BINF2111 – Introduction to Bioinformatics Computing

Github and Research Cluster – git'ing ready

Richard Allen White III, PhD
RAW Lab
Lecture 12 – Tuesday Oct 5th, 2021

Learning Objectives

- Review quiz/bonus

- Review lab 5

- Make a github account

- Github terminal authentication

- Github functions

- Quiz 12

Quiz 11 answers

```
Function_name[ ]{
command
}
```

This is a the proper syntax for a bash function?

T or F

Quiz 11 answers

```
Function_name[ ]{
command
}
```

This is a the proper syntax for a bash function?

T or F

Quiz 11 answers

```
for ((i = 0 ; i < 100 ; i++))
do
    command $i
done
```

This is a regular style for loop in bash?

T or F

Quiz 11 answers

```
for ((i = 0 ; i < 100 ; i++))
do
    command $i
done
```

This is a regular style for loop in bash?

T or F

Bonus 10

- Write a function that will return the number of lines it has in it?

Bonus 10

- Write a function that will return the number of lines it has in It?

```
1 #!/bin/bash
2
3 lines_in_file () {
4     cat $1 | wc -l
5 }
6
7 num_lines=$( lines_in_file $1 )
8
9 echo Your file $1 has $num_lines lines in it.
```

Lab 5 Question 1

Write a bash script that prints any range of lines you give it from a file.

For example, bash script.sh file.tsv (prints lines 2 to 5). Place code here and make an executable script.

HOW WOULD YOU DO THIS?

Lab 5 Question 1

Write a bash script that prints any range of lines you give it from a file.

For example, bash script.sh file.tsv (prints lines 2 to 5). Place code here and make an executable script.

HOW WOULD YOU DO THIS?

```
1 #!/bin/sh
2
3 filename="$1"
4 firstline=$2
5 lastline=$3
6 linestoprint=$(($lastline-$firstline+1))
7
8 tail -n +$firstline "$filename" | head -n $linestoprint
```

Another way?

Lab 5 Question 1

Write a bash script that prints any range of lines you give it from a file.

For example, bash script.sh file.tsv (prints lines 2 to 5). Place code here and make an executable script.

HOW WOULD YOU DO THIS?

```
1 #!/bin/bash
2
3 range_start=$2
4 range_end=$3
5
6 cat --number $1 | head -$range_end | tail +$range_start
```

Another way?

Lab 5 Question 1

Write a bash script that prints any range of lines you give it from a file.

For example, bash script.sh file.tsv (prints lines 2 to 5). Place code here and make an executable script.

HOW WOULD YOU DO THIS?

```
1 #!/bin/bash
2
3 filename=$1
4 firstline=$2
5 lastline=$3
6
7 sed -ne "${firstline},${lastline}p;${lastline}q" < ${filename}
```

Lab 5 Question 2

Oh NO! You tried to save your file from a text editor and it got corrupted (corrupted.fastq).You must now find where the file is corrupted by looking at what is missing in the file?

Write a command or script (better) to find the corrupted lines then print their line numbers.

HOW WOULD YOU DO THIS?

```
1 #!/bin/bash
2
3 diff $1 $2 --color
```

Another way?

Lab 5 Question 2

Oh NO! You tried to save your file from a text editor and it got corrupted (corrupted.fastq).You must now find where the file is corrupted by looking at what is missing in the file?

Write a command or script (better) to find the corrupted lines then print their line numbers.

OUTPUT

```
(base) docwhite@system76-pc:~/Desktop/BINF2111/data$ bash Lab5_q2.sh MultiN.fastq corrupted.fastq
93,94c93
< @M00618:17:000000000-A31W7:1:1101:13859:2209 1:N:0:21
< GGAGGCGAGAGCGGAGACGACGGTGATCTGGGAGATCGACGAGGGGCAGAAGCGCATCGAGGGCNNNNACGAGGCCGGTCGCCCCGACCCCGCCTACGCGGCCGCGCTCGGGCTCC
GCCCTGCCCTCGCTCGTGCGGATCGCGACCGCACGCTCCGTCATGCGCG
---
> TGATCTGGGAGATCGACGAGGGGCAGAAGCGCATCGAGGGCNNNNACGAGGCCGGTCGCCCCGACCCCGCCTACGCGGCCGCGCTCGGGCTCCGCCCCGCGCCGCTCGGCCGTCGG
CGCGACCGCACGCTCCGTCATGCGCG
1421,1424c1420
< @M00618:17:000000000-A31W7:1:1101:15910:4449 1:N:0:21
< CACGAGGCGGTGCGCGAGCGCCGGCACGGCGAGCGCCTTGACGTCGTCGGGGGTCACGTAGTCGCGGCCGTGGATCGCTGCCCACGCGCGCGCCGTCCGCACGAGCGCGAGCGCGG
ACGTGGACGCCGCGCACCTGGTCCGCGAGCGGGCCGGACGGCCATGTGC
< +
< AAAAAAAADEDDDDDGGGGGGHHHHHHHHHHHHHHHGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGG
GA84?88*''5'.'..'8:*****...5'4.''''''''''''''*****
---
> AAADEDDDDDDGGGGGGHHHHHHHHHHHHHHHHGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGG
88*''5'.'..'8:*****...5'4.''''''''''''''*****
```

Lab 5 Question 2

Oh NO! You tried to save your file from a text editor and it got corrupted (corrupted.fastq).You must now find where the file is corrupted by looking at what is missing in the file?

Write a command or script (better) to find the corrupted lines then print their line numbers.

```
#!/bin/bash

original=$1
corrupted=$2
original_count=`grep "@" $original | wc -l`
corrupted_count=`grep "@" $corrupted | wc -l`

Corrupt_check(){
    diff $original $corrupted --color
}

lines_diff=$(Corrupt_check)

output=$(($original_count-$corrupted_count))

echo -n "Your original file has:"
echo $original_count
echo -n "Your corrupted file has:"
echo $corrupted_count
echo -n "Difference in line numbers:"
echo $output
echo -n "Difference (diff) command found these lines to be different:"
echo $lines_diff
```

Lab 5 Question 2

Oh NO! You tried to save your file from a text editor and it got corrupted (corrupted.fastq).You must now find where the file is corrupted by looking at what is missing in the file?

Write a command or script (better) to find the corrupted lines then print their line numbers.

OUTPUT

```
(base) docwhite@system76-pc:~/Desktop/BINF2111/data$ bash lab5_q2.sh MultiN.fastq corrupted.fastq
Your original file has:3463
Your corrupted file has:3461
Difference in line numbers:2
Difference (diff) command found these lines to be different:93,94c93 < @M00618:17:000000000-A31W7:1:11
CGACCCCGCCTACGCGGCCGCGCTCGGGCTCCGCCCCGCGCCGCTCGGCCGTCGCGCGCTCGCGTCGATCATCGAGCTCTCGCTCTACGCGCTGCTGCAGCTC
GGGCNNNNACGAGGCCGGTCGCCCCGACCCCGCCTACGCGGCCGCGCTCGGGCTCCGCCCCGCGCCGCTCGGCCGTCGCGCGCTCGCGTCGATCATCGAGCTC
00618:17:000000000-A31W7:1:1101:15910:4449 1:N:0:21 < CACGAGGCGGTGCGCGAGCGCCGGCACGGCGAGCGCCTTGACGTCGTCC
GTCGCGTCGACGAGCCGCGCGACGTAGTCGACGATCGACGCCTCGACGTGGACGCCGCGCACCTGGTCCGCGAGCGGGCCGGACGGCCATGTGC < + < AA
GGGGGGGGGGGGGGGAGGGGGGGECEEDGGDGGGGGGDA>GGEGDGGD<DGGGGGGGGDGGDAD>AGGGG<8?CC?8.4<>E.<C8>'.9CEGA84?88*''5
```

Lab 5 Question 3

Write a bash script or command to find sequences with >4 N's in the sequence (MultiN.fastq), have it print the header of the sequence and the sequence itself.

```
#!/bin/bash

input=$1

N_detector(){
    grep -B 1 "NNNN" $input
}

output=$(N_detector)

echo $output
```

Lab 5 Question 3

Write a bash script or command to find sequences with >4 N's in the sequence (MultiN.fastq), have it print the header of the sequence and the sequence itself.

OUTPUT

```
CGGGCCGCTGGGACTACCTGTTCTCGATGATCAAGNCCTTCCGGCNNCGCNGGCCNGACTNCGTGCTGCCGGANCGCGCNNNGCNNANGCTGCNCGCNCNNCNCNNCNNNNNNNNNGGACTATCTGCGGT
(base) docwhite@system76-pc:~/Desktop/BINF2111/data$ bash lab5_q3.sh MultiN.fastq >lab5_q3_out.txt
(base) docwhite@system76-pc:~/Desktop/BINF2111/data$ more lab5_q3_out.txt
@M00618:17:000000000-A31W7:1:1101:15235:1847 1:N:0:21 GTGCAGCTGCTGAAGTCGTTCACATTCCTGCCGTCGGCTCGCATCGCCGAGCTCGAGCGCGCGACGGCCGAGGNNNNGTTCCGCCGCGAGGCGCAGCGCGAGCTCGCGCTGCAGGCGACCGCCTTCATCCACGGTCGCGCCGCGACGGAGCAGGCGA
TCGCCGCATCGGAGGCGCTCTTCGGTGCCGGGGATCTCGCGGCGCTCGACGCCGGCACCCTCGCAGACTCGGTTTGGCGTGAGCCGCAGGGCGC -- @M00618:17:000000000-A31W7:1:1101:17204:1912 1:N:0:21 CGCAGCATCAAGAACCGCACCGGCATCGGCGAGGCGGGCCGCTCGATCGTGCGCCTCGC
GCTCGANNNNNNNNGCGCTCGCGTCGACCTCGCCGGCGCCCGCGTGCTGCTCGTCGGCACCGGCCGCTTCGCGGGCGTCTCGCTCGCCGAGCTGCGCCGCCGTGGCGCGAAGCACATCGAGGTGTGGAGGCCCTCGAACCGCGGCGAGGGCATCGCCCGCAGCCACTCGGTAGAGGAAGTACCACCGGACCT -- @M00618:17:0000
00000-A31W7:1:1101:13859:2209 1:N:0:21 GGAGGCGAGAGCGGAGACGACGGTGATCTGGGAGATCGACGAGGGGCAGAAGCGCATCGAGGGCNNNNACGAGGCCGGTCGCCCCGACCCCGCCTACGCGGCCGCGCTCGGGCTCCGCCCCGCGCCGCTCGGCCGTCGCGCGCTCGCGTCGATCATCGAGCTCTCGCTCTAC
GCGCTGCTGCAGCTGCCGTACTGGCTCGTCGCCCTGCCCTCGCTCGTGCGGATCGCGACCGCACGCTCCGTCATGCGCG -- @M00618:17:000000000-A31W7:1:1101:20727:2816 1:N:0:21 GTGCCCATGATCTCGGCCGCGCGGACCGCGAGGCCCGCGATGCGCTCGAGCTTCGCCTCGAGCNNNNNNNNNNC
CNGNAACGCGAGCCGGGCGTTCCAGCCGACGGCGCCGCGGTTGAAGGCGAGCATCGTCTCCGCGTCGCGGATGAGGACGTCGTCCCACGGCACGAGCACGTCGTCGAACACGACGAGCGCGACGTGCTCCTGGAAGCGCCGCGCGAGCGGCTGGTCTCAAGAATGCCCGCCCTGCCC -- @M00618:17:000000000-A31W7:1:1
101:23681:4443 1:N:0:21 GCGACGACGATGAACTCGCTCGCCTGCTCGAGCTCCTCGTGCGTGTCGAGCGTGNNNNCGTAGNNNNNNNNNNNANNNGTNACGCGNNAGANNGCGATCTCGNNNNNGTCCANCNNNCNGTTGCCGATGAGCGTCAGCAGCAGGTCGAAGGGGCCGTCGAAGNNGTNNNNCNCGNGGCGGAAGCCGC
GNCGCTCCTCCNCNGCNGCCGACTCCCCNGCNCCTGACGCGCCGGGTAGCCCTCGGCCGATG -- @M00618:17:000000000-A31W7:1:1101:24313:5033 1:N:0:21 GCCCCGACCTGCCGGTGCGCCGCGTCGTGCTGACGGGCGACGGCGCGAGCCTCGGNNNNNTNNNNNNNNNNNNNNNNNNNTNNTGNNCTT
NNCCATCGACNGNNNNGACCNNNNNNNNTCGGTGCTGACGAAGGAGCGCGACCGCCCTCGCACGANCGGCGCGGCGAACCGAGCGNCCGTCGCGCTGGGCCTGNCGATGCNNCAGATCTGAAGNGGCNANNCCTGNACTCCCCCACGGCGCGTCTCTACG -- @M00618:17:000000000-A31W7:1:1101:26789:7963 1:
N:0:21 CGCGTCGGCGGCGTGCCGGGGCCGGTGATCGAGACGCAGGGGCTCGAGAAGCGATTCGGGCGCGNNNNNNNNNTCGACGGGCTCGACCTCGTCGTCGAGCAGGGGCAGGTGCACGGCTTCCTGGGCCCGAACGGCGCCGGCAAGTCGACGGCGATCCGGGTGCTGCTCGGGATGCTGCGTGCGACGGGCGGCGAGGCGAGCGTC
TTCGGCATGCACCCGTGTCGCGACGCAGCGAATCAACCGGCTGCTGG -- @M00618:17:000000000-A31W7:1:1101:27801:9262 1:N:0:21 TCGGCCGTCTCGCCGCCGACGAGCGCCGTGCCGGTCGCCTCGCACGCCTCGGCGATGCCGCGNNNNNNNNNNNNNNNNNNTCGGGCACGACGCGCCCGCACGCGA
TGTAGTNNNNNATGAACAGCGGCTTCGCGCCGACCACGACGATGTCGTCGACGACCATGCCGACGAGGTCCTGCCCGATCGTGTCGTGCTTGTCGATCGCCTGGGCGAGCGCGAGCTTCGTGCNGAACTTACGTCGGAGCCGGGC -- @M00618:17:000000000-A31W7:1:1101:27772:9473 1:N:0:21 TGCCTCAA
GCTGCTGCGCGCCTTCGAGTTCGGGCGCATCGCGTTCCGCATCGCCGACGCGCTNNNNNNNNNNNNNNNNNACTACCACGCCGACGGCCGGCACATCACGTTCCNNACCGCGCCGGGCACGAAGCTCGCGGGGGTGCTGCTCGCCGACGACGTCGTGTTCGAGATCGACCACATCGGCGAGCGCGAGGCGTGGAGCGTTGTCGCCCACGGC
AGGGCGCGCCGGATGATCCAACAGCAGAGCCT -- @M00618:17:000000000-A31W7:1:1101:28149:9970 1:N:0:21 GATCGGCTCGGCATCGATCGCGGCACGATCCGGGCGACCGTGCTCATCGAGACGNNNNNNGNNNNNNNNNNNNNNNNNNNNNNCTNNGGCNNCNGNGCGANCNNNNGGCCNNNNNNNNCG
CGGGCCGCTGGGACTACCTGTTCTCGATGATCAAGNCCTTCCGGCNNCGCNGGCCNGACTNCGTGCTGCCGGANCGCGCNNNGCNNANGCTGCNCGCNCNNCNCNNCNNNNNNNNNGGACTATCTGCGGT
```

Lab 5 Question 4

Write a bash script that counts the number of ATG (starts), Serine (S), Arginine (R), and TAA, TAG, TGA (stops) from the example2.fasta file then converts them into amino acid M, S, R, and * for TAA, TAG, TGA for stop codon.

Remember that ATG encodes for methonine so they only count as start from the beginning of the sequence or the end for the stops. Serine and Arginine can be throughout the sequence.

HOW WOULD YOU DO THIS?

Lab 5 Question 4

```bash
#!/bin/bash

input=$1

#Count start codons
count_starts(){
    grep "^ATG" $input | wc -l
}

number_starts=$(count_starts)

#Count stop codons
count_stops(){
    egrep "TAA$|TAG$|TGA$" $input | wc -l
}

number_stops=$(count_stops)

#Count codons for arginine
count_arg(){
    egrep "CGT|CGC|CGA|CGG|AGA|AGG" $input | wc -l
}

number_arg=$(count_arg)

#Count codons for serine
count_ser(){
    egrep "AGT|AGC|TCT|TCC|TCA|TCG" $input | wc -l
}

number_ser=$(count_ser)

#final table
echo -n "Number of starts: "
echo $number_starts
echo -n "Number of stops: "
echo $number_stops
echo -n "Number of Arginine: "
echo $number_arg
echo -n "Number of Serine: "
echo $number_ser

#Count codons for arginine
count_arg(){
    egrep "CGT|CGC|CGA|CGG|AGA|AGG" $input | wc -l
}

number_arg=$(count_arg)

#Count codons for serine
count_ser(){
    egrep "AGT|AGC|TCT|TCC|TCA|TCG" $input | wc -l
}

number_ser=$(count_ser)

#final table
echo -n "Number of starts: "
echo $number_starts
echo -n "Number of stops: "
echo $number_stops
echo -n "Number of Arginine: "
echo $number_arg
echo -n "Number of Serine: "
echo $number_ser

#Convert codons
convert_cod(){
    sed 's/^ATG/M/g' $input |
    sed 's/TAA$/*/g' |
    sed 's/TAG$/*/g' |
    sed 's/TGA$/*/g' |
    sed 's/CGT/S/g' |
    sed 's/CGC/S/g' |
    sed 's/CGA/S/g' |
    sed 's/CGG/S/g' |
    sed 's/AGA/S/g' |
    sed 's/AGG/S/g' |
    sed 's/CGT/R/g' |
    sed 's/CGC/R/g' |
    sed 's/CGA/R/g' |
    sed 's/CGG/R/g' |
    sed 's/AGA/R/g' |
    sed 's/AGG/R/g'
}
convert=$(convert_cod)

echo "Converting codons to listed amino acids: "
echo $convert
```

Lab 5 Question 4

Write a bash script that counts the number of ATG (starts), Serine (S), Arginine (R), and TAA, TAG, TGA (stops) from the example2.fasta file then converts them into amino acid M, S, R, and * for TAA, TAG, TGA for stop codon.

Remember that ATG encodes for methonine so they only count as start from the beginning of the sequence or the end for the stops. Serine and Arginine can be throughout the sequence.

OUTPUT

```
ATGCTAAGGCTATCNNGACAACTGACTAAATAG
(base) docwhite@system76-pc:~/Desktop/BINF2111/data$ bash lab5_q4.sh example2.fasta
Number of starts: 7
Number of stops: 7
Number of Arginine: 7
Number of Serine: 5
Converting codons to listed amino acids:
>chr1_geneA MCTASCTATCTTGACAACTGACTGCC* >chr1_geneB MCTASCTATGTTGGCAACTGACTCCC* >chr1_geneC MCTASCTACCTTGACAACTGACTGGG* >chr1_geneD MAAASCTATCTTGACAACTGACTCCC* >chr1_geneX MCTASCTATCTTGATTTCTGACTTTT* >chr
1_geneY MGGGGGGCTATCTTGACAACTGACTGCG* >chr1_geneZ MCTASCTATCNNGACAACTGACTAAA*
```

Lab 5 Question 5

Write a bash script that prints username, current directory, the location of our desktop directory, and the date/time

```
1  #!/bin/bash
2
3  echo -n "My user name is: "
4  whoami
5  echo -n "My current directory is: "
6  pwd
7  echo -n "My Desktop directory is: "
8  echo $HOME/Desktop
9  echo -n "The date and time is: "
10 date
```

Lab 5 Question 5

Write a bash script that prints username, current directory, the location of our desktop directory, and the date/time

OUTPUT

```
(base) docwhite@system76-pc:~/Desktop/BINF2111/data$ bash lab5_q5.sh
My user name is: docwhite
My current directory is: /home/docwhite/Desktop/BINF2111/data
My Desktop directory is: /home/docwhite/Desktop
The date and time is: Mon Oct  4 22:18:06 EDT 2021
```

Github

- What is github?

Github

- What is github?

GitHub, Inc. is a provider of Internet hosting for software development and version control using Git. It offers the distributed version control and source code management functionality of Git, plus its own features.
Founded 2008 (56 million, owned by microsoft)

Github

- Alternatives to github?
  - gitlab (https://about.gitlab.com/)
  - bitbucket (https://bitbucket.org/)

Github

- Uses of git:
  - Code collaboration
  - Version control
  - Safe storage of code

Github – Create an Account

github.com

- Click “Sign Up” at the top-right
- Use your UNCC email
- 2-5 collaborators is fine
- For Features I suggest “collaborative coding”
- Free account is OK, It can also be upgraded for free:
  - https://education.github.com/discount_requests/student_application
  - Select: “University of North Carolina at Charlotte”
  - Make sure to use your @uncc.edu address for this

Github – Example Markdown (Readme)

```
# Metaome Stats: Calculating denovo assembly statistics from metaomes
==============================================

## Installing
- pip installation <br /> (LINE BREAKS)
`pip install MetaomeStats`

- conda installation <br /> (LINE BREAKS)
`conda install -c bioconda MetaomeStats`

- source (github)

```bash
git clone https://github.com/raw-lab/metaome_stats
cd metaome_stats
python setup.py
```

Usage examples

```bash
```

https://github.com/raw-lab/metaome_stats
```

Github – terminal authentication

- Github no longer allows password authentication. There are two options to authenticate
  - ssh authentication (more involved setup)
  - Personal Access Token (Easier setup)
  - https://docs.github.com/en/authentication/keeping-your-account-and-data-secure/creating-a-personal-access-token
  - When git asks you for a password, use the created token instead of your password

  The command on the terminal “git config --global credential.helper store” will tell git to store your password, it saves it as plain text (no encryption) so it is not secure. This is the reason for the Personal Access Token, giving limited access to anyone that may steal this token. SSH is much better, we may set that up on a future date.

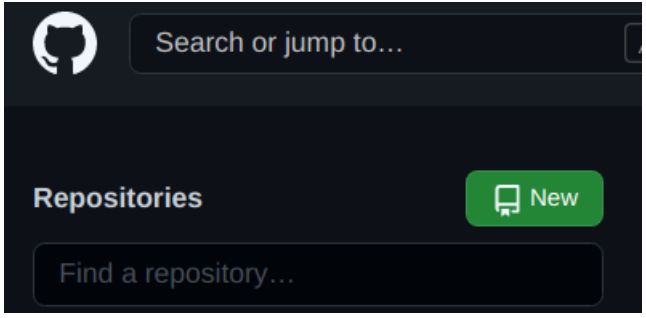
Github – Basic Functions

- Pulling data from github
  - git clone https://github.com/....
  - Git pull (for updating repo after someone else makes changes)
- Git needs to authenticate before you can push, or pull from a private repo.
- Push to github (assuming you are in root directory of the code)
  - git add .
  - git commit –m “Description of changes”
  - git push

Github – Basic Functions

- Show changes
  - git status
  - git fetch (checks server for updates without ‘pulling’ them)
- Create a new branch (for working on fixes/features)
  - git checkout –b <branch name> (i.e. dev)
- Switch between branches
  - git checkout <name>
- List local and remote branches
  - git branch -a

Github – Practice: Hello World



- Create a new repository on github
  - Repository name: hello_world
  - Select public and add the README file
- Now to clone it in your home directory
  - On the terminal type ‘cd’ to make sure you are in your home
  - git clone https://github.com/<username>hello_world
  - Type ‘ls’ to confirm the new folder “hello_world” is there
- Congratulations on creating your first github repo!

Github – Practice: Making Changes

- Change the README file:
  - cd hello_world
  - ls
- You should see README.md in the directory. With your favorite text editor change the readme file (vim, nano, gedit, echo…)
  - Add the line '## created by <your name>" and save the file
- Push your changes: (type 'git status' after each command)
  - git add README.md
  - git commit –m "Added author to README"
  - git push
- Now on your browser at https://github.com/<username>/hello_world you will see your updated README file

Quiz 12

- On canvas now

Bonus 11

- Write a bash script that converts this into a tsv file (specific files and all files)?

With this file:
Rat,steven,bear,Xi
Olf,thor,flower,Ton
Lazy,aaron,larry,jose
old,thor,flower,Ton