Teaching Guide: NH-INBRE Bioinformatics Course Curriculum

Devin Thomas[1]
Jordan Ramsdell[1]
W. Kelley Thomas[1]

January 4, 2017

[1]UNH HCGS, supported by New Hampshire-INBRE through an Institutional Development Award (IDeA), P20GM103506, from the National Institute of General Medical Sciences of the NIH

Contents

Introduction

This course has been designed to adaptable from short workshops to full courses as part of NH-INBRE's goal to "Engage large numbers of undergraduate students in authentic scientific discovery using modern tools of genetics and Bioinformatics." The core of this curriculum is introducing students to bash, and the Bioinformatics tools that they can run with bash. This document is intended to provide reference material for instructors teaching this course. Including suggestions for structuring the course, as well as addressing some of the more common issues encountered in teaching.

Chapter 1

Designing a Curriculum

This chapter should help inform your decision of how to adapt this course to your specific needs. These suggestions assume that the end goal is for the students to assemble and annotate reads from a microbe they have had sequenced. If the end goal is instead purely to teach your students bash or a different pipeline that will likely change the relative value of different modules. These modules do not require that each student have their own data, however this will make it a more engaging experience.
We have broken the modules into three color coded categories: Green will correspond to elements of the curriculum that should not be missed. These are core skills which your students need to know to have a working knowledge of bash. Orange will correspond to elements of the curriculum that are important, but can be skipped to speed up the course. Red will correspond to elements which may form part of a richer computer science experience, but may not directly relate to Bioinformatics.
When possible time estimates will be included to help guide choices.

1.1 Example Curricula

1.2 Workshop

This workshop curricula is designed for a workshop where the goal is to get the students working with their own data as quickly as possible. This assumes that the students will not have time before to run a flipped class, so the structure should revolve around working through the videos, then working on their own data. If time is extremely tight note that they may not have enough time to actually run an assembly, if this is the case consider running assemblies for them beforehand.

Table 1.1: Modules and suggestion levels, with corresponding worksheet sections and video lengths

Name	Worksheet	Time
Bioinformatics in a Biology Curriculum	N/A	Video: 8 min
Principles of Genomics	N/A	Video: 10 min
The Generation of Genomic Data	N/A	Video: 5 min
Data Structures and Conventions	N/A	Video: 5 min
Sequencing and Assembling a Genome	N/A	Video: 5 min
Genome Annotation and Inferring Function	N/A	Video: 5 min
Logging on to ron	refer to online resources for putty and ssh	Video: 5 min
Anatomy of a command	1.1	Video: 4 min
Finding help	1.2	Video: 6 min
Pathways and directories	1.3, paths_worksheet.pdf	Video: 6 min
Moving about	1.4	Video: 10 min
Move copy and delete	1.5	Video: 14 min
Odds and ends	1.6	Video: 7 min
Text editors: nano	1.7	Video: 5 min
Text viewing	1.8	Video: 7 min
Grep	1.9	Video: 16 min
Pipes and Redirecting	1.10	Video: 15 min
Globbing and Wildcards	1.11	Video: 13 min
Variables	1.12	Video: 6 min
Configuration	1.13	Video: 6 min
File Permissions	1.14	Video: 12 min
Shell scripts	1.15	Video: 13 min
Nohup	N/A	Video: 10 min
Trimmomatic	Pipeline Worksheets	Video: 13 min, runtime: short
Spades	Pipeline Worksheets	Video: 9 min, runtime: long
Quast	Pipeline Worksheets	Video: 5 min, runtime: short
Prokka	Pipeline Worksheets	Video: 8 min, runtime: medium

Table 1.2: Workshop curriculum, concentrates on following through the videos as a group, then running the pipeline on their data.

Name	Worksheet	Time
Bioinformatics in a Biology Curriculum	N/A	Video: 8 min
Principles of Genomics	N/A	Video: 10 min
The Generation of Genomic Data	N/A	Video: 5 min
Data Structures and Conventions	N/A	Video: 5 min
Sequencing and Assembling a Genome	N/A	Video: 5 min
Genome Annotation and Inferring Function	N/A	Video: 5 min
Logging on to ron	refer to online resources for putty and ssh	Video: 5 min
Anatomy of a command	None	Video: 4 min
Finding help	None	Video: 6 min
Pathways and directories	None	Video: 6 min
Moving about	None	Video: 10 min
Move copy and delete	None	Video: 14 min
Text editors: nano	None	Video: 5 min
Text viewing	None	Video: 7 min
Grep	None	Video: 16 min
Trimmomatic	Pipeline Worksheets	Video: 13 min, runtime: short
Spades	Pipeline Worksheets	Video: 9 min, runtime: long
Quast	Pipeline Worksheets	Video: 5 min, runtime: short
Prokka	Pipeline Worksheets	Video: 8 min, runtime: medium

Table 1.3: Laboratory Curriculum, Tries to give a more rigorous approach to bash, while being as efficient as possible

Name	Worksheet	Time
Bioinformatics in a Biology Curriculum	N/A	Video: 8 min
Principles of Genomics	N/A	Video: 10 min
The Generation of Genomic Data	N/A	Video: 5 min
Data Structures and Conventions	N/A	Video: 5 min
Sequencing and Assembling a Genome	N/A	Video: 5 min
Genome Annotation and Inferring Function	N/A	Video: 5 min
Logging on to ron	refer to online resources for putty and ssh	Video: 5 min
Anatomy of a command	1.1	Video: 4 min
Finding help	None	Video: 6 min
Pathways and directories	paths_worksheet.pdf	Video: 6 min
Moving about	1.4	Video: 10 min
Move copy and delete	None	Video: 14 min
Odds and ends	1.6	Video: 7 min
Text editors: nano	None	Video: 5 min
Text viewing	1.8	Video: 7 min
Grep	1.9	Video: 16 min
Pipes and Redirecting	1.10	Video: 15 min
Globbing and Wildcards	None	Video: 13 min
Variables	None	Video: 6 min
Shell scripts	1.15 Very dependant on time	Video: 13 min
Nohup	N/A	Video: 10 min
Trimmomatic	Pipeline Worksheets	Video: 13 min, runtime: short
Spades	Pipeline Worksheets	Video: 9 min, runtime: long
Quast	Pipeline Worksheets	Video: 5 min, runtime: short
Prokka	Pipeline Worksheets	Video: 8 min, runtime: medium

1.2.1 Laboratory

This "Lab" curricula is intended to fit a lab style situation, where the total class hours are not yet enough to devote time to learning bash for bash's sake but where the students can be assigned videos to watch, and the class can be flipped. In general divide the sections up roughly proportional to their video length, in class use material from the worksheets to reinforce what they learned in the videos. Consider assigning reading from the worksheets and leaving just the problems for in class. The general goal with this curriculum it to efficiently cherry pick sections, especially from the worksheet which are the most useful and give the most return for time invested.

1.2.2 Course

This course curricula should be appropriate for fitting these modules in as a substantial part of a course. As with the lab it is very beneficial to run this as a flipped course, focusing class time on working through examples, worksheets or homework. Home works and a project are available. Consider having the students work on some sort of final project, either related to the assembly and annotation of their samples, or another project letting them have an experience applying what they have learned to biology.

1.3 Recommendations for Biosafety

One of the core activities of a lab or course version of the curriculum is the opportunity to discover and characterize a new organism. Fundamentally while it is true that they are not culturing known pathogens and by definition these are naturally occurring in the environment they are bringing the microbes to high concentration. As a result it is recommended to error on the side of safety and use standard BSL-2 practices. For the most part, these are normal universal precautions (treating the microbial agent as if potentially infectious) used whenever you have "unknowns".
Two resources that serve as guidelines for handling of microbiological agents:
Biosafety in Microbiological and Biomedical Laboratories (CDC/NIH): `http://www.cdc.gov/biosafety/publications/bmbl5/bmbl.pdf`
ASM Guidelines for Biosafety in Teaching Laboratories: `http://www.asm.org/images/asm_biosafety_guidelines-FINAL.pdf`

1.4 Publishing as a Goal

One of the important aspects of this curriculum is that the students are doing real science and making unique discoveries. This is both and exciting opportunity and a responsibility.

Because each of this microbes is almost certainly going to represent a unique organisms that is new to science the data can and should be published and made accessible to the broader scientific community. There are two basic paths that can be followed. First, all reasonably complete data sets should be submitted as draft assemblies to GenBank. As an even higher goal we encourage students and faculty to publish their high quality draft assemblies genomes as genome announcements `http://genomea.asm.org/`. Toward these goals it is important (mandatory) that a minimum set of metadata be collected to accompany each new genome. A worksheet describing the kinds of information can be downloaded from the course website. Each student should make a metadata record for each microbe. This data will accompany the submission to GenBank.

Table 1.4: Full course curriculum

Name	Worksheet	Time
Bioinformatics in a Biology Curriculum	N/A	Video: 8 min
Principles of Genomics	N/A	Video: 10 min
The Generation of Genomic Data	N/A	Video: 5 min
Data Structures and Conventions	N/A	Video: 5 min
Sequencing and Assembling a Genome	N/A	Video: 5 min
Genome Annotation and Inferring Function	N/A	Video: 5 min
Logging on to ron	refer to online resources for putty and ssh	Video: 5 min
Anatomy of a command	1.1	Video: 4 min
Finding help	1.2	Video: 6 min
Pathways and directories	1.3, paths_worksheet.pdf	Video: 6 min
Moving about	1.4	Video: 10 min
Move copy and delete	1.5	Video: 14 min
Odds and ends	1.6	Video: 7 min
Text editors: nano	1.7	Video: 5 min
Text viewing	1.8	Video: 7 min
Grep	1.9	Video: 16 min
Pipes and Redirecting	1.10	Video: 15 min
Globbing and Wildcards	1.11	Video: 13 min
Variables	1.12	Video: 6 min
Configuration	1.13	Video: 6 min
File Permissions	1.14	Video: 12 min
Shell scripts	1.15	Video: 13 min
Nohup	N/A	Video: 10 min
Trimmomatic	Pipeline Worksheets	Video: 13 min, runtime: short
Spades	Pipeline Worksheets	Video: 9 min, runtime: long
Quast	Pipeline Worksheets	Video: 5 min, runtime: short
Prokka	Pipeline Worksheets	Video: 8 min, runtime: medium

Chapter 2

Preparation

Once you have at least a loose plan for what modules you would like to use, it is now time to prepare. The focus of this section is to lay out both what you need to do and our suggestions for a smooth experience.

2.1 Getting accounts on ron

The server ron.sr.unh.edu has been built to facilitate this curriculum. It has been configured to ensure that all of the software referenced by this material works properly. Additionally beyond this it has a full compliment of Bioinformatics tools (sampled from tools used by researchers here at UNH). If you would like to use software not installed on ron as part of your class please contact us and we would be happy to make it available on ron. This course is setup around and possible because of Open Source software. One of the many downsides of using closed source software is the limiting licensing structure, which will usually prevent us from making any closed source products available on ron. Please feel free to contact us for suggestions of free and open source alternatives.

In order for you and your students to access and use ron, you need to provide us with a spreadsheet with three columns: column 1 should be the students name, column two the students preferred username this should be similar to their student email, and column three their email. Please do not include any fields with commas, colons etc. If you have not already obtained accounts for yourself and any other instructors make sure you include rows for yourselves.

The setup for classes on ron is that you and your students will all be given regular user accounts on ron which will be all in a group together. This means that your whole class has permissions to view and collaborate on files in each others directories. This is extremely useful, but keep in mind that your students have the same access to your files as you do to theirs.

2.1.1 time

This should take little to no time depending on the structure of your available class list. It will however take time on our end, so please do this well before you actually need the access.

2.2 Check your ability to connect to ron

If there is a specific classroom or location you and your students will be accessing ron together from please go there and make sure that you are able to log into ron from that location. This serves two purposes, it ensures that you are able to log in yourself, and if there are any issues with the connection it lets us debug it without a classroom of students waiting on it. Most bioinformatics is done by connecting to servers from a laptop. We encourage you to have your students use their own laptops, as it will let them learn how they can do this for any future servers they may need to connect to. It also enables them to work from anywhere they can find an internet connection.
Classes have had issues with WiFi setups that were not powerful and/or stable enough, this will manifest as students getting disconnected occasionally seemingly at random. If you know that the WiFi in the location you will be using is frequently complained about try and be proactive about either finding a different location or having IT upgrade the infrastructure.
We have taken several steps to ensure that ron can be accessed from any stable internet connection. This flexibility is only possible when we put in place other strong security measures. One of these is our decision to default you and your students to very strong passwords. We recommend that you use either that password, or a similar strength one. We also recommend that when entering your password you copy and paste. This is because after several unsuccessful attempts your account will be locked for 15 min (like a smart-phone).

2.2.1 time

Assuming no trouble this should take only a few seconds or minutes. It may take slightly longer if this is your first time logging into a server.

2.3 Go through the material yourself

Your knowledge of the material you are teaching is the single biggest factor for how smoothly this will go. If possible it can be particularly useful to have experience logging into ron on both a UNIX (mac or Linux) system via ssh and a windows computer via putty. The experience can be quite different and the putty installation and usage can be confusing because it relies on a GUI.

2.3.1 Time

The core bash curriculum is 2:30 of videos, 3:15 including the videos on specific Bioinformatics tools. Plan to spend slightly longer as you follow along with Jordan.

2.4 Go through additional course material yourself

Time and interest permitting please consider going through some or all of the rest of the available material. It can help make the difference between you being able to answer questions like "how do I pull headers out of a fasta file with grep" and questions like "Why do I use grep to pull out headers from a fasta file, and how else could I do it?" Remember that the setup for this course as applied to workshops is a crash course, and you may want to go more in depth for yourself.

2.4.1 time

The worksheet is roughly 10 pages of raw text, use that to estimate how long it would take to skim it. Consider this the lower bound. Our class of Grad students took about 10 hours of class time to completely work through the worksheets, including the paths_worksheet. Consider this the upper limit for using purely the resources made available by this course.

2.5 Ensure the students have computers to use

If the students will be bringing their own laptop be sure to remind them. If they will be using computers provided by you make sure that they work to log into ron with. When the option is present Linux is vastly preferable to OS X which is preferable to windows. Windows computers that you do not have administrative control over may not be able to connect to ron.
It is always good to have a few spare laptops available in case someone forgets a computer, or brings one which does not work. Certain hybrid tablets, like iPads and windows RT tablets and android tablets can be mistakenly brought by students who think they will work, but may not work with the tools we use. Often they could work but the setup will take much longer than is worth spending. Any computer made in the past decade should be more than capable of running everything we do, and learning to install Ubuntu on an old laptop can be a informative and fun experience for someone learning about computers, it is also quite easy in most cases.

2.5.1 time

Student provided computers or a Linux/mac computer lab should take a few minutes. Making a windows lab work may take longer or be nearly as quick. Setting up

computers will take substantially longer.

2.6 Have Helpers

TA's / random experienced students pulled from the hallway will make it much easier to help students when a few of them get stuck. Often there will be several students with different issues at the same time, and having helpers who have more experience than the students can help keep the class moving.
Leave no one behind: when working with students in an online exercise it is important to keep everyone engaged. Find a way to get students to let you know when something does not work. Helpers can be very good for this. Always encourage your students to work together problem solving, helping each other is a great way to learn. It is highly recommended that they have their own microbe and genome to work with. It is critical for student engagement for them all to have their own computer/project to work on.

Chapter 3

Modules

Each section will have some general tips for how to ensure it runs smoothly. It will then have a list of questions and answers meant to highlight some of the concepts the students should learn in this section. Thirdly there is a rough time estimate, remember to include some time for discussion!

3.1 Bash Basics: Learning to walk

3.1.1 Log on to ron

Logging onto ron is the first thing that should be done. If possible have the students log on to ron at least once before the workshop. This step is much much shorter if the students have already logged in to ron, on their own before the workshop. If a student is unable to log in, because of a unsupported operating system or any issue that you can not easily and quickly fix strongly consider giving them a loaner system to work on. Again we would like to reiterate the importance of your students having strong secure passwords.

Learning Goals

I What does ssh/PuTTY do? *It securely connects me to a remote server and allows me to give the remote server commands*

II Am I on ron or my own computer? *Think of it like ron it a remote control car, and your computer is the controller.*

III Why am I bothering with all this remote stuff when I have a computer already? *It took me (Devin) hours to install the software we use in this course on ron. If you have a Linux based operating system you might also be able to do it in hours, otherwise you will likely spend much much much longer. Ron is also many orders of magnitude more powerful than the most powerful laptops in existence.*

IV How can I tell if I am on ron? *Your prompt will look like* `uname@ron:~$`*, if it says* `names-macbook-pro:~ uname` *then you are on your macbook*

V Why should I have a secure password? *It is your responsibility to protect your account, the best and easiest way to do this is have a secure password.*

Time:

Less than a minute if the students have already logged in, and have all their account information easily accessible. It has taken nearly an hour to do this when the students were not prepared beforehand. The video for this is 5 min.

3.1.2 Anatomy of a command

This first section walks through the basic structure of commands in bash. The first five sections are crucial to being able to function at the most basic level in bash, and if you have any time to spare these first sections should be the priority, doing otherwise will have similar results to trying to teach a baby to run before they know how to walk. That being said these should be some of the shortest sections, and if your students are catching on quickly or already have some bash experience these should be extremely quick. Use the very basic examples from this section to make sure that everyone is following along and trying out the examples that Jordan covers in the videos.

Learning Goals

I What is a command? *A command is a name that bash knows how to interpret as instructions, either as a text script in bash, or python (like spades.py) or as binary 1s and 0s.*

II What is an option? *An option is something that a command looks for which modifies the behavior of the command*

III How does bash tell the difference between commands and options? *In most cases your command, and each following option will be broken up by white space. The position of these chunks of text is how bash knows what to interpret as what.*

IV Is bash case sensitive? *It is case sensitive, unlike windows cmd.exe and powershell.*

V What is the command prompt? *When we say command prompt we usually mean the text that appears before commands when you are entering them, often of the form* `uname@ron:path$` *the presence of this indicates that ron is awaiting instructions from you.*

Time:

The video is 4 min long

3.1.3 Finding Help

Learning to problem solve, and how to find help when working with bash is a crucial step towards being able to program independently. Try and encourage students to find solutions themselves with the methods they learn in this section whenever possible.

Learning Goals

I What are the standard ways of finding documentation? *Google, man, the -h and –help options, whatis*

II Which method should I prioritize? *Unless your question is very straightforward like: What was the option for ls to display hidden files? You will almost always have the easiest time by using a search engine online to find a applicable solution.*

3.1.4 Pathways and Directories

This is the first concept being covered which may be foreign to a large portion of the students. Spending time now on making sure that they understand the concepts of directories, files, absolute and relative paths now will pay dividends later, as from our experiences improper paths is the biggest source of errors for learning students.
Some of the more visual learners have been helped by having a physical example in front of them to practice with. This demo can be made by nesting boxes and containers to represent directories and placing small objects or pieces of paper inside the containers to represent files.
The paths worksheet was specifically created to help students understand the concept of paths and we recommend going through it time permitting.

Learning Goals

I Where am I? *If ron's file system is a tree you are always in exactly one place(branch).*

II I just logged in where am I? *You are in your home directory*

III Where is that thing? *let's say your coffee is on your desk, and you are sitting at your desk, the absolute path of your coffee is country, state, town, building, your desk. The relative path is "right in front of you".*

Time:

Video is 6 minutes, but plan on spending as much time as possible on this subject as it will pay off later.

3.1.5 Moving About

Provided that the student understand paths this should be a simple application of that knowledge. Consider using this section as practice for paths.

Learning Goals

I How do I move around on ron? *cd, which stands for change directory.*

II Do I need to move around? *need? no. But it is often much easier to cd around rather than typing big long relative paths.*

Time:

Video is 6 min, if the students were able to get the hang of paths it should not take much longer than the video.

3.1.6 Move Copy and Delete

Like "Moving About" the speed of this section will rely heavily on how well the students understood paths. After this section students should begin to feel like they can actually do something useful in bash. The sections following will begin to show specific tools and how they are used in the context of Bioinformatics.

Learning Goals

I What do all three of these functions have in common? *They all take paths.*

II How do I rename a file? *Move it to a new path, pointing it at the same directory just with a different filename!*

III Oops I accidentally deleted my homework, where is the recycle bin? *rm and mv are permanent, always be cautious when using them.*

IV How much damage can I do? *In theory you do not have the privileges to do any damage to the server as a whole, you can wipe out all of your own files though.*

Time:

Video is 10 min, the actual time will again depend heavily on the student's understanding of paths.

3.2 Tools in bash: A GNU way of thinking

3.2.1 Odds and Ends

For the most part this section is full of quick quality of life tips. If you do decide to skip this section strongly consider explaining at least Ctrl-C.

Learning Goals

I Stop stop stop! *ctrl-C*

II That didn't work!! *Try hitting q*

III That did not work either!!! *Just "kill it with fire", hit the x to close your terminal or putty session and start a new one.*

IV What is top? *It is like task manager in windows, it shows CPU and memory usage*

V Wow Joe is hogging ron, I can't even see my processes in top because he has so many! *If you only want to see your own processes use* `htop -u $USER`

VI What's that htop you just mentioned? *It is a prettier version of top, it is not always installed but when it is I prefer it.*

Time:

Video is 14 min, it should not take much longer than the video.

3.2.2 Text Editors

Knowing how to use a file editor is a fundamental skill in bash, the video provides a nice introduction to nano while the worksheet includes quick demos with vim and Emacs as well.

Learning Goals

I I'm on some strange new server, which of these should I try. *Pretty much any computer that isn't windows will have nano and vi, nano is easier so try that first.*

II Why can't I just edit the files on my own computer instead? *You can! It is just much more involved.*

III I like word, put word on ron. *No! Graphical applications are very annoying to setup for people using windows laptops. Additionally all of the files we are working with need to be formatted in plain text, something that windows editors, including notepad do not do properly.*

Time:

Video is 7 min, video on editing without nano is 5 min it should not take much longer unless they also do the worksheet examples.

3.2.3 Viewing Text

You can always just use editors to view text, so this section is not critical. It is very often much easier and preferable to use tools like head and less when appropriate. We strongly recommend going through this section.

Learning Goals

I Wow more is way better than less! *No it is not, never use more unless you find yourself on a computer so old that it does not have less*

II Why bother with less when I have nano? *Why read a book when I can have the author dictate it to me? Less is lighter and faster than a text editor, and it means you will not accidentally edit your data files!*

III I looked at a file and it is full of windings!! *It is probably a compressed file, use less*

Time:

Video is 7 min

3.2.4 grep

Grepping is so important it gets to be a real verb.

Learning Goals

I What is grep good at? *I have a bunch of lines of something, and I want to know something about the lines with —– in them.*

II What is grep not good at? *Grep can do a lot of stuff, but there is a point where what you are doing is so complicated you should really be using a different tool.*

Time:

Video is 16 min, This is an important subject so it is a good time to check understanding.

3.2.5 Pipes and Redirecting

Pipes and output redirection are fundamental parts of the UNIX philosophy, small simple tools chained together to solve complex problems.

Learning Goals

I What is a pipe? *A pipe is a hollow cylinder meant to convey a liquid or gas.*

II Why are |'s called pipes? *The information of your program flows from command to command, each connected by a pipe.*

III What are the downsides of pipes? *They can make your code very very difficult to parse. Always keep this in mind.*

Time:

Video is 15 min

3.2.6 Globbing and Wildcards

Globbing and wildcards is absolutely crucial in certain situations, "I have 1000 files how can I open the ones starting with Sample_12?" If you are so lucky as to have multiple files worth of reads you will need to use wildcards for trimmomatic.

Learning Goals

I Why glob? *We use glob when we want to specify the path to a bunch of files and folders at once.*

II What does * represent? *It represents any number of anything.*

III What does ? represent? *Exactly one of anything.*

Time:

Video is 13 min

3.2.7 Variables

Again, crucial for bash, not crucial for running the spades/prokka/quast pipeline.

Learning Goals

I Why are variables useful? *We can store values to use multiple times, or to use as input that is easier to do with variables than pipes.*

II I have a variable X, what is the difference between X and $X? *X is just X, but $X is saying, "give me the value held by X"*

Time:

Video is 6 min

3.2.8 Configuration

This is good to know for anyone that plans on using a Linux based system long term, however it is definitely not necessary.

Learning Goals

I Where do I change the color of my terminal to make it all cool like Jordan's? *On your own laptop, look at the settings/profiles for your terminal or PuTTY*

II What do I change to get Jordan's super cool prompt? *The PS1 variable controls what you are prompted with edit it in your .bashrc.*

III How are things usually configured in Linux? *There are almost always plain text files, like the .bashrc or .nanorc somewhere which hold the configuration for your programs.*

Time:

Video is 12 min

3.2.9 File Permissions

This is important to know if you plan on working on other systems, or long term with bash. However the accounts you get on ron are configured so that you and your students are all in the same group, so you should be able to copy stuff around between your accounts without issues in most cases.

Learning Goals

I What is a user? *You are a user, if you log into ron you have control over anything that you own.*

II What is a group? *A group is a collection of users that share something in common. In the case of Ron each class is its own group, each with their own home directories. We do this so your students can easily share files between themselves.*

Time:

Video is 13 min

3.2.10 Shell Scripts

If the focus of this program was learning to use bash this would be the most important section.

Learning Goals

I What is the shebang? `#! /bin/sh or #! /bin/bash`

II What does it do? *When bash runs a file it looks at the first line to look at what program it should call to run it. /bin/sh says to use the default shell, /bin/bash says to use the Bourne Again SHell.*

III What else do I need other than a shebang? *You should also use chmod +x to make the file executable.*

IV I want my script to take inputs? *Yes eventually! 1, 2, 3... are declared as variables when the script is run, they contain the first, second, third... arguments passed to the script.*

V What if I need all the arguments but I don't know how many there will be? *@ contains all the arguments as one long list*

VI This is a lot of work I'd rather just run everything interactively. *Making a script is like following a lab protocol, you can accomplish something without doing it, but it isn't very useful until you record what you actually have done.*

Time:

Video is 10 min, it really needs more examples to catch on though.

3.3 Bioinformatics in bash: putting it all to use

3.3.1 Trimmomatic

Trimming with trimmomatic is the first step in our Bioinformatics pipeline.

Learning Goals

I What does trimmomatic do? *It trims your raw reads for quality, hopefully improving your final assembly.*

II How much data are you giving to spades? *Look at the number of reads, each read is 250 bp long,* $\frac{250*reads}{genomesize}$ *will give you average coverage*

Time:

Video 13 min runtime: very quick

3.3.2 Spades Assembly

The assembly will provide the data for quast and prokka.

Learning Goals

I What does spades do? *It assembles your reads into contigs.*

Time:

Video is 9 min runtime: long, expect most of the runtime of the pipeline to be assembly

3.3.3 Quast

Quast provides metrics for the assembly done by spades

Learning Goals

I What are some easy things too look at in quast? *Look at the distribution of your contigs, are there some that are a significant portion of a bacterial genome long? Are the biggest ones only a few thousand bases long? These can tell you something about your assembly.*

Time:

Video is 5 min runtime: quick

3.3.4 Prokka

Prokka annotates your assembly.

Learning Goals

I I ran prokka, what now? *Look at the results! This is where the biology lives.*

Time:

Video is 8 min runtime: medium

Chapter 4

Debriefing

Congratulations! You did it (or you are reading ahead which is good too!) This chapter will talk about what you need to do, and what we would really appreciate you doing for us.

4.1 Close up loose ends on ron

We would like to be as proactive as possible with clearing out unused accounts on ron. Please contact us with a rough idea of the following:

1. Is there data or results you or your students would like from ron?
2. When are your students done with the course, and when can we remove their accounts? (this does not need to include deletion of their files, just removal of their login)
3. When are you done with your account? We can keep it for your next class, or you can start with a fresh one next time.

For the security of ron, to lower the number of active accounts that could potentially be compromised we will be quite proactive about deactivating accounts after they are no longer being used. If you do not contact us about this post class do not assume that the accounts or data will be there. We will however try our best to accommodate any requests you or your students might have.

4.2 Give us feedback

Learning your experience, and your students experiences with these modules is extremely helpful for us. Please let us know any feedback you have from grammar errors in a worksheet to broad changes you would make.

Chapter 5

Troubleshooting

5.1 Assembling

5.1.1 Spades stopped working after working fine for a while. Crashing with some weird scary looking stack dump.

Check your reads, calculate the expected coverage, if you have only got a tiny number of reads then Spades may just crash partway through trying to assemble. If this is the case, try using the untrimmed reads instead.

5.2 Unable to login to ron

5.2.1 permission denied when entering my login information

In rough order of likely hood these are the possible causes:

1. You have entered your username or password wrong
2. You have entered your password wrong to many times in the past 15 min, and have been locked off of ron. Contact us or wait 15 min.

5.2.2 timed out after being successfully connected "session has timed out" or no response to input and no updating output

This issue comes from your computer's internet connection going offline, simply close the putty/terminal window and open a new instance. You can avoid this by not letting your computer fall asleep, and improving the WiFi infrastructure where you are located.

5.2.3 timeout error or ssh doesn't return anything and just sits there doing nothing WHEN CONNECTING

This is usually caused by some sort of firewall blocking the connection somewhere between you and ron.
Rerun ssh, but with the -vvv option in addition to all the other options you normally use. This will turn on very very verbose reporting and can help us diagnose the issue. Once you have done this please contact us ASAP so we can begin working on the issue.

5.3 I can't assemble because my WiFi drops too often!

remember to use nohup when appropriate!

5.4 I submit a command and it doesn't do anything

No news is good news! usually this just means your program ran exactly as you would expect.

Chapter 6

FAQ

6.1 PuTTY: the website has a bunch of links, which do I download?

The download page for PuTTY has a large number of options to download.
PuTTY is only necessary for windows computers, if you have a mac or Linux system you do not use putty, you use ssh to connect to ron.
To install PuTTY you have two options. The preferable option is to use the download in the "A Windows MSI installer package for everything except PuTTYtel".
This is an installer, running it will prompt you through installing PuTTY, once you have completed the installation you can run the PuTTY program that was installed.
If for any reason this does not work you can try the second method.
Download the putty.exe download in "For Windows on Intel x86", this is just the putty executable, there is no installation, to run PuTTY simply run the file you download.

6.1.1 Help us help you

Be specific when describing the issue you have encountered. "We can't connect to ron" could be a huge number of different things, whenever possible send us the commands and output that begot the error.

6.2 Why does my terminal look different than Jordan's in the videos?

Jordan has spent untold hours fiddling with his settings to get his terminal to look just the way he likes it. It is purely cosmetic. Feel free to explore the settings for the terminal program or PuTTY you are using on your computer. Some settings in the .bashrc, especially the PS1 can be changed on your ron account, feel free to do so!