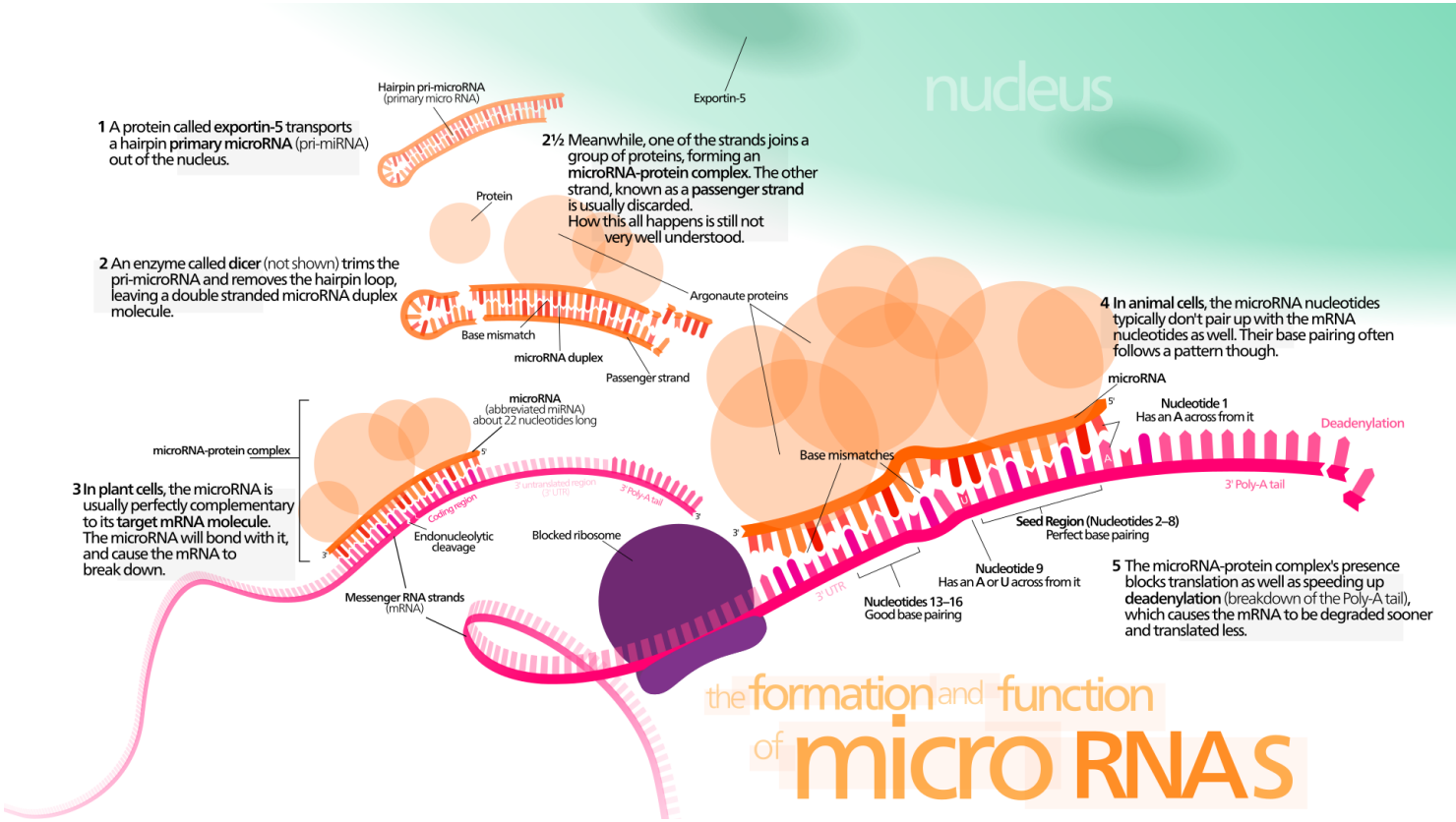
Next generation sequencing analysis of microRNAs

Jessy Slota
PhD Student
Department of Medical Microbiology and Infectious Diseases, University of Manitoba, Winnipeg, Canada
Prion Diseases Section, National Microbiology Laboratory, Public Health Agency of Canada
jessy.slota@phac-aspc.gc.ca; slotaj@myumanitoba.ca

Micro-RNAs (miRNAs)

- Small, non-coding RNAs
- ~22 nt in length
- Regulate gene expression through RISC
- Found in circulating fluids

Bartel, David P. *Cell* vol. 173,1 (2018): 20-51. doi:10.1016/j.cell.2018.03.006

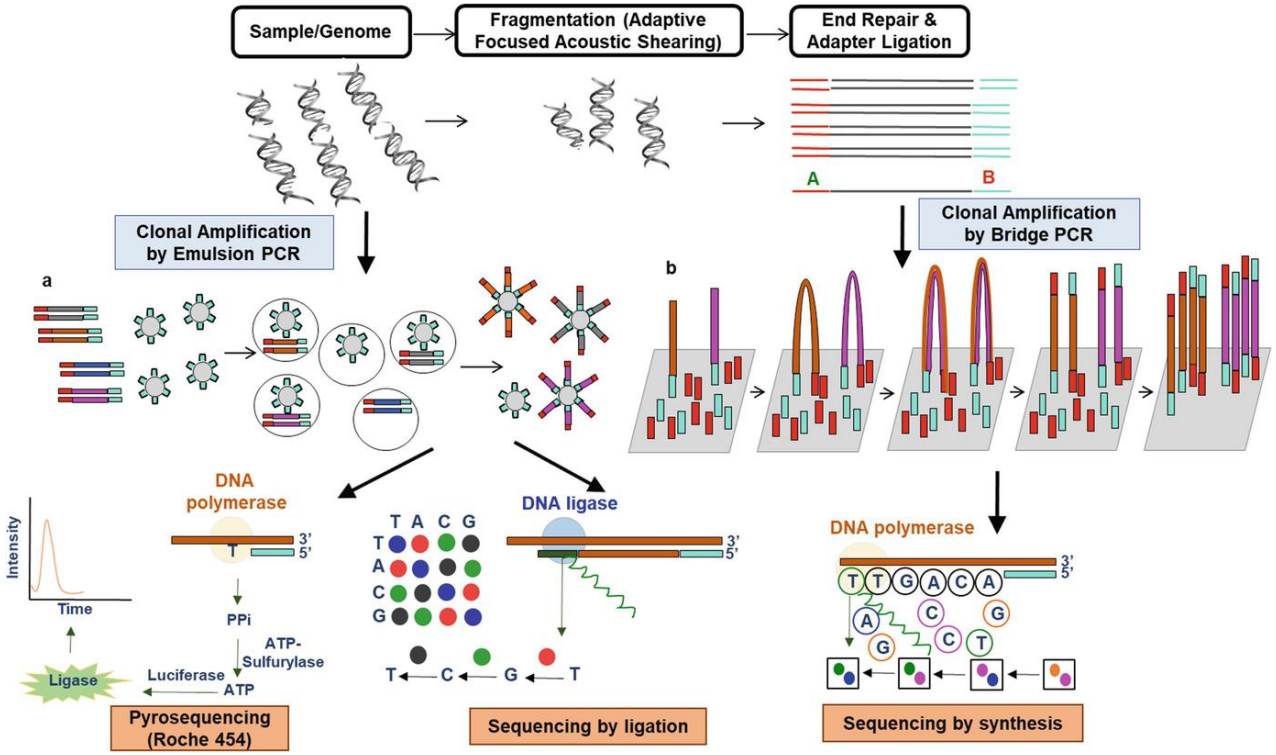


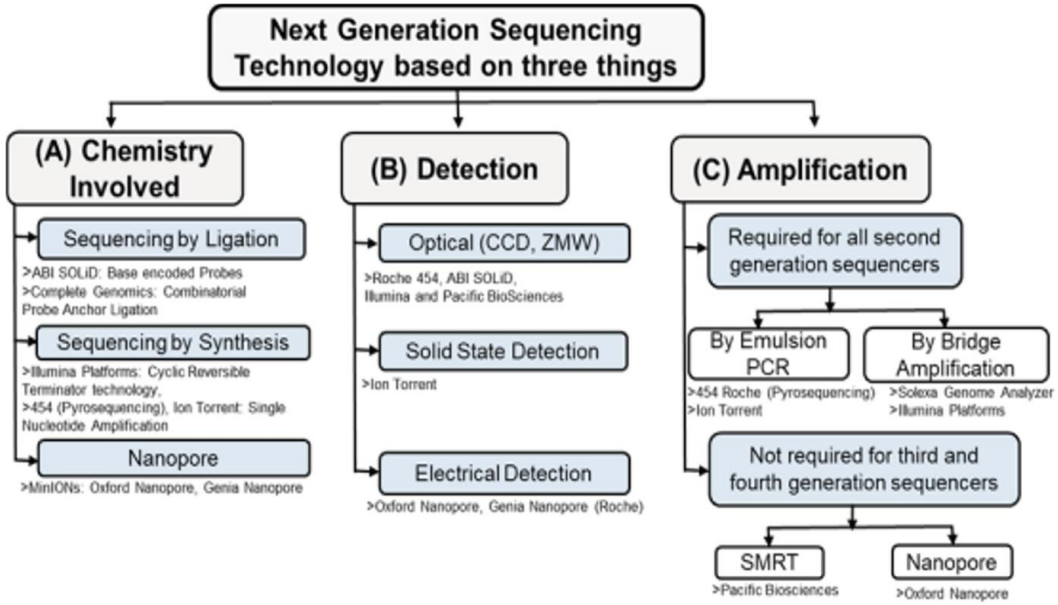
By Kelvinsong - Own work, CC BY 3.0, https://commons.wikimedia.org/w/index.php?curid=23311105

Next generation sequencing (NGS)

- Allows for high-throughput sequencing of DNA or RNA
- Sequencing of RNA can be used to measure gene expression
- Can also measure abundance of miRNAs through small-RNA sequencing
- **Why sequence miRNAs?**
  - Global identification of miRNAs with altered abundance
  - Biomarker discovery
  - Combine with mRNA seq for miRNA-target identification

Next Generation Sequencing (NGS)





Gupta N., Verma V.K. (2019) Microorganisms for Sustainability, vol 17. Springer, Singapore.
https://doi.org/10.1007/978-981-13-8844-6_15

NGS of miRNAs

1. RNA extraction… **depends on tissue** being used
   - E.g., TRIZol reagent, various kits for different types of tissues or biological fluids
   - Assess quality with Agilent Bioanalyzer
2. Library preparation
   - E.g., Illumina small RNA library preparation
   - Check quality of library on Agilent Bioanalyzer or TapeStation etc.
   - Library Fractionation (size selection **122–166 bp**) with Pippin HT (SageScience)
3. Sequencing
   - E.g., Illumina NextSeq 500/550
   - usually performed by sequencing core
4. Primary output is raw `fastq` files

Kolanowska M., Kubiak A., Jażdżewski K., Wójcicka A. (2018) In: Ørom U. (eds) miRNA Biogenesis. Methods in Molecular Biology, vol 1823. Springer, New York, NY. https://doi-org.uml.idm.oclc.org/10.1007/978-1-4939-8624-8_8

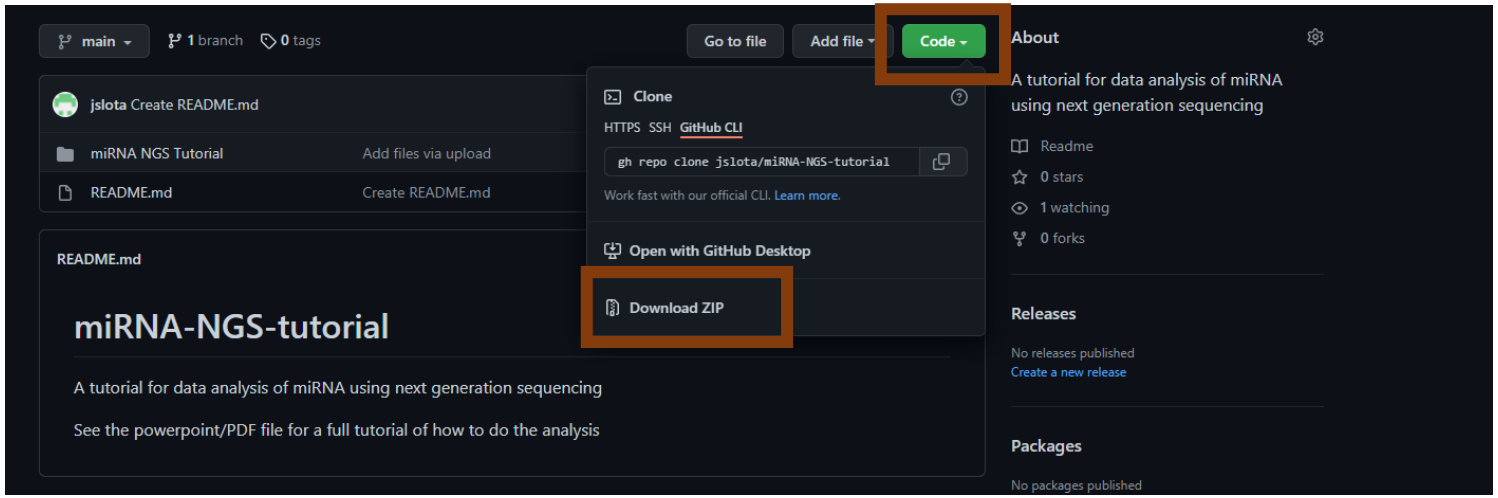
NGS data analysis

- “I have some data, **now what?**”
- Different tools available… commercially available vs publicly available
- The following tutorial will only use publicly available tools:
- Pre-processing with **Galaxy** – mapping reads to miRNAs
- Downstream analysis with **R** – statistics and data visualization (e.g., differential expression analysis)

Tutorial: basic NGS analysis of miRNAs

- **Part I:** Preprocessing in Galaxy
  - Assessing sequencing read quality with FastQC
  - Removing sequencing adapters with Cutadapt
  - Cleaning sequencing reads with Trimmomatic
  - Aligning to reference genome with Bowtie2
  - Mapping reads to miRNAs and counting with FeatureCounts
- **Part II:** Downstream Analysis with R
  - Raw reads count processing
  - Normalization and differential expression analysis with DESeq2
  - Examples of common data visualizations

Getting started: download tutorial data directory



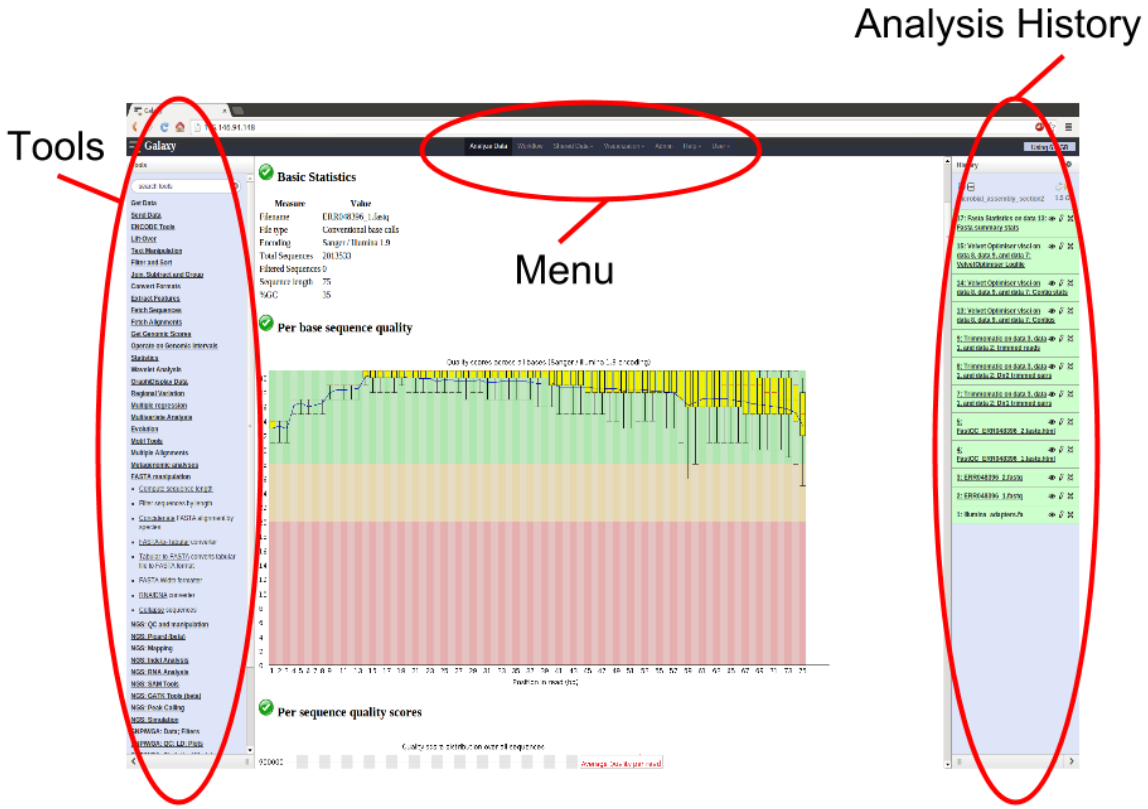
- Find the tutorial at: https://github.com/jslota/miRNA-NGS-tutorial
- Can manually download Zip file with all tutorial materials
- Unzip to your preferred location to get folder with all materials

Part I: Pre-processing sequencing reads in Galaxy

- **Part I:** Preprocessing in Galaxy
  1. Downloading data from NCBI SRA
  2. Assessing sequencing read quality with FastQC
  3. Removing sequencing adapters with Cutadapt
  4. Cleaning sequencing reads with Trimmomatic
  5. Aligning to reference genome with Bowtie2
  6. Mapping reads to miRNAs and counting with FeatureCounts
  7. Download read count files (and re-upload as individual files)
  8. Differential expression analysis with DESeq2 within Galaxy

Galaxy platform

- Publicly available
- Different versions… can try free version online
- Need to set up free account for this tutorial
- Access at: https://usegalaxy.org/



Step 1: Collect some data

We will use some of the data from this publication…

SCIENTIFIC REPORTS
natureresearch

OPEN

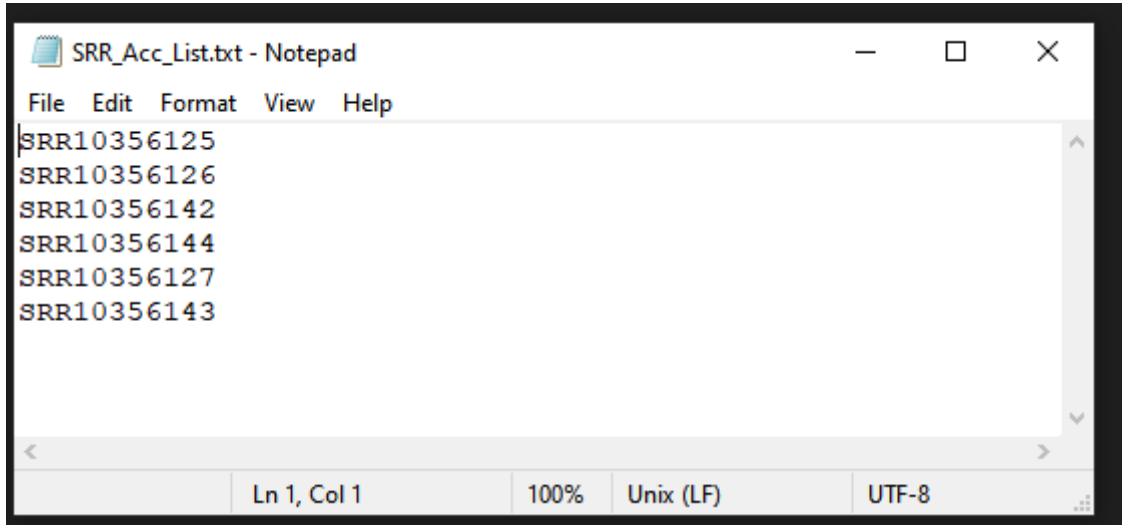
Identification of circulating microRNA signatures as potential biomarkers in the serum of elk infected with chronic wasting disease

Jessy A. Slota[1,2], Sarah J. Medina[1], Megan Klassen[1], Damian Gorski[4], Christine M. Mesa[1], Catherine Robertson[1], Gordon Mitchell[5], Michael B. Coulthart[3], Sandra Pritzkow[4], Claudio Soto[4] & Stephanie A. Booth[1,2*]

Chronic wasting disease (CWD) is an emerging infectious prion disorder that is spreading rapidly in wild populations of cervids in North America. The risk of zoonotic transmission of CWD is as yet unclear but a high priority must be to minimize further spread of the disease. No simple diagnostic tests are available to detect CWD quickly or in live animals; therefore, easily accessible biomarkers may be useful in identifying infected animals. MicroRNAs (miRNAs) are a class of small, non-coding RNA molecules that circulate in blood and are promising biomarkers for several infectious diseases. In this study we used next-generation sequencing to characterize the serum miRNA profiles of 35 naturally infected elk that tested positive for CWD in addition to 35 elk that tested negative for CWD. A total of 21 miRNAs that are highly conserved amongst mammals were altered in abundance in sera, irrespective of hemolysis in the samples. A number of these miRNAs have previously been associated with prion diseases. Receiver operating characteristic (ROC) curve analysis was performed to evaluate the discriminative potential of these miRNAs as biomarkers for the diagnosis of CWD. We also determined that a subgroup of 6 of these miRNAs were consistently altered in abundance in serum from hamsters experimentally infected with scrapie. This suggests that common miRNA candidate biomarkers could be selected for prion diseases in multiple species. Furthermore, Kyoto Encyclopedia of Genes and Genomes (KEGG) pathway analyses pointed to a strong correlation for 3 of these miRNAs, miR-148a-3p, miR-186-5p, miR-30e-3p, with prion disease.

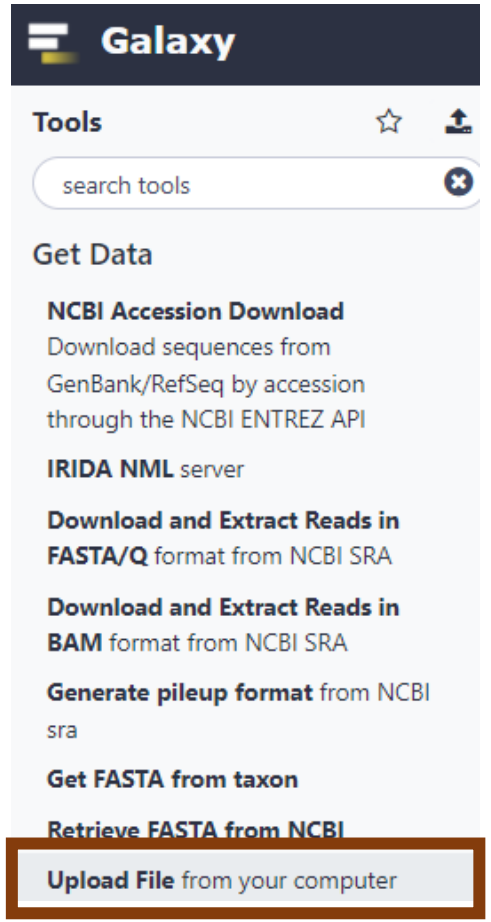
Slota, J.A., Medina, S.J., Klassen, M. *et al. Sci Rep* **9,** 19705 (2019).
https://doi.org/10.1038/s41598-019-56249-6

Step 1: Collect some data

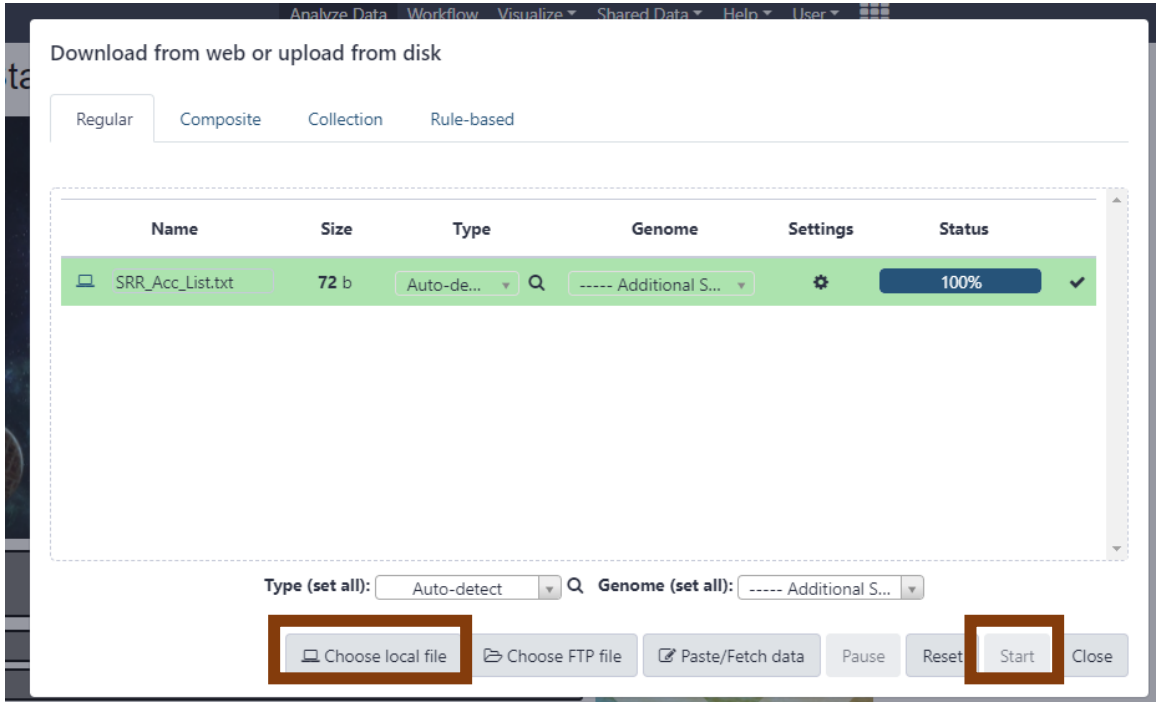


- Publicly available data from NCBI SRA (https://www.ncbi.nlm.nih.gov/sra)
- Data will be imported into Galaxy using this list of SRA accession numbers that can be found in the downloaded tutorial directory

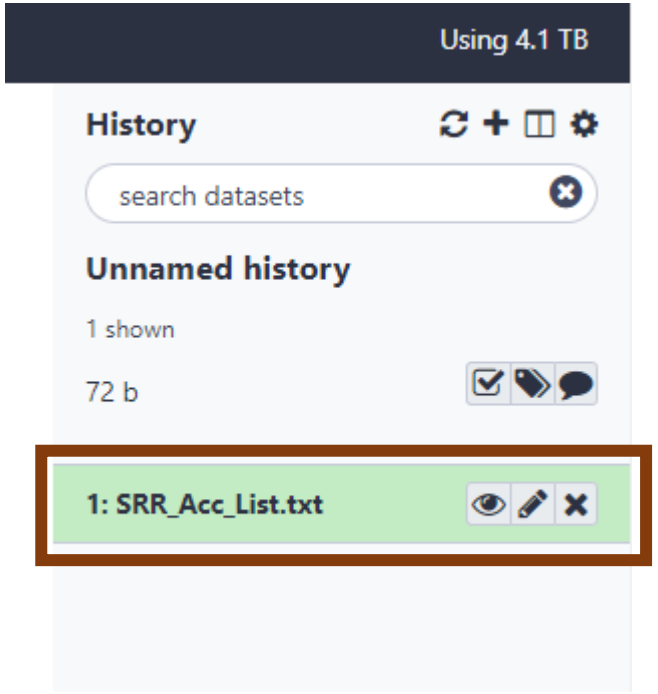
Step 1: Collect some data



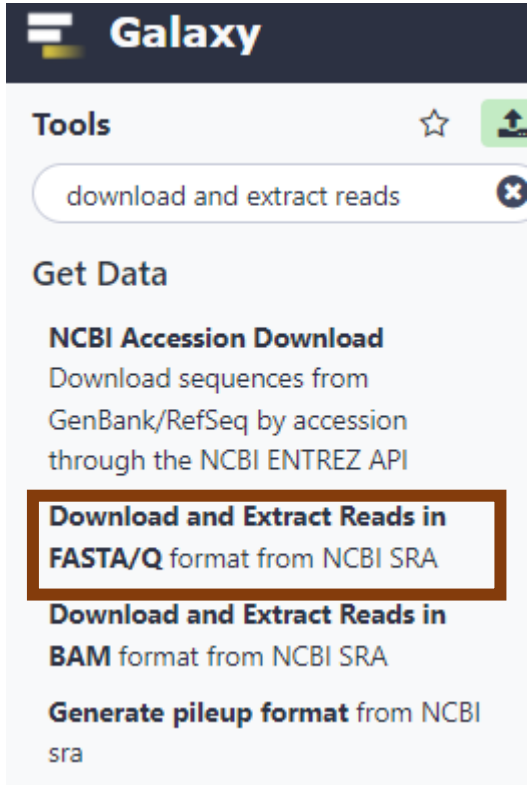
Type "**Upload file**" into Tools search bar



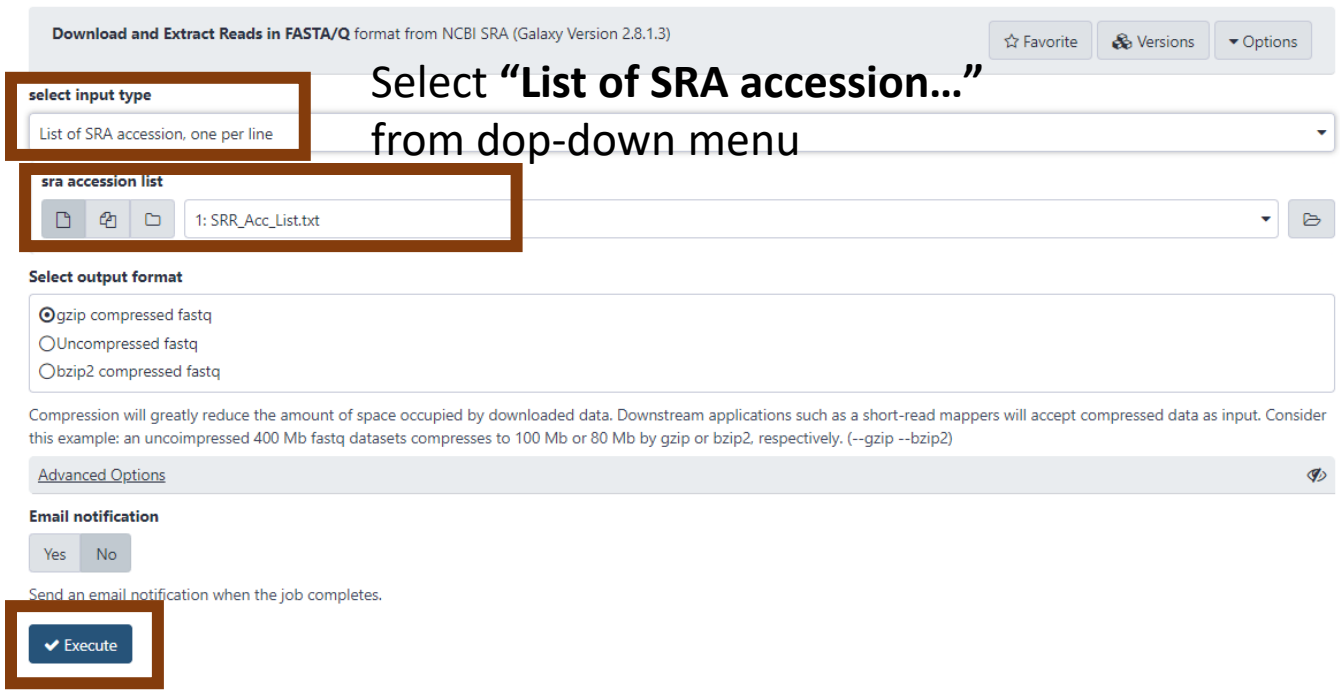
Browse and select "**SRR_Acc_List.txt**" from wherever it is saved



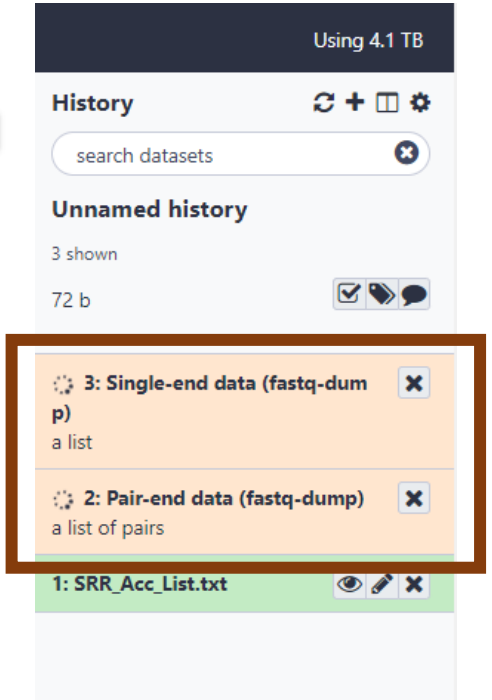
Step 1: Collect some data



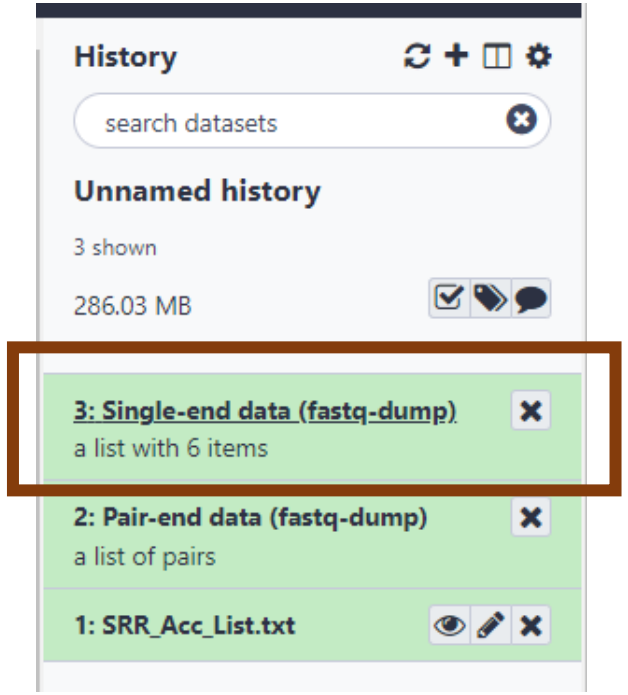
Type "**Download and Extract Reads in FASTA/Q**" into tools search bar



Select **"List of SRA accession…"** from dop-down menu

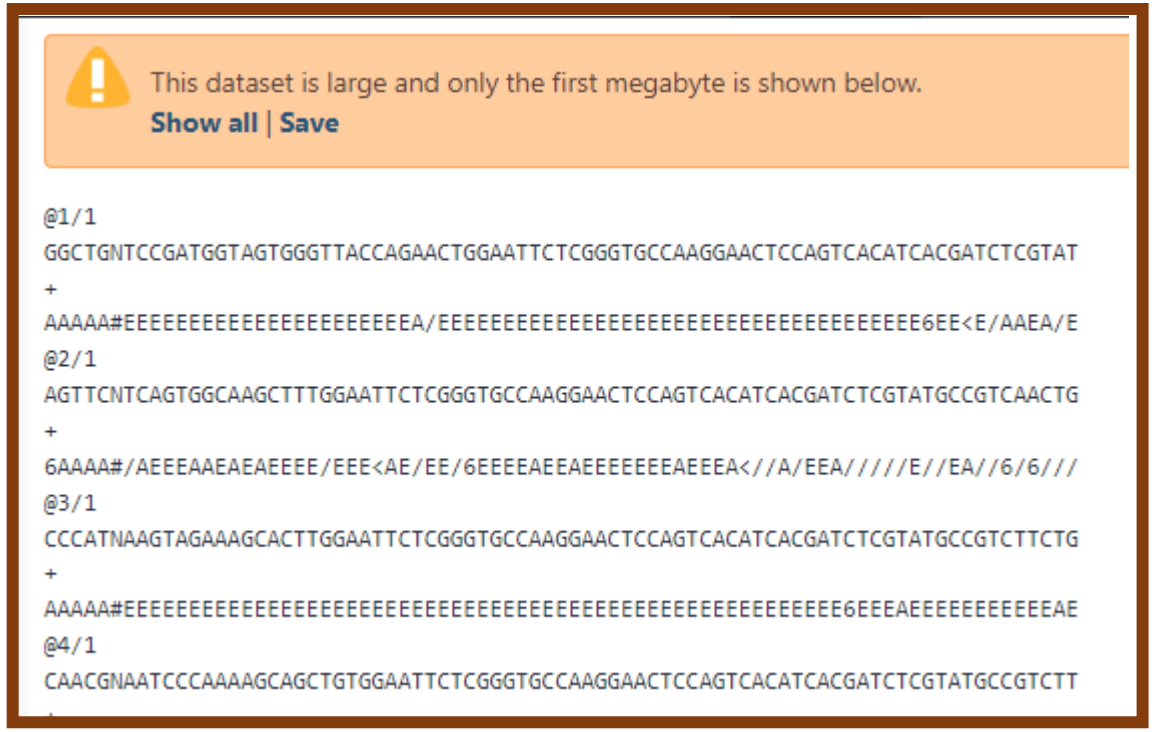


Step 1: Collect some data



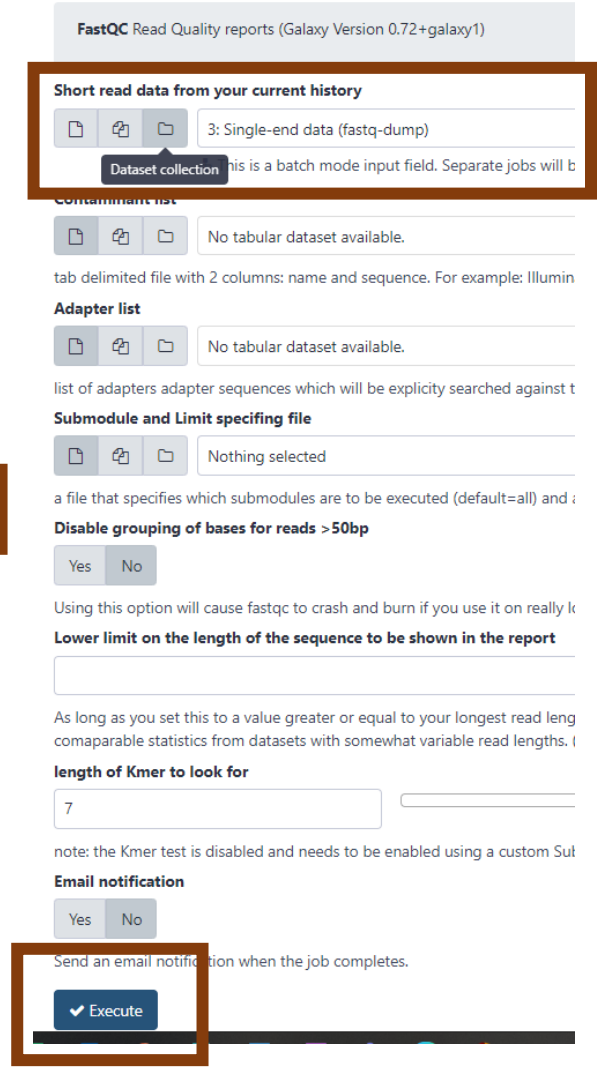
Data is in the format of a **collection**
Click on the collection to examine individual datasets

Click “View data”



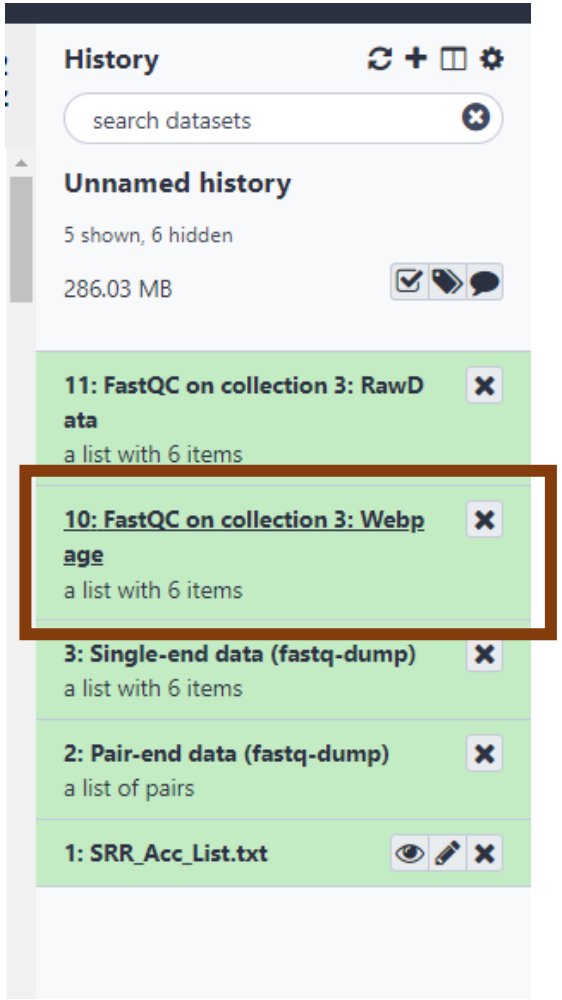
Data is in **Fastq** format

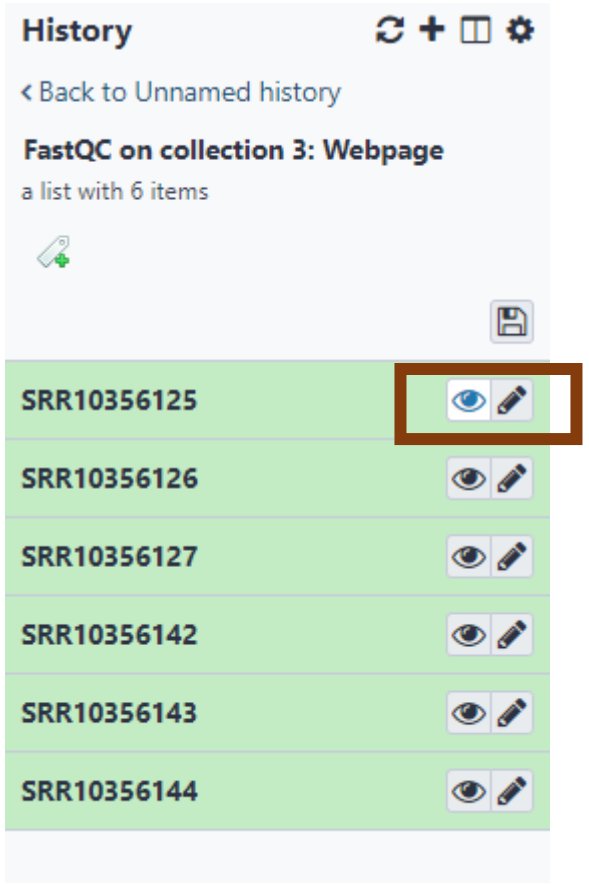
Step 2: Assess data quality with FastQC



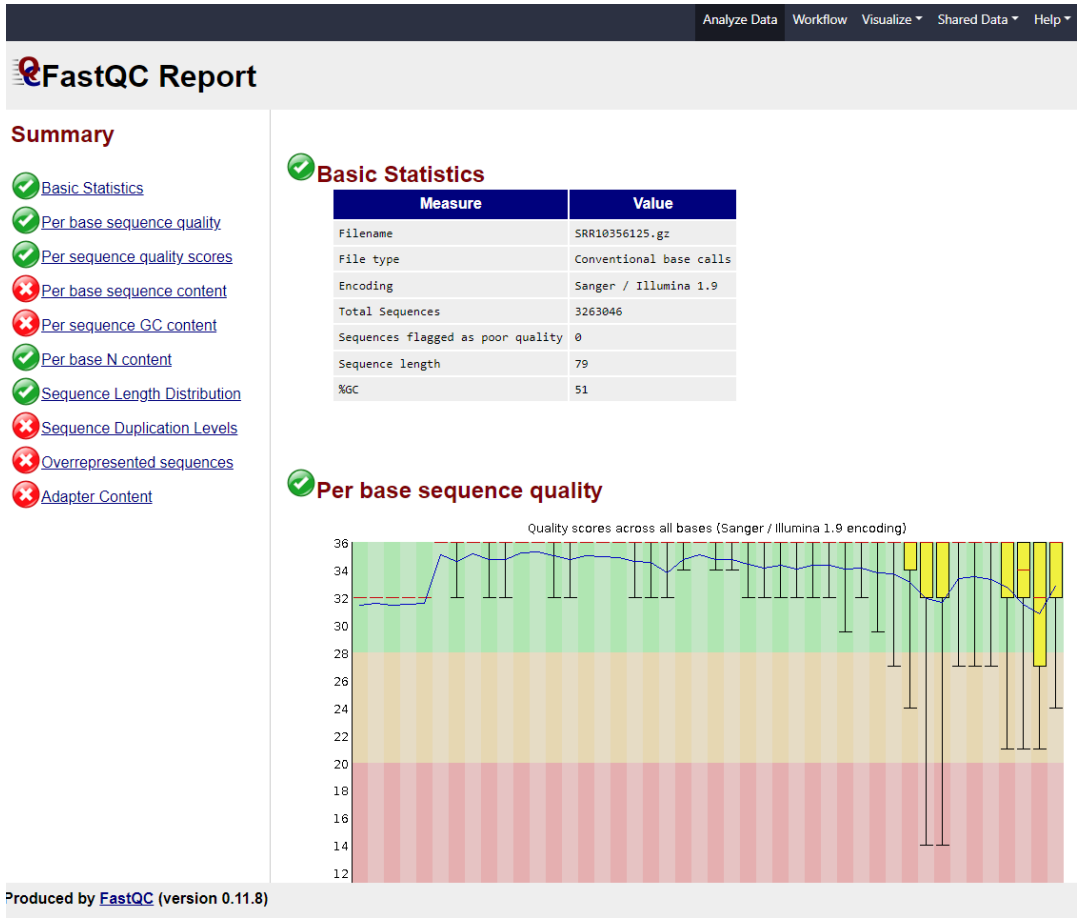
Select “**dataset collection**” Icon
Then select “**Single-end data**…” from drop-down

Step 2: Assess data quality



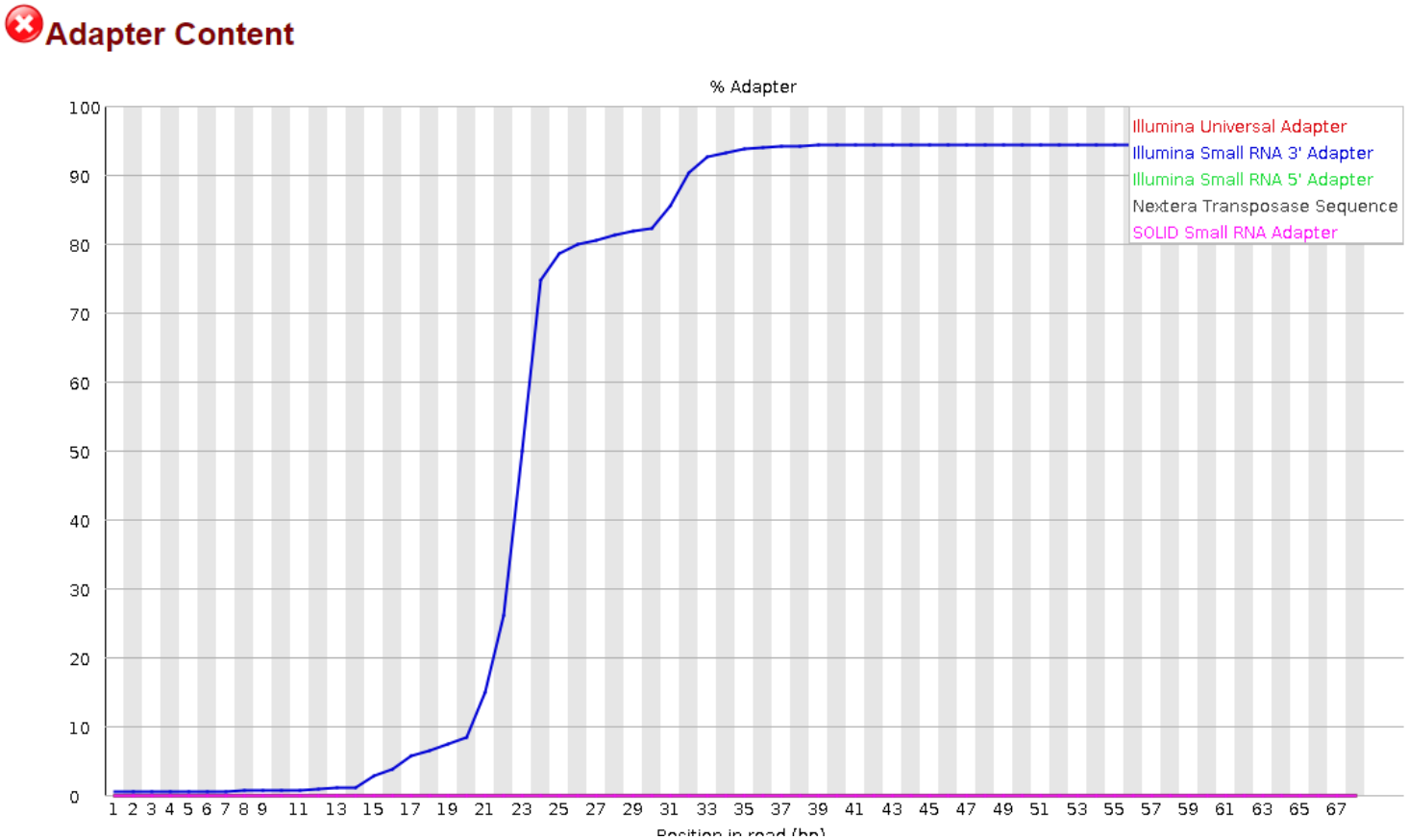


Click “View data”



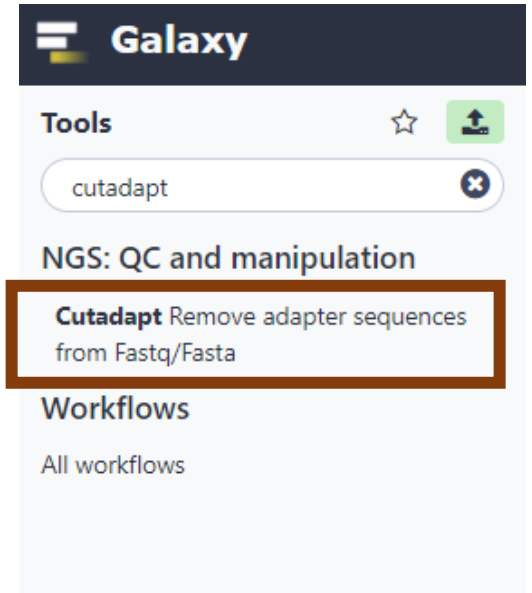
Explore various quality reports…

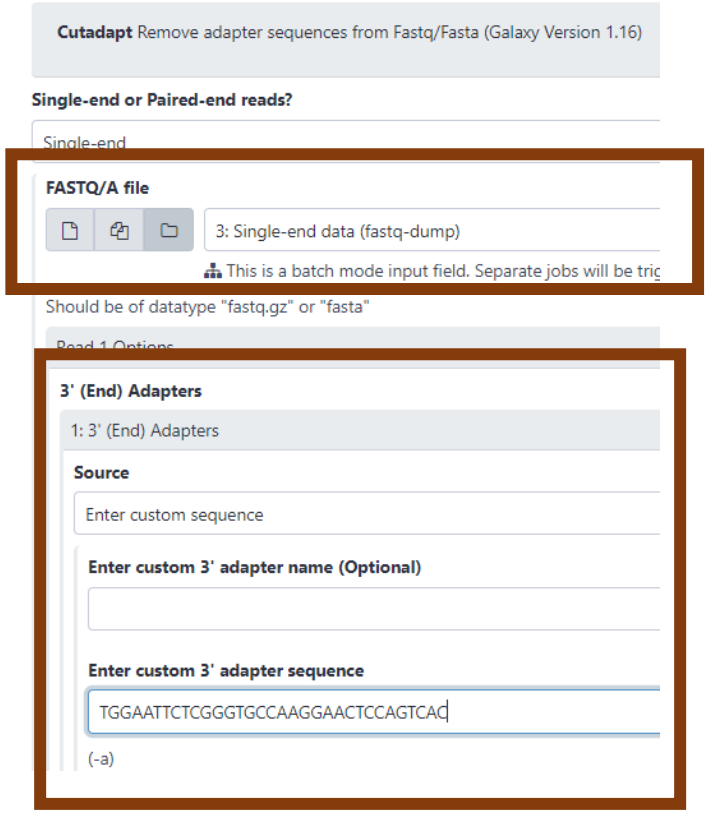
Step 2: Assess data quality



From FastQC results, we can see Illumina adapter contamination…

Step 3: Remove sequencing adapters with Cutadapt

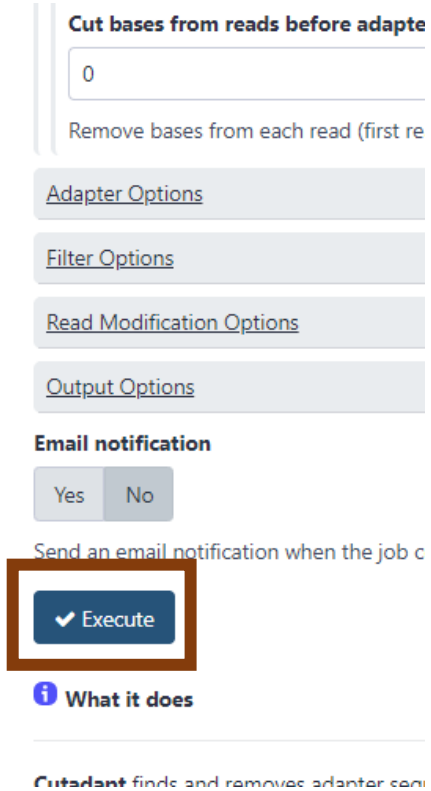


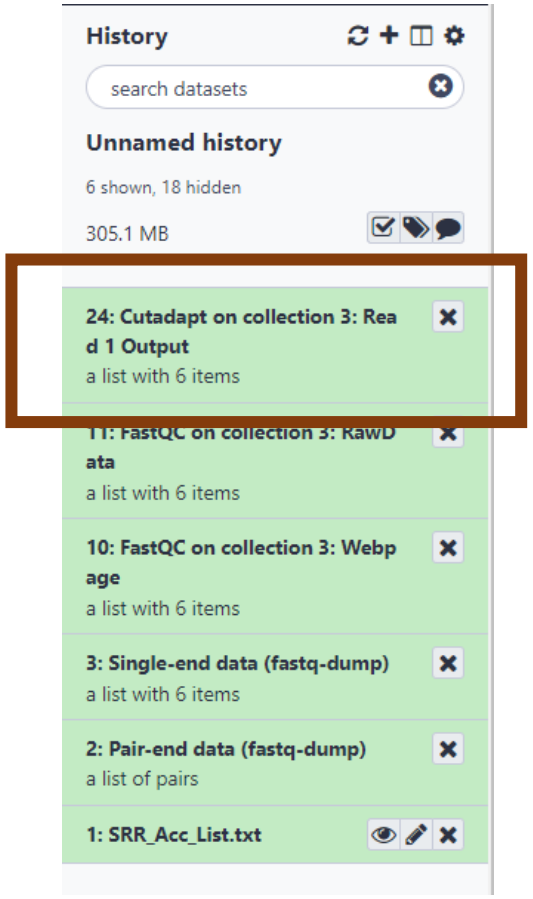


Select “**dataset collection**” Icon
Then select “**Single-end data**..” from drop-down

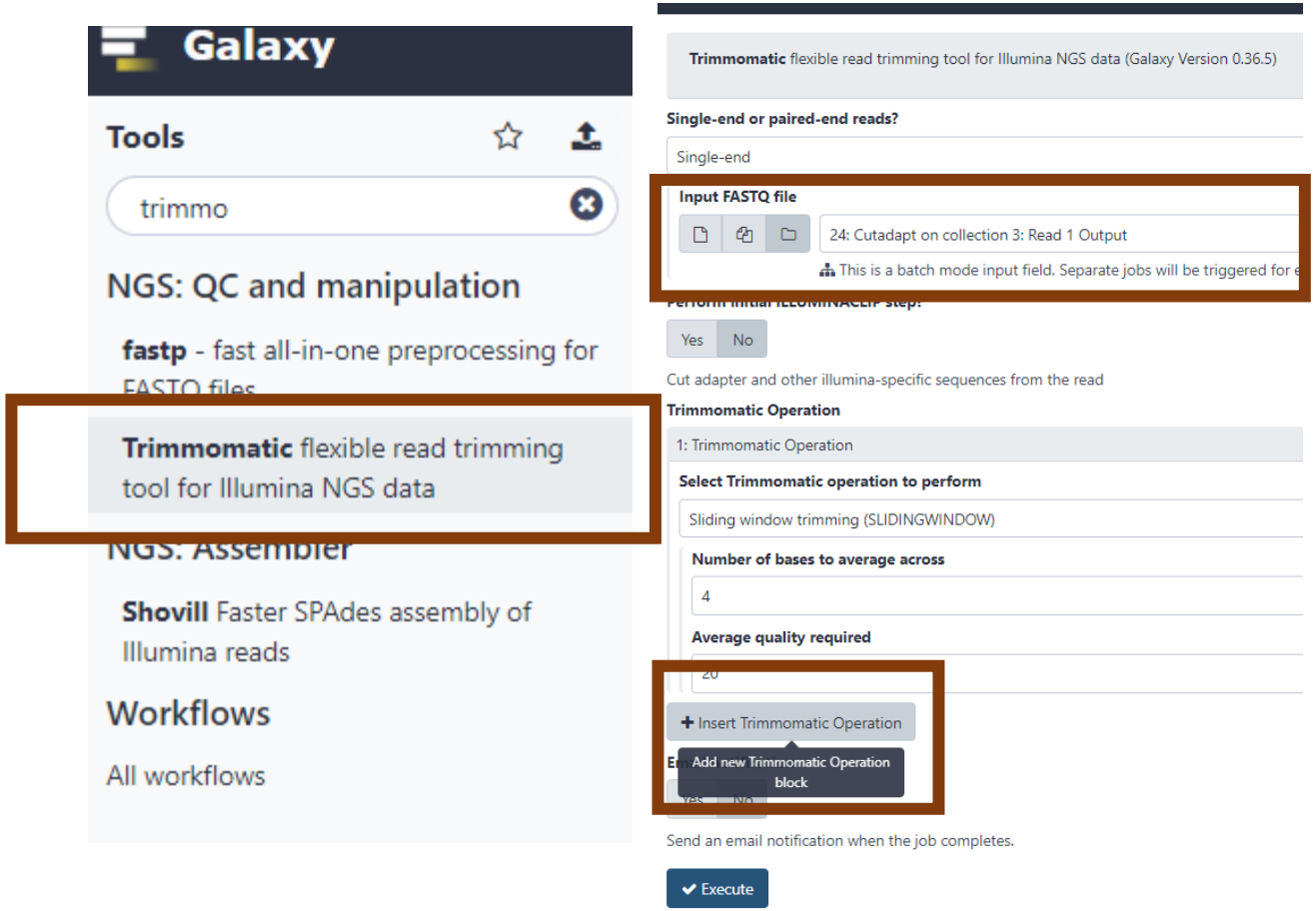
Copy/paste the 3’ adapter sequence:
TGGAATTCTCGGGTGCCAAGGAACTCCAGTCAC

Step 3: Remove sequencing adapters with Cutadapt





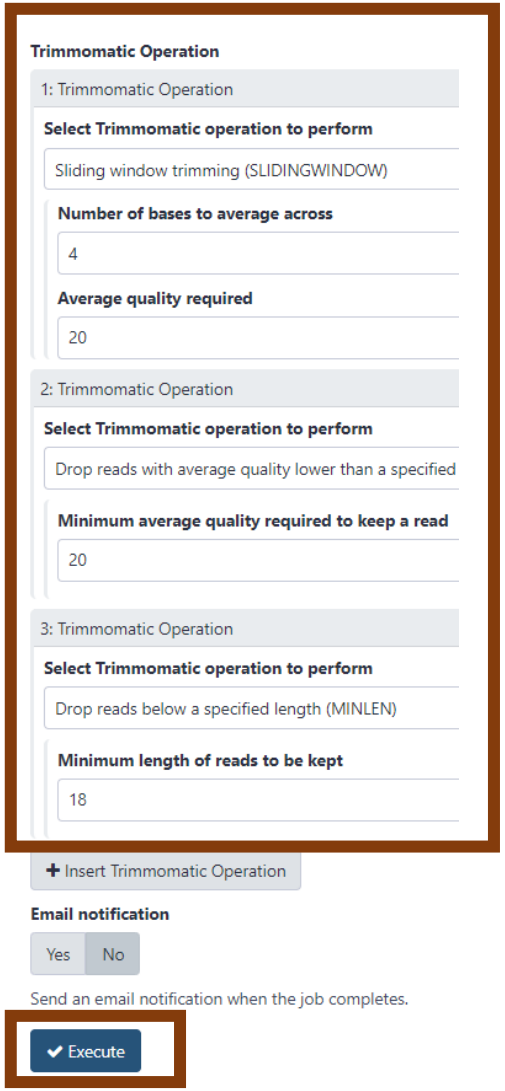
Step 4: Clean reads with Trimmomatic



Select “**dataset collection**” Icon
Then select “**Cutadapt on collection**…”
from drop-down

Click “**Insert Trimmomatic Operation**”
twice for a total of 3 “operations”

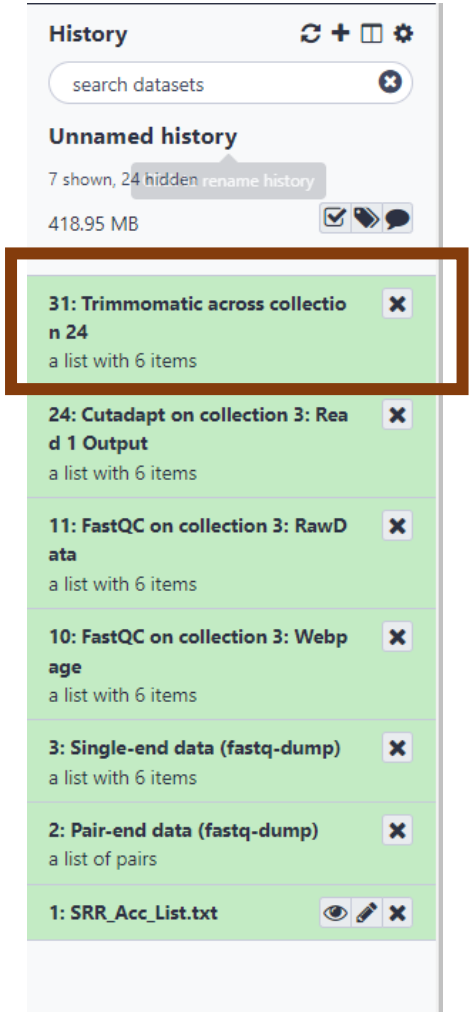
Step 4: Clean reads with Trimmomatic



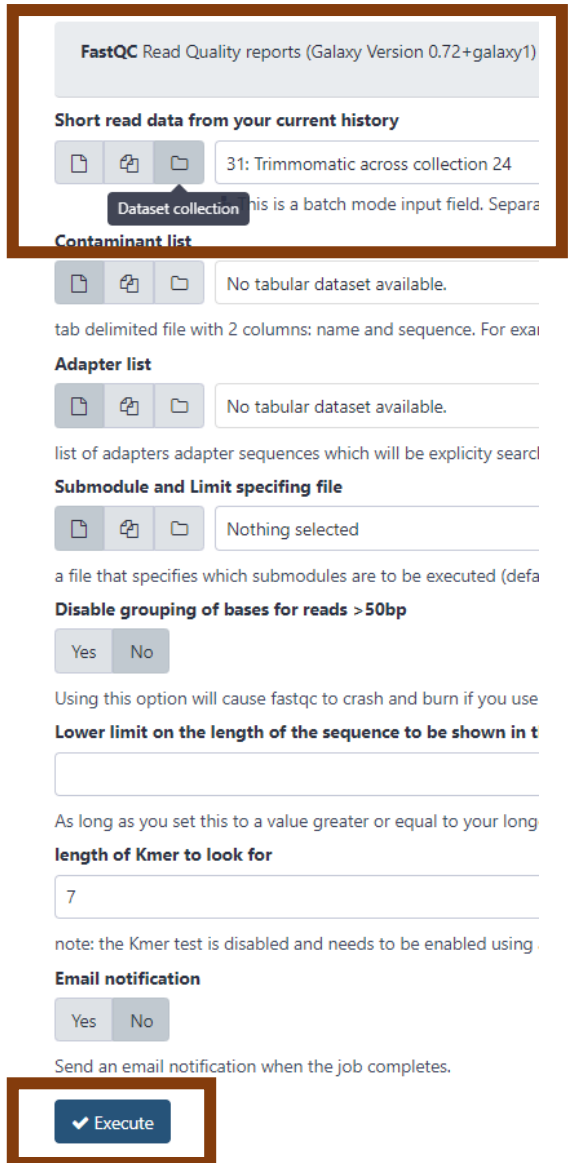
1st operation – leave as **defaults**

2nd operation – "**Drop reads with average quality**…" Min average quality = **20**

3rd operation – "**Drop reads below length**…" min length = **18**
(How long should miRNA reads be?)



Step 4: Clean reads with Trimmomatic

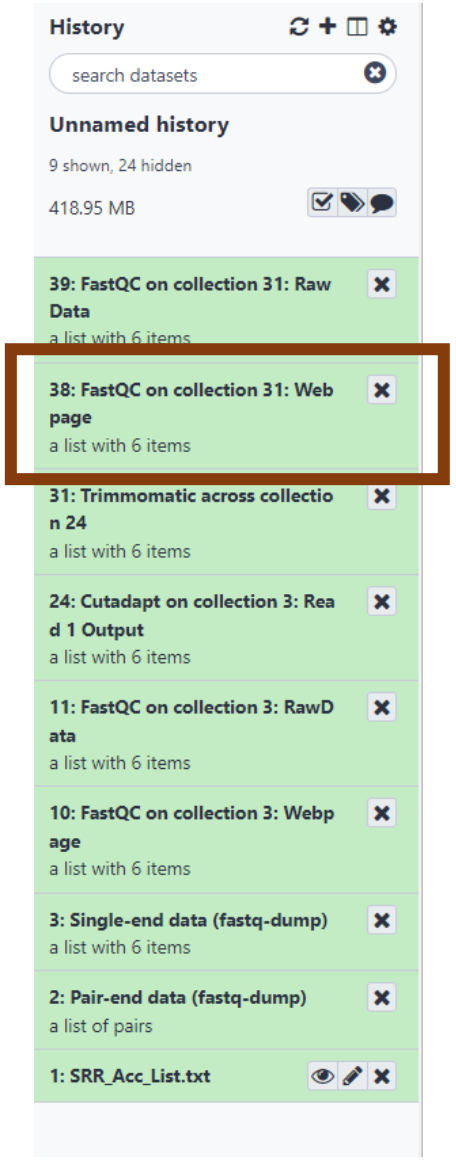


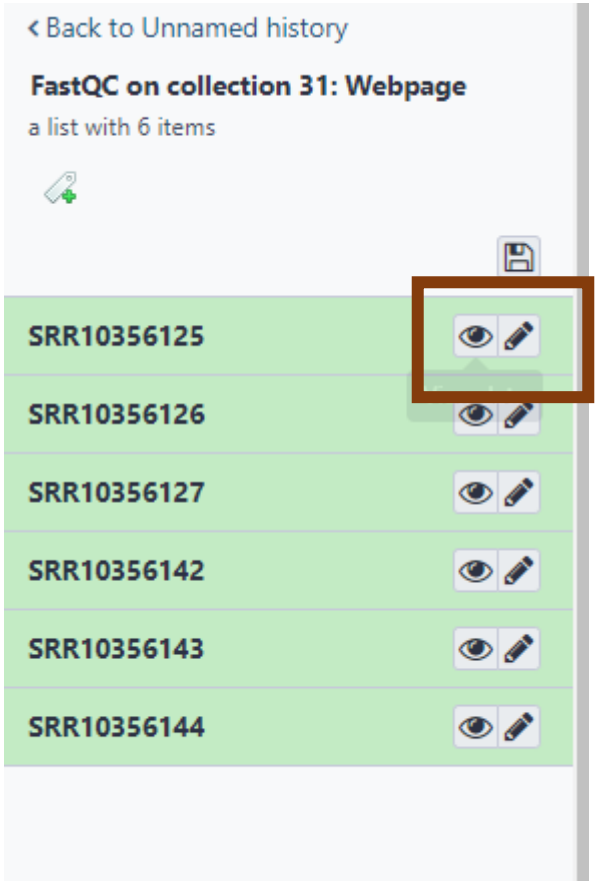
Select "**dataset collection**" Icon
Then select "**Trimmomatic across collection…**" from drop-down

Use **FastQC** to check quality of cleaned reads (Same as Step 2)

Step 4: Clean reads with Trimmomatic

Examine **FastQC** results as in step 2:





Step 4: Clean reads with Trimmomatic

FastQC Report

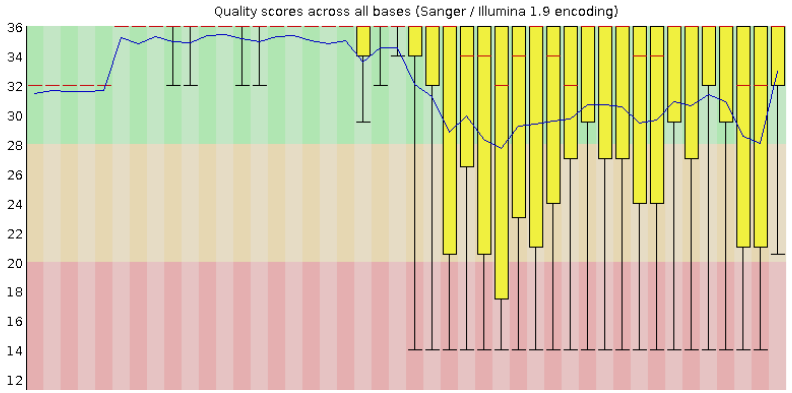
Summary

- Basic Statistics
- Per base sequence quality
- Per sequence quality scores
- Per base sequence content
- Per sequence GC content
- Per base N content
- Sequence Length Distribution
- Sequence Duplication Levels
- Overrepresented sequences
- Adapter Content

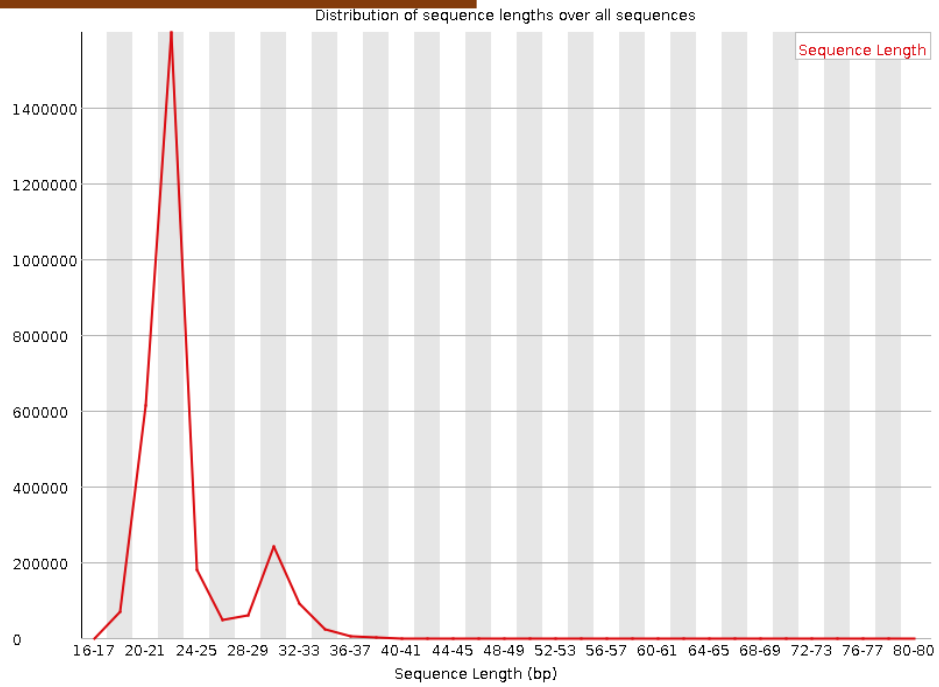
Basic Statistics

Measure	Value
Filename	SRR10356125.gz
File type	Conventional base calls
Encoding	Sanger / Illumina 1.9
Total Sequences	2979146
Sequences flagged as poor quality	0
Sequence length	18-79
%GC	51

Per base sequence quality

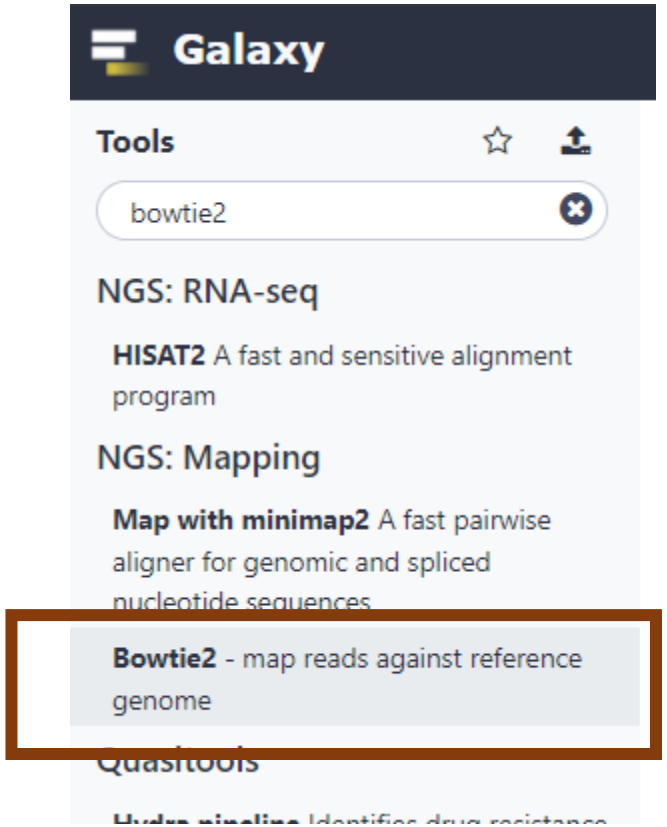


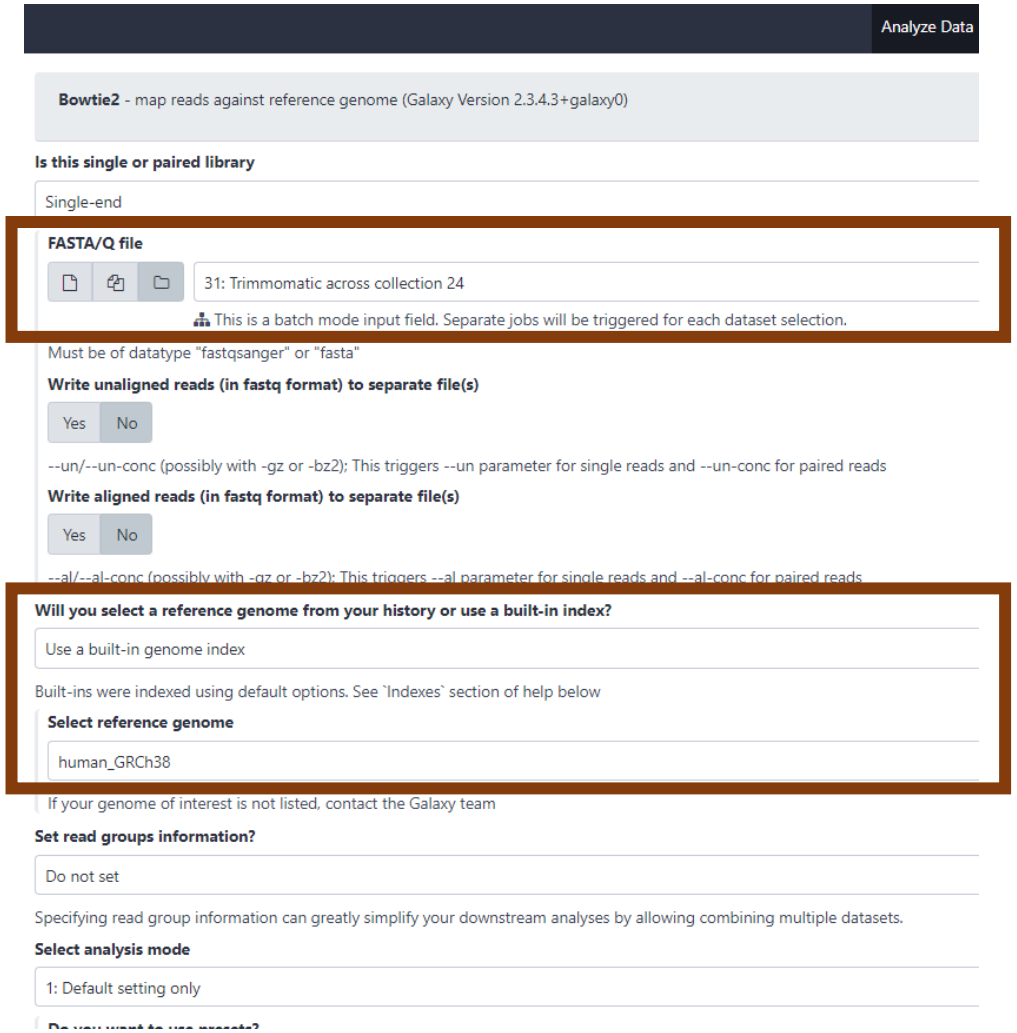
Sequence Length Distribution



Does the sequence length distribution make sense for **miRNAs**?

Step 5: Align to reference genome with Bowtie2

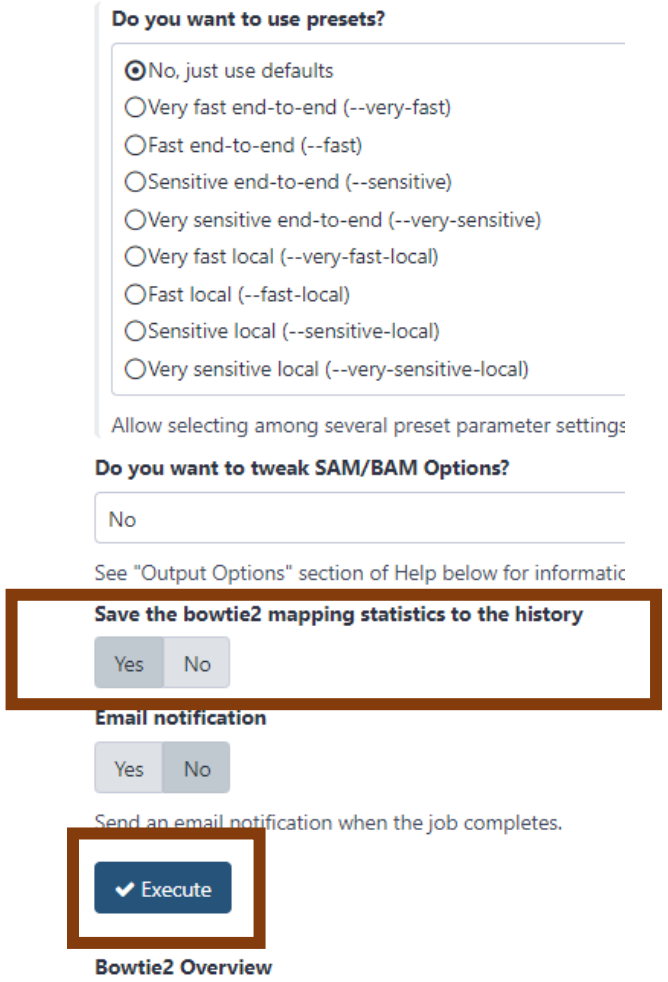


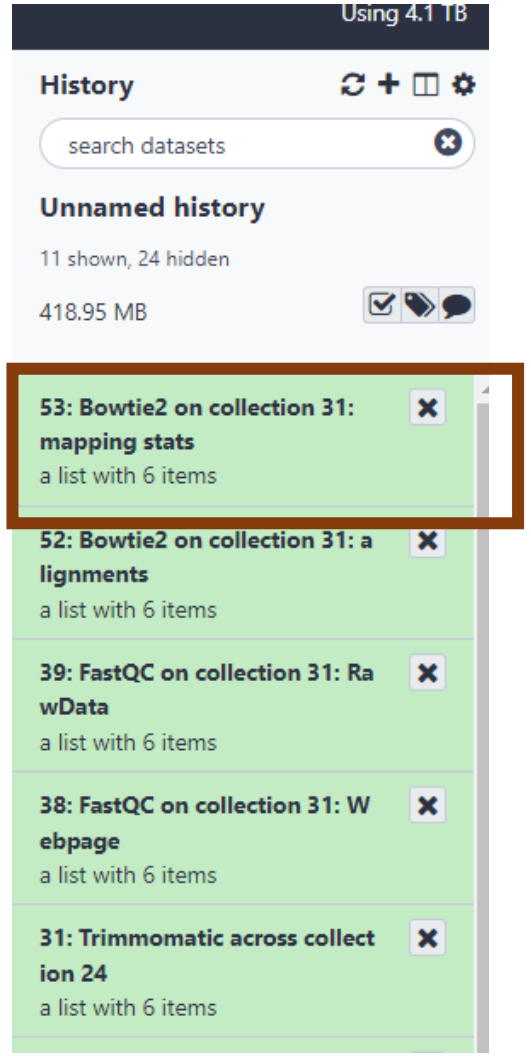


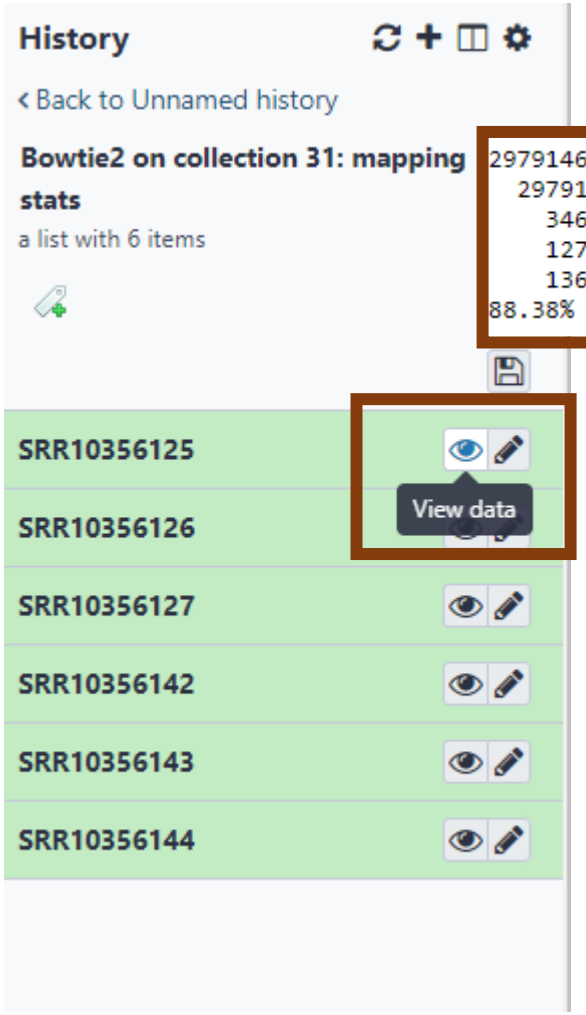
Select "**dataset collection**" Icon Then select "**Trimmomatic across collection…**" from drop-down

Select "**Built in genome index**" and choose **Human genome** from dropdown ***Note, we are using human genome, because elk annotation is not available*

Step 5: Align to reference genome with Bowtie2





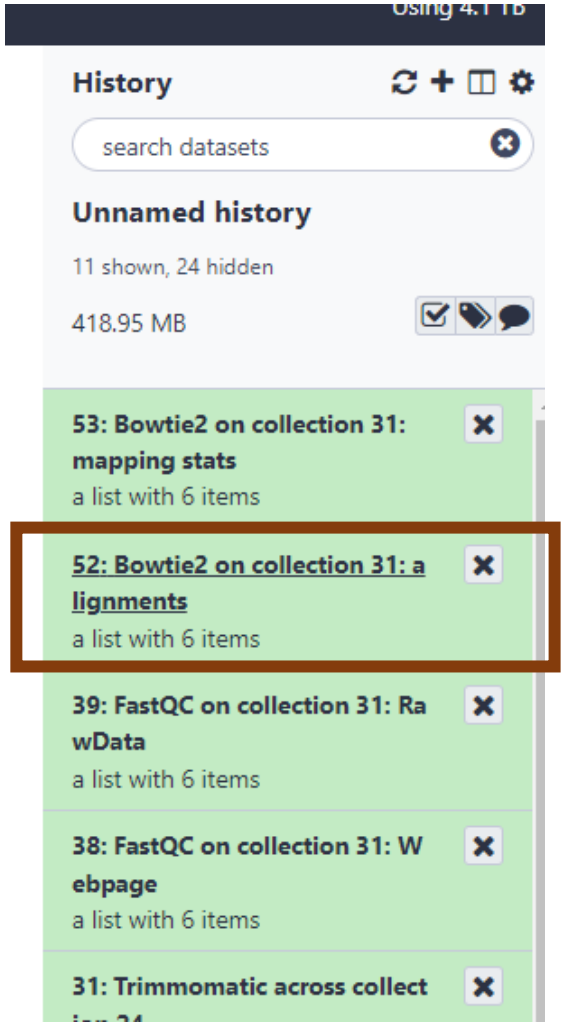


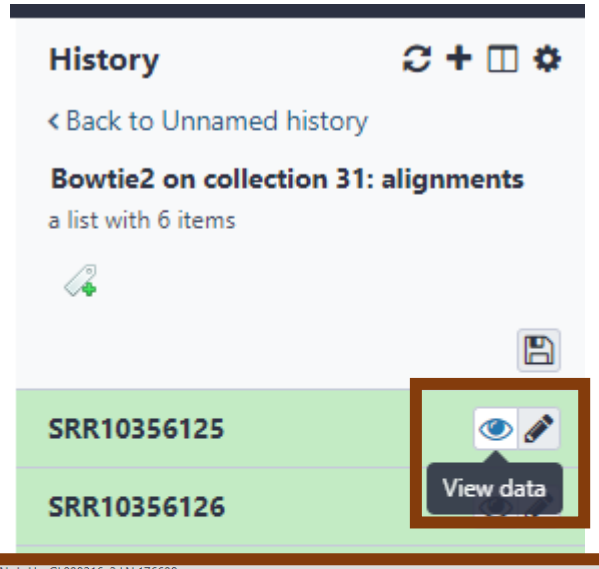

```
2979146 reads; of these:
  2979146 (100.00%) were unpaired; of these:
    346086 (11.62%) aligned 0 times
    1270093 (42.63%) aligned exactly 1 time
    1362967 (45.75%) aligned >1 times
88.38% overall alignment rate
```

View **mapping statistics**… notice that there is a high level of multi-mapped reads

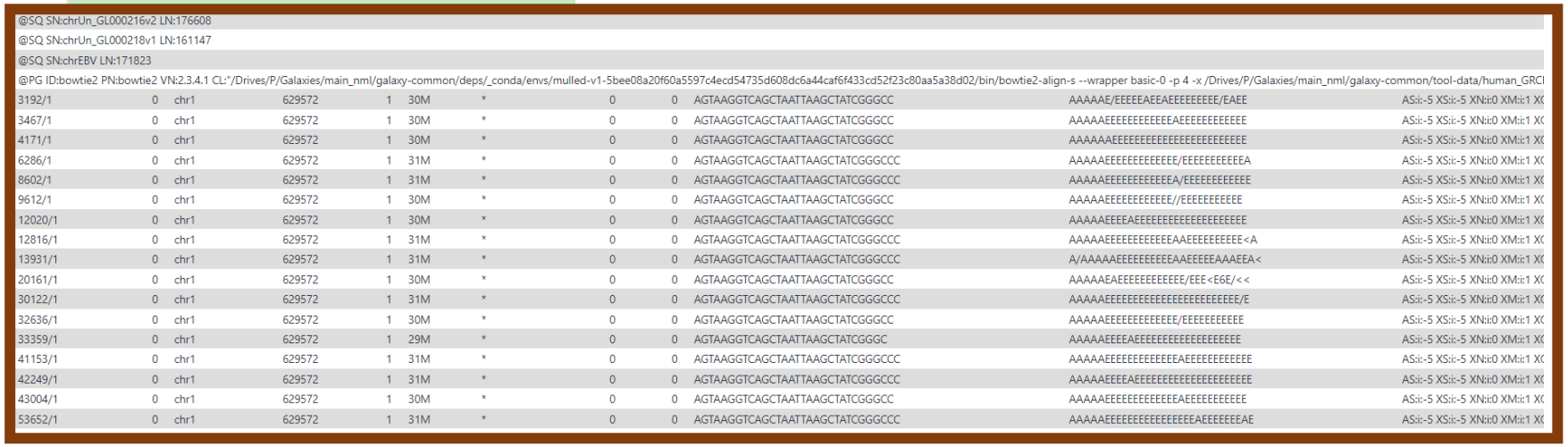
(Does this make sense for miRNAs?)

Step 5: Align to reference genome with Bowtie2



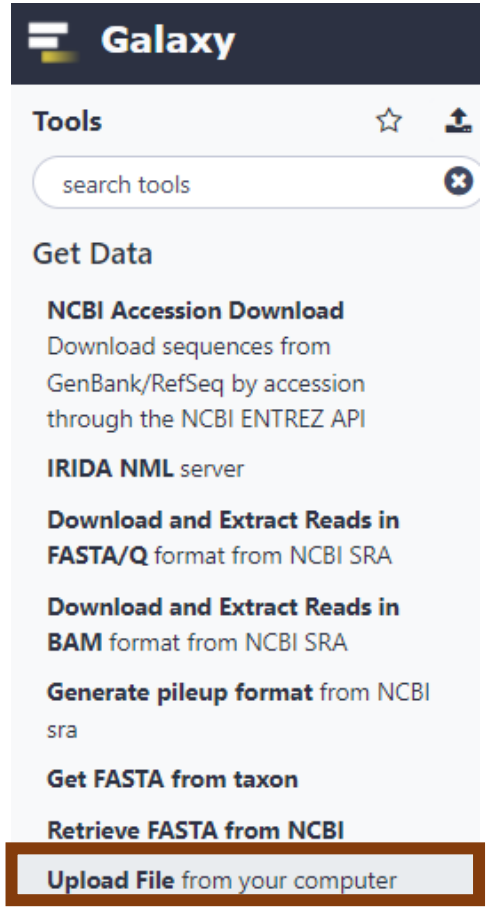


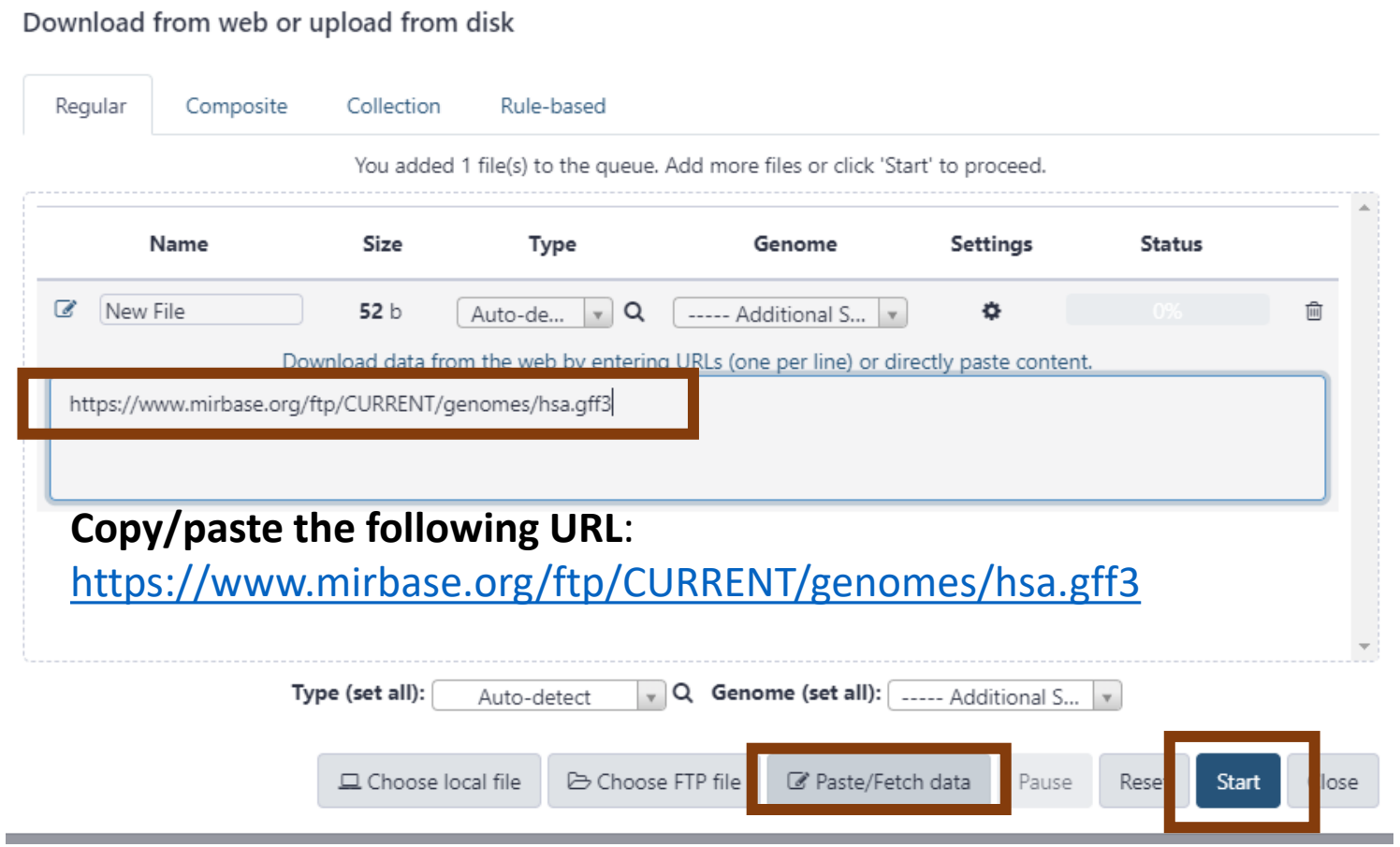
View alignment file in “BAM” format



```
@SQ SN:chrUn_GL000216v2 LN:176608
@SQ SN:chrUn_GL000218v1 LN:161147
@SQ SN:chrEBV LN:171823
@PG ID:bowtie2 PN:bowtie2 VN:2.3.4.1 CL:"/Drives/P/Galaxies/main_nml/galaxy-common/deps/_conda/envs/mulled-v1-5bee08a20f60a5597c4ecd54735d608dc6a44caf6f433cd52f23c80aa5a38d02/bin/bowtie2-align-s --wrapper basic-0 -p 4 -x /Drives/P/Galaxies/main_nml/galaxy-common/tool-data/human_GRCI
3192/1    0  chr1  629572  1  30M  *  0  0  AGTAAGGTCAGCTAATTAAGCTATCGGGCC    AAAAAE/EEEEEAEEAEEEEEEEEE/EAEE    AS:i:-5 XS:i:-5 XN:i:0 XM:i:1 XC
3467/1    0  chr1  629572  1  30M  *  0  0  AGTAAGGTCAGCTAATTAAGCTATCGGGCC    AAAAAEEEEEEEEEEEEAEEEEEEEEEEE    AS:i:-5 XS:i:-5 XN:i:0 XM:i:1 XC
4171/1    0  chr1  629572  1  30M  *  0  0  AGTAAGGTCAGCTAATTAAGCTATCGGGCC    AAAAAAEEEEEEEEEEEEEEEEEEEEEEEE    AS:i:-5 XS:i:-5 XN:i:0 XM:i:1 XC
6286/1    0  chr1  629572  1  31M  *  0  0  AGTAAGGTCAGCTAATTAAGCTATCGGGCCC    AAAAAEEEEEEEEEEEE/EEEEEEEEEEA    AS:i:-5 XS:i:-5 XN:i:0 XM:i:1 XC
8602/1    0  chr1  629572  1  31M  *  0  0  AGTAAGGTCAGCTAATTAAGCTATCGGGCCC    AAAAAEEEEEEEEEEEEA/EEEEEEEEEEE    AS:i:-5 XS:i:-5 XN:i:0 XM:i:1 XC
9612/1    0  chr1  629572  1  30M  *  0  0  AGTAAGGTCAGCTAATTAAGCTATCGGGCC    AAAAAEEEEEEEEEEEE//EEEEEEEEEE    AS:i:-5 XS:i:-5 XN:i:0 XM:i:1 XC
12020/1   0  chr1  629572  1  30M  *  0  0  AGTAAGGTCAGCTAATTAAGCTATCGGGCC    AAAAAEEEEAEEEEEEEEEEEEEEEEEE    AS:i:-5 XS:i:-5 XN:i:0 XM:i:1 XC
12816/1   0  chr1  629572  1  31M  *  0  0  AGTAAGGTCAGCTAATTAAGCTATCGGGCCC    AAAAAEEEEEEEEEEEEAAEEEEEEEEEE<A    AS:i:-5 XS:i:-5 XN:i:0 XM:i:1 XC
13931/1   0  chr1  629572  1  31M  *  0  0  AGTAAGGTCAGCTAATTAAGCTATCGGGCCC    A/AAAAAEEEEEEEEEEAAEEEEEAAAEEA<    AS:i:-5 XS:i:-5 XN:i:0 XM:i:1 XC
20161/1   0  chr1  629572  1  30M  *  0  0  AGTAAGGTCAGCTAATTAAGCTATCGGGCC    AAAAAEAEEEEEEEEEEE/EEE<E6E/<<    AS:i:-5 XS:i:-5 XN:i:0 XM:i:1 XC
30122/1   0  chr1  629572  1  31M  *  0  0  AGTAAGGTCAGCTAATTAAGCTATCGGGCCC    AAAAAEEEEEEEEEEEEEEEEEEEEEE/E    AS:i:-5 XS:i:-5 XN:i:0 XM:i:1 XC
32636/1   0  chr1  629572  1  30M  *  0  0  AGTAAGGTCAGCTAATTAAGCTATCGGGCC    AAAAAEEEEEEEEEEEE/EEEEEEEEEE    AS:i:-5 XS:i:-5 XN:i:0 XM:i:1 XC
33359/1   0  chr1  629572  1  29M  *  0  0  AGTAAGGTCAGCTAATTAAGCTATCGGGC    AAAAAEEEEAEEEEEEEEEEEEEEEEE    AS:i:-5 XS:i:-5 XN:i:0 XM:i:1 XC
41153/1   0  chr1  629572  1  31M  *  0  0  AGTAAGGTCAGCTAATTAAGCTATCGGGCCC    AAAAAEEEEEEEEEEEEAEEEEEEEEEEE    AS:i:-5 XS:i:-5 XN:i:0 XM:i:1 XC
42249/1   0  chr1  629572  1  31M  *  0  0  AGTAAGGTCAGCTAATTAAGCTATCGGGCCC    AAAAAEEEEAEEEEEEEEEEEEEEEEEE    AS:i:-5 XS:i:-5 XN:i:0 XM:i:1 XC
43004/1   0  chr1  629572  1  30M  *  0  0  AGTAAGGTCAGCTAATTAAGCTATCGGGCC    AAAAAEEEEEEEEEEEEAEEEEEEEEEE    AS:i:-5 XS:i:-5 XN:i:0 XM:i:1 XC
53652/1   0  chr1  629572  1  31M  *  0  0  AGTAAGGTCAGCTAATTAAGCTATCGGGCCC    AAAAAEEEEEEEEEEEEEEEEAEEEEEEEAE    AS:i:-5 XS:i:-5 XN:i:0 XM:i:1 XC
```

Step 6: Map and count miRNA reads with FeatureCounts





Copy/paste the following URL:
https://www.mirbase.org/ftp/CURRENT/genomes/hsa.gff3

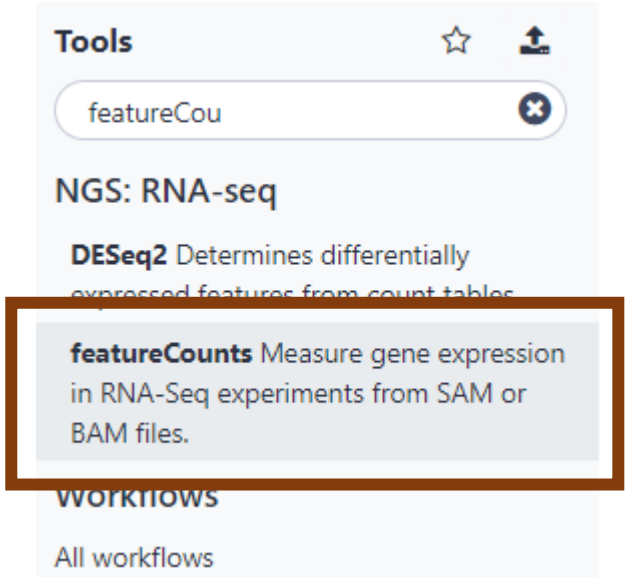
Upload **reference annotation** file in GFF format

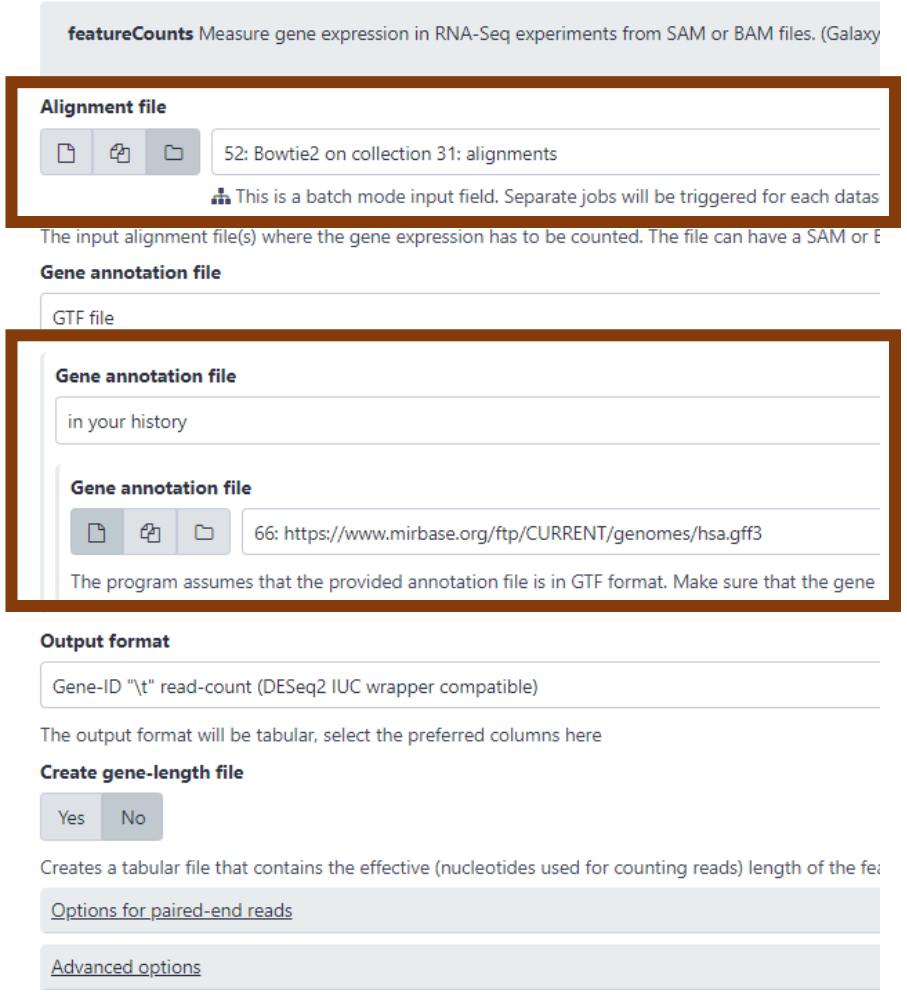
Step 6: Map and count miRNA reads with FeatureCounts

Seqid	Source	Type	Start	End	Score	Strand	Phase	Attributes
##gff-version 3								
##date 2018-3-5								
#								
# Chromosomal coordinates of Homo sapiens microRNAs								
# microRNAs: miRBase v22								
# genome-build-id: GRCh38								
# genome-build-accession: NCBI_Assembly:GCA_000001405.15								
#								
# Hairpin precursor sequences have type "miRNA_primary_transcript".								
# Note, these sequences do not represent the full primary transcript,								
# rather a predicted stem-loop portion that includes the precursor								
# miRNA. Mature sequences have type "miRNA".								
#								
chr1	.	miRNA_primary_transcript	17369	17436	.	-	.	ID=MI0022705;Alias=MI0022705;Name=hsa-mir-6859-1
chr1	.	miRNA	17409	17431	.	-	.	ID=MIMAT0027618;Alias=MIMAT0027618;Name=hsa-miR-6859-5p;Derives_from=MI0022705
chr1	.	miRNA	17369	17391	.	-	.	ID=MIMAT0027619;Alias=MIMAT0027619;Name=hsa-miR-6859-3p;Derives_from=MI0022705
chr1	.	miRNA_primary_transcript	30366	30503	.	+	.	ID=MI0006363;Alias=MI0006363;Name=hsa-mir-1302-2
chr1	.	miRNA	30438	30458	.	+	.	ID=MIMAT0005890;Alias=MIMAT0005890;Name=hsa-miR-1302;Derives_from=MI0006363
chr1	.	miRNA_primary_transcript	187891	187958	.	-	.	ID=MI0026420;Alias=MI0026420;Name=hsa-mir-6859-2
chr1	.	miRNA	187931	187953	.	-	.	ID=MIMAT0027618_1;Alias=MIMAT0027618;Name=hsa-miR-6859-5p;Derives_from=MI0026420
chr1	.	miRNA	187891	187913	.	-	.	ID=MIMAT0027619_1;Alias=MIMAT0027619;Name=hsa-miR-6859-3p;Derives_from=MI0026420
chr1	.	miRNA_primary_transcript	632615	632685	.	-	.	ID=MI0039740;Alias=MI0039740;Name=hsa-mir-12136
chr1	.	miRNA	632668	632685	.	-	.	ID=MIMAT0049032;Alias=MIMAT0049032;Name=hsa-miR-12136;Derives_from=MI0039740
chr1	.	miRNA_primary_transcript	1167104	1167198	.	+	.	ID=MI0000342;Alias=MI0000342;Name=hsa-mir-200b
chr1	.	miRNA	1167124	1167145	.	+	.	ID=MIMAT0004571;Alias=MIMAT0004571;Name=hsa-miR-200b-5p;Derives_from=MI0000342
chr1	.	miRNA	1167160	1167181	.	+	.	ID=MIMAT0000318;Alias=MIMAT0000318;Name=hsa-miR-200b-3p;Derives_from=MI0000342
chr1	.	miRNA_primary_transcript	1167863	1167952	.	+	.	ID=MI0000737;Alias=MI0000737;Name=hsa-mir-200a
chr1	.	miRNA	1167878	1167899	.	+	.	ID=MIMAT0001620;Alias=MIMAT0001620;Name=hsa-miR-200a-5p;Derives_from=MI0000737
chr1	.	miRNA	1167916	1167937	.	+	.	ID=MIMAT0000682;Alias=MIMAT0000682;Name=hsa-miR-200a-3p;Derives_from=MI0000737
chr1	.	miRNA_primary_transcript	1169005	1169087	.	+	.	ID=MI0001641;Alias=MI0001641;Name=hsa-mir-429
chr1	.	miRNA	1169055	1169076	.	+	.	ID=MIMAT0001536;Alias=MIMAT0001536;Name=hsa-miR-429;Derives_from=MI0001641
chr1	.	miRNA_primary_transcript	1296110	1296170	.	-	.	ID=MI0022571;Alias=MI0022571;Name=hsa-mir-6726
chr1	.	miRNA	1296145	1296165	.	-	.	ID=MIMAT0027353;Alias=MIMAT0027353;Name=hsa-miR-6726-5p;Derives_from=MI0022571
chr1	.	miRNA	1296110	1296129	.	-	.	ID=MIMAT0027354;Alias=MIMAT0027354;Name=hsa-miR-6726-3p;Derives_from=MI0022571

View annotation file in GFF format

Step 6: Map and count miRNA reads with FeatureCounts

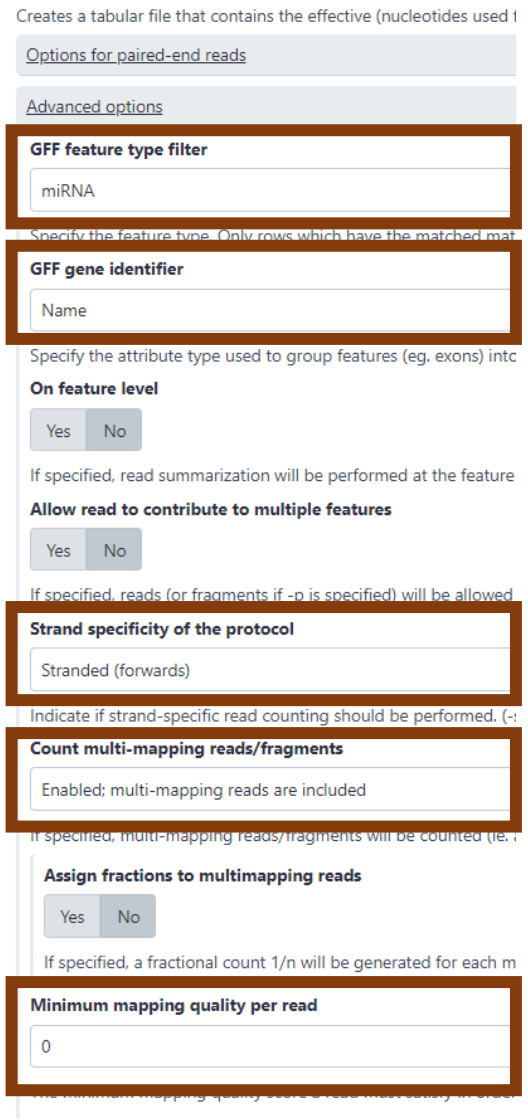




Select “**dataset collection**” Icon
Then select “**Bowtie2 on collection…**” from drop-down

Select “in your history”
Then select **miRbase GFF file** from drop-down

Step 6: Map and count miRNA reads with FeatureCounts



Change GFF feature type to “**miRNA**”

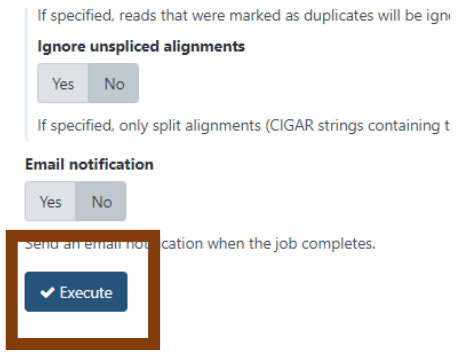
Change GFF gene identifier to “**Name**”

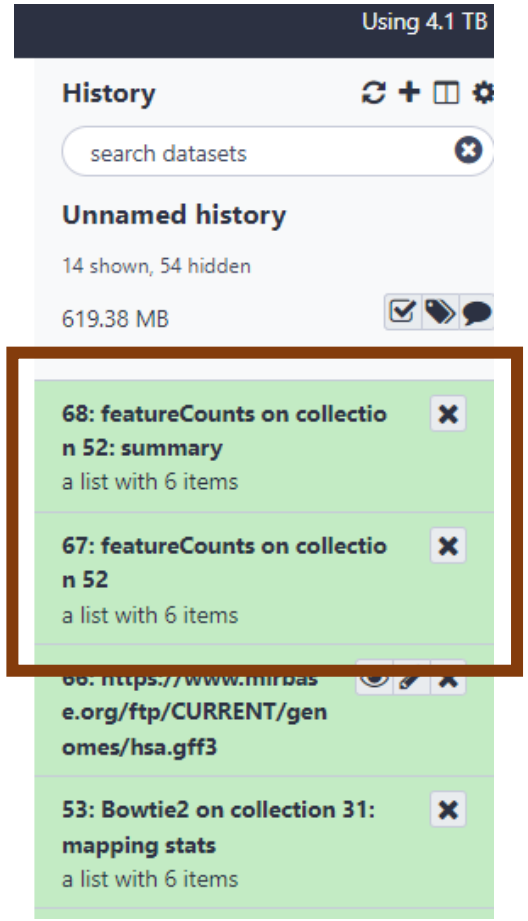
Change strand specificity to “**Stranded (forwards)**”

Enable **multi-mapped** read counting

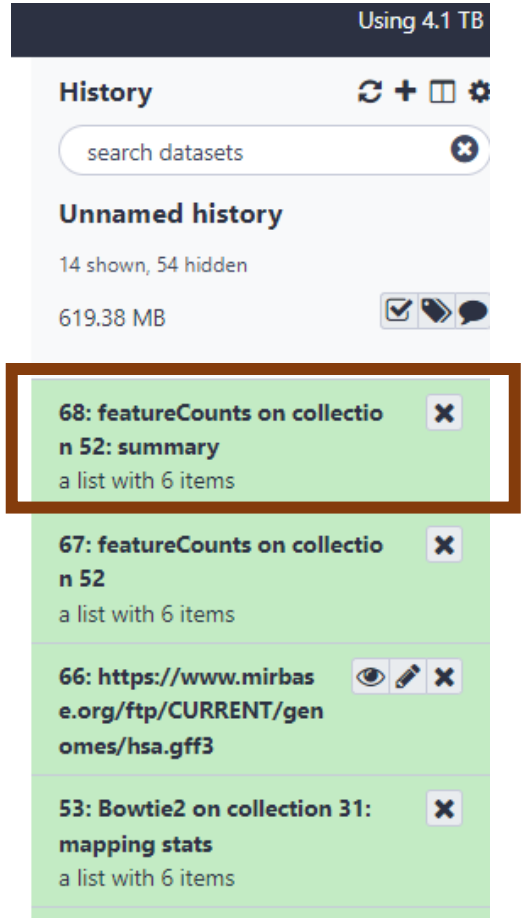
Set minimum read quality to 0
(This is because of how bowtie2 deals with multi-mapped read)

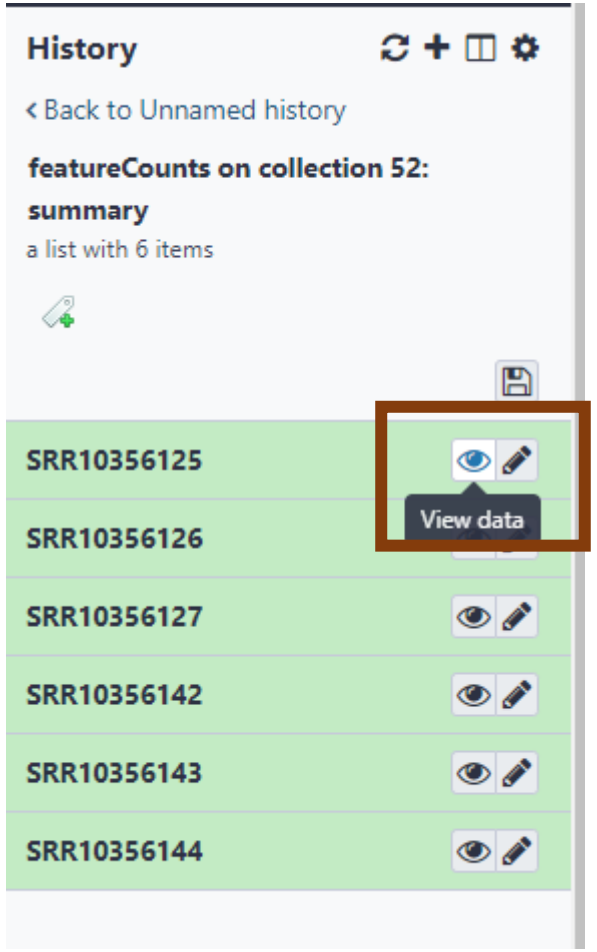
Step 6: Map and count miRNA reads with FeatureCounts





Step 6: Map and count miRNA reads with FeatureCounts

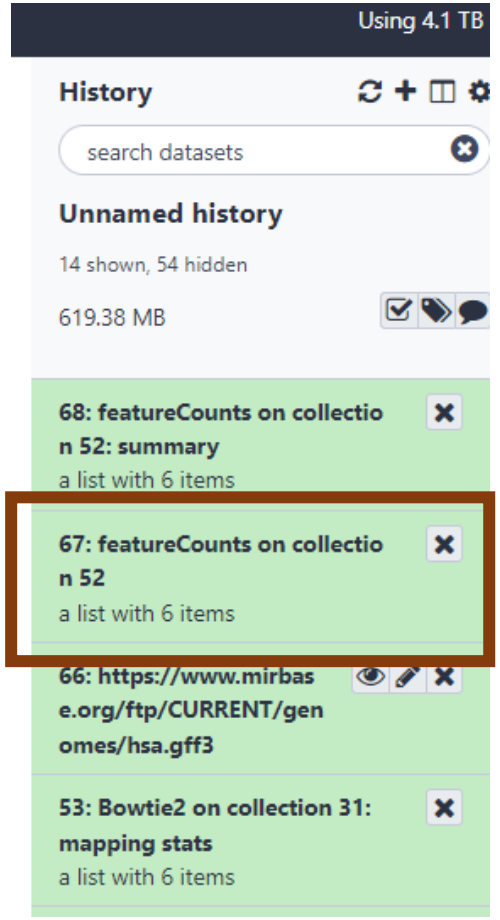


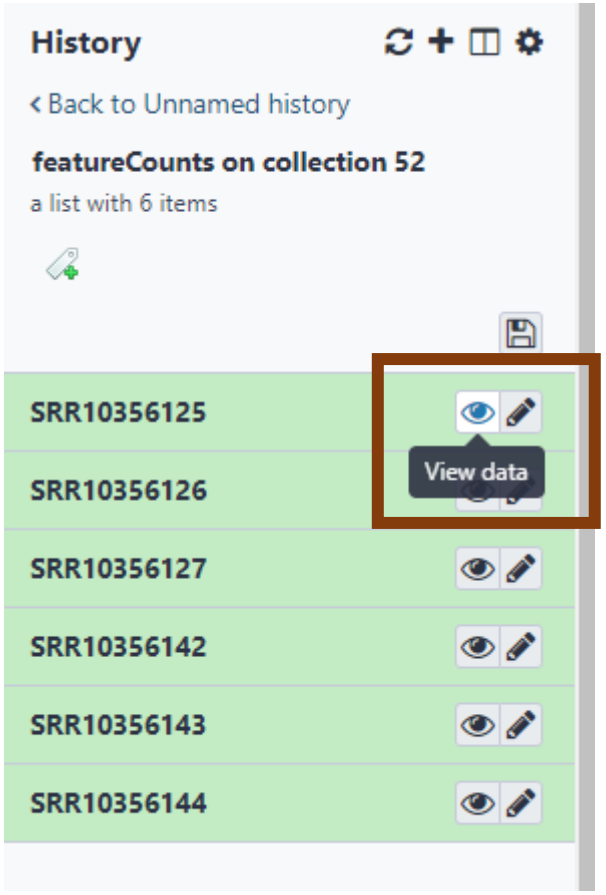


Status	SRR10356125
Assigned	2149693
Unassigned_Unmapped	346086
Unassigned_MappingQuality	0
Unassigned_Chimera	0
Unassigned_FragmentLength	0
Unassigned_Duplicate	0
Unassigned_MultiMapping	0
Unassigned_Secondary	0
Unassigned_Nonjunction	0
Unassigned_NoFeatures	483367
Unassigned_Overlapping_Length	0
Unassigned_Ambiguity	0

Examine mapping statistics

Step 6: Map and count miRNA reads with FeatureCounts

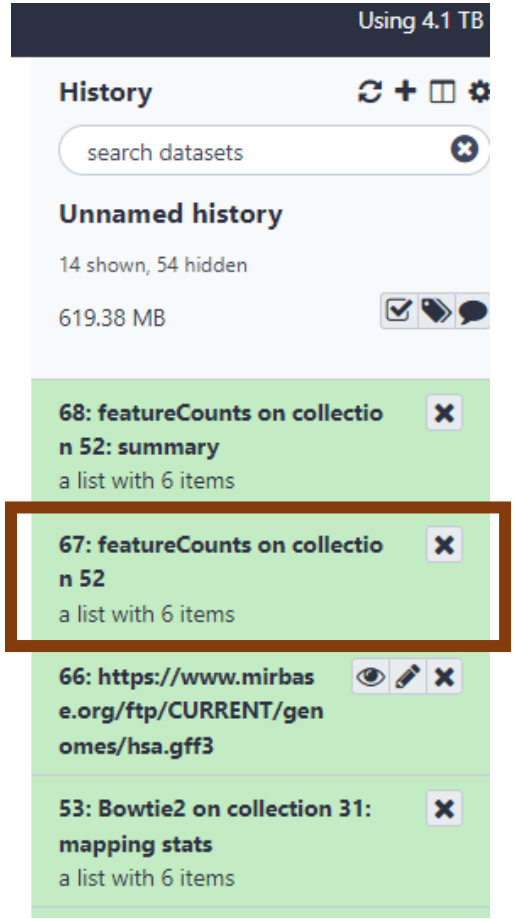


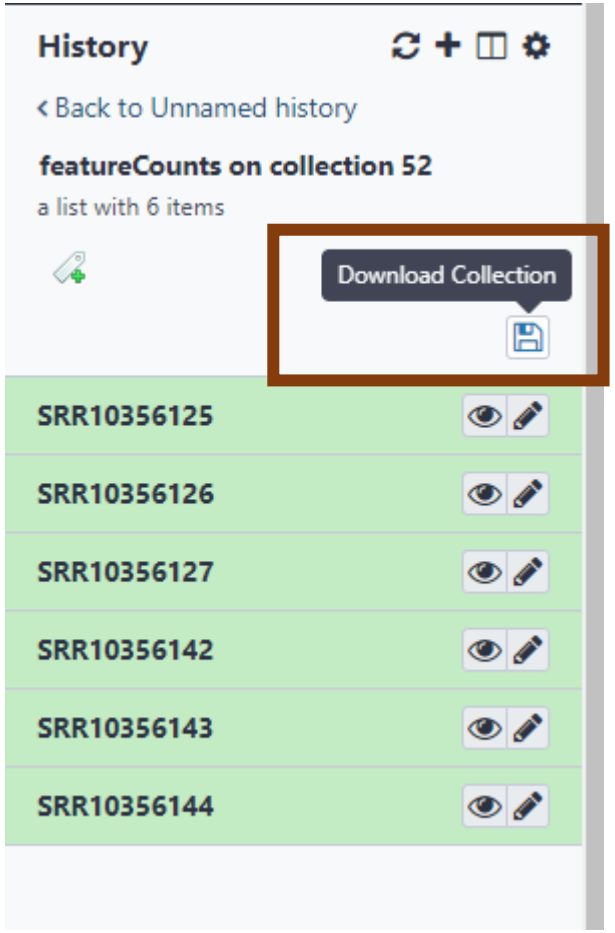


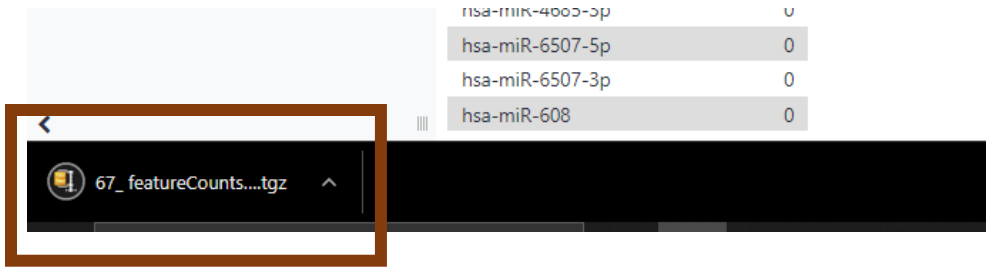
Geneid	SRR10356125
hsa-miR-6859-5p	0
hsa-miR-6859-3p	0
hsa-miR-1302	0
hsa-miR-12136	0
hsa-miR-200b-5p	0
hsa-miR-200b-3p	1
hsa-miR-200a-5p	0
hsa-miR-200a-3p	5
hsa-miR-429	0
hsa-miR-6726-5p	0
hsa-miR-6726-3p	0
hsa-miR-6727-5p	0
hsa-miR-6727-3p	0
hsa-miR-6808-5p	0
hsa-miR-6808-3p	0
hsa-miR-4251	0
hsa-miR-551a	0
hsa-miR-4689	0
hsa-miR-4252	0
hsa-miR-6728-5p	0
hsa-miR-6728-3p	0
hsa-miR-34a-5p	2

Examine read count files

Step 7: Download read count files for further analysis

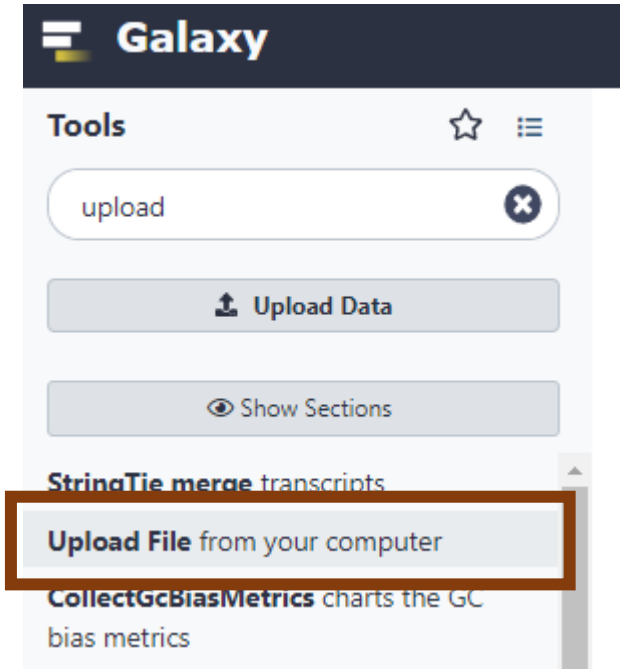


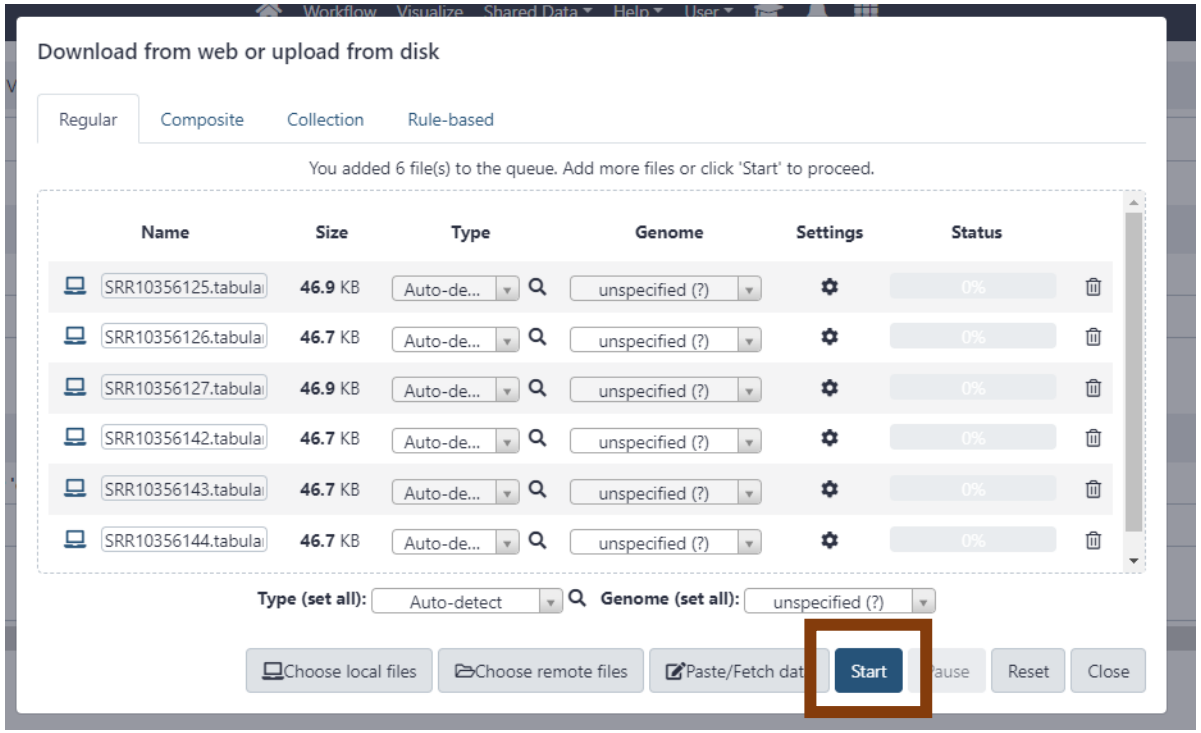




Download and unzip/extract raw read count files

Step 8: Differential expression analysis with DESeq2 in Galaxy

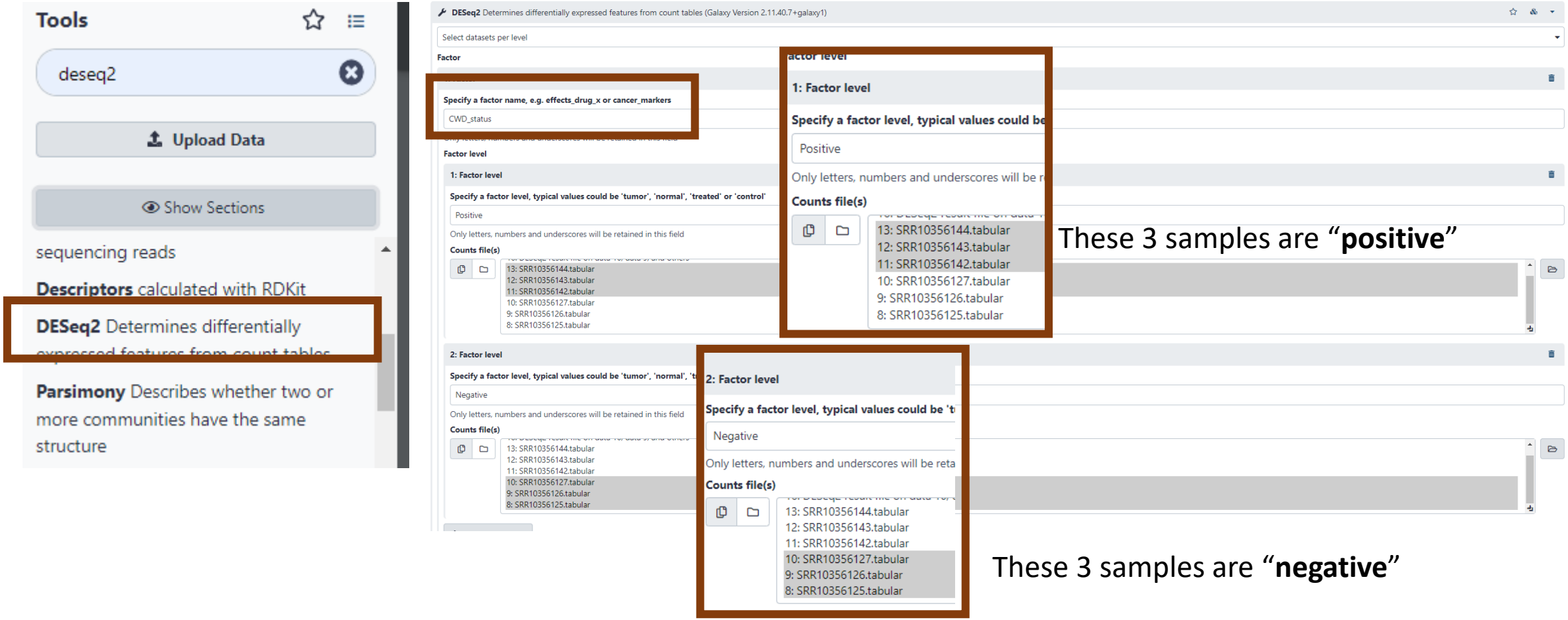




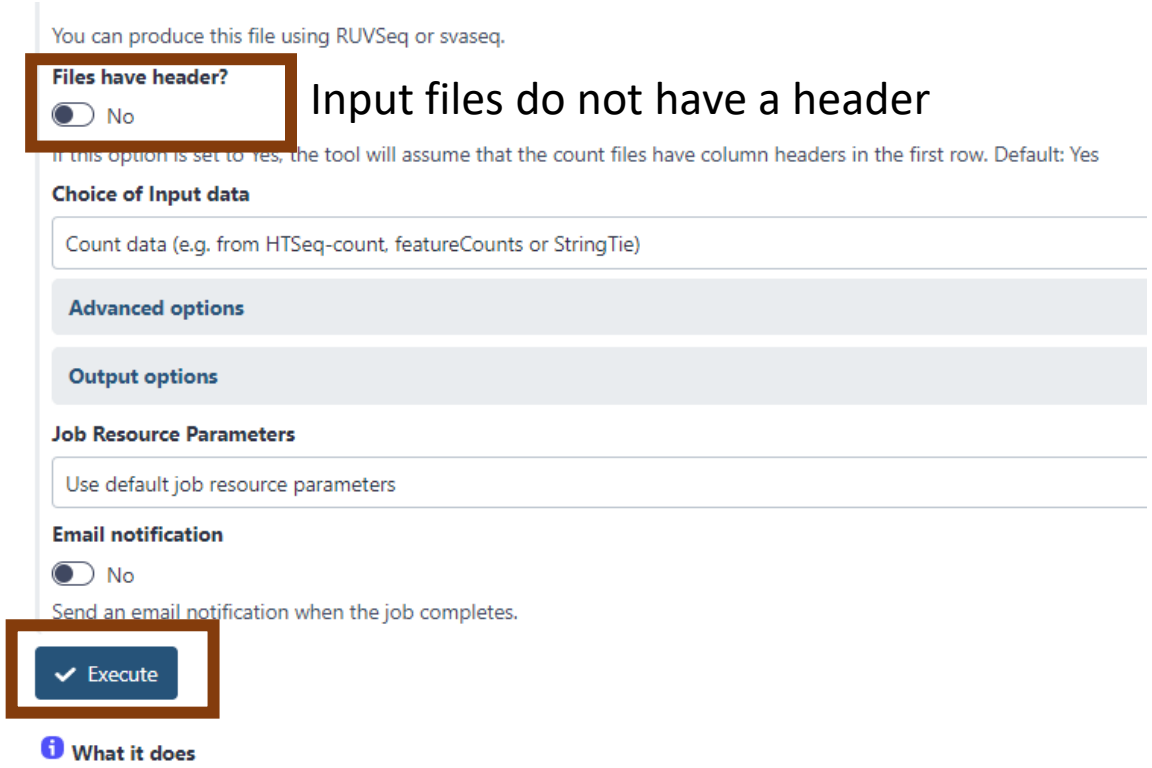
Re-upload raw read count files as individual files, instead of a collection

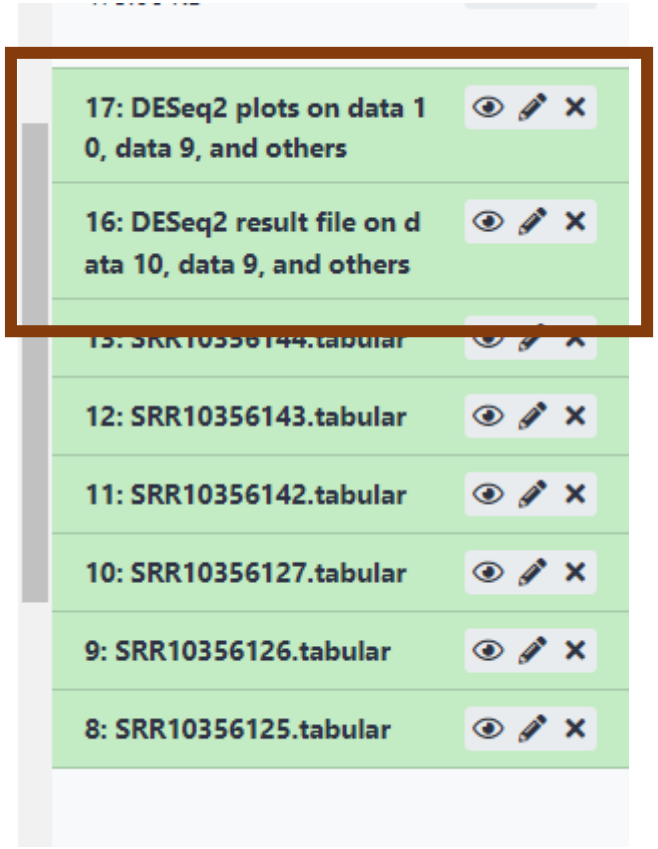
Step 8: Differential expression analysis with DESeq2 in Galaxy

Factor Name = **CWD_Status**

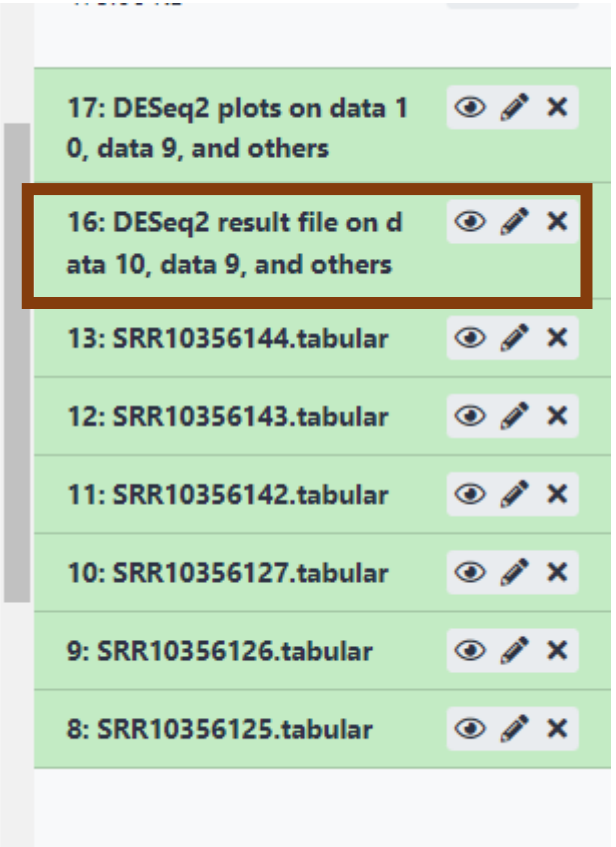


Step 8: Differential expression analysis with DESeq2 in Galaxy





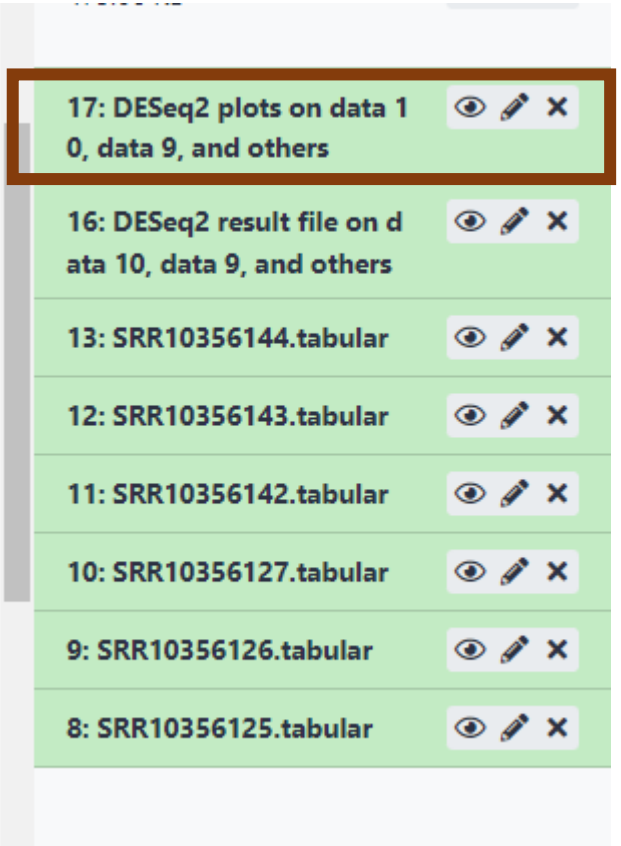
Step 8: Differential expression analysis with DESeq2 in Galaxy

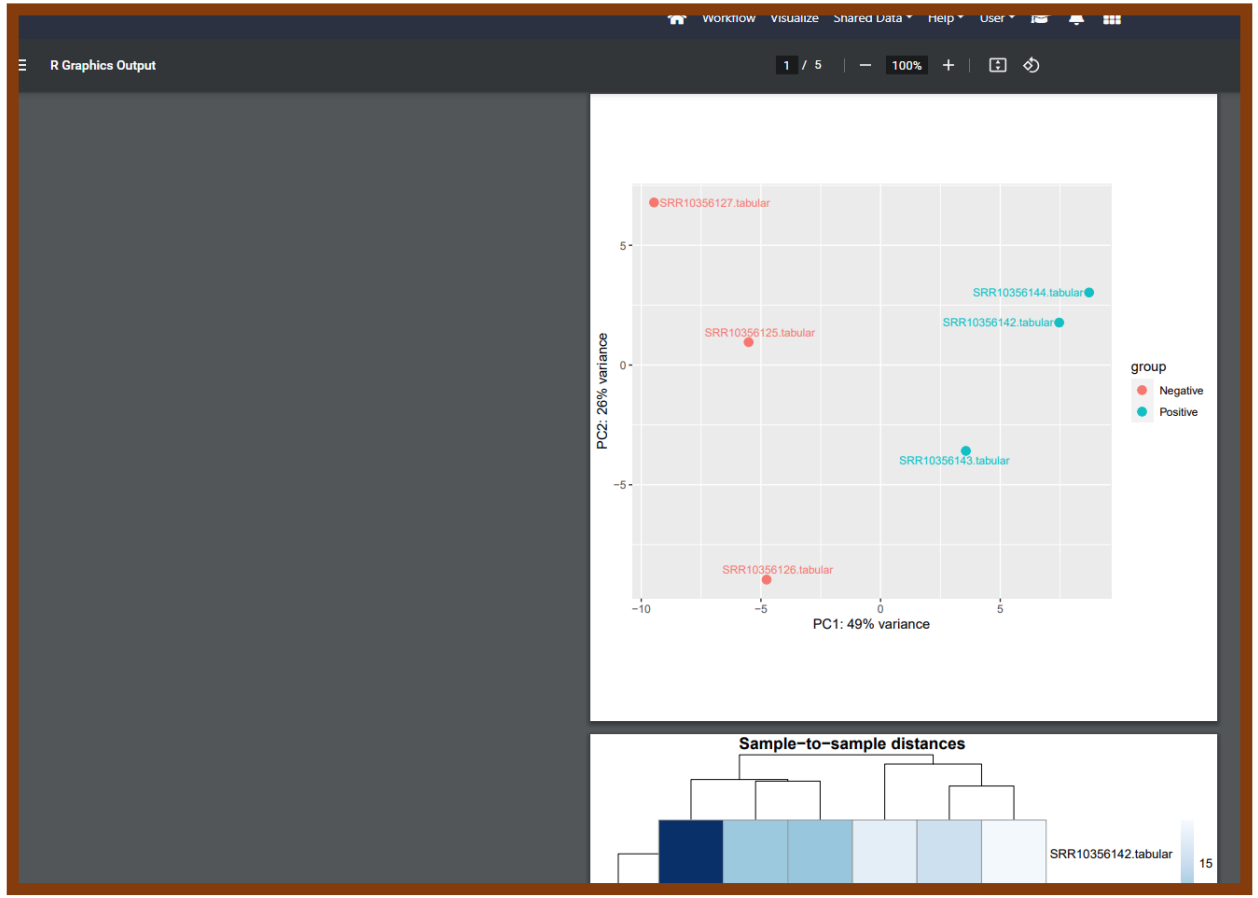


GeneID	Base mean	log2(FC)	StdErr	Wald-Stats	P-value	P-adj
hsa-miR-144-5p	256.334411435762	4.03572974461951	0.509595099187083	7.91948303870543	2.38500096580741e-15	5.60475226964741e-13
hsa-miR-375-3p	1530.82399828163	4.02965836923229	0.606270106563184	6.64663872687961	2.99861824441712e-11	2.94775520856156e-09
hsa-miR-185-5p	200.697555088063	4.50851675336146	0.681753599736636	6.61311763532032	3.76309175561051e-11	2.94775520856156e-09
hsa-miR-486-5p	767656.041359639	3.29959068102232	0.540860163445729	6.10063543966926	1.05647618400484e-09	6.20679758102844e-08
hsa-miR-107	4529.11854670314	2.7340159694676	0.478737653288701	5.71088559816885	1.12389780540217e-08	5.28231968539021e-07
hsa-miR-103a-3p	9291.40871253028	2.64615614586663	0.470716535901716	5.62154915759989	1.89252629252658e-08	7.41239464572909e-07
hsa-miR-125b-5p	42.6781725115144	-3.81025135724564	0.705115328070426	-5.4037278804767	6.52699409200655e-08	2.19120515945934e-06
hsa-miR-125a-5p	225.362195319651	-2.78984505055407	0.547866040035147	-5.09220292313627	3.5392700475753e-07	1.03966057647525e-05
hsa-miR-185-3p	22.1888417148245	3.58467677028873	0.7252146880113	4.94291804833506	7.69618412421234e-07	2.00955918798878e-05
hsa-miR-223-3p	1246.71302037906	-2.06816457103587	0.453311111717051	-4.56235136880294	5.05839009801464e-06	0.000118872167303344
hsa-miR-199b-5p	112.822032004768	2.08988529051403	0.462872154480673	4.5150378355743	6.33054624514435e-06	0.000135243487964448
hsa-miR-451a	10083.3521501164	1.81253011494869	0.422331834059908	4.29172032220423	1.77294168302527e-05	0.000320595551386441
hsa-miR-25-3p	55069.7606639701	2.26925756216252	0.528761157782266	4.29164950708605	1.77350730554201e-05	0.000320595551386441

Examine results file

Step 8: Differential expression analysis with DESeq2 in Galaxy





Examine plots

Part II: Downstream analysis in R (more advanced)

1. Raw read count processing
2. Normalization and differential expression analysis with DESeq2
3. Common data visualizations

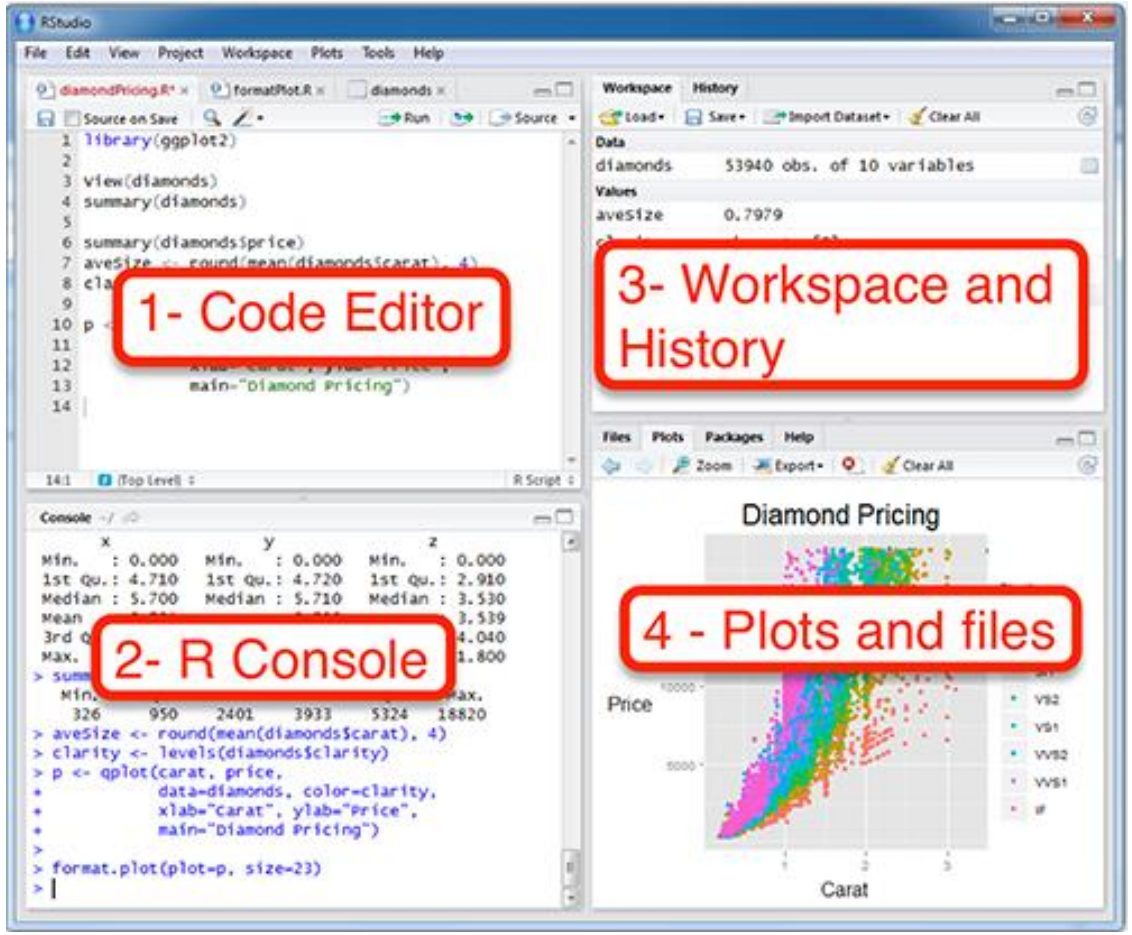
R coding environment & R Studio

What is R?
"R is a programming language for statistical computing and graphics"

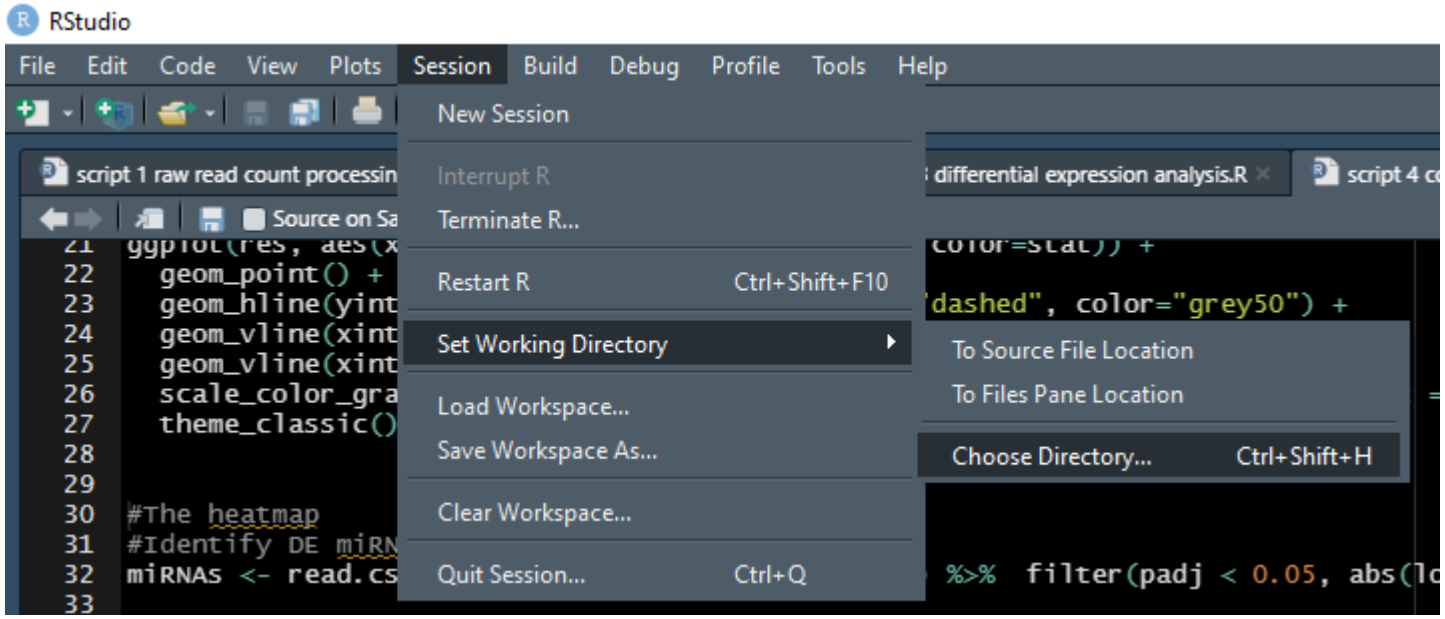
Download **R** from https://www.r-project.org/

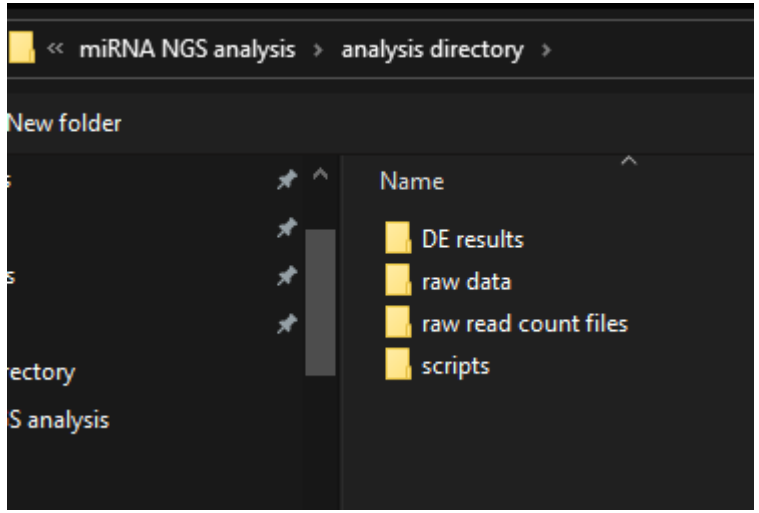
Download **R studio** from https://www.rstudio.com/products/rstudio/download/

New to R?
https://support.rstudio.com/hc/en-us/articles/201141096-Getting-Started-with-R



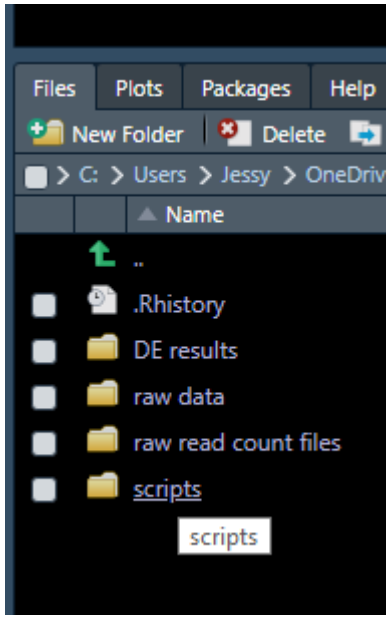
Set working directory in R Studio

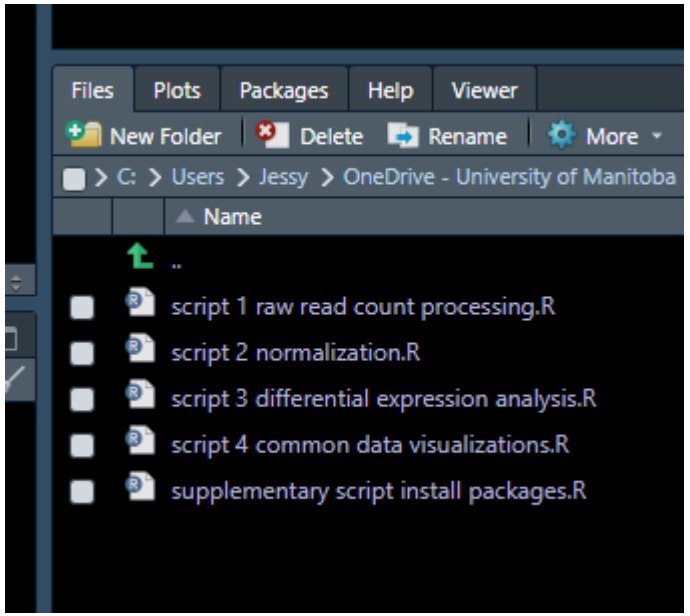




The **working directory** lets R know where your files are saved, you simple specify the folder that will be used for the analysis. In this case it is the `**analysis directory**` folder provided in the tutorial.

Open R script files into R Studio





Files can be examined and opened in the “Plots and files” pane within R Studio. Open the provided **R scripts** by simply clicking on the file.

Download required R packages

An **R package** is a reproducible unit of R code that usually contains a set of pre-defined functions used to complete a particular type of analysis.

supplementary script install packages.R | script 1 raw read c

Source on Save

```
#Install packages through CRAN
install.packages("ggplot2")
install.packages("RColorBrewer")
install.packages("dplyr")
install.packages("pheatmap")

#Install DESeq2 through BiocManager
install.packages("BiocManager")
BiocManager::install("DESeq2")
```

Scripts are opened in R studio. Individual lines of code can be run by pressing **`ctrl + enter`**

Script 1: Pre-processing raw read count files

script 1 raw read count processing.R | script 2 normalization.R | script 3 differential expression analysis.R | script 4 common data visualizations.R

Source on Save

```r
#miRNA NGS data analysis
#
#Script 1
#
#Raw read count processing

###Collect all raw read count files and merge into one matrix
data_files <- Sys.glob("raw read count files/*.tabular") #store paths for all raw read count files
tmp <- list() #create an empty list to store each file
for (i in data_files) {
  X <- gsub(".tabular.*", "", gsub(".*raw read count files/", "", i)) #extract sample name from file name and store in "X"
  tmp[[X]] <- read.delim(i, row.names = 1, header = FALSE) #load read count file "i"
  colnames(tmp[[X]]) <- X #rename column with sample name
  print(X) #print sample name to track progress in console
}
read_counts <- do.call(cbind, tmp) #do.call function collapses all objects within list into one data frame

#Clean up read count file
read_counts <- read_counts[rowMeans(read_counts)>0,] # remove all transcripts that were not detected
read_counts <- read_counts[order(rowMeans(read_counts), decreasing = TRUE),] # order transcripts based on average raw read count

#Manually make a data frame containing sample information
sample_info <- data.frame(Sample = colnames(read_counts),
                          CWD_status = c(rep("Neg", 3), rep("Pos", 3)))

#Save files for further analysis
if (dir.exists("raw data") == FALSE) { dir.create("raw data") }
write.csv(read_counts, "raw data/raw_read_counts.csv")
write.csv(sample_info, "raw data/sample_info.csv")
```

Run each line of code by pressing `**ctrl + enter**`. Pay attention to what is happening in the console and environment. Files that are loaded in the `environment` of R studio can be examined by clicking on them.

Script 1: Pre-processing raw read count files

Input: (x6)

	V2
hsa-miR-6859-5p	0
hsa-miR-6859-3p	0
hsa-miR-1302	0
hsa-miR-12136	0
hsa-miR-200b-5p	0
hsa-miR-200b-3p	0
hsa-miR-200a-5p	0
hsa-miR-200a-3p	1
hsa-miR-429	0
hsa-miR-6726-5p	0
hsa-miR-6726-3p	0
hsa-miR-6727-5p	0
hsa-miR-6727-3p	0
hsa-miR-6808-5p	0

Output:

	SRR10356125	SRR10356126	SRR10356127	SRR10356142	SRR10356143	SRR10356144
hsa-miR-486-5p	398669	70960	316241	1175615	405887	748382
hsa-miR-191-5p	449776	124957	248597	65468	72379	45908
hsa-miR-423-5p	316543	33840	226684	169837	66163	127436
hsa-miR-142-5p	139973	45982	109468	28867	44975	24944
hsa-miR-22-3p	122713	38722	105241	13749	44072	14978
hsa-miR-25-3p	56498	9574	44564	88626	33123	36732
hsa-miR-16-5p	76560	8317	60443	23148	18099	27028
hsa-miR-92a-3p	61304	14322	41601	32927	18763	12571
hsa-miR-148a-3p	41482	12246	51872	4224	5529	1414
hsa-miR-143-3p	14738	81	96933	92	228	120
hsa-miR-27b-3p	31915	8268	43163	3723	9890	3260
hsa-miR-6529-5p	31697	3540	25879	15659	6682	7850
hsa-miR-192-5p	19760	5274	17513	12242	10896	5767

This script takes the 6 separate read-count files from feature counts and merges them together into a single matrix, which is then saved as a `.csv` file and can be examined in excel.

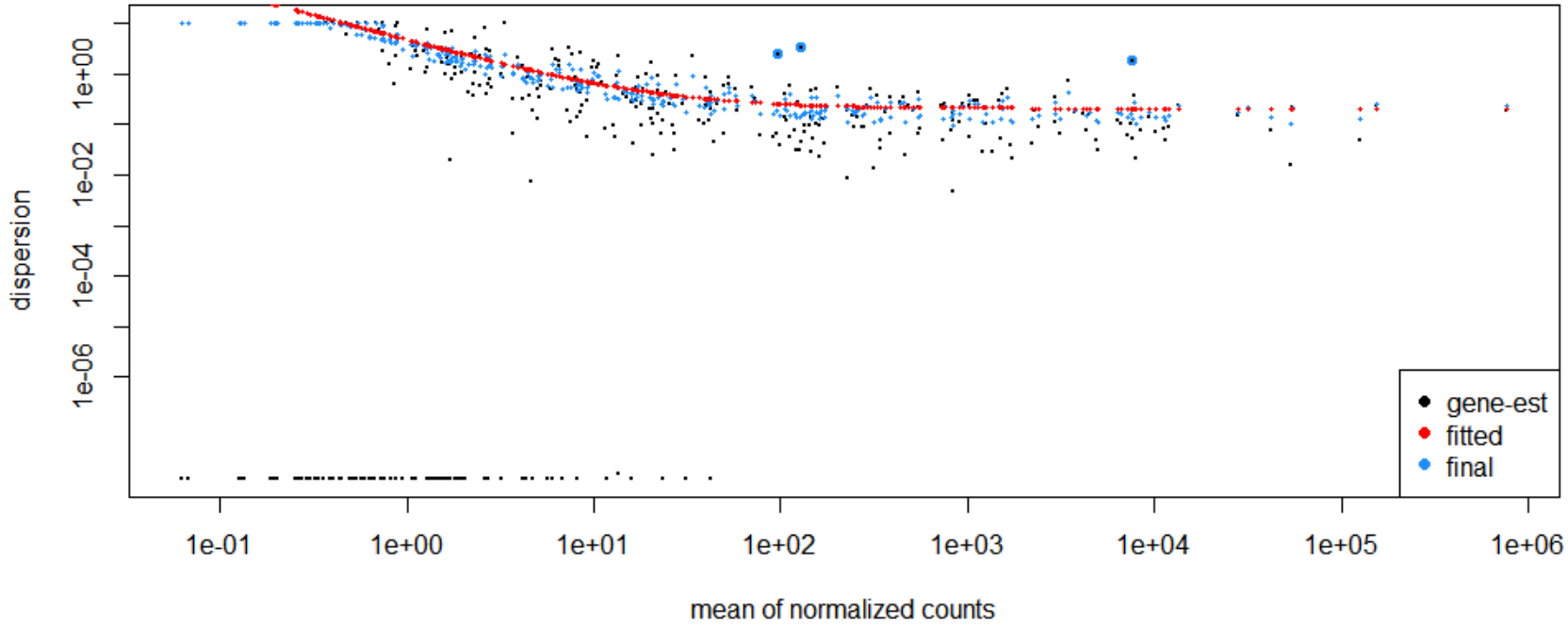
Script 2: Normalization with DESeq2

```
#miRNA NGS data analysis
#
#Script 2
#
#Normalizing the data and assessing variation


library(DESeq2)
library(ggplot2)

#load raw data
read_counts <- read.csv("raw data/raw_read_counts.csv", row.names = 1)
sample_info <- read.csv("raw data/sample_info.csv", row.names = 1)
rownames(sample_info) <- sample_info$Sample

#make sure samples are in order
summary(col DESeqDataSetFromMatrix(countData, colData, design, tidy = FALSE, ignoreRank =
            FALSE, ...)

#make DEseq data object
dds <- DESeqDataSetFromMatrix(countData = read_counts, colData = sample_info, design = ~CWD_status)
dds <- DESeq(dds)

#plot dispersion estimates to examine normalization
plotDispEsts(dds)
```

Run each line of code by pressing `**ctrl + enter**`. Pay attention to what is happening in the console and environment. Files that are loaded in the `environment` of R studio can be examined by clicking on them.

The **DESeq2 R** package is popular for differential expression analysis of RNAseq data. A full description of how to use DESeq2 can be found at
http://bioconductor.org/packages/devel/bioc/vignettes/DESeq2/inst/doc/DESeq2.html

Script 2: Normalization with DESeq2



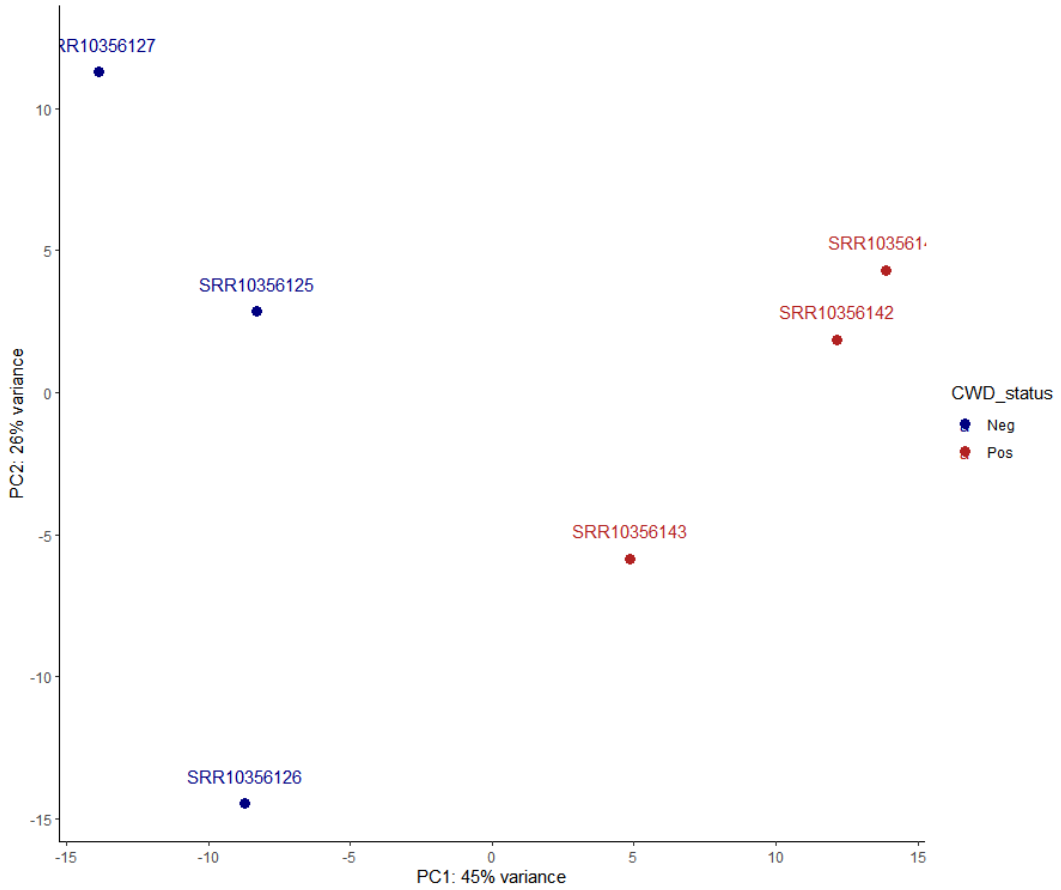
This plot illustrates the “negative binomial model of gene-fitted dispersion estimates” used by DESeq2 to normalize the data. This is a global normalization method that is used by other RNAseq analysis R packages such as edgeR.

Common data visualizations: PCA (Script 2)

```
#extract normalized read counts
norm_counts <- varianceStabilizingTransformation(dds)

#make a basic PCA plot
plotPCA(norm_counts, intgroup="CWD_status")

#make a custom PCA plot with ggplot
pcaData <- plotPCA(norm_counts, intgroup=c("Sample", "CWD_status"), returnData=TRUE)
percentVar <- round(100 * attr(pcaData, "percentVar"))
ggplot(pcaData, aes(PC1, PC2, color=CWD_status, label=Sample)) +
  geom_point(size=3) +
  geom_text(nudge_y = 1) +
  xlab(paste0("PC1: ",percentVar[1],"% variance")) +
  ylab(paste0("PC2: ",percentVar[2],"% variance")) +
  scale_color_manual(values = c("navy", "firebrick")) +
  coord_fixed() +
  theme_classic()

#Save normalized read counts for visualization later
write.csv(assay(norm_counts), "raw data/normalized_read_counts.csv")
```

Run each line of code by pressing `**ctrl + enter**`. Pay attention to what is happening in the console and environment. Files that are loaded in the `environment` of R studio can be examined by clicking on them.

Common data visualizations: PCA (Script 2)



Principle component analysis, or PCA, is a dimensionality reduction technique used to visualize variation in the data in 2-dimensions. It is often used to examine sources of variation in gene expression data.

Script 3: Differential expression analysis with DESeq2

```
library(DESeq2)

#load raw data
read_counts <- read.csv("raw data/raw_read_counts.csv", row.names = 1)
sample_info <- read.csv("raw data/sample_info.csv", row.names = 1)
rownames(sample_info) <- sample_info$Sample

#make DESeq data object
dds <- DESeqDataSetFromMatrix(countData = read_counts, colData = sample_info, design = ~CWD_status)
dds <- DESeq(dds)

#get differential expression results
resultsNames(dds)
res <- results(object = dds, contrast = c("CWD_status", "Pos", "Neg"))

#clean up results file
res <- res[order(res$padj),]
res <- na.omit(res)
res <-as.data.frame(res)

summary(res$padj < 0.05)

#Save differential expression results
if (dir.exists("DE results")==FALSE) { dir.create("DE results") }
write.csv(res, "DE results/CWD_DE_miRNAs.csv")
```

Run each line of code by pressing `**ctrl + enter**`. Pay attention to what is happening in the console and environment. Files that are loaded in the `environment` of R studio can be examined by clicking on them.

Script 3: Differential expression analysis with DESeq2

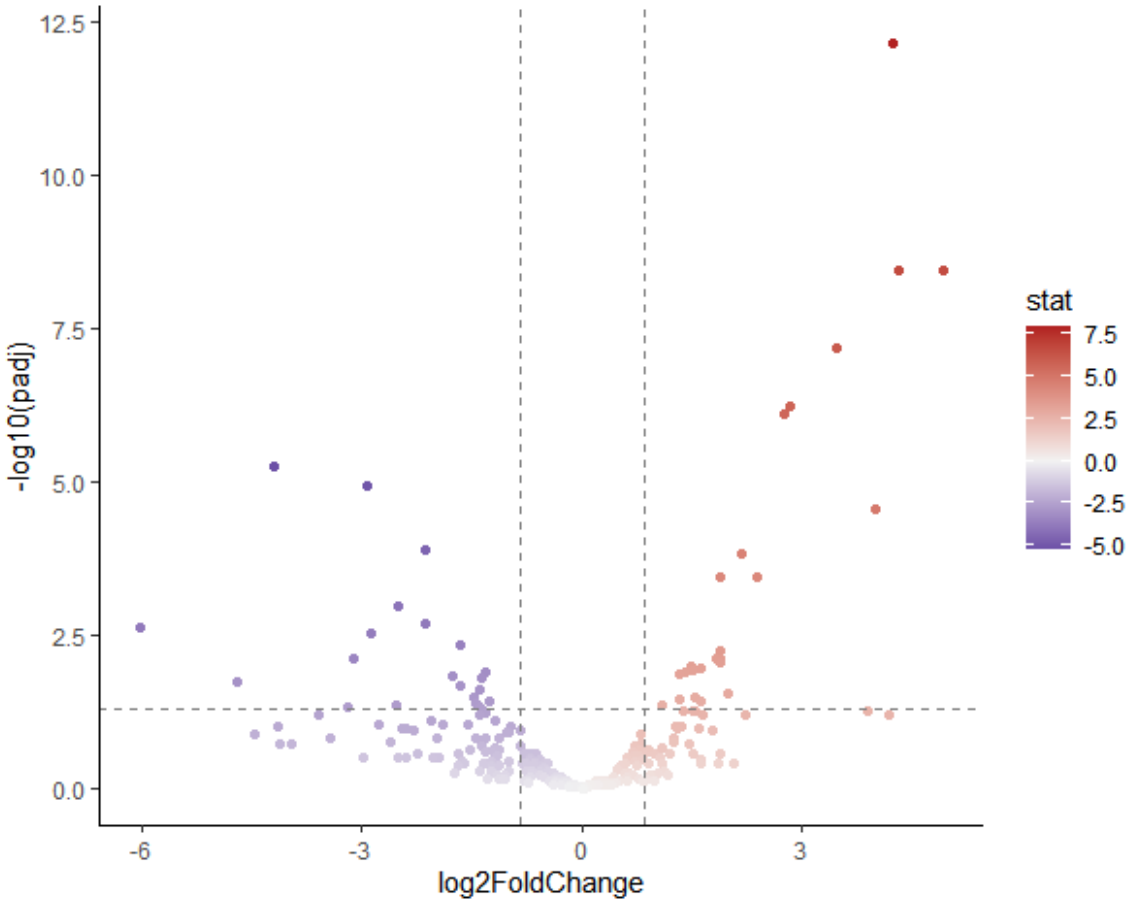
Results file:

	baseMean	log2FoldChange	lfcSE	stat	pvalue	padj
hsa-miR-144-5p	2.563344e+02	4.2183261	0.5337497	7.9031908	2.718531e-15	6.905068e-13
hsa-miR-375-3p	1.530824e+03	4.3010739	0.6471957	6.6457082	3.017625e-11	3.608444e-09
hsa-miR-185-5p	2.006976e+02	4.9145387	0.7452286	6.5946730	4.261942e-11	3.608444e-09
hsa-miR-486-5p	7.676560e+05	3.4711205	0.5689435	6.1009932	1.054114e-09	6.693623e-08
hsa-miR-107	4.529119e+03	2.8427122	0.4977763	5.7108228	1.124312e-08	5.711507e-07
hsa-miR-103a-3p	9.291409e+03	2.7475877	0.4887584	5.6215661	1.892341e-08	8.010911e-07
hsa-miR-125b-5p	4.267817e+01	-4.1982603	0.8006333	-5.2436741	1.574101e-07	5.711738e-06
hsa-miR-125a-5p	2.253622e+02	-2.9383044	0.5781449	-5.0822977	3.728963e-07	1.183946e-05
hsa-miR-185-3p	2.218884e+01	3.9829770	0.8127359	4.9007024	9.549461e-07	2.695070e-05
hsa-miR-223-3p	1.246713e+03	-2.1411961	0.4693917	-4.5616407	5.075544e-06	1.289188e-04
hsa-miR-199b-5p	1.128220e+02	2.1673227	0.4801270	4.5140616	6.359774e-06	1.468530e-04
hsa-miR-25-3p	5.506976e+04	2.3813877	0.5548757	4.2917498	1.772706e-05	3.463799e-04
hsa-miR-451a	1.008335e+04	1.8675975	0.4351612	4.2917368	1.772810e-05	3.463799e-04
hsa-miR-222-3p	4.586697e+02	-2.5124478	0.6242641	-4.0246550	5.705890e-05	1.035212e-03
hsa-miR-99b-5p	2.297306e+02	-2.1312760	0.5546841	-3.8423238	1.218749e-04	2.063748e-03
hsa-miR-143-3p	7.673291e+03	-6.0147614	1.5887402	-3.7858684	1.531728e-04	2.288581e-03

Common data visualizations: Volcano plot (Script 4)

```r

library(ggplot2)
library(RColorBrewer)
library(pheatmap)
library(dplyr)

#The volcano plot
#Load differential expression results from 150 dpi
res <- read.csv("DE results/CWD_DE_miRNAs.csv")

#a basic volcano plot
ggplot(res, aes(x=log2FoldChange, y=-log10(padj))) +
  geom_point()

#a nicer volcano plot
ggplot(res, aes(x=log2FoldChange, y=-log10(padj), color=stat)) +
  geom_point() +
  geom_hline(yintercept = -log10(0.05), linetype="dashed", color="grey50") +
  geom_vline(xintercept = 0.85, linetype="dashed", color="grey50") +
  geom_vline(xintercept = -0.85, linetype="dashed", color="grey50") +
  scale_color_gradient2(low = "navy", high = "firebrick", mid="grey95", midpoint = 0) +
  theme_classic()

```

Common data visualizations: Volcano plot (Script 4)

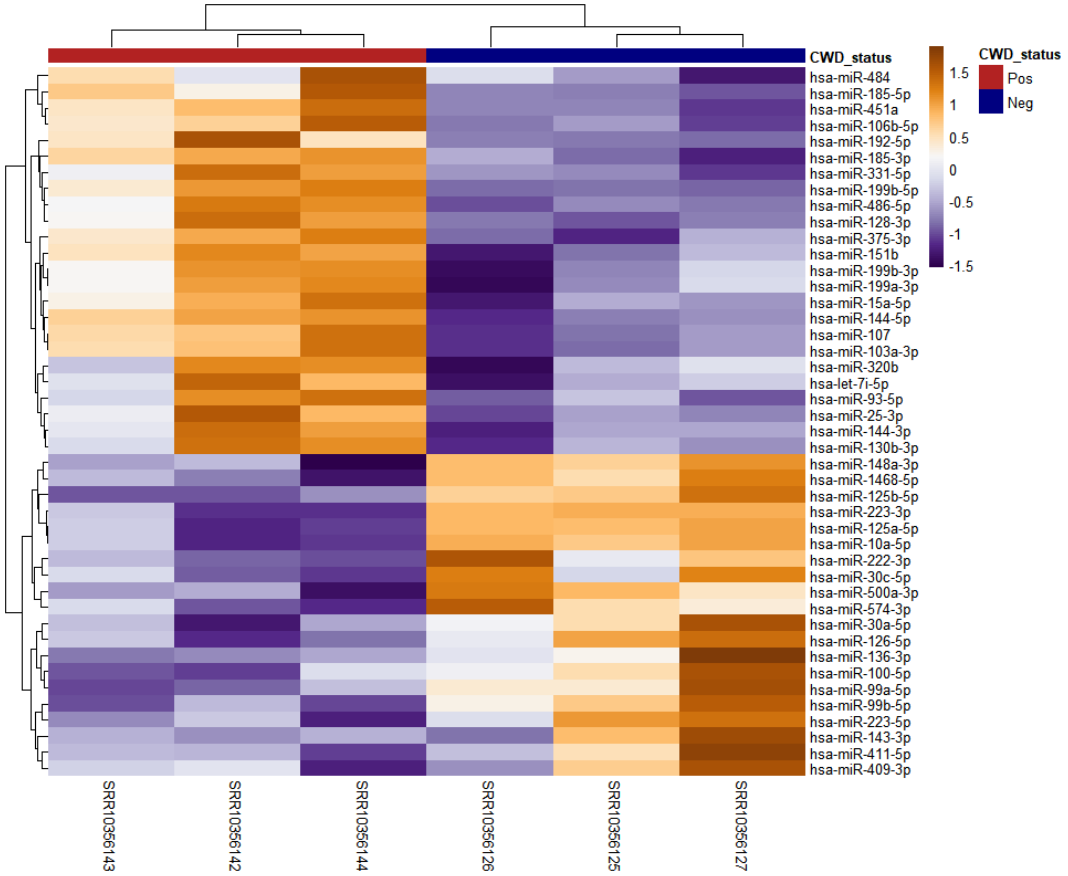


A volcano plot show statistical significance (p-value) versus magnitude (fold change) and can quickly show changes in large datasets

Common data visualizations: Heatmap (Script 4)

```r
#The heatmap
#Identify DE miRNAs
miRNAs <- read.csv("DE results/CWD_DE_miRNAs.csv") %>%  filter(padj < 0.05, abs(log2FoldChange) > 0.85, baseMean > 15) %>% pull(X)

#We will use normalized read-counts to calculate z-scores
zscores <- read.csv("raw data/normalized_read_counts.csv", row.names = 1)
zscores <- as.matrix(zscores[miRNAs,])
zscores <- (zscores-rowMeans(zscores))/matrixStats::rowSds(zscores)

#basic heatmap with hierachichal clustering
pheatmap(zscores)

#nicer heatmap

#specify additional variables required by pheatmap
plot_colors <- rev(colorRampPalette(brewer.pal(11,"PuOr"))(100))
column_annotation <- read.csv("raw data/sample_info.csv", row.names = 1)
column_annotation <- data.frame(row.names = column_annotation$Sample,
                                CWD_status = column_annotation$CWD_status)
annotation_colors <- list(`CWD_status`=c(`Pos`="firebrick", `Neg`="navy"))

pheatmap(zscores, color = plot_colors, annotation_col = column_annotation, annotation_colors = annotation_colors,
         treeheight_row = 25, treeheight_col = 25, border_color = FALSE)
```

Common data visualizations: Heatmap (Script 4)



Heatmaps are commonly used to examine gene expression across samples. Hierarchical clustering of genes and samples can make it easier to identify patterns in the gene expression data.

Further analysis

- Which genes are targeted by these miRNAs?
- miRNA target prediction online tools:
  - http://carolina.imis.athena-innovation.gr/diana_tools/web/index.php?r=tarbasev8%2Findex
  - http://snf-515788.vm.okeanos.grnet.gr/
  - http://mirdb.org/
  - http://www.targetscan.org/vert_80/

Summary

- **Part I:** Preprocessing in Galaxy
  - Assessing sequencing read quality with FastQC
  - Removing sequencing adapters with Cutadapt
  - Cleaning sequencing reads with Trimmomatic
  - Aligning to reference genome with Bowtie2
  - Mapping reads to miRNAs and counting with FeatureCounts
- **Part II:** Downstream Analysis with R
  - Raw reads count processing
  - Normalization and differential expression analysis with DESeq2
  - Examples of common data visualizations