Microbiome

Microbiome Analysis
Metagenomics
Metatranscriptomics

Bioinformatics

Genome Assembly & Annotation
SNP & Variant Detection
Microbiome & Metagenomic analysis

Genomics

Genome Sequencing
Next Gen Genetic Mapping
RNA-Seq
Bisulfite Sequencing

CAGEF

University of Toronto
Centre for the Analysis of
Genome Evolution & Function

Proteomics

Mass Determination
Protein Identification
Post-translational modification

cagef.utoronto.ca
cagef@utoronto.ca

CSB1021HF LEC0131

FUNDAMENTALS OF GENOMIC DATA SCIENCE

0.0.0 Module 1: Exploring Genomic File Formats

0.1.0 About Fundamentals of Genomic Data Science

Fundamentals of Genomic Data Science is brought to you by the **Centre for the Analysis of Genome Evolution & Function (CAGEF)** bioinformatics training initiative. This course was developed based on feedback on the needs and interests of the Department of Cell & Systems Biology and the Department of Ecology and Evolutionary Biology.

The structure of this course is a "code-along", hands-on style! A few hours prior to each lecture, materials will be made available for download at QUERCUS (https://q.utoronto.ca/). The teaching materials will consist of a weekly PDF that you can use to follow along with the instructor along with any datasets that you'll need to complete the module. This learning approach will allow you to spend the time coding and not taking notes!

As we go along, there will be some in-class challenge questions for you to solve. Post lecture assessments will also be available for each module, building upon the concepts learned in class (see syllabus for grading scheme and percentages of the final mark).

0.1.1   Where is this course going?

We'll take a blank slate approach here to learning genomic data science and assume you know nothing about programming or working directly with next generation sequencing data. From the beginning of this course to the end we want to guide you from potential scenarios like:

- You don't know what to do with a set of raw sequencing files fresh from a facility like CAGEF.

- You've been handed a legacy pipeline to analyse your data or maintain for the lab, but you don't know what it runs or how.
- You plan on generating high-throughput data but there are no bioinformaticians around to help you out.

and get you to the point where you can:

- Recognize the basic tools in sequence analysis.
- Plan and write your own data analysis pipelines.
- Explain your data analysis methods to labmates, supervisors, and other colleagues.

0.1.2   How do we get there?

In the first half of this course, we'll focus on how to generate analysis pipelines using the Galaxy platform – a user-friendly graphical interface that provides access to common sequence analysis tools. After we are comfortable with these tools, we'll look at life through the lens of a command-line interface. It is here that we will learn the basics of file manipulation and how to program scripts that can carry out multiple tasks for us. From there we'll revisit tools from the first half and learn skills to make your data analysis journey easier.

0.2.0 Goals of the module

1. Set-up your computer for this course.
2. Learn how to download genomic data.
3. Explore the formats of common genomic data.
4. Learn how to manipulate data files in a text editor.
5. Learn how to use the Broad Institute's Integrated Genomics Viewer (IGV).

0.3.0 Pre-class readings or Coursera modules

Each week we strongly encourage you to complete the assigned Coursera modules and/or readings **before** class. These are meant to provide you with sufficient background material on each week's module so that we can focus on the act of "doing" something with that data rather than spend a lot of time on the origins of it. You'll find a section outlining the next set of Coursera modules and readings **at the end** of each module.

0.3.1   Go to www.coursera.org and sign up for an account with your e-mail.

0.3.2   Search the following course and enroll to audit:

- Command Line Tools for Genomic Data Science, Johns Hopkins University.

0.4.0 Setting up your working directory

We suggest that you create a new directory (folder) for this course directly off your root directory called "**FGDS**". Working from your root directory is not necessary, but it will make some of the aspects of the course a little easier to manage. For MacOS users, we suggest you create this as a subfolder in your <u>**user**</u> directory.

0.4.1   Within this directory, create another directory called "**Module1**". This is where we will store the data used in this week's module.

0.4.2   Create a subdirectory called "**downloads**" to store the initial files as we download them before decompressing and working with them in later steps.

1.0.0 Obtaining Genomic Data from NCBI

1.1.0 Locating genomic data

Navigating NCBI can, at times be a daunting experience. Today we'll start with locating genomic data using the *Haemophilus influenzae* str. KW20 genome as an example.

> 1.1.1   Go to *https://www.ncbi.nlm.nih.gov/*. Note that most NCBI pages have a *search bar (All Databases)* near the top of the page. This box can be used to search a wide range of different data.

> 1.1.2   Search "*Haemophilus influenzae*" in the **Genome** database through the *search bar* (omit the quotation marks". This can be accessed by pulling down the **All Databases** menu.
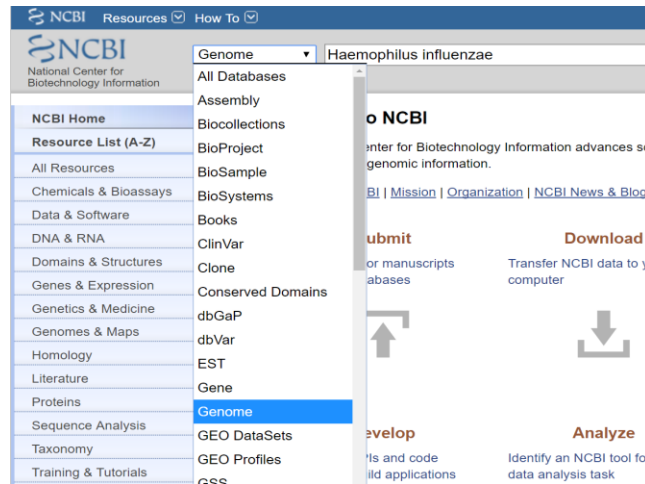
> 1.1.3   Find the **Organism Overview** section and select the **Genome Assembly and Annotation report** hyperlink, which will take you to a page with all the sequenced strains of *H. influenzae* that are available on NCBI.

> 1.1.4   In the search box, search for "Haemophilus influenzae Rd KW20" or "Rd KW20".

Now we've located our genome of interest. From here there are several pieces of information we can use to learn about our genome of interest such as the origins of the biological sample, size of the sequenced genome, and a list of coding sequence locations.

1.2.0 Downloading genome assembly files

What we are ultimately interested in from this report is the genome assembly itself which will hold files that include annotations of features as well as DNA, RNA and protein sequences if applicable.

> 1.2.1   From the genome assembly report, under the *Assembly* column, click on the accession **GCA_000027305.1** hyperlink which will take us to the assembly report.

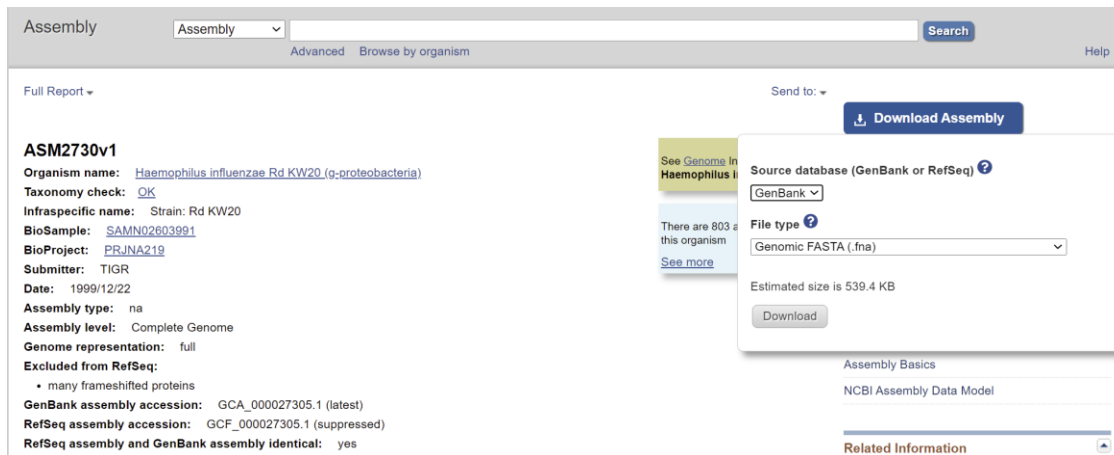> 1.2.2   Select **Download Assembly** (blue button on the top right corner).

1.2.3 Select **GenBank** from the *Source database* dropdown menu.

1.2.4 Select **All file types** from the *File type* dropdown menu and hit the **Download** button.

This step will download a zip file to the directory of your choice. Open the archived file using 7-Zip (https://www.7-zip.org/) or any other available program. Alternatively, you can download the files below separately from the **Annotations, Features and products** subheader under *File type*.

1.2.5 Once the archive is opened, you will see many files with the prefix **GCA_000027305.1_ASM2730v1_**. You can transfer the following files to your **Module1/downloads** working directory:

File suffix	Description
cds_from_genomic.fna.gz	All annotated coding sequences in nucleotide format
feature_table.txt.gz	Tab delimited file of locations and attributes for annotated features.
genomic.fna.gz	Genome assembly in fasta format.
genomic.gbff.gz	GenBank flat file format with all genomic sequences in the assembly.
genomic.gff.gz	Genome annotation in gff3 format.
protein.faa.gz	All annotated protein products in amino acid format.
rna_from_genomic.fna.gz	All annotated RNA features in nucleotide format.

4

1.3.0 Download sequencing files from the NCBI Sequence Read Archive (SRA)

Now that we've downloaded some annotations from NCBI, we are also interested in collecting the original sequencing data used to assemble the genome. We'll use this information in later sessions to dig deeper into the influenza genome. In this case we'll be downloading files in the **fastq** format.

    1.3.1   From the _NCBI search bar_, select **SRA** from the dropdown menu and search for "_Haemophilus influenzae_ Rd KW20".

    1.3.2   Locate and select the entry for **KW20 whole genome sequencing** (SRX026485).

    1.3.3   Under the _Runs_ section select the **SRR065202** hyperlink to go to the SRA information page for our genome. More information about SRA accessions can be found in **Appendix 1**.

This will take us to the SRA home page for our chosen experiment. It will look slightly different from the NCBI homepage. From this archive we will download the **fastq** files from the original sequencing run.

    1.3.4   Locate the _FASTA/FASTQ download_ tab for this dataset and then click on the **FASTQ** icon in the **Download** section near the bottom. Note that this will begin downloading a ~600 Mb data file.

Note that on the _Reads_ and _FASTA/FASTQ_ tab you can perform some fun filtering based on sequences. For instance, if you are only interested in a subset of data based on specific sequences (like a motif or barcode) you can try to filter exclusively for those reads. The same kind of filtering can be applied at the time of downloading your sequences.

1.4.0 Extract your downloaded files

All the files we've been downloading should now be saved into a directory such as **~/Module1/downloads**. These will need to be moved to the **Module1** analysis directory and unpacked. Do this using Finder if you're on a Mac and File Explorer if you're on Windows.

IMPORTANT: Leave the **fastq** file for now because it will take some time to download.

1.4.1 Move files from your **downloads** directory to your FGDS directory by dragging and dropping.

1.4.2 Unzip all of the files in this directory.

1.4.2.1 Mac
- Select all files and double click to extract.
- Select all gz files and move them to your trash.

1.4.2.2 Windows 10
- Select all files > Compressed Folder Tools > Extract
- Select all gz files and move them to your trash.
- If this doesn't work on your system, you will need to download 7zip: https://www.wikihow.com/Extract-a-Gz-File

1.4.3 Change all of your filenames to something more intuitive in your Finder/File Explorer window. Note that there are several common extensions for genomic files. A complete list is provided in **Appendix 2**.

Download name	New name	Description/format
GCA_000027305.1_ASM2730v1_cds_from_genomic.fna		gene file, fasta
GCA_000027305.1_ASM2730v1_feature_table.txt		feature file, text
GCA_000027305.1_ASM2730v1_genomic.fna		genome file, fasta
GCA_000027305.1_ASM2730v1_genomic.gbff		genbank file, text
GCA_000027305.1_ASM2730v1_genomic.gff		general feature format file, text
GCA_000027305.1_ASM2730v1_protein.faa		protein file, fasta
GCA_000027305.1_ASM2730v1_rna_from_genomic.fna		RNA file, fasta

2.0.0 Exploring and manipulating genomic data using a text editor

As we discussed at the beginning of today's class, most genomic data is going to be stored in flat text files. These are easy to access from the command line and you will learn some of the simple tools for accessing and manipulating genomic data at the command line later in this course. For now, we are going to work with text editors to access and manipulate our genomic data. For Mac, one of the best and most broadly used text editors is **BBEdit**. For Windows, **Notepad++** is often the text editor of choice.

> 2.0.1 Download and install the appropriate text editor for your system from one of the following links.
> - Mac: https://www.barebones.com/products/textwrangler/download.html
> - Windows: https://notepad-plus-plus.org/

2.1.0 Open and explore assembly files

Open and explore all the assembly and annotation files from your **Module1** directory. This is most easily accomplished by dragging and dropping into your editing application or directly from the editing program.

> 2.1.1 File > Open
>
> 2.1.2 Select "Show Hidden Items".
>
> 2.1.3 Find the directory where you saved you *H. influenza* files and open them.

We will now use some very simple Regular Expressions (RegEx) to search and edit some of these genomic data files. RegEx is an absurdly powerful way to perform text pattern searching and is an essential aspect of bioinformatics and nearly all aspects of computer science. We don't have time to explore RegEx in depth in this course, but we will use a few simple examples to illustrate its power. We provide more information on RegEx in **Appendix 3**.

2.2.0 Search for the beginning of new entries in a fasta file

We can count the number of contigs present in the genome assembly (**HinfKW20_genomic.fna**) and the number of genes in the gene file (**HinfKW20_cds.fna**) using a simple regular expression search.

> 2.2.1 Open up the Find/Search box (See below: Left - BBEdit; Right – Notepad++).
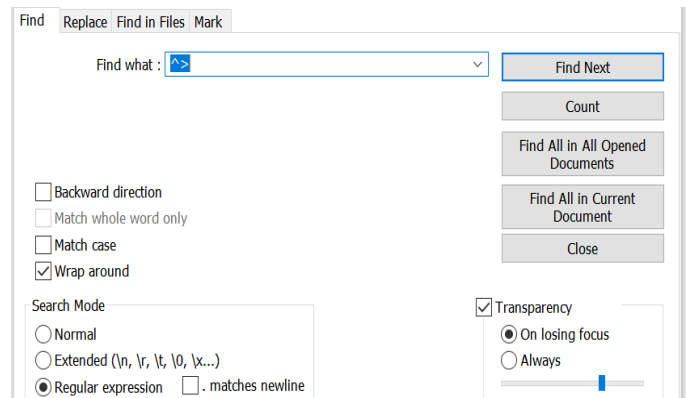>
> - Press Cmd + F (Mac)
> - Ctrl + F (Notepad++)
>
> *Make sure you have selected Grep in BBEdit and Regular Expression in Notepad++.*
>
> 2.2.2 Find the start of a line with the ^>
>
> - The ^ specifies the beginning of the line, so you are searching for all lines that start with >.
>
> 2.2.3 Select **Find All** (BBEdit) or **Count** (Notepad ++). How many coding sequences are there in each file?

2.3.0 Count the number of genes in a gene file that use an ATG start codon

Now that we know how many entries are in our **HinfKW20_cds.fna** file we can search and see how many of those begin with an ATG start codon.

> 2.3.1   Find all entries with this expression: `^>.+\nATG`

- `.` is a regular expression that means anything.
- `+` is a quantifier that means one or more times.
- `\r\n` and `\n` are end-of-line (EOL) characters in Notepad++ and BBEdit, respectively. Be aware that different programs may use different end-of-line characters, and if you don't use the right one, the operation will not work. **Both MacOS X and Unix-based files use "\n" rather than "\r"**. If you're working on a file that was created in Windows, we strongly recommend you to change the EOL character to the Unix format. You can do this in Notepad++ by: Edit > EOL Conversion > Unix

2.4.0 Replace all start codons in a gene file with **ATG**

Suppose we wanted to replace the first codon in all of our sequences with an ATG start codon sequence. We can also accomplish that with a find/replace command.

> 2.4.1   Use the Find window's *Replace* tab

> 2.4.2   *Find what*: `^(>.+\n)\w{3}`

- Again, use "\n" instead of "\r" for Linux/Windows.
- The parentheses around `>.+` means capture this information and hold it in a temporary variable. The first set of parentheses will be held in variable `\1`, the second set (if present) would be held in variable `\2`, and so on.
- `\w` matches any word character, including digits and underscores.
- `{3}` specifies the number of word characters (in this case) to match. `\w{3}` could also have been written as `\w\w\w`.

> 2.4.3   *Replace with*: `\1ATG`

- `\1` is the temporary holding variable. This will copy everything captured in the parentheses of the find statement. In this case that's the *header* of our fasta entry that we are putting back as part of the replacement statement.

- ATG tells the replace statement to write "ATG". Since you did not capture the first three characters (ie \w{3}) in the find statement, this means that the replace will overwrite those characters with ATG.

2.4.4   Select the **Replace All** button.

2.5.0 Convert your CDS file into tab-delimited format

2.5.1   *Find what*: (^>.+$); *Replace with*: \1\t

- ^ specifies the beginning of the line while $ specifies the end of the line.
- "\t" is a wildcard for the tab character.

2.5.2   Select the **Replace All** button.

2.5.3   *Find what*: \n([^>]); *Replace with*: \1

- [^>] means any character that is not > (not in the square bracket). Note that the ^ has a different meaning when used in the square bracket.
- This line finds every occurrence of the return character followed by anything that isn't the ">"
- Remember to replace the "\r" with "\n" when working in Linux.
- Now you replace what is found above with nothing, so all lines that aren't fasta headers (starting with the ">") are concatenated together on one line.

2.6.0 Additional RegEx resources

These are just a couple of the manipulations that can be done with your Text Editors. Regular expressions are an incredibly useful tool for genomic data science, so try to familiarize yourself with the usage of some basic expressions in the coming weeks. Here are some resources:

- http://www.rexegg.com/regex-quickstart.html
- https://regexr.com/

3.0.0 Exploring genomic data with the Integrated Genome Viewer (IGV)

By now your fastq file should have finished downloading. Let's move that to the **Module1** directory and rename it. We won't need to complete this step until next week when we'll actually use this data so just be sure to complete this before next class. The unzipped file will actually take quite a bit of space: 4.6 Gb uncompressed!

3.0.1   Move the file by dragging it from your **Downloads** directory to your **Module1** directory.

3.0.2   Unzip the file

5.0.2.1 Mac
- Select all files and double click to extract.
- Select all gz files and move them to your trash.

5.0.2.2 Windows 10
- Select all files > Compressed Folder Tools > Extract
- Select all gz files and move them to your trash.
- Use 7zip if necessary.

3.0.3   Rename the file **sra_data.fastq.gz**

Download name	New name	Description/format
sra_data.fastq.gz	HinfKW20_data.fastq	gene file, fasta

3.1.0 Download and install the Integrated Genome Viewer

3.1.1   You can download IGV from the Broad Institute at the following link:
https://software.broadinstitute.org/software/igv/download

Make sure to choose the correct version for your operating system.

Extensive information about the IGV can be found at
http://software.broadinstitute.org/software/igv/UserGuide.

3.1.2   **If IGV does not open properly**, you likely have an incompatible version of Java. Follow the instructions below to uninstall your current version of Java, then install Java 8 and redownload IGV.

3.1.2.1 Mac

- Uninstall Java (Finder > Documents > Search "JavaApp" > Move to Trash > Empty Trash).
- Install Java 8 (https://java.com/en/download > Free Java Download > Agree and Start Free Download > Follow Installation Instructions).
- Install IGV (https://software.broadinstitute.org/software/igv/download > IGV for Mac > Double Click on ZIP File).

3.1.2.2 Windows

- Uninstall Java (Control Panel > Uninstall or Change a Program > Select All Java Versions > Uninstall)
- Install Java 8 (https://java.com/en/download > Free Java Download > Agree and Start Free Download > Follow Installation Instructions).

- Install IGV (http://software.broadinstitute.org/software/igv/download_snapshot >IGV Java 8 for Windows > Run .bat file).

3.2.0 Use IGV to explore genome-sequencing data with alignment files

IGV allows you to import and explore genomic data from flat text files in a user-friendly format. Today we will be looking at a reference alignment file to learn some of the basic functionality of IGV. You will need to download two additional files from the Quercus website (Module1 > Data) for this section. These files were generated by aligning the reads from our fastq file to the *H. influenzae* KW20 reference genome.

- Sorted BAM alignment file – HinfKW20.sorted.bam
- Index for the sorted BAM alignment file – HinfKW20.sorted.bam.bai

3.3.0 Open IGV and generate a genome file from a reference genome

Now that we have the associated .bam and .bam.bai files for our genome, we can use IGV to actually look at the reference sequence and how the reads from our downloaded SRA fastq files align to the genome. We'll be using the H. influenzae KW20 reference data that we've been collecting throughout the module.

3.3.1 Open IGV.

3.3.2 Load a genome file using ***Genomes > Load Genome from File…***

3.3.3 From your data directory select ⬛⬛⬛⬛⬛⬛⬛⬛⬛⬛ and then select *Open*

This action will load the initial genome information from our data and we'll see the contig name for our genome (L42023.1)

3.3.4 Load the gene names and locations using ***File > Load from File…***

3.3.5 From the data directory select ⬛⬛⬛⬛⬛⬛⬛⬛⬛⬛

This will load a new track *onto* your loaded genome that will show the various open reading frames in the genome. Remember that a **.gff** file is a *general feature format* so we've basically annotated our genome with the gene locations!

3.4.0 Load and explore the reference alignment that you've downloaded in IGV

Next. we will use that **.bam** file we downloaded from Quercus to annotate the location of all the sequencing reads from that very large fastq we downloaded from the SRA. The **.bam** file was made using a program called **samtools**, where we generated an annotation of the fastq reads and by aligning them within our reference genome (aka **HinfKW20_genomic.fna**). We'll learn more about how to do this in later sessions! For now, we'll load the alignment information.

3.4.1 **File > Load From File…**

3.4.2 Select the sorted bam file (⬛⬛⬛⬛⬛⬛⬛⬛⬛⬛

3.5.0 Review some basic tools that you can use to explore the alignment in IGV

3.5.1 Identify the displayed tracks by looking at the information presented in the left pane.

3.5.2 Right click on a track to expand it.

3.5.3 Zoom in and out with the (+) and (-) zoom slider.
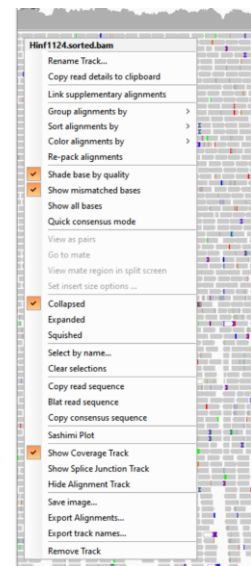
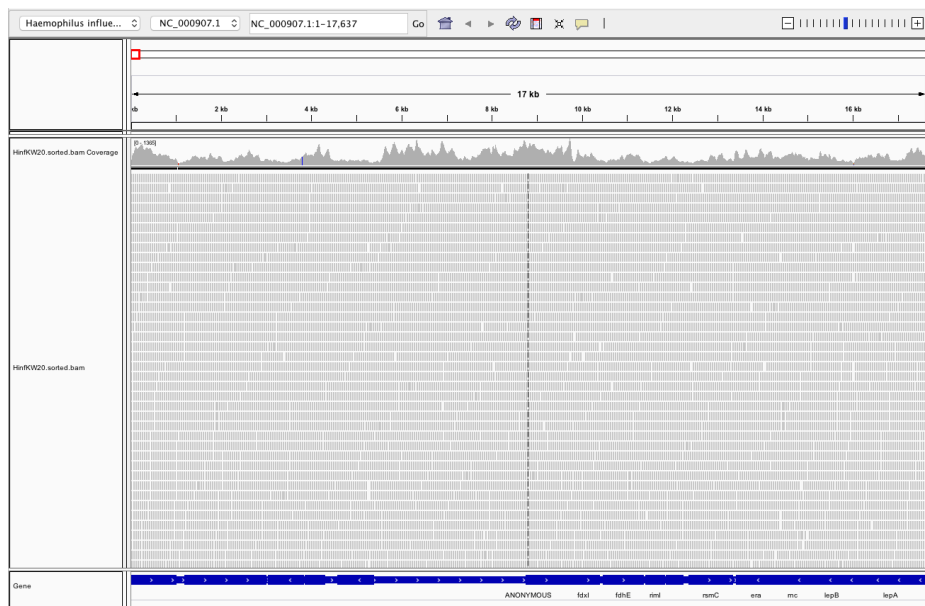3.5.4 Shift the window by clicking and dragging.

3.5.5   Scroll over elements to obtain more information about the region, read, or gene. You can modify this function with the comment symbol in the toolbar.

3.5.6   Right-click to bring up context-sensitive options for displaying or analyzing the data.

3.5.7   Search for specific regions in the search box (contig: baseStart-baseEnd).

3.5.8   Find examples of locations where reads identify different bases (coloured letters), where there are indels (solid line, I), and where reads can be mapped to multiple locations (white fill).

3.6.0 Find a particular motif in the reference genome

3.6.1   **Tools > Find Motif…**

3.6.2   Enter sequence and track names and select OK.

3.6.3   Remove the tracks by left clicking and selecting "Remove Tracks".

3.7.0 Saving your session in its current state

3.7.1   **File > Save Session…**

3.7.2   Name the session and click OK.

3.7.3   Reopen the session with **File > Open Session…**

University of Toronto Centre for the Analysis of Genome Evolution & Function (cagef.utoronto.ca) Licensed under CC BY-NC-SA

Human hg18	Genome drop-down box: Loads a genome. more...
chr1	Chromosome drop-down box: Zooms to a chromosome. more...
chr14:51,263,791–104,979,925 Go	Search box: Displays the chromosome location being shown. To scroll to a different location, enter the gene name, locus, or track name and click Go. more...
(home icon)	Whole genome view: Zooms to whole genome view. more...
◄ ►	Moves backward and forward through views of the genome like the back and forward buttons in a web browser.
(refresh icon)	Refreshes the display.
(region icon)	Defines a region of interest on the chromosome. more...
(fit icon)	Reduces the row height on all tracks to fit all data for the region in view into the window; will also expand tracks (to their maximum preferred size) to fill the view, if needed.
(popup icon)	Controls the popup information behavior. Options include displaying the information as the cursor hovers over an item, or when the item is clicked. The popup can also be disabled.
⊟ |||||||||||||||||||||| ⊞	Zoom Slider: Zooms in and out on a chromosome. more...

Quick visual reference of IGV menu-bar tools

4.0.0 Class summary

That concludes our first lecture and introduction to working with genomic data. Altogether we've explored the following:

- Common sequencing and data file types.
- Navigating the NCBI to download genome assembly data.
- Navigating the Sequence Read Archive to download raw sequencing data.
- Using regular expressions to search and replace data.
- Visualizing genomes with the Integrated Genome Viewer.

4.1.0 Post-lecture assessment (10% of final grade)

Soon after this lecture, a homework assignment will be made available on Quercus in the assignment section. It will build on the ideas and/or data generated within this lecture. Each homework assignment will be worth 10% of your final mark. If you have assignment-related questions, please try the following steps in the order presented:

- Check the internet for a solution – read forums and learn to navigate for answers.
- Generate a discussion on Quercus outlining what you've tried so far and see if other students can contribute to a solution.
- Contact course teaching assistants or the instructor.

4.2.0 Suggested class preparation for Module 2

Next week we will begin exploring the Galaxy interface and learning how to perform quality control and genome assembly on data from sources like the SRA. To prepare for this, we suggest the following Galaxy Tutorial:

- Galaxy 101 for everyone:
    - https://gallantries.github.io/video-library/videos/introduction/galaxy-intro-101-everyone/tutorial/
    - This is a ~1.25h video tutorial that you can watch/listen to ahead of time to familiarize yourself with the ideas of Galaxy.
    - We will go over much of this in class but the focus will be on navigating and using Galaxy.

CAGEF | **University of Toronto**
Centre for the Analysis of
Genome Evolution & Function

5.0.0 Appendix 1: SRA accessions

Accession Prefix	Accession Name	Definition	Example
SRA	SRA submission accession	The submission accession represents a virtual container that holds the objects represented by the other five accessions and is used to track the submission in the archive.	Since the SRA accession number is an artificial packaging construct, there is no example available since the SRA accession number has no specific response page
SRP	SRA study accession	A Study is an object that contains the project metadata describing a sequencing study or project. Imported from BioProject.	HTML
SRX	SRA experiment accession	An Experiment is an object that contains the metadata describing the library, platform selection, and processing parameters involved in a particular sequencing experiment.	HTML
SRR	SRA run accession	A Run is an object that contains actual sequencing data for a particular sequencing experiment. Experiments may contain many Runs depending on the number of sequencing instrument runs that were needed.	HTML
SRS	SRA sample accession	A Sample is an object that contains the metadata describing the physical sample upon which a sequencing experiment was performed. Imported from BioSample.	HTML
SRZ	SRA analysis accession	An analysis is an object that contains a sequence data analysis BAM file and the metadata describing the sequence analysis.	

The first letter of the accession prefix shows which INSDC archive the data originated from (S =NCBI-SRA, E = EMBL-SRA, D = DDBJ-SRA:

Data originating from EMBL-EBI:

Accession Prefix	Accession Name
ERA	ERA submission accession
ERP	ERA study accession
ERX	ERA experiment accession
ERR	ERA run accession
ERS	ERA sample accession
ERZ	ERA analysis accession

Data originating from DDBJ:

Accession Prefix	Definition
DRA	DRA submission accession
DRP	DRA study accession
DRX	DRA experiment accession
DRR	DRA run accession
DRS	DRA sample accession
DRZ	DRA analysis accession

More info at: https://www.ncbi.nlm.nih.gov/books/NBK56913/

- search for "What do the different SRA accessions represent?"

6.0.0 Appendix 2: Genomic file extensions

File Extension	File Type
.asn	genome record in asn.1 format
.faa	protein sequences in fasta format, text file
.ffn	protein coding portions of the genome segments
.fna	genome fasta sequence
.frn	rna coding portions of the genome segments
.gb	Genbank
.gbk	genome in Genbank file format
.gff	General file format used for storing annotations like genome features
.gpff	Genbank protein
.ptt	protein table
.rnt	rna table
.rpt	summary report
.val	binary file (genome project)
.tar	TAR archive, a common Linux archive file format.
.gz	gzip, a compressed format. Not the same as .zip
.tar.gz	gzipped tar, usually a tar file that was subsequently gzipped
.tgz	gzipped tar, usually gzipped by the tar application
.zip	Zip, a common Windows compression format

Additional sources:

https://bioinformatics.uconn.edu/resources-and-events/tutorials-2/file-formats-tutorial/#

7.0.0 Appendix 3: Regular expressions cheat sheet

regular expressions

Anchors

^	Start of line +
\A	Start of string +
$	End of line +
\Z	End of string +
\b	Word boundary +
\B	Not word boundary +
\<	Start of word
\>	End of word

Character Classes

\c	Control character
\s	White space
\S	Not white space
\d	Digit
\D	Not digit
\w	Word
\W	Not word
\xhh	Hexadecimal character hh
\Oxxx	Octal character xxx

POSIX Character Classes

[:upper:]	Upper case letters
[:lower:]	Lower case letters
[:alpha:]	All letters
[:alnum:]	Digits and letters
[:digit:]	Digits
[:xdigit:]	Hexadecimal digits
[:punct:]	Punctuation
[:blank:]	Space and tab
[:space:]	Blank characters
[:cntrl:]	Control characters
[:graph:]	Printed characters
[:print:]	Printed characters and spaces
[:word:]	Digits, letters and underscore

Assertions

?=	Lookahead assertion +
?!	Negative lookahead +
?<=	Lookbehind assertion +
?!= or ?<!	Negative lookbehind +
?>	Once-only Subexpression
?()	Condition [if then]
?()|	Condition [if then else]
?#	Comment

Note Items marked + should work in most regular expression implementations.

Sample Patterns

([A-Za-z0-9-]+)	Letters, numbers and hyphens
(\d{1,2}\/\d{1,2}\/\d{4})	Date (e.g. 21/3/2006)
([^\s]+(?=\.(jpg|gif|png))\.\2)	jpg, gif or png image
(^[1-9]{1}$\|^[1-4]{1}[0-9]{1}$|^50$)	Any number from 1 to 50 inclusive
(#?([A-Fa-f0-9]){3}(([A-Fa-f0-9]){3})?)	Valid hexadecimal colour code
((?=.*\d)(?=.*[a-z])(?=.*[A-Z]).{8,15})	8 to 15 character string with at least one upper case letter, one lower case letter, and one digit (useful for passwords).
(\w+@[a-zA-Z_]+?\.[a-zA-Z]{2,6})	Email addresses
(\<(/?[^\>]+)\>)	HTML Tags

Note These patterns are intended for reference purposes and have not been extensively tested. Please use with caution and test thoroughly before use.

Quantifiers

*	0 or more +
*?	0 or more, ungreedy +
+	1 or more +
+?	1 or more, ungreedy +
?	0 or 1 +
??	0 or 1, ungreedy +
{3}	Exactly 3 +
{3,}	3 or more +
{3,5}	3, 4 or 5 +
{3,5}?	3, 4 or 5, ungreedy +

Special Characters

\	Escape Character +
\n	New line +
\r	Carriage return +
\t	Tab +
\v	Vertical tab +
\f	Form feed +
\a	Alarm
[\b]	Backspace
\e	Escape
\N{name}	Named Character

String Replacement (Backreferences)

$n	nth non-passive group
$2	"xyz" in /^(abc(xyz))$/
$1	"xyz" in /^(?:abc)(xyz)$/
$`	Before matched string
$'	After matched string
$+	Last matched string
$&	Entire matched string
$_	Entire input string
$$	Literal "$"

Ranges

.	Any character except new line (\n) +
(a|b)	a or b +
(...)	Group +
(?:...)	Passive Group +
[abc]	Range (a or b or c) +
[^abc]	Not a or b or c +
[a-q]	Letter between a and q +
[A-Q]	Upper case letter between A and Q +
[0-7]	Digit between 0 and 7 +
\n	nth group/subpattern +

Note Ranges are inclusive.

Pattern Modifiers

g	Global match
i	Case-insensitive
m	Multiple lines
s	Treat string as single line
x	Allow comments and white space in pattern
e	Evaluate replacement
U	Ungreedy pattern

Metacharacters (must be escaped)

^	[	.
$	{	*
(	\	+
)	|	?
<	>	

Available free from AddedBytes.com