```