Galaxy Tutorial – Basics of Data Analysis using Galaxy platform

Part of

(SW4) The Galaxy Platform for Multi-Omic Data Analysis and Informatics

Dave Clements
Galaxy Project, Johns Hopkins University

8:30-10:00am
Saturday February 20, 2016

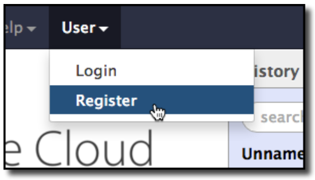
ABRF 2016

A short version of this material, used as a workshop handout is available at http://bit.ly.gxyabrf16intro
This, longer version of the same material is available on-line at http://bit.ly/gxyabrf16intro-long

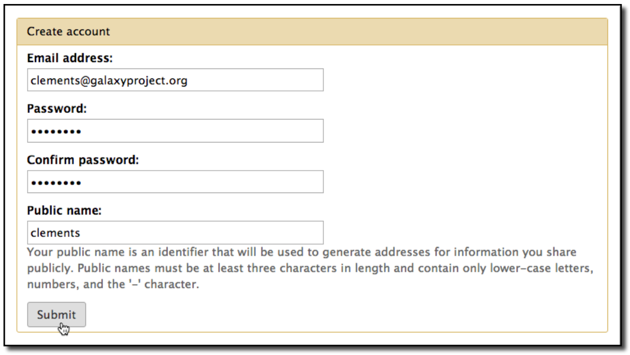
Create an account

You don't have to have an account to use most Galaxy instances, but having an account allows you to use the full set of Galaxy features, including:

- Save and work with multiple histories.
- Save and run *workflows*, rerunnable pipelines for specific analyses.
- Save and work with custom genomes
- Share and publish your analyses with others.



You'll want to have an account today. To create an account, *click* on the User pulldown and *select* **Register**.


Create account

Email address:
clements@galaxyproject.org

Password:
••••••••

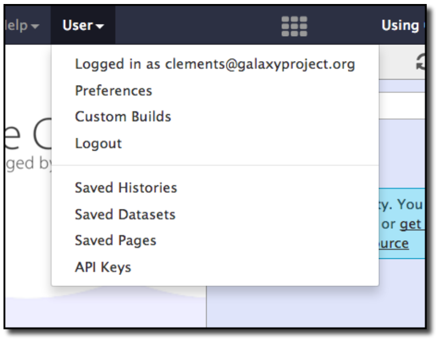
Confirm password:
••••••••

Public name:
clements

Your public name is an identifier that will be used to generate addresses for information you share publicly. Public names must be at least three characters in length and contain only lower-case letters, numbers, and the '-' character.

Submit

Use an email address you can remember, **and a low security password**. Note that these servers use HTTP, not HTPPS, and therefore your password is going out on the net unencrypted. (Galaxy can use HTTPS and most servers do, but the default cloud installation uses HTTP.)



Reload the page by *clicking* **Analyze Data** or **Return to the home page**. The User pulldown now has several options that are enabled only for logged in users. These include the ability to define your own reference genomes, have more than one saved history, and to access Galaxy via scripts/programs.

Will this create an account on other Galaxy servers?

No. At this time there is no sharing of user accounts between different Galaxy instances. You need to create an account on each server you use.

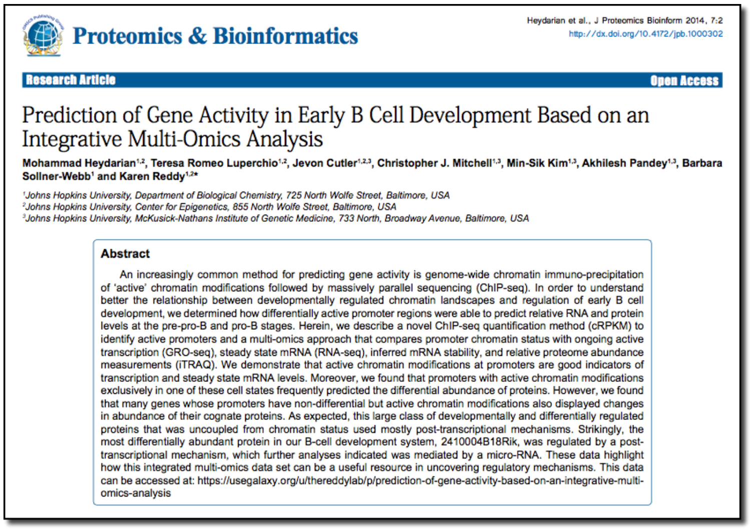
Our goal: Identify transcripts using RNA-Seq in pre-pro-B and pro-B mouse cells

We are going to use RNA-seq data to detect transcripts, possibly novel transcripts, in pre-pro-B and pro-B mouse cells.

This data comes from

> Heydarian *et al.*, (2014) Prediction of Gene Activity in Early B Cell Development Based on an Integrative Multi-Omics Analysis. *J Proteomics Bioinform* 7: 050-063. doi:10.4172/jpb.1000302

The paper uses RNA-Seq, ChIP-Seq, GRO-Seq, and iTRAQ techniques to "demonstrate that active chromatin modifications at promoters are good indicators of transcription and steady state mRNA levels."



Proteomics & Bioinformatics

Heydarian et al., J Proteomics Bioinform 2014, 7:2
http://dx.doi.org/10.4172/jpb.1000302

Research Article — Open Access

Prediction of Gene Activity in Early B Cell Development Based on an Integrative Multi-Omics Analysis

Mohammad Heydarian[1,2], Teresa Romeo Luperchio[1,2], Jevon Cutler[1,2,3], Christopher J. Mitchell[1,3], Min-Sik Kim[1,3], Akhilesh Pandey[1,3], Barbara Sollner-Webb[1] and Karen Reddy[1,2]*

[1]Johns Hopkins University, Department of Biological Chemistry, 725 North Wolfe Street, Baltimore, USA
[2]Johns Hopkins University, Center for Epigenetics, 855 North Wolfe Street, Baltimore, USA
[3]Johns Hopkins University, McKusick-Nathans Institute of Genetic Medicine, 733 North, Broadway Avenue, Baltimore, USA

Abstract

An increasingly common method for predicting gene activity is genome-wide chromatin immuno-precipitation of 'active' chromatin modifications followed by massively parallel sequencing (ChIP-seq). In order to understand better the relationship between developmentally regulated chromatin landscapes and regulation of early B cell development, we determined how differentially active promoter regions were able to predict relative RNA and protein levels at the pre-pro-B and pro-B stages. Herein, we describe a novel ChIP-seq quantification method (cRPKM) to identify active promoters and a multi-omics approach that compares promoter chromatin status with ongoing active transcription (GRO-seq), steady state mRNA (RNA-seq), inferred mRNA stability, and relative proteome abundance measurements (iTRAQ). We demonstrate that active chromatin modifications at promoters are good indicators of transcription and steady state mRNA levels. Moreover, we found that promoters with active chromatin modifications exclusively in one of these cell states frequently predicted the differential abundance of proteins. However, we found that many genes whose promoters have non-differential but active chromatin modifications also displayed changes in abundance of their cognate proteins. As expected, this large class of developmentally and differentially regulated proteins that was uncoupled from chromatin status used mostly post-transcriptional mechanisms. Strikingly, the most differentially abundant protein in our B-cell development system, 2410004B18Rik, was regulated by a post-transcriptional mechanism, which further analyses indicated was mediated by a micro-RNA. These data highlight how this integrated multi-omics data set can be a useful resource in uncovering regulatory mechanisms. This data can be accessed at: https://usegalaxy.org/u/thereddylab/p/prediction-of-gene-activity-based-on-an-integrative-multi-omics-analysis

We will use this experiment in several contexts today. We'll start with the RNA-Seq data.

The data

We have two RNA-Seq datasets, one from mouse pre-pro-B cells, and one from mouse pro-B cells. Each dataset consists of ~90 million single-end reads. Sequencing was done an Illumina HiSeq 2000 and all reads are 97bp long.

180 million reads might be just what is needed for this experiment, but it is way too many reads for a hands-on workshop. Therefore, we are going to use striiped down datasets consisting of reads that tend to map to the particular regions of the genome we'll focus on today, plus some additional reads, just to make things interesting.

> *Note that while reducing the datasets will make tool execution very fast, it is a dangerous cheat. Many NGS tools are built on statistical models that assume they are being given data for the entire genome. With our data that is decidedly not the case. Downsampling your data is still a useful technique for early experimentation when building your analysis, but results and assumptions should be checked when scaling up your analysis to complete datasets.*
>
> *Also, in this session we are using single-end reads for transcript assembly in eukaryotes. We are doing this today to allow us to use the same example across the different sessions. However, in general you are much better off using paired-end data for transcript assembly in eukaryotes.*

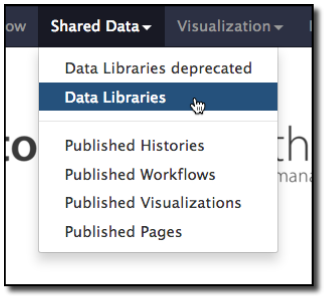
Get the data / Data libraries

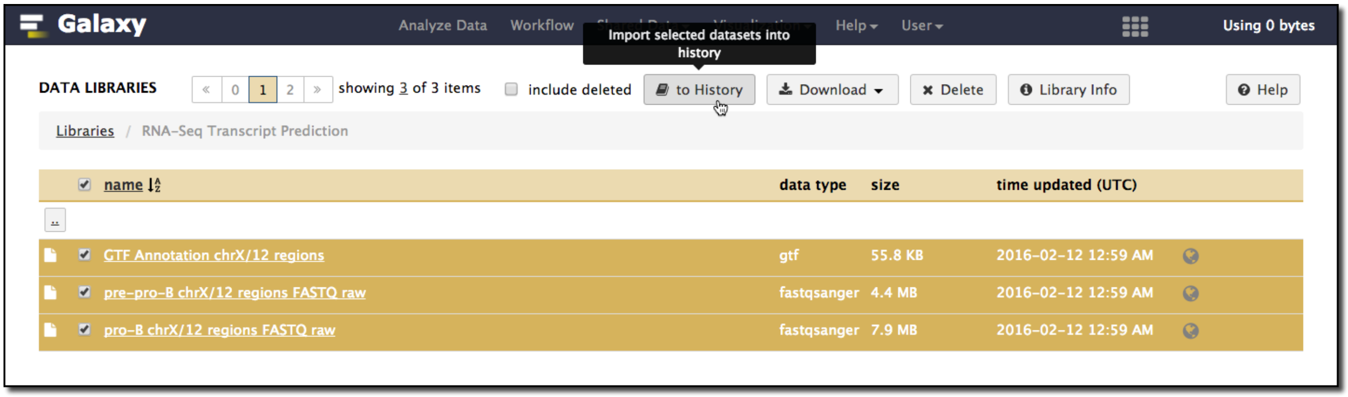
Our reduced datasets are available in a *data library* on your server. Data libraries are maintained by the Galaxy server's administrators and contain datasets that are useful to users of that server. For example, the MiSSiSSiPPi server is focuses on RNA-seq analysis, especially small RNA, and includes several viral datasets in it's data libraries (see https://mississippi.snv.jussieu.fr/library/list).

Let's import our two RNA-Seq datasets and the genome annotation for the relevant parts of the genome. *Click* on the **Shared Data** pull-down, and then *select* **Data Libraries**.

This shows the list of Data Libraries defined on this server.

Click on the **RNA-Seq Transcript Prediction** library and select all 3 datasets in this library and then *click* on **to History** to import them into your current history.

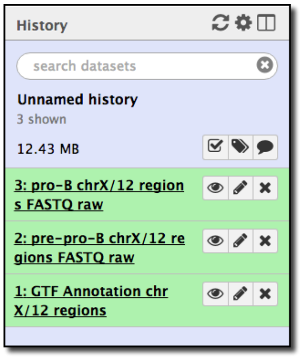




	name	data type	size	time updated (UTC)
☑	GTF Annotation chrX/12 regions	gtf	55.8 KB	2016-02-12 12:59 AM
☑	pre-pro-B chrX/12 regions FASTQ raw	fastqsanger	4.4 MB	2016-02-12 12:59 AM
☑	pro-B chrX/12 regions FASTQ raw	fastqsanger	7.9 MB	2016-02-12 12:59 AM

This will ask which history to import these datasets into. So far, you only have one ("Unnamed history"). *Click* **Import** to add them to that history.

Now *click* **Analyze Data** to see what we have.



Unnamed history
3 shown
12.43 MB

3: pro-B chrX/12 regions FASTQ raw

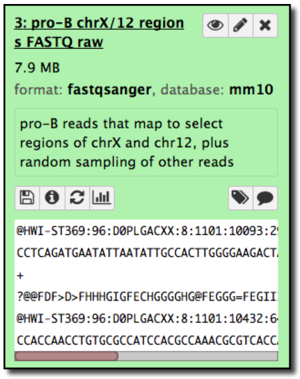
2: pre-pro-B chrX/12 regions FASTQ raw

1: GTF Annotation chrX/12 regions

FASTQ datasets

And what we have is a history consisting of three datasets. Two of them are the raw read datasets for the pre-pro-B and pro-B cells. These datasets are in FASTQ format.

To see a preview of any dataset, click on the dataset name. This displays some metadata about the dataset, and if it's in a text-based format, it will also show the first few lines of the dataset.



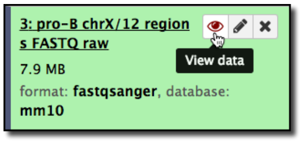
3: pro-B chrX/12 regions FASTQ raw

7.9 MB

format: **fastqsanger**, database: **mm10**

pro-B reads that map to select regions of chrX and chr12, plus random sampling of other reads

```
@HWI-ST369:96:D0PLGACXX:8:1101:10093:2
CCTCAGATGAATATTAATATTGCCACTTGGGGAAGACT
+
?@@FDF>D>FHHHGIGFECHGGGHG@FEGGG=FEGII
@HWI-ST369:96:D0PLGACXX:8:1101:10432:6
CCACCAACCTGTGCGCCATCCACGCCAAACGCGTCACC
```



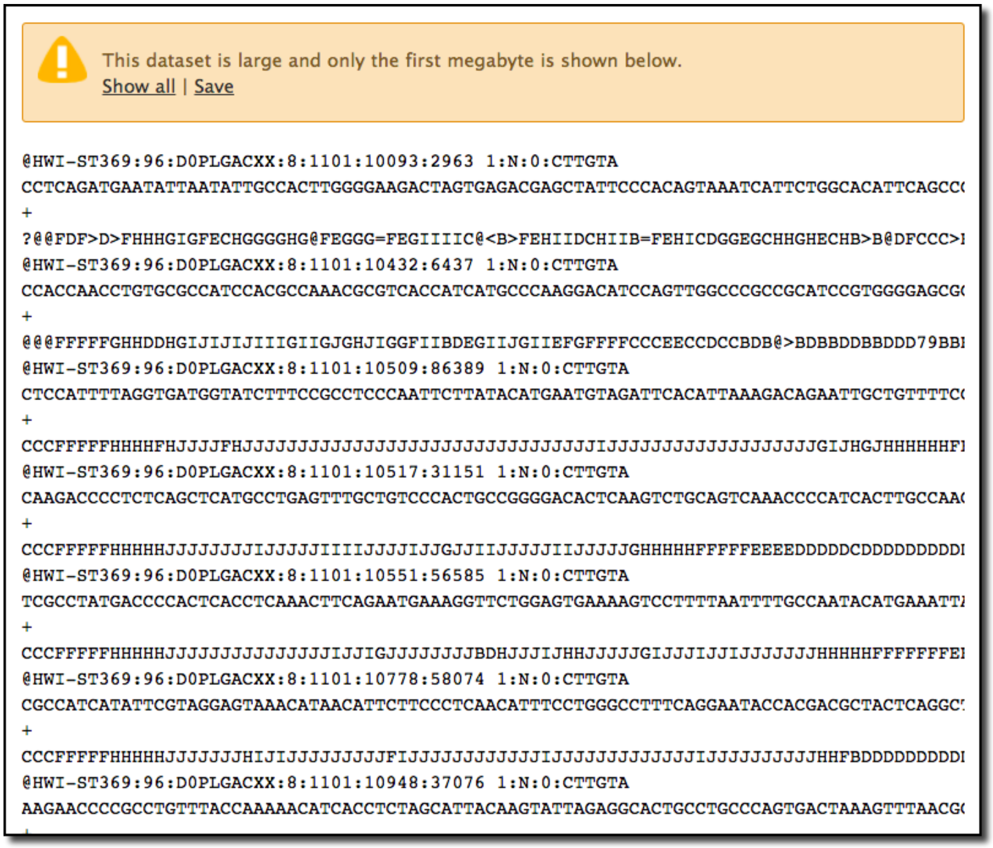
3: pro-B chrX/12 regions FASTQ raw

7.9 MB

format: **fastqsanger**, database: **mm10**

To view the full dataset, *poke it in the eye*.

This displays the content of this FASTQ dataset in the middle panel.



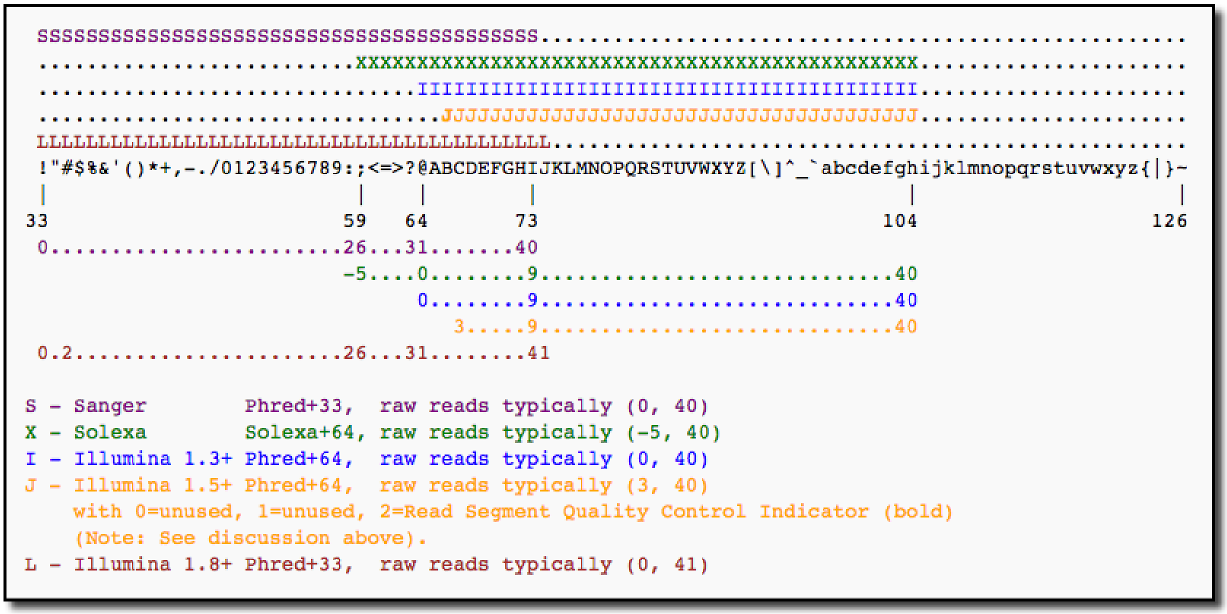
This dataset is large and only the first megabyte is shown below.
Show all | Save

```
@HWI-ST369:96:D0PLGACXX:8:1101:10093:2963 1:N:0:CTTGTA
CCTCAGATGAATATTAATATTGCCACTTGGGGAAGACTAGTGAGACGAGCTATTCCCACAGTAAATCATTCTGGCACATTCAGCC
+
?@@FDF>D>FHHHGIGFECHGGGGHG@FEGGG=FEGIIIIC@<B>FEHIIDCHIIB=FEHICDGGEGCHHGHECHB>B@DFCCC>
@HWI-ST369:96:D0PLGACXX:8:1101:10432:6437 1:N:0:CTTGTA
CCACCAACCTGTGCGCCATCCACGCCAAACGCGTCACCATCATGCCCAAGGACATCCAGTTGGCCCGCCGCATCCGTGGGGAGCG
+
@@@FFFFFGHHDDHGIJIJIJIIIGIIGJGHJIGGFIIBDEGIIJGIIEFGFFFCCCEECCDCCBDB@>BDBBDDBBDDD79BB
@HWI-ST369:96:D0PLGACXX:8:1101:10509:86389 1:N:0:CTTGTA
CTCCATTTTAGGTGATGGTATCTTTCCGCCTCCCAATTCTTATACATGAATGTAGATTCACATTAAAGACAGAATTGCTGTTTTC
+
CCCFFFFFHHHHFHJJJJFHJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJIJJJJJJJJJJJJJJJJJJGIJHGJHHHHHHF
@HWI-ST369:96:D0PLGACXX:8:1101:10517:31151 1:N:0:CTTGTA
CAAGACCCCTCTCAGCTCATGCCTGAGTTTGCTGTCCCACTGCCGGGGACACTCAAGTCTGCAGTCAAACCCCATCACTTGCCAA
+
CCCFFFFFHHHHHJJJJJJJIJJJJJIIIIJJJJIJJGJJIIJJJJJIIJJJJJGHHHHHFFFFFEEEEDDDDDCDDDDDDDDDD
@HWI-ST369:96:D0PLGACXX:8:1101:10551:56585 1:N:0:CTTGTA
TCGCCTATGACCCCACTCACCTCAAACTTCAGAATGAAAGGTTCTGGAGTGAAAAGTCCTTTTAATTTTGCCAATACATGAAATT
+
CCCFFFFFHHHHHJJJJJJJJJJJJJJIJJIGJJJJJJJBDHJJJIJHHJJJJJGIJJJIJJIJJJJJJHHHHHFFFFFFFE
@HWI-ST369:96:D0PLGACXX:8:1101:10778:58074 1:N:0:CTTGTA
CGCCATCATATTCGTAGGAGTAAACATAACATTCTTCCCTCAACATTTCCTGGGCCTTTCAGGAATACCACGACGCTACTCAGGC
+
CCCFFFFFHHHHHJJJJJJHIJIJJJJJJJJJFIJJJJJJJJJJJJIJJJJJJJJJJJJJIJJJJJJJJJJHHFBDDDDDDDDDD
@HWI-ST369:96:D0PLGACXX:8:1101:10948:37076 1:N:0:CTTGTA
AAGAACCCCGCCTGTTTACCAAAAACATCACCTCTAGCATTACAAGTATTAGAGGCACTGCCTGCCCAGTGACTAAAGTTTAACG
```

FASTQ Format

Each FASTQ entry is 4 lines long and incorporates three types of information:

- Line 1 starts with an @ and is the read’s identifier
- Line 2 shows the called bases in this read
- Line 3 is a separator (the + character).
- Line 4 shows the instrument’s confidence in each of the base calls.



```
 SSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSS.....................................................
 ..........................XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX......................
 ...............................IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII......................
 .................................JJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJ......................
 LLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLL....................................................
 !"#$%&'()*+,-./0123456789:;<=>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_`abcdefghijklmnopqrstuvwxyz{|}~
 |                         |    |        |                              |                     |
33                        59   64       73                             104                  126
 0........................26...31.......40
                          -5....0........9.............................40
                                0........9.............................40
                                   3.....9.............................40
 0.2......................26...31........41

S - Sanger        Phred+33,  raw reads typically (0, 40)
X - Solexa        Solexa+64, raw reads typically (-5, 40)
I - Illumina 1.3+ Phred+64,  raw reads typically (0, 40)
J - Illumina 1.5+ Phred+64,  raw reads typically (3, 40)
    with 0=unused, 1=unused, 2=Read Segment Quality Control Indicator (bold)
    (Note: See discussion above).
L - Illumina 1.8+ Phred+33,  raw reads typically (0, 41)
```

https://en.wikipedia.org/wiki/FASTQ_format

The ID and bases are straightforward, but the quality scores look like gibberish and require some explanation.

The scores assigned to each base call are *Phred quality scores*. Phred scores are logarithmic and represent the likelihood that the given base call is incorrect. Most datasets use Sanger scoring which typically ranges from 0 to 40. Some reference points in that spectrum:

Score	**Instrument thinks it be wrong**	
10	10% of the time	(1 out of 10 times)
20	1% of the time	(1 out of 100 times)
30	0.1% of the time	(1 out of 1000 times)
40	0.01% of the time	(1 out of 10000 times)

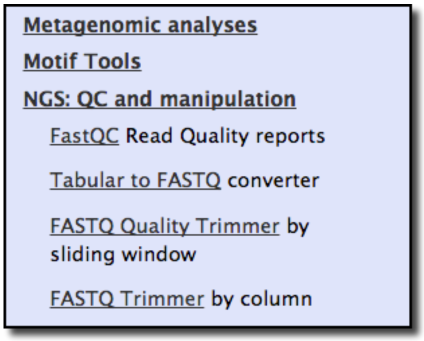
We don't see these scores in the FASTQ datasets because each score is encoded to a single character according to the FASTQSanger convention. FASTQSanger maps scores to a range of adjacent ASCII characters, starting with `!` for 0, and ending with `I` for 40.

This encoding to a single character means there is a per-column correspondence between a called base, and its corresponding quality score: the *nth* called base will be in column *n* of the called bases line, and its score will be in column *n* of the score line.

Do you ever need to know, for example, that `B` means a quality score of 33? No. The tools will take care of this. However, if you see a quality line that looks like swearing in comics ("#$%&!") then that's not good.

Check FASTQ dataset quality

That's what FASTQ datasets are. Let's now get an idea of how good these datasets are, and if we should do anything before using them in our analysis.

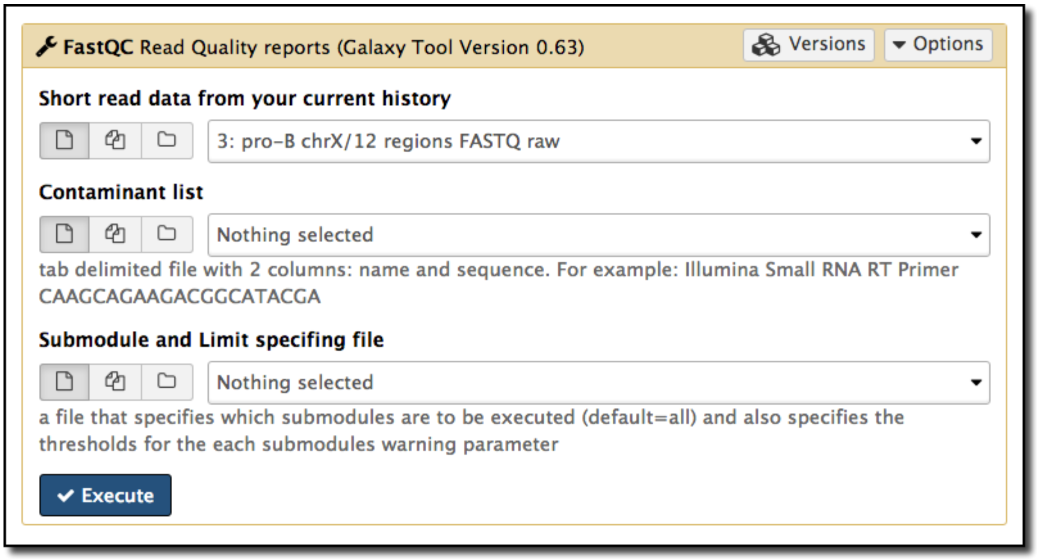

Metagenomic analyses
Motif Tools
NGS: QC and manipulation
FastQC Read Quality reports
Tabular to FASTQ converter
FASTQ Quality Trimmer by sliding window
FASTQ Trimmer by column

There are two ways to find tools on a Galaxy server. If you know the name of the tool you can enter it in the tool panel search box. You can also try searching for more general terms like "trim" or "quality." Searching with more general terms will be hit or miss.

You can also just browse each Toolbox. In this case we are investigating the quality of an NGS dataset, so the **NGS: QC and Manipulation** toolbox looks the most promising. *Click* on the toolbox name to see the tools in it. In this case **FastQC Read Quality reports** is right at the top, and that sounds exactly like what we are looking for. *Click* on it.

FastQC is a widely used tool for summarizing the quality of FASTQ datasets. It's good at presenting the big picture, and at identifying specific areas that may need to be addressed.

This is a typical, if simple, Galaxy tool form. All tool forms have the same look and use common elements. A particular strength of Galaxy is that it presents a wide range of disparate tools in a uniform fashion.



FastQC Read Quality reports (Galaxy Tool Version 0.63)

Short read data from your current history

3: pro-B chrX/12 regions FASTQ raw

Contaminant list

Nothing selected

tab delimited file with 2 columns: name and sequence. For example: Illumina Small RNA RT Primer CAAGCAGAAGACGGCATACGA

Submodule and Limit specifing file

Nothing selected

a file that specifies which submodules are to be executed (default=all) and also specifies the thresholds for the each submodules warning parameter

Tool pages generally have these sections

- *Versions and options* - If multiple versions are supported, there will be link to run the others. And the options menu enables you see more about the tool and how it is wrapped in Galaxy.
- *The form* - where you configure and run the tool
- *Documentation* - Describes the tool and provides help. This section often include links to a tool's external documentation.
- *Citation* - A paper, if there is one, for this tool.

Run our first tool: FastQC

FastQC has only three parameters that can be set. Let's set only the top one, **Short read data from your current history**. *Click* this option's dataset pulldown menu and select **2: pre-pro-B chrX/12 regions FASTQ raw**.

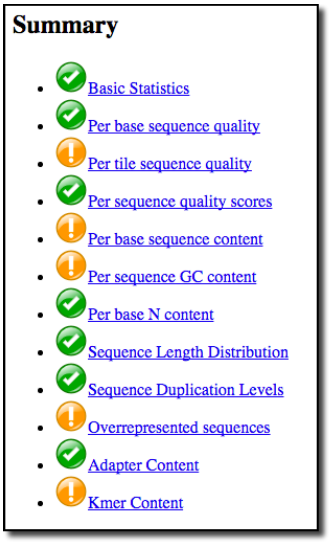
> *Galaxy tools know what types of datasets they can operate on. When the tool form is loaded your history is searched for compatible datasets and all such datasets are shown in the tool's dataset pulldown menus. In this case, FastQC recognized that is can analyze the quality of the two FASTQ datasets, but not the GTF dataset.*

We aren't going to set the contaminant list or limit the submodules/tests that FastQC runs. We'll request that it run all of its tests.

Click **Execute**.

Now a quick succession of things (hopefully) happens

1. A big green box appears in the middle panel and two small gray boxes appear at the top of the history. These mean that the task has been successfully queued: It's ready to run, but isn't running yet.
2. The gray boxes in the history are replaced with yellow boxes. This means the job has started and is actively running
3. The yellow box are replaced with green boxes indicating that the tool finished successfully.



Summary

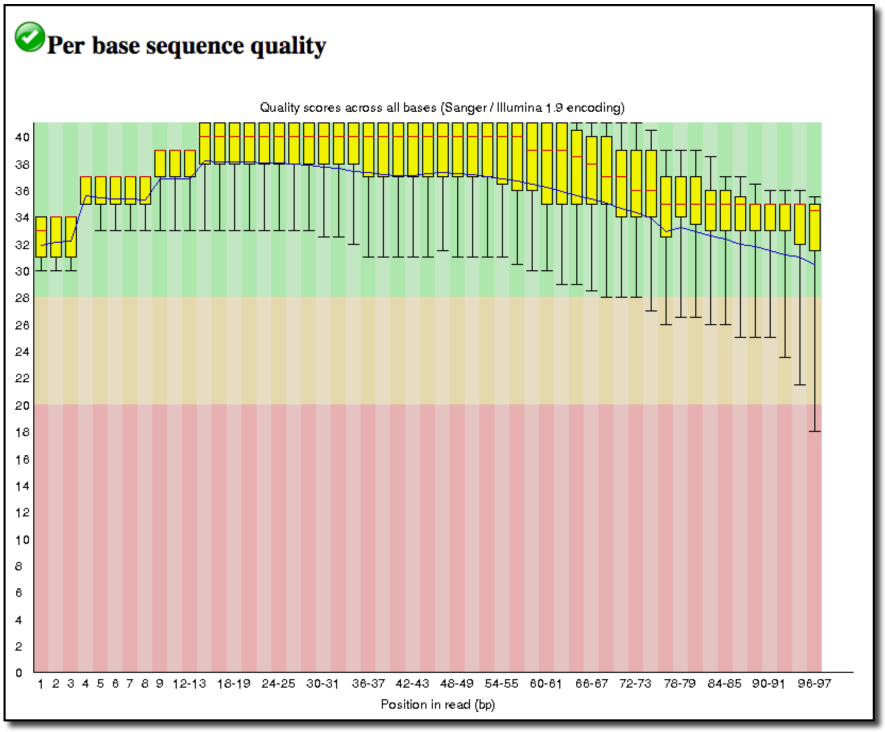
- ✓ Basic Statistics
- ✓ Per base sequence quality
- ⚠ Per tile sequence quality
- ✓ Per sequence quality scores
- ⚠ Per base sequence content
- ⚠ Per sequence GC content
- ✓ Per base N content
- ✓ Sequence Length Distribution
- ✓ Sequence Duplication Levels
- ⚠ Overrepresented sequences
- ✓ Adapter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

Once a tool has finished running you can preview the output datasets (by clicking on the dataset name) or view the data itself (by poking it in the eye). *Poke* the **FastQC Web page** dataset *in the eye*.

This shows 12 different reports about the pre-pro-B FASTQ dataset.

For each test it runs, FastQC has predefined thresholds for what constitutes good, not-so-good, and bad. This dataset has passed 7 of the tests and got warning on the rest. Let's take a look at some of these tests.

Per base sequence quality (boxplot)



✓ **Per base sequence quality**

Boxplots are the most common way to summarize the quality of a FASTQ dataset.

- **X axis is the position in the read.** That is, the first column summarizes all the base calls in the first position of all the reads; the second column summarizes the second call in all the reads and so on. Starting at position 10, the information is aggregated for every two bases.
- **Y axis is the Phred quality score**, from 0 to 40.
- Each column summarizes the quality scores across all reads at that position
  - **Yellow boxes bound the middle 50%** of quality scores at each position in the read.
  - **Whiskers bound the middle 80%** of quality scores at that position.
  - **Blue line is the average score** at each position
  - **Red lines are the median score** at each position.

This dataset shows a typical pattern for Illumina data. Quality dips at the front, but rises quickly, then stays good and slowly decays at the end, with the lowest quality at the end.

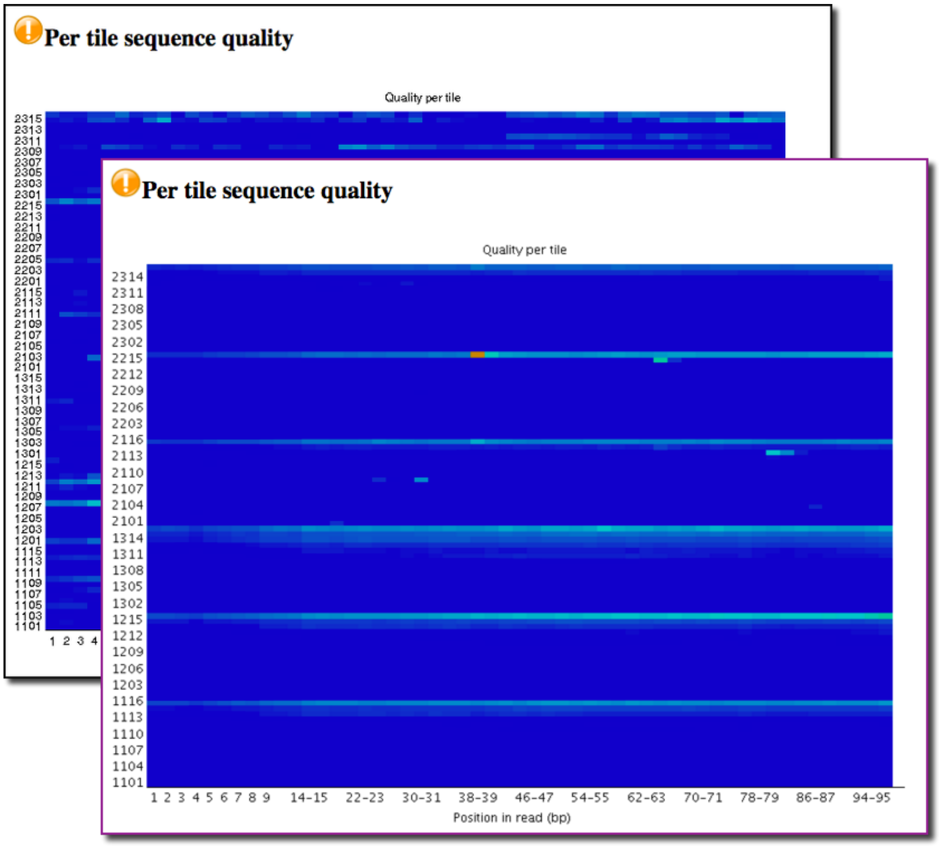
This dataset is also pretty good. It's only at the last two positions that more than 10% of the quality scores fall below 20 (the *errors occur more than 1% of the time* boundary). However, we can't tell from this graph where the bottom 10% of reads are in any of the positions. They could all be just below (or at) the whiskers, or they could be 2's.

Per tile sequence quality

This graph is particularly significant for this crowd. It shows something that your clients have no control over but that you do: where on the slide the low quality reads tended to happen. As core staff you can use this information to identify trouble spots with the sequencing run.

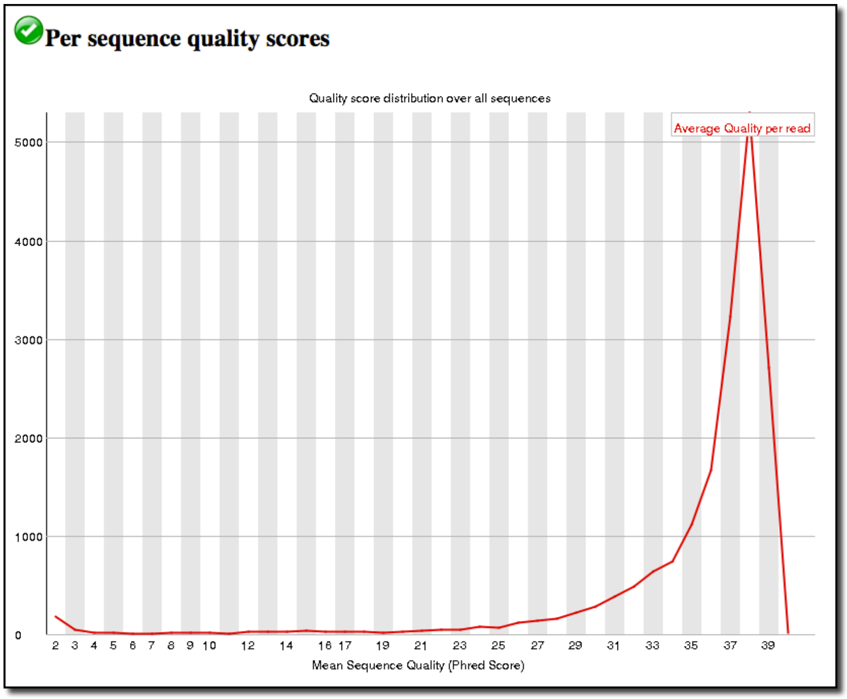
However, as core staff you should also have access to analysis tools from your sequencing vendor that enable you to find this same information and more.

For Illumina data, tile information is embedded in the ID line of FASTQ entries.



Note: The graphic for the complete ~90 million read dataset is superimposed over our sample dataset graph here. With the larger dataset, it's much clearer what's happening.

Per sequence quality scores

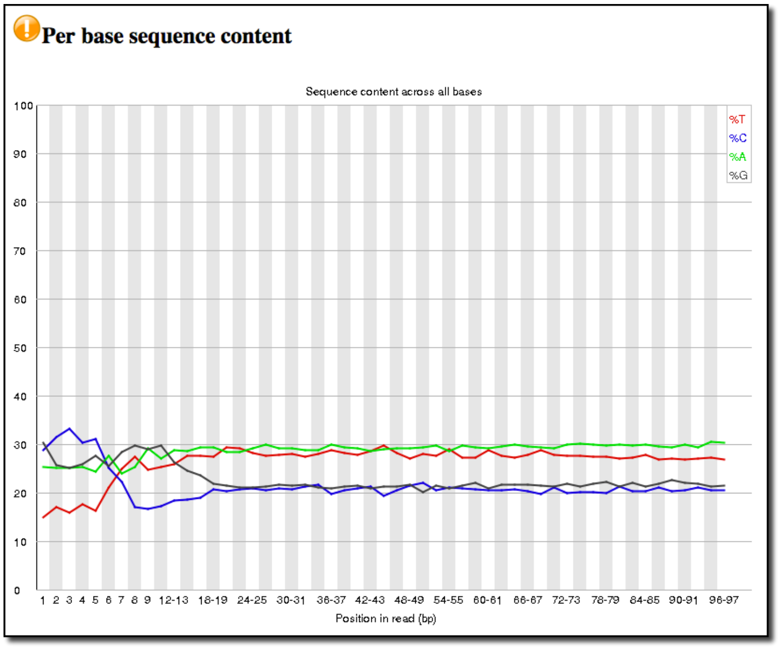


Remember this statement from the boxplot explanation above:

However, we can't tell from this graph where the bottom 10% of reads are in any of the positions.

This graph gives us an idea just how bad (or good) our bottom 10% is. Here, the X axis is the quality score, and the Y axis is the number of times that score occurs across the entire dataset. The scores in this dataset are pretty good and there are only a small number of calls with confidence below 20. However, there is a small spike at 2. We might want to get rid of these low quality calls.

Per base sequence content



This graph is good at highlighting if you might have untrimmed adapter or barcode problems.

In an ideal world the lines in this graph would be flat. Flat lines mean there is no bias for particular bases at particular positions. Alas, these lines are not flat, *especially at the beginning*. What's going on here?

The turbulence at the front is caused by the use of random hexamers in Illumina library preparation. See

> Hansen, *et al.*, "Biases in Illumina transcriptome sequencing caused by random hexamer priming" *Nucleic Acids Research*, Volume 38, Issue 12 (2010)

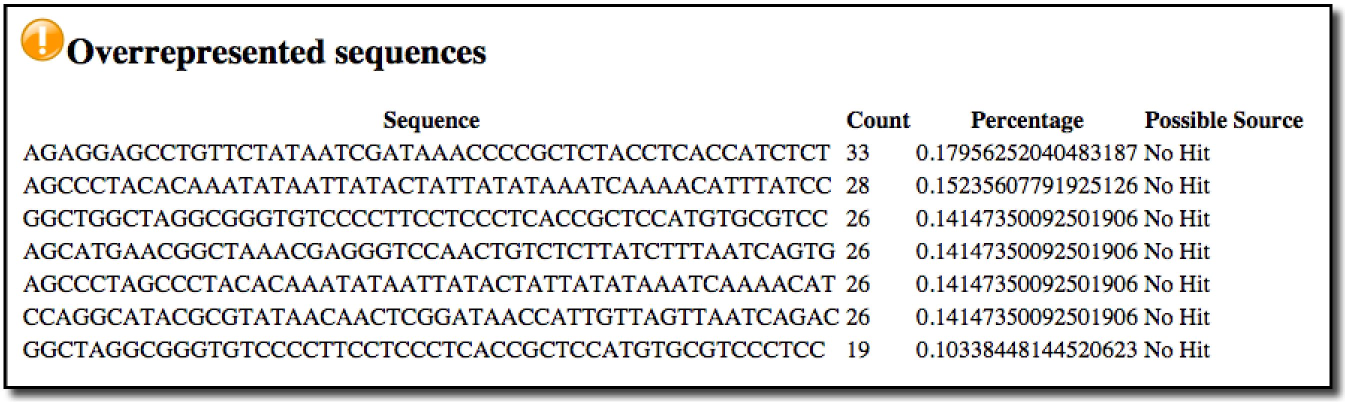
for details. Basically Illumina library prep is slightly less than random. The summary is: *It's not our fault.*

Should we do anything about it? No. We could trim the leading 12 or so base calls, but

1. this sequence is real - it actually exists in the organism, and
2. trimming only hides the bias, and it makes the mapping less accurate.

So we'll leave it in.

Overrepresented sequences

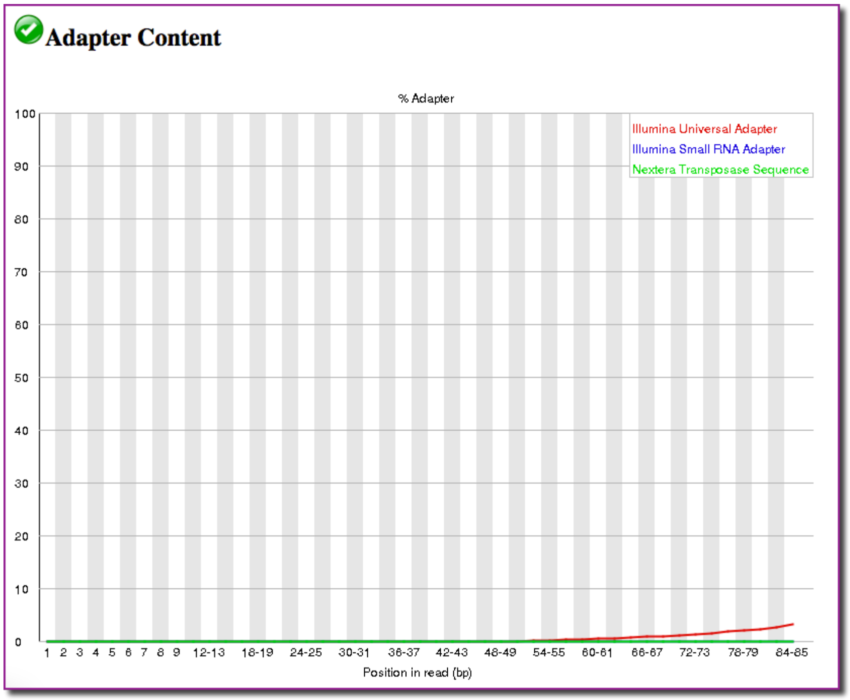


Overrepresented sequences

Sequence	Count	Percentage	Possible Source
AGAGGAGCCTGTTCTATAATCGATAAACCCCGCTCTACCTCACCATCTCT	33	0.17956252040483187	No Hit
AGCCCTACACAAATATAATTATACTATTATATAAATCAAAACATTTATCC	28	0.15235607791925126	No Hit
GGCTGGCTAGGCGGGTGTCCCCTTCCTCCCTCACCGCTCCATGTGCGTCC	26	0.14147350092501906	No Hit
AGCATGAACGGCTAAACGAGGGTCCAACTGTCTCTTATCTTTAATCAGTG	26	0.14147350092501906	No Hit
AGCCCTAGCCCTACACAAATATAATTATACTATTATATAAATCAAAACAT	26	0.14147350092501906	No Hit
CCAGGCATACGCGTATAACAACTCGGATAACCATTGTTAGTTAATCAGAC	26	0.14147350092501906	No Hit
GGCTAGGCGGGTGTCCCCTTCCTCCCTCACCGCTCCATGTGCGTCCCTCC	19	0.10338448144520623	No Hit

A couple of things to note here:

- FastQC was written to analyze DNA datasets, not RNA datasets. With RNA we expect some overrepresented sequence, especially for housekeeping genes.
- This is Mitochondrial 12S/16S sequence.



Adapter Content

There's something going on here. It looks like at the end of the reads the dataset is 3 to 4 percent untrimmed adapter. This falls within FastQC's tolerance, but maybe not ours. We might want to clip those off.

And the rest

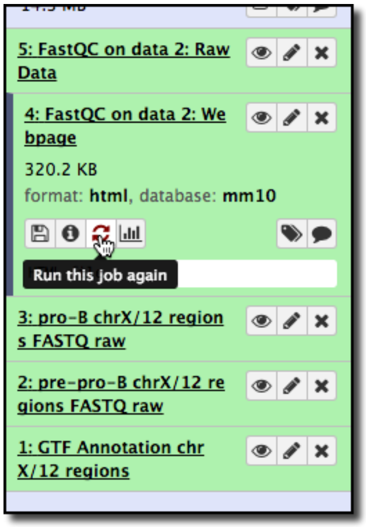
The remaining tests are either not informative (look at Sequence Length Distribution), flat out good (Per base N content), or look significantly different when the whole dataset is scanned (Per sequence GC content, Sequence Duplication Levels, Kmer Content). We'll ignore all these today.

Rerun FastQC on pro-B dataset

Now let's run FastQC on the other dataset, the pro-B reads. We could rerun FastQC by finding it in the Tool panel again, or, we could rerun FastQC from the History panel. Let's do that.

Preview one of the **FastQC** datasets in your history by *clicking* on its name. In the preview *click* on the **Run this job again** looping arrow icon. This launches the tool form in the middle panel and pre-populates it with the same settings it was run with before.

"Re-run" has several uses in Galaxy. First, it's an easy way to see exactly how this dataset was generated. It's also a good way to tweak the tool's parameters and then rerun it with almost the same settings as before. That's what we'll do.


5: FastQC on data 2: Raw Data

4: FastQC on data 2: Webpage

320.2 KB

format: html, database: mm10

Run this job again

3: pro-B chrX/12 regions FASTQ raw

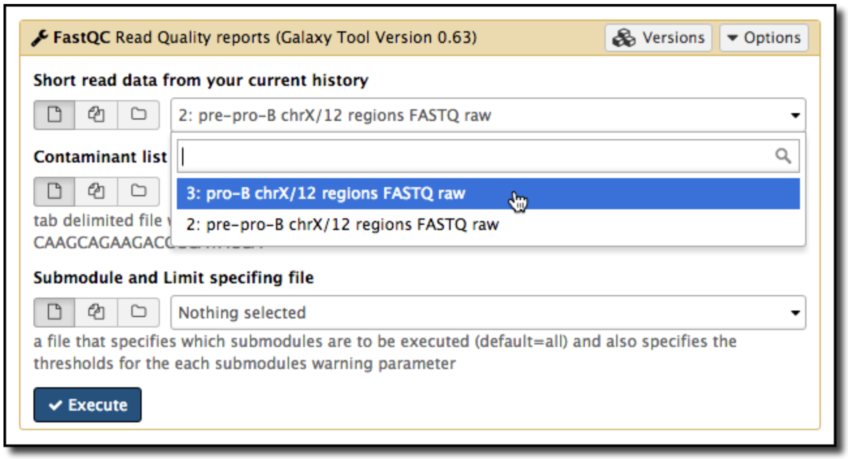
2: pre-pro-B chrX/12 regions FASTQ raw

1: GTF Annotation chrX/12 regions

In the middle panel *Click* the dataset pulldown menu and *select* **3: pro-B chrX/12 regions FASTQ raw**. Then *click* **Execute**.

Poke the new **FastQC Webpage** *in the eye* to review the results.

The quality is similar.


FastQC Read Quality reports (Galaxy Tool Version 0.63)

Short read data from your current history

2: pre-pro-B chrX/12 regions FASTQ raw

Contaminant list

3: pro-B chrX/12 regions FASTQ raw

2: pre-pro-B chrX/12 regions FASTQ raw

tab delimited file w
CAAGCAGAAGACC

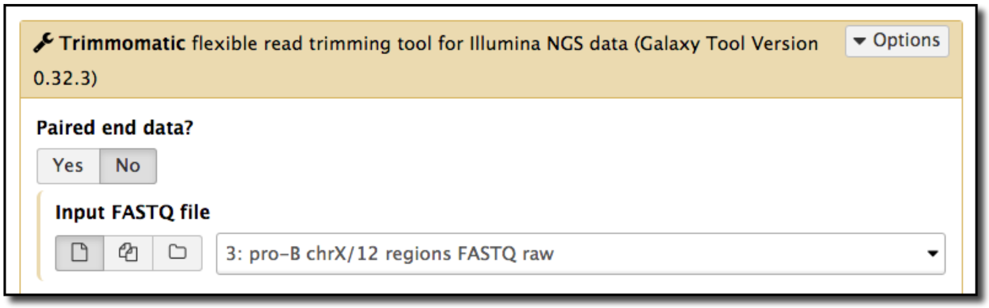
Submodule and Limit specifing file

Nothing selected

a file that specifies which submodules are to be executed (default=all) and also specifies the thresholds for the each submodules warning parameter

Improve the quality: Trimmomatic

We identified that things were pretty good, but that there were some quality issues with the data. Let's use Trimmomatic to separate the good from the bad. Trimmomatic supports several common trimming approaches for NGS data. We'll use three: ILLUMINACLIP, SLIDINGWINDOW and MINLEN.



Trimmomatic flexible read trimming tool for Illumina NGS data (Galaxy Tool Version 0.32.3)

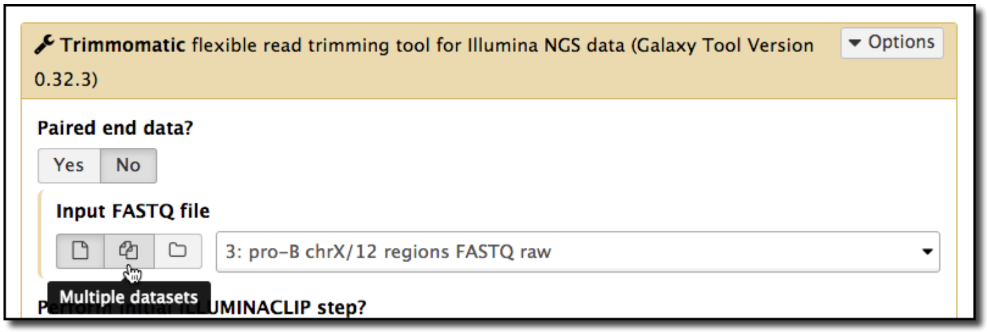
Paired end data?

Input FASTQ file: 3: pro-B chrX/12 regions FASTQ raw

FastQC was the first tool in the **NGS: QC and manipulation** toolbox. **Trimmomatic** is the last. *Click* on it to launch the tool form in the middle panel.

First, *change* **Paired end data?** to **No**. This changes the form to ask for a single dataset instead of two.

Processing Multiple datasets with one submission



Trimmomatic flexible read trimming tool for Illumina NGS data (Galaxy Tool Version 0.32.3)

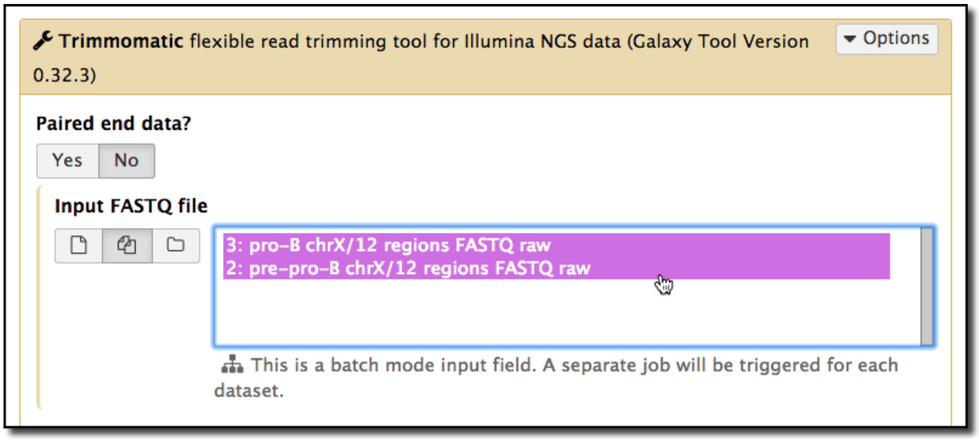
Paired end data?

Input FASTQ file: 3: pro-B chrX/12 regions FASTQ raw

At this point we could run Trimmomatic the same way we ran FastQC - by running it on one dataset, and then re-running it on the second.

However, we can achieve the same effect more efficiently by using the **Multiple datasets** feature, another shortcut available in Galaxy.

To enable this *click* on the **Multiple datasets icon** under **Input FASTQ file**.



Trimmomatic flexible read trimming tool for Illumina NGS data (Galaxy Tool Version 0.32.3)

Paired end data?

Input FASTQ file

3: pro-B chrX/12 regions FASTQ raw
2: pre-pro-B chrX/12 regions FASTQ raw

This is a batch mode input field. A separate job will be triggered for each dataset.

This changes the parameter selection from a pull-down to a multi-select field. *Select* both datasets.

This tells Galaxy to run Trimmomatic twice, once on each dataset. It's identical to what we did with FastQC and the re-run button, but is quicker and less error prone.

Select Trimmomatic Filters

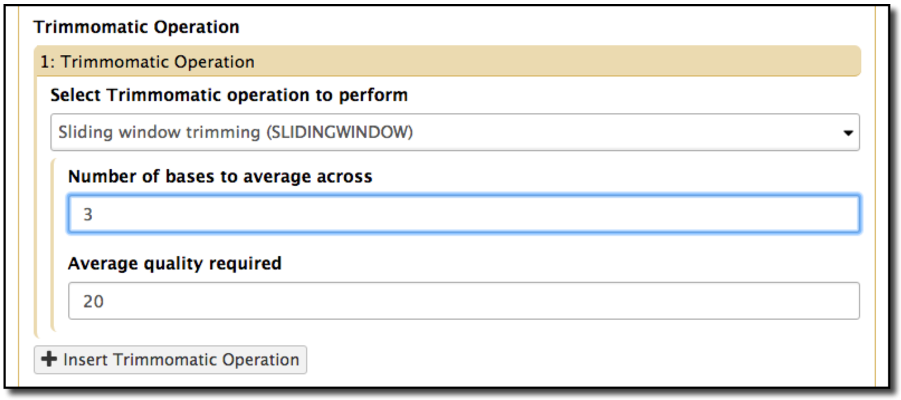
We want to use three different filters in Trimmomatic:

1. **ILLUMINACLIP** - This will trim known Illumina adapters from reads in the dataset**.**
2. **SLIDINGWINDOW** - Cutting once the average quality within a sliding window falls below a threshold. Trims from both ends
3. **MINLEN** - Drop any reads that are shorter than a given length after trimming.

Trimmomatic has 7 different types of trims it can run. It can run as many or as few of these as you want, and you get to the pick the order the tests are run in.

To enable the adapter trimming step, *select* **Yes** under **Perform initial ILLUMINACLIP step?**. This updates the form to show several parameters for this step. *Change* the **Adapter sequences to use** pulldown to **TruSeq3 (single-ended, for MiSeq and HiSeq)** (these data were generated using a MiSeq). Leave the other parameters at the default.

The form defaults to a single SLIDINGWINDOW step. This is what we want, but let's update it to use the same window definition as the paper. *Change* the **Number of bases to average across** to **3**.



Trimmomatic Operation

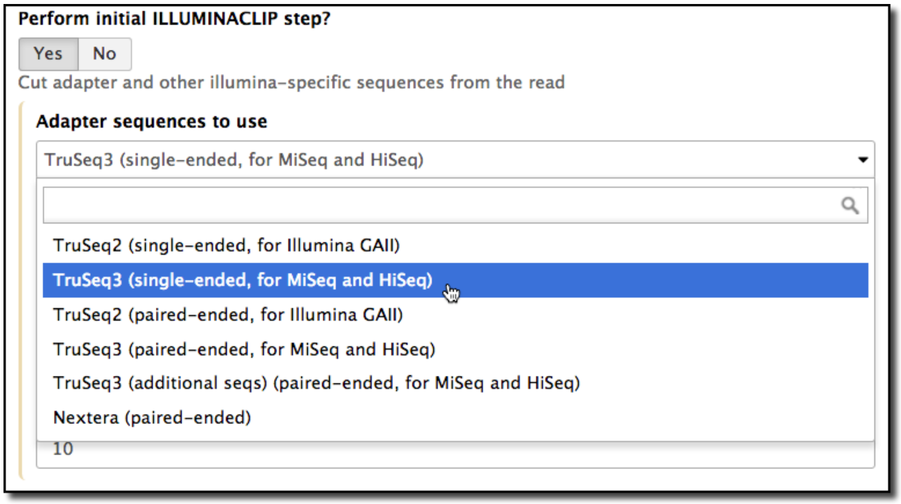
1: Trimmomatic Operation

Select Trimmomatic operation to perform: Sliding window trimming (SLIDINGWINDOW)

Number of bases to average across: 3

Average quality required: 20

Once the sliding window step is run, some of the reads may too short to be meaningful and would just become noise in our analysis. (In fact, some of them might also be entirely trimmed because they never reach minimum requirements.) To eliminate any short reads we need to add another operation.



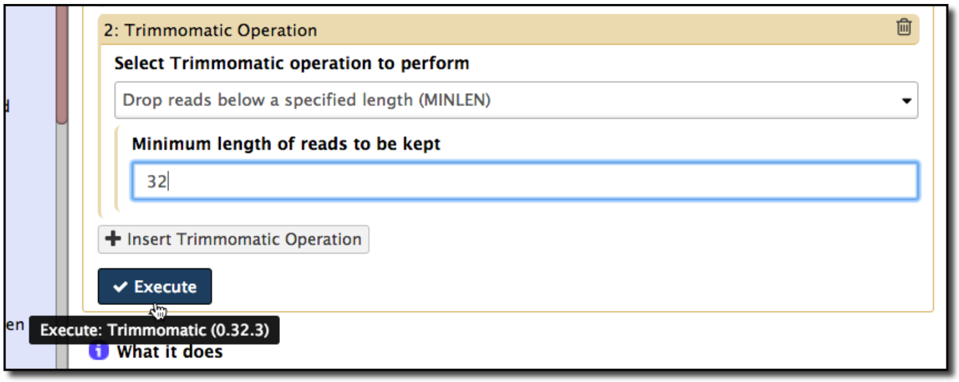
Perform initial ILLUMINACLIP step? Yes No

Cut adapter and other illumina-specific sequences from the read

Adapter sequences to use: TruSeq3 (single-ended, for MiSeq and HiSeq)

TruSeq2 (single-ended, for Illumina GAII)
TruSeq3 (single-ended, for MiSeq and HiSeq)
TruSeq2 (paired-ended, for Illumina GAII)
TruSeq3 (paired-ended, for MiSeq and HiSeq)
TruSeq3 (additional seqs) (paired-ended, for MiSeq and HiSeq)
Nextera (paired-ended)
10

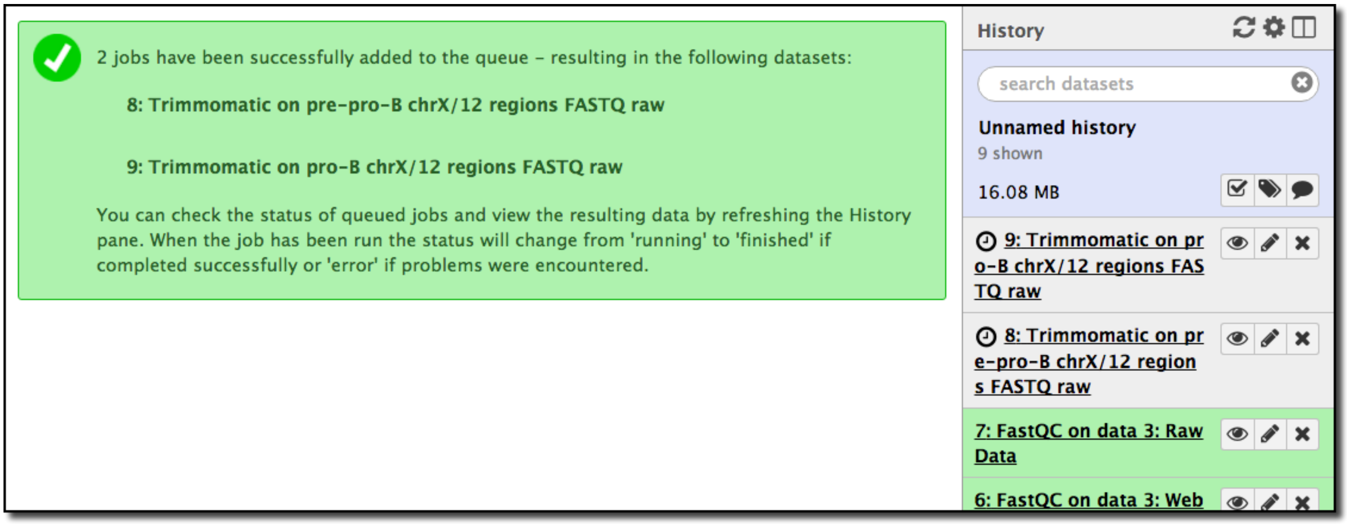
Click on **+ Insert Trimmomatic Operation**. This adds a second operation. *Change* **Select Trimmomatic operation to perform** to **Drop reads below a specified length (MINLEN)** and *set* **Minimum length of reads to be kept** to **32**. Now that all 3 steps are set up, *click* **Execute**.



2: Trimmomatic Operation

Select Trimmomatic operation to perform: Drop reads below a specified length (MINLEN)

Minimum length of reads to be kept: 32



2 jobs have been successfully added to the queue – resulting in the following datasets:

8: Trimmomatic on pre-pro-B chrX/12 regions FASTQ raw

9: Trimmomatic on pro-B chrX/12 regions FASTQ raw

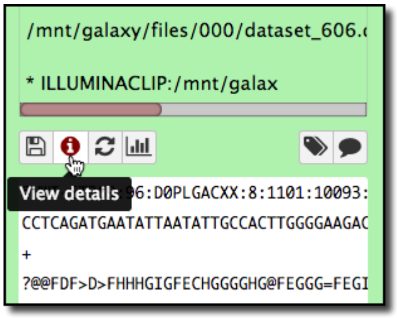
You can check the status of queued jobs and view the resulting data by refreshing the History pane. When the job has been run the status will change from 'running' to 'finished' if completed successfully or 'error' if problems were encountered.

History
search datasets
Unnamed history
9 shown
16.08 MB
9: Trimmomatic on pro-B chrX/12 regions FASTQ raw
8: Trimmomatic on pre-pro-B chrX/12 regions FASTQ raw
7: FastQC on data 3: Raw Data
6: FastQC on data 3: Web

As requested, clicking Execute queues two runs of Trimmomatic, one for each input dataset.

Trimmomatic Results

There are several ways to determine how much Trimmomatic trimmed, and how much it improved the quality of the input datasets. The easiest way is to preview the before and after datasets, and compare their size. The pre-pro-B dataset dropped from 4.4mb to 4.1mb and the pro-B dropped from 7.9mb to 7.4mb.



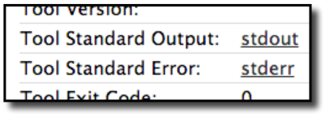
/mnt/galaxy/files/000/dataset_606.d

* ILLUMINACLIP:/mnt/galax

```
96:D0PLGACXX:8:1101:10093:
CCTCAGATGAATATTAATATTGCCACTTGGGGAAGAC
+
?@@FDF>D>FHHHGIGFECHGGGGHG@FEGGG=FEGI
```

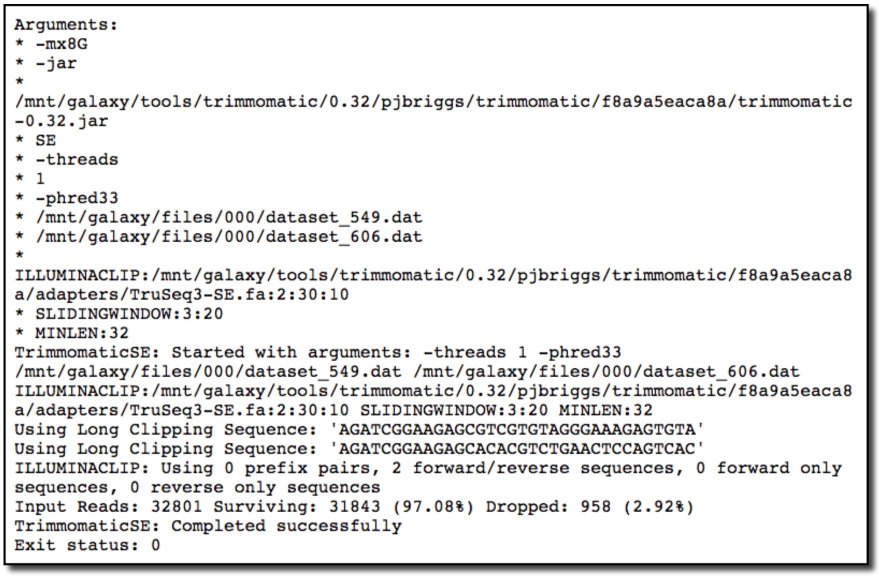
To get a more detailed picture we'll have to do some digging. Many tools provide runtime summaries that are shown in the dataset preview. Trimmomatic does this, but the summary is truncated. To see the full summary, *click* the ⓘ **icon (View details)** in the dataset preview.

This displays the metadata for this dataset, including the exact tool that was run and the parameter settings. Two links are included with the metadata for **stdout** and **stderr**. These are standard (thus the *std*) files in Unix that are used to record *standard output* and *standard error* from the tool. Standard output lists non-error outputs from the program, while standard error lists warnings and errors. (Well, that how it's *supposed* to be.)



Tool Version:
Tool Standard Output: stdout
Tool Standard Error: stderr
Tool Exit Code: 0

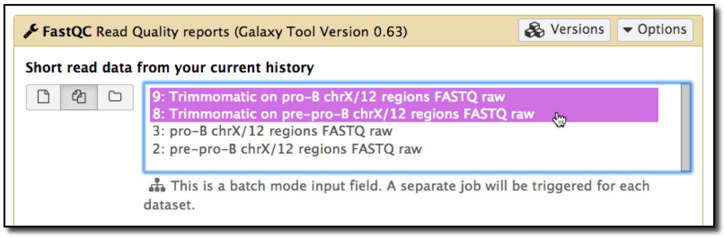
Click on the **stdout** link to see the full summary. This contains a little more information about what was trimmed, including that 958 reads were dropped altogether from the pro-B dataset.



```
Arguments:
* -mx8G
* -jar
*
/mnt/galaxy/tools/trimmomatic/0.32/pjbriggs/trimmomatic/f8a9a5eaca8a/trimmomatic
-0.32.jar
* SE
* -threads
* 1
* -phred33
* /mnt/galaxy/files/000/dataset_549.dat
* /mnt/galaxy/files/000/dataset_606.dat
*
ILLUMINACLIP:/mnt/galaxy/tools/trimmomatic/0.32/pjbriggs/trimmomatic/f8a9a5eaca8
a/adapters/TruSeq3-SE.fa:2:30:10
* SLIDINGWINDOW:3:20
* MINLEN:32
TrimmomaticSE: Started with arguments: -threads 1 -phred33
/mnt/galaxy/files/000/dataset_549.dat /mnt/galaxy/files/000/dataset_606.dat
ILLUMINACLIP:/mnt/galaxy/tools/trimmomatic/0.32/pjbriggs/trimmomatic/f8a9a5eaca8
a/adapters/TruSeq3-SE.fa:2:30:10 SLIDINGWINDOW:3:20 MINLEN:32
Using Long Clipping Sequence: 'AGATCGGAAGAGCGTCGTGTAGGGAAAGAGTGTA'
Using Long Clipping Sequence: 'AGATCGGAAGAGCACACGTCTGAACTCCAGTCAC'
ILLUMINACLIP: Using 0 prefix pairs, 2 forward/reverse sequences, 0 forward only
sequences, 0 reverse only sequences
Input Reads: 32801 Surviving: 31843 (97.08%) Dropped: 958 (2.92%)
TrimmomaticSE: Completed successfully
Exit status: 0
```

That's informative, but to get a feel for the overall quality of the post-Trimmomatic, let's run FastQC again on both and compare those reports with the original reports.

Click **FastQC** in the tool panel, *select* the **Multiple datasets icon** under **Short read data from your current history**, and then *select* both **Trimmomatic on …** datasets. *Click* **Execute.**



FastQC Read Quality reports (Galaxy Tool Version 0.63)

Short read data from your current history

9: Trimmomatic on pro-B chrX/12 regions FASTQ raw
8: Trimmomatic on pre-pro-B chrX/12 regions FASTQ raw
3: pro-B chrX/12 regions FASTQ raw
2: pre-pro-B chrX/12 regions FASTQ raw

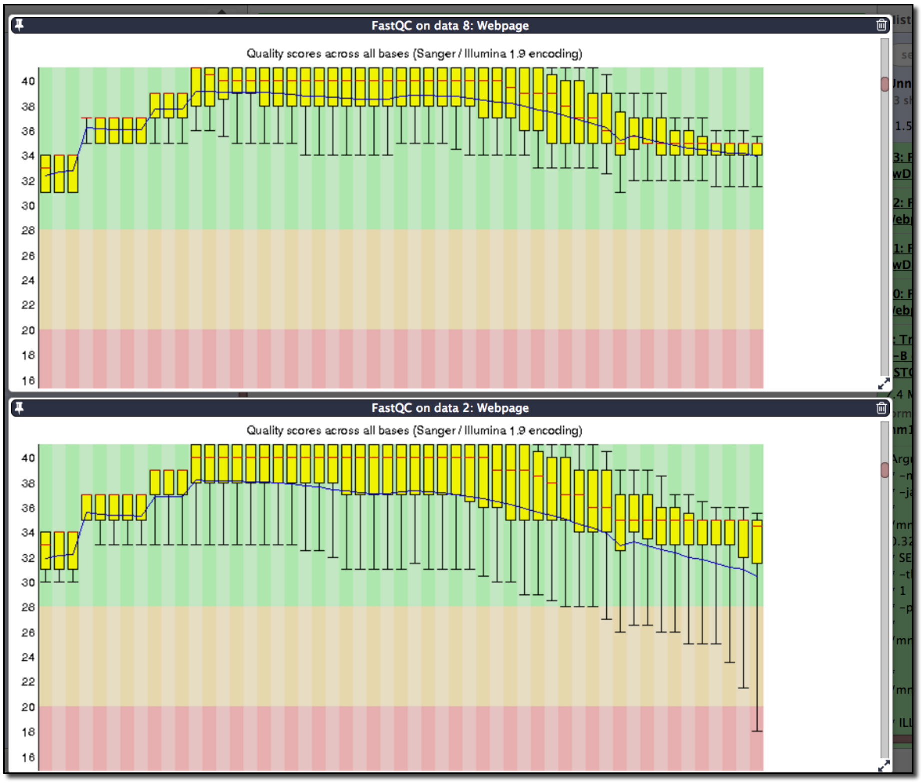
This is a batch mode input field. A separate job will be triggered for each dataset.

> *Note: We aren't using paired-end data today, but if we were Trimmomatic would automatically maintain read pairings throughout all steps.*

Using Galaxy Scratchpad to view multiple datasets

We want to compare pre and post-Trimmomatic FastQC reports. We could do this by alternately poking the two datasets in the eye, scrolling down to the graph we are reviewing and repeating, until, sadly, we would go insane. To avoid this fate, you can use Scratchpad to view multiple datasets simultaneously.

To enable Scratchpad, *click* on the grid icon (▦) in the top menu bar. This turns it yellow (▦)! Now, *scroll down* in your history to first FastQC report on the pre-pro-B data (most likely named **4: FastQC on data 2: Webpage**) and *poke it in the eye*.



FastQC on data 8: Webpage
Quality scores across all bases (Sanger / Illumina 1.9 encoding)
40
38
36
34
32
30
28
26
24
22
20
18
16
FastQC on data 2: Webpage
Quality scores across all bases (Sanger / Illumina 1.9 encoding)
40
38
36
34
32
30
28
26
24
22
20
18
16

Rather than display the dataset in the middle panel, the dataset is now displayed on an overlap of the entire browser window. *Click* somewhere outside the displayed dataset. This will remove the overlap and return you to the standard view. *Scroll up* the history to the second FastQC report on the pre-pro-B data (most likely the fourth dataset from the top) and *poke it in the eye*.

After some scrolling and resizing you can now simultaneously compare the two reports. Additional datasets can be added as desired.

We see that there has been a sizable improvement in the quality of this dataset. Note that the lowest quality score we now have anywhere is 25.

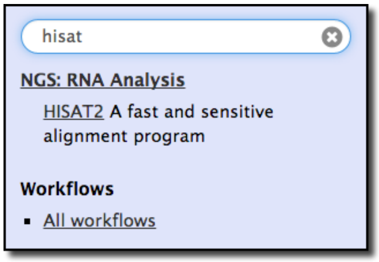
To switch back to the standard view either click outside the displayed datasets or poke the yellow eye (👁2). To disable the Scratchbook display, click on the yellow grid icon.

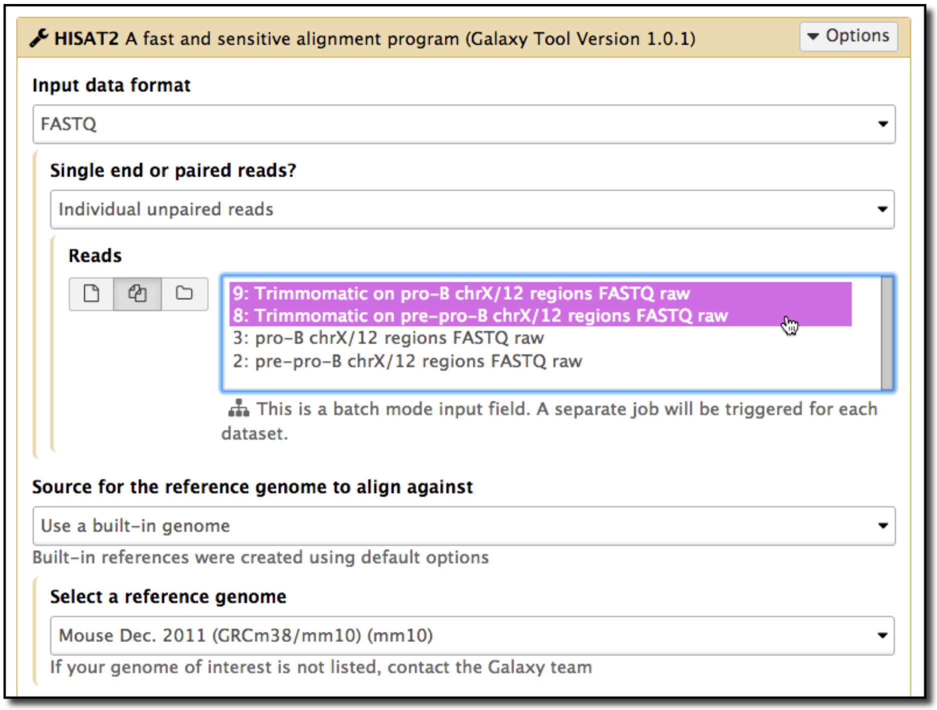
Map the reads

The data has now been cleaned and we are ready to map them against the mouse reference genome. Mapping RNA-Seq data involves finding the most likely place in the genome where each read came from, while also having to consider that reads may need to be broken up across introns. This is no small task.

HISAT2

We are going to use HISAT2 to do this. These servers also have Tophat2 installed on them, and we could use that too (although HISAT is newer and claimed to be much faster, and more sensitive). *Search* for **HISAT** in the tool panel. *Click* on it to bring up the HISAT2 tool form.





HISAT2 A fast and sensitive alignment program (Galaxy Tool Version 1.0.1)

Input data format
FASTQ

Single end or paired reads?
Individual unpaired reads

Reads
9: Trimmomatic on pro-B chrX/12 regions FASTQ raw
8: Trimmomatic on pre-pro-B chrX/12 regions FASTQ raw
3: pro-B chrX/12 regions FASTQ raw
2: pre-pro-B chrX/12 regions FASTQ raw

This is a batch mode input field. A separate job will be triggered for each dataset.

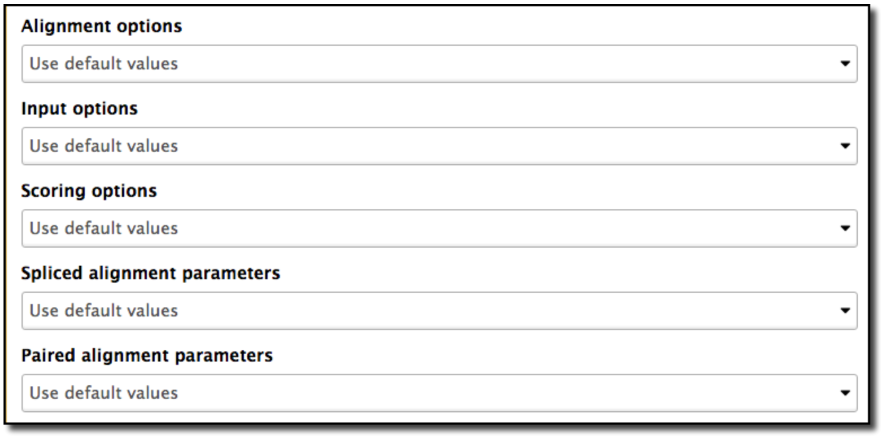
Source for the reference genome to align against
Use a built-in genome
Built-in references were created using default options

Select a reference genome
Mouse Dec. 2011 (GRCm38/mm10) (mm10)
If your genome of interest is not listed, contact the Galaxy team

Change **Single end or paired reads?** to **Individual unpaired reads** and then *select* the **Multiple datasets icon** under **Reads**. *Select* both **Trimmomatic on …** datasets. (This will fire off two independent HISAT runs - one for each dataset.)

Leave the **reference genome** as **mm10**. (A production server would have many reference genomes here.)

HISAT, like many of the "heavy lifting" tools on Galaxy, has a wealth of run-time options. The HISAT tool wrapper (which is what defines a tool to Galaxy) takes a common approach of providing defaults by placing them in sections that can be expanded as needed. HISAT has 5 such sections:



Alignment options
Use default values

Input options
Use default values

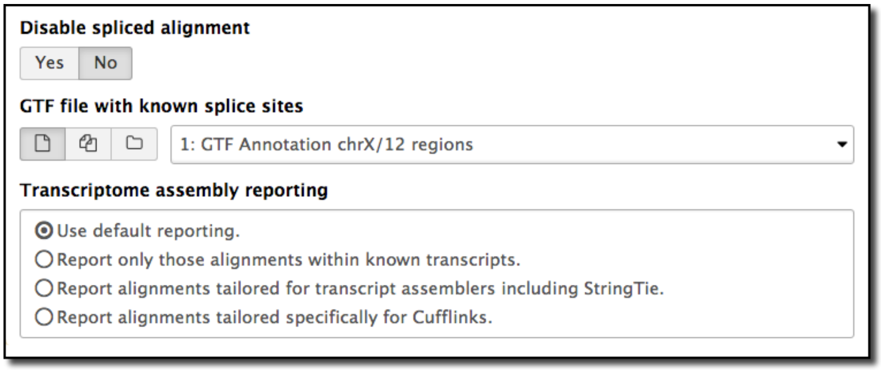
Scoring options
Use default values

Spliced alignment parameters
Use default values

Paired alignment parameters
Use default values

If you expand all of these sections, HISAT options go from 1 screen to almost 6 screens. Understanding all these options before running a tool is the right thing to do, but it's also daunting. One of Galaxy's strengths is that it allows you to *experiment with tools* and *learn them incrementally*. For today's exercise we are going to start with setting just one parameter.

Change **Spliced alignment parameters** to **Specify spliced alignment parameters**. This displays almost 15 additional options, most of them giving you fine-grained control over how reads that are split across multiple exons are scored. Ignore all those and *scroll* down to **GTF file with known splice sites** and *select* **1: GTF Annotation chrX/12 regions**.



Disable spliced alignment

Yes No

GTF file with known splice sites

1: GTF Annotation chrX/12 regions

Transcriptome assembly reporting

● Use default reporting.
○ Report only those alignments within known transcripts.
○ Report alignments tailored for transcript assemblers including StringTie.
○ Report alignments tailored specifically for Cufflinks.

The GTF dataset contains genome annotation for the regions of chromosome X and chromosome 12 that we'll be focusing on today. Providing this to HISAT2 gives the tool guidance (but not dogma) on where intron/exon boundaries are. This information is used when aligning reads with small anchors on one side of a splice site.

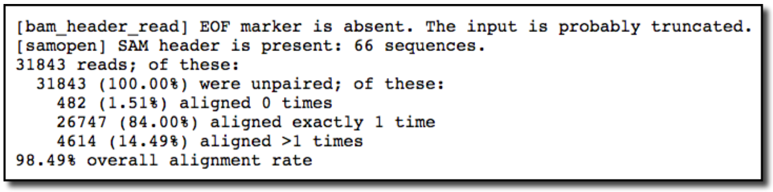
> *Note: If we were doing transcript assembly on the entire datasets we might consider selecting the Report alignments tailored for transcript assemblers including StringTie option under Transcriptome assembly reporting. This helps optimize compute and memory usage in transcript assemblers.*

Click **Execute**.

HISAT2 Results

This creates a mapped reads dataset for each of the two inputs. These files are in BAM format, an efficient binary format for representing mapped data. There is an equivalent text format called SAM (and Galaxy has tools for converting between the two).

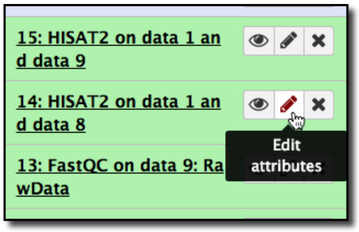
How did the data do? Again, like Trimmomatic, there is a summary of the run in the dataset preview, but again like Trimmomatic that summary is truncated. To see the full summary *click* the **ⓘ icon (View details)** in the dataset summary. This time click on the **stderr** link. This tells us that our overall alignment rates are over 98% for both datasets. (The alignment rates on the full datasets are not this good.)



```
[bam_header_read] EOF marker is absent. The input is probably truncated.
[samopen] SAM header is present: 66 sequences.
31843 reads; of these:
  31843 (100.00%) were unpaired; of these:
    482 (1.51%) aligned 0 times
    26747 (84.00%) aligned exactly 1 time
    4614 (14.49%) aligned >1 times
98.49% overall alignment rate
```

Updating dataset and history metadata: Renaming

Up until now, all but the FastQC datasets in our history have had reasonable names. However, the HISAT2 output datasets have names like **HISAT2 on data 1 and data 8**. Everything in those names is true, but it's not as helpful as could be. It would be much better to know, for example, which of those HISAT outputs is for the pro-B and the pre-pro-B datasets.



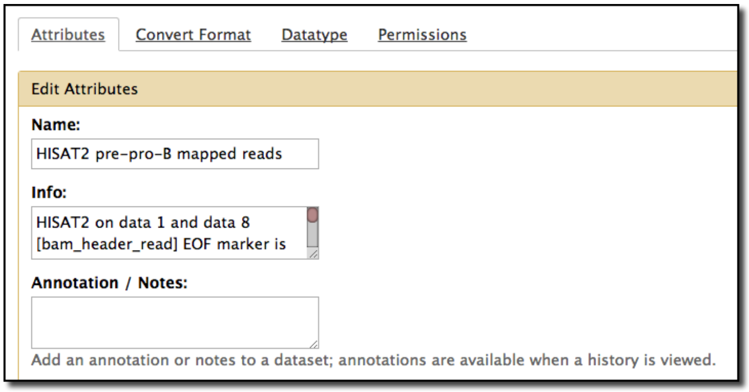
15: HISAT2 on data 1 and data 9

14: HISAT2 on data 1 and data 8

13: FastQC on data 9: RawData

To address this let's update the HISAT2 datasets' metadata. *Click* on the **pencil (Edit attributes) icon** for the pre-pro-B **HISAT2 dataset**.

Since all the information in the current name is accurate, *move* it to the **Info** field. Now, what to name it? I'll suggest **HISAT2 pre-pro-B mapped reads**. This is kinda long, but tells us very quickly what this is. *Click* **Save** and then *repeat* with the **pro-B dataset.**



Edit Attributes

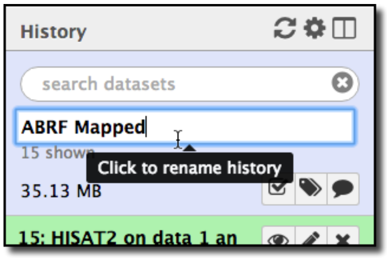
Name:
HISAT2 pre-pro-B mapped reads

Info:
HISAT2 on data 1 and data 8
[bam_header_read] EOF marker is

Annotation / Notes:

Add an annotation or notes to a dataset; annotations are available when a history is viewed.

While we are at it, we should also rename our history. Galaxy is quite happy to support 15 (or 150 or …) histories under the same user all named **Unnamed history**. However, when you come back to Galaxy a month or a week, or even a day later, that is just not helpful. A best practice is to always name your histories. To do this, *click* on **Unnamed history**, *enter* a new and informative history name, and then *press* **return** or **enter**.



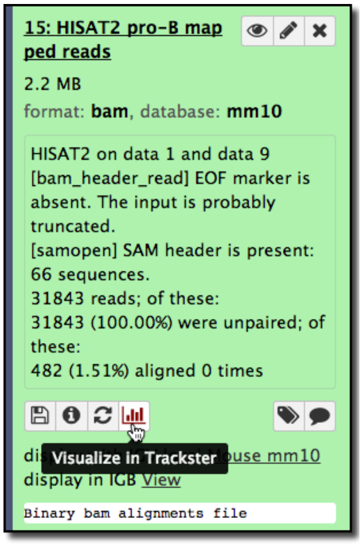
ABRF Mapped
15 shown
35.13 MB
15: HISAT2 on data 1 an

But what do the HISAT2 mappings look like?

We have some idea for how many reads mapped, but we still don't know much beyond that. There are a couple of things we could do here. There are summary tools that provide more summary statistics on things like mapping state (**Flagstat**) and how many reads mapped to each chromosome (**IdxStats**).

We can also visually inspect the mappings in the context of the genomic regions they mapped to. Later today you'll use the IGV desktop program to do just that. Right now, let's learn how to do this inside Galaxy, using the Trackster visualization tool.

To launch Trackster, open a preview of one of the HISAT2 datasets and *click* on the **Visualize in Trackster icon**.

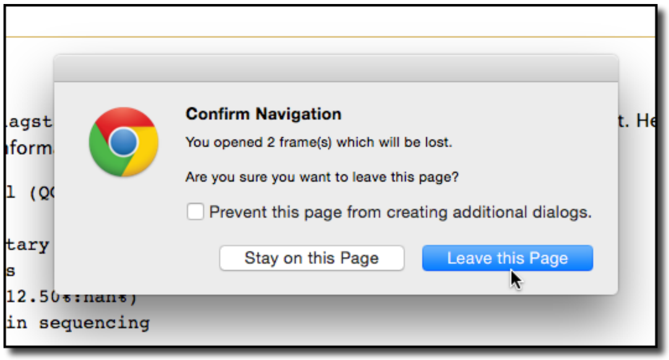


15: HISAT2 pro-B mapped reads
2.2 MB
format: bam, database: mm10

HISAT2 on data 1 and data 9
[bam_header_read] EOF marker is absent. The input is probably truncated.
[samopen] SAM header is present: 66 sequences.
31843 reads; of these:
31843 (100.00%) were unpaired; of these:
482 (1.51%) aligned 0 times

Binary bam alignments file

Note: If you haven't reloaded the page since working with Scratchbook (see above), you may see a warning like



Confirm Navigation

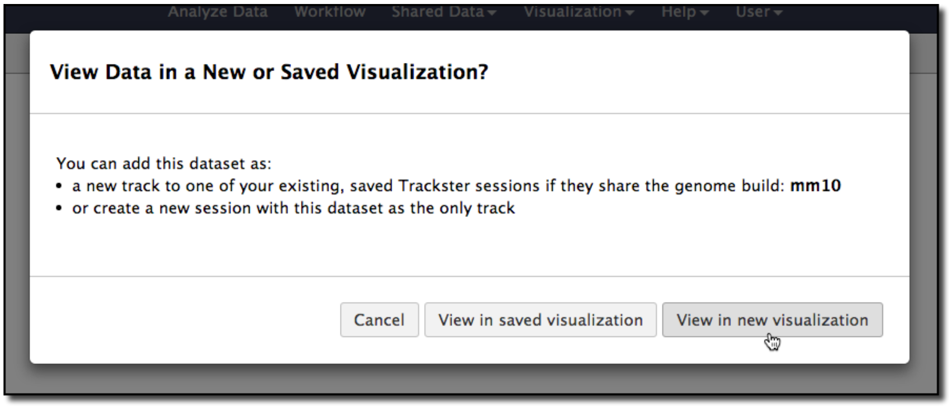
You opened 2 frame(s) which will be lost.

Are you sure you want to leave this page?

□ Prevent this page from creating additional dialogs.

Click ***Leave this Page****. Your Scratchpad will be cleared when you do this.*

This will clear all 3 panels and display:



Analyze Data Workflow Shared Data Visualization Help User

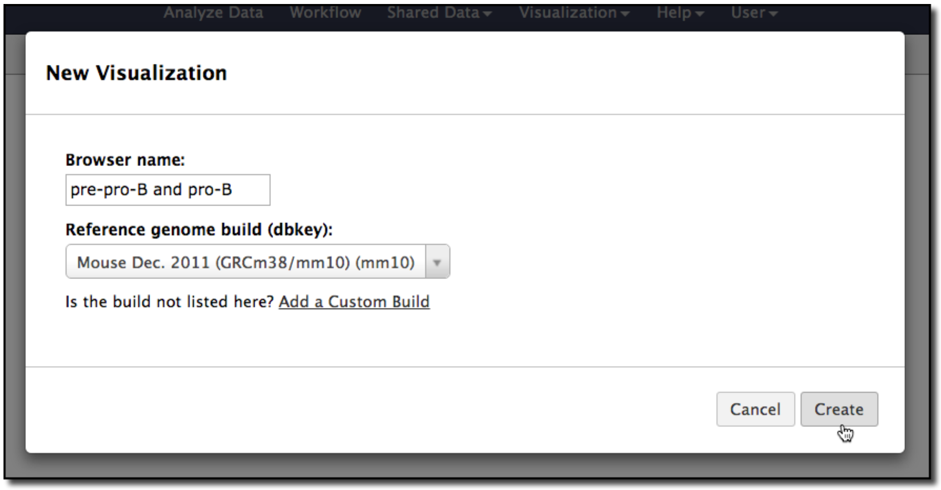
View Data in a New or Saved Visualization?

You can add this dataset as:

- a new track to one of your existing, saved Trackster sessions if they share the genome build: **mm10**
- or create a new session with this dataset as the only track

Cancel | View in saved visualization | View in new visualization

Select **View in new visualization** and then give the visualization a meaningful name and *click* **Create**.



Analyze Data Workflow Shared Data Visualization Help User

New Visualization

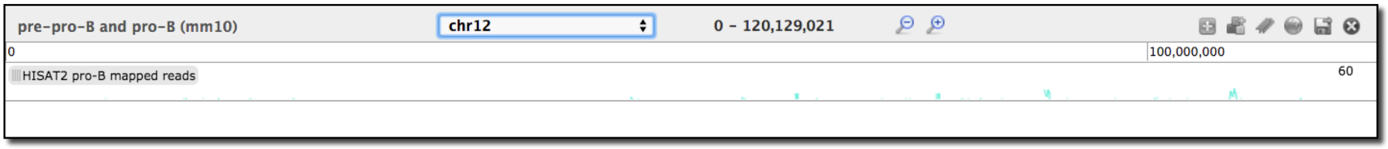
Browser name: pre-pro-B and pro-B

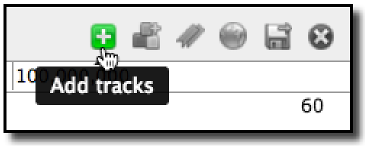
Reference genome build (dbkey): Mouse Dec. 2011 (GRCm38/mm10) (mm10)

Is the build not listed here? Add a Custom Build

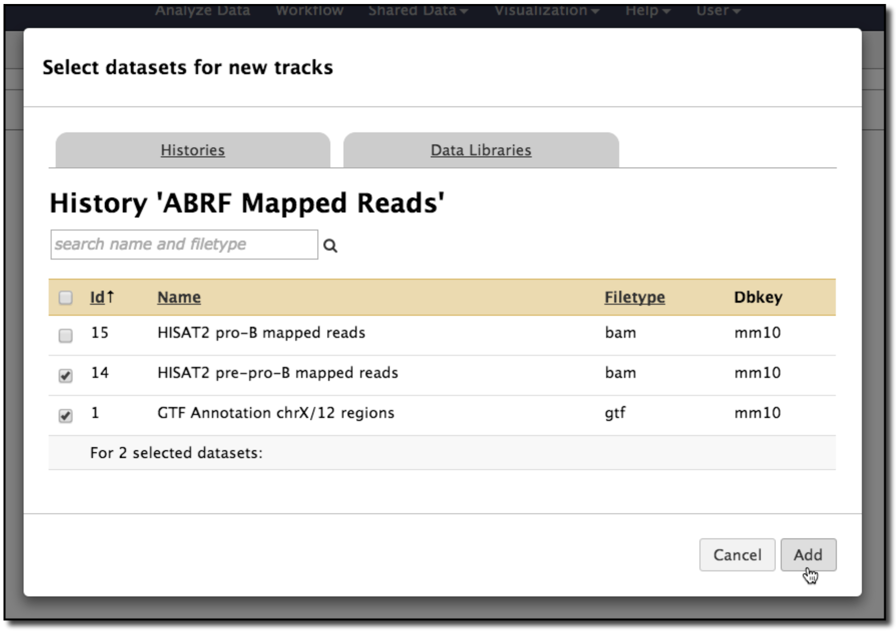
Cancel | Create

This will bring up a new Trackster browser with a status message saying that it is indexing the dataset for display. While the indexing is happening switch from chr1 to chr12 (or chrX), where we have mapped data.





Once that track has loaded (or while it is loading), let's add other datasets to the display. *Click* on the **+ (Add tracks) icon** in the upper right, and then select the other mapped dataset and the reference annotation.



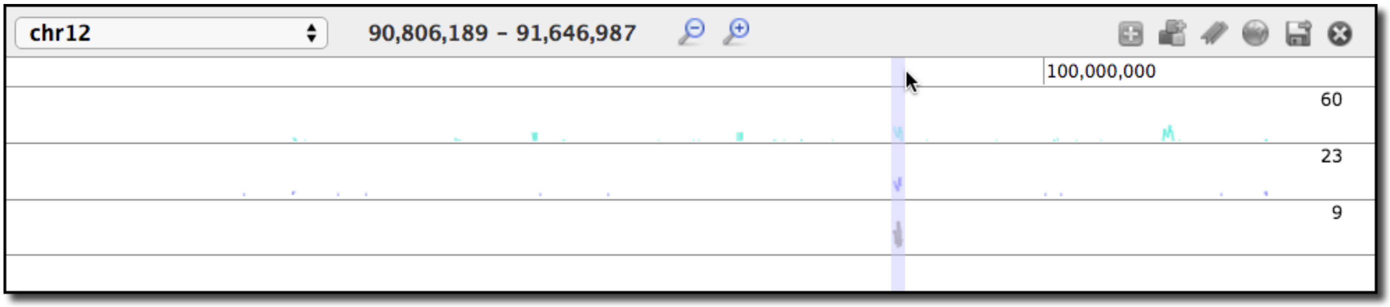
Select datasets for new tracks

History 'ABRF Mapped Reads'

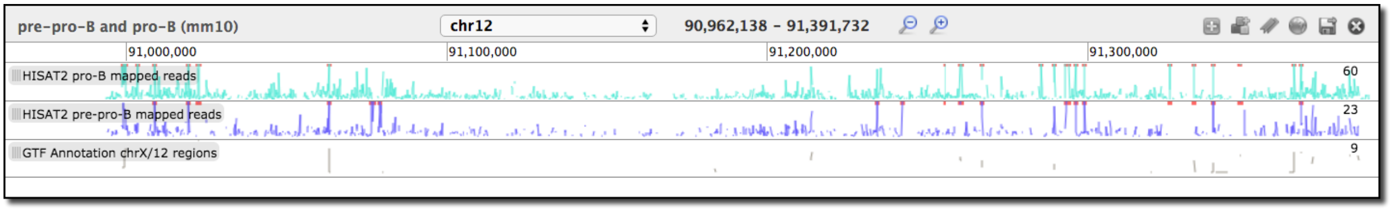
	Id↑	Name	Filetype	Dbkey
□	15	HISAT2 pro-B mapped reads	bam	mm10
☑	14	HISAT2 pre-pro-B mapped reads	bam	mm10
☑	1	GTF Annotation chrX/12 regions	gtf	mm10

For 2 selected datasets:

Once those new tracks have loaded, let's zoom in on the region of interest - the area with the annotations. To zoom in, *click and drag* on the coordinates track (near the top) and cover the entire region with annotation.

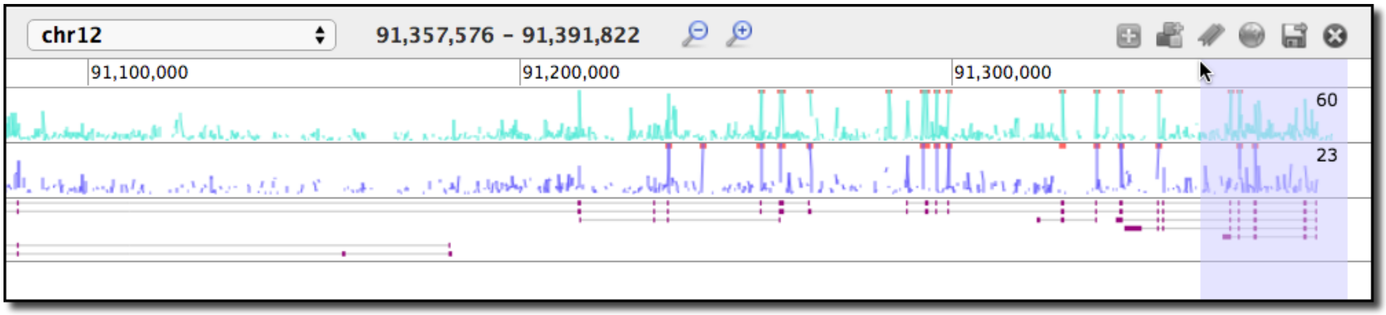


You may have to repeat that a few times to get the region of interest but when you are done you'll see something like this:


pre-pro-B and pro-B (mm10)
chr12
90,962,138 – 91,391,732
91,000,000
91,100,000
91,200,000
91,300,000
HISAT2 pro-B mapped reads
60
HISAT2 pre-pro-B mapped reads
23
GTF Annotation chrX/12 regions
9

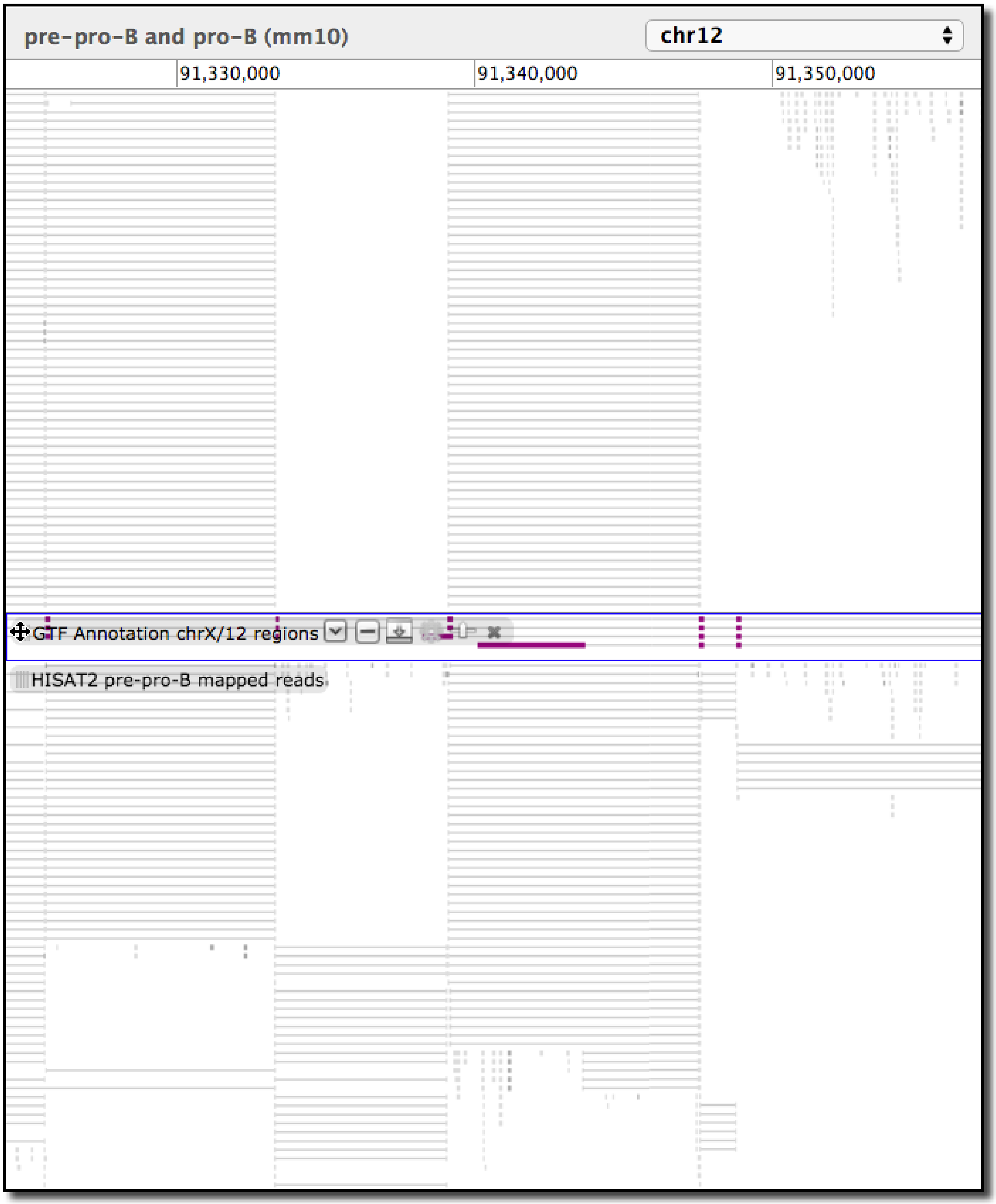
The top 2 tracks show the read depth for any position in this region, and the bottom track shows the annotation (but not very well). First, let's make the annotation more recognizable. *Hover* over the **GTF Annotation chrX/12 regions** track and then *click* on the ∨ **Set display mode icon** and *select* **Squish**. This changes the display to show the intron/exon structure of all the transcripts in the annotation file.

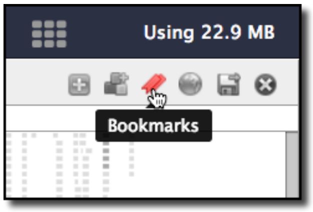
Now, to get a better idea of what the aligned reads look like, zoom in the last 5 or so introns on the far right.



This changes the display mode for all tracks and now we can see individual reads, including where splice junction were introduced to map the reads. However, it's now difficult to see the annotation.

Drag the annotation track to the middle by *clicking and dragging* the **|||| vertical lines icon** at the left of the track title upwards to be between the two BAM tracks. You can see that many of the reads align well with the annotation, but also that some do not, and that some show splice junctions were introduced in locations outside the annotations.

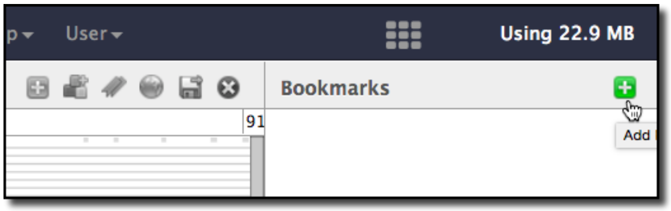

pre-pro-B and pro-B (mm10)
chr12
91,330,000
91,340,000
91,350,000
GTF Annotation chrX/12 regions
HISAT2 pre-pro-B mapped reads

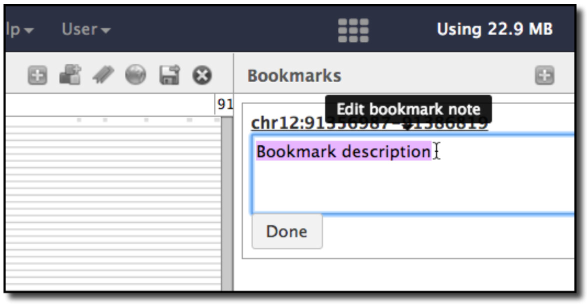


We'll want to come back to this location once we have predicted transcripts for these datasets.

Bookmark this location by *clicking* on the **bookmark ribbon icon** in the upper right corner. Then *click* the **+ (Add bookmark) icon** and give the bookmark a meaningful description.

Finally, *click* the **save icon** to save your visualization.

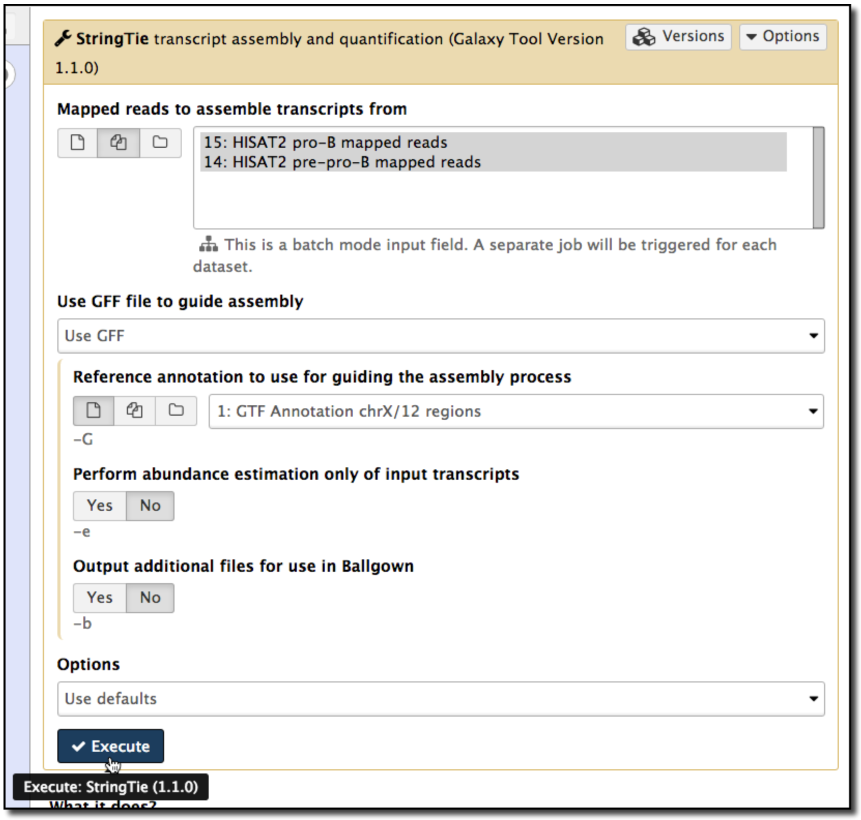




Transcript Prediction with StringTie

We've cleaned our data and mapped it the reference genome. Now let's see if we can predict the structure of transcripts occurring in the two datasets. HISAT2 aligned the reads with the genome, and split reads when that resulted in the best mapping. However, HISAT2 makes no prediction about how those reads might assemble into transcripts. That job is handled by StringTie.

Search for **StringTie** in the tool panel and then *click* on it to launch the tool form. *Select* **Multiple datasets** under **Mapped reads to assemble transcripts from**, and then *select* both **HISAT2** datasets. *Change* **Use GFF file to guide assembly** to **Use GFF**. Leave the rest of the options with defaults and *click* **Execute**.



StringTie transcript assembly and quantification (Galaxy Tool Version 1.1.0)

Mapped reads to assemble transcripts from
15: HISAT2 pro-B mapped reads
14: HISAT2 pre-pro-B mapped reads
This is a batch mode input field. A separate job will be triggered for each dataset.

Use GFF file to guide assembly: Use GFF

Reference annotation to use for guiding the assembly process: 1: GTF Annotation chrX/12 regions
-G

Perform abundance estimation only of input transcripts: Yes No
-e

Output additional files for use in Ballgown: Yes No
-b

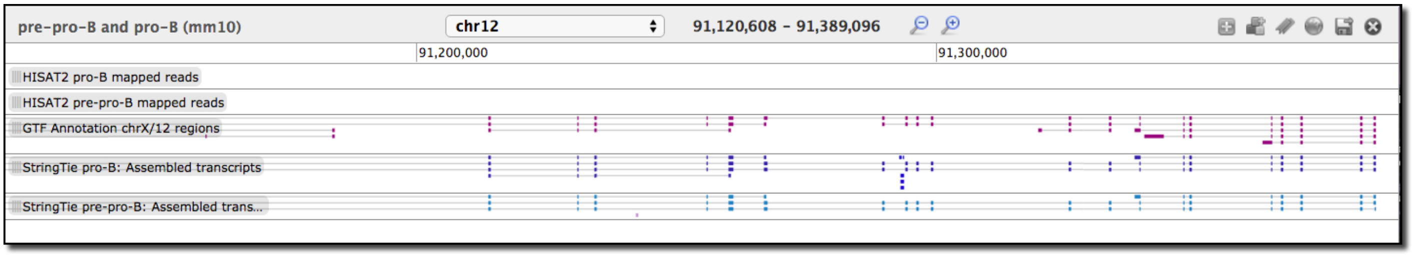
Options: Use defaults

StringTie generates 3 output datasets for each input dataset:

- Coverage - All the transcripts in the reference GTF that are fully covered by reads.
- Gene abundance estimates - This is created but it's empty because we did not ask for it in the advanced options section.
- Assembled transcripts - This is the one we care most about.

Rename the two **assembled transcripts** datasets to include **pre-pro-B** or **pro-B** in the dataset name. Once they have informative names, add the assembled transcripts datasets to the visualization you created and saved after running HISAT. The process is the same, except this time, you will view in a saved visualization.

Once you are in trackster, add the other assembled transcripts dataset. Now hide the content of the two mapped reads datasets and zoom out a few times. Set the display mode for both new datasets to Squish.


pre-pro-B and pro-B (mm10)
chr12
91,120,608 - 91,389,096
91,200,000
91,300,000
HISAT2 pro-B mapped reads
HISAT2 pre-pro-B mapped reads
GTF Annotation chrX/12 regions
StringTie pro-B: Assembled transcripts
StringTie pre-pro-B: Assembled trans...

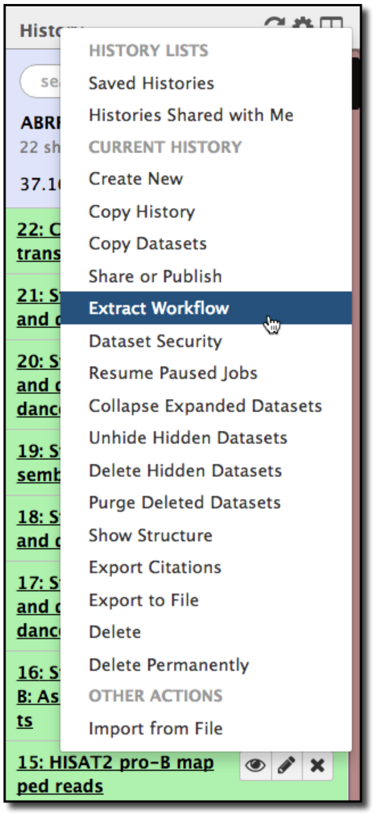
A couple of things to note about this region:

- the pro-B assembled transcripts have some transcripts that are not in the reference, and even some new exons
- the pre-pro-b assembled transcripts are largely a subset of the pro-B predicted transcripts.
- several annotated transcripts and exons are not detected at all in these datasets.

Save the visualization and go back to **Analyze Data**.

And there is more!

Workflows: repeating an analysis

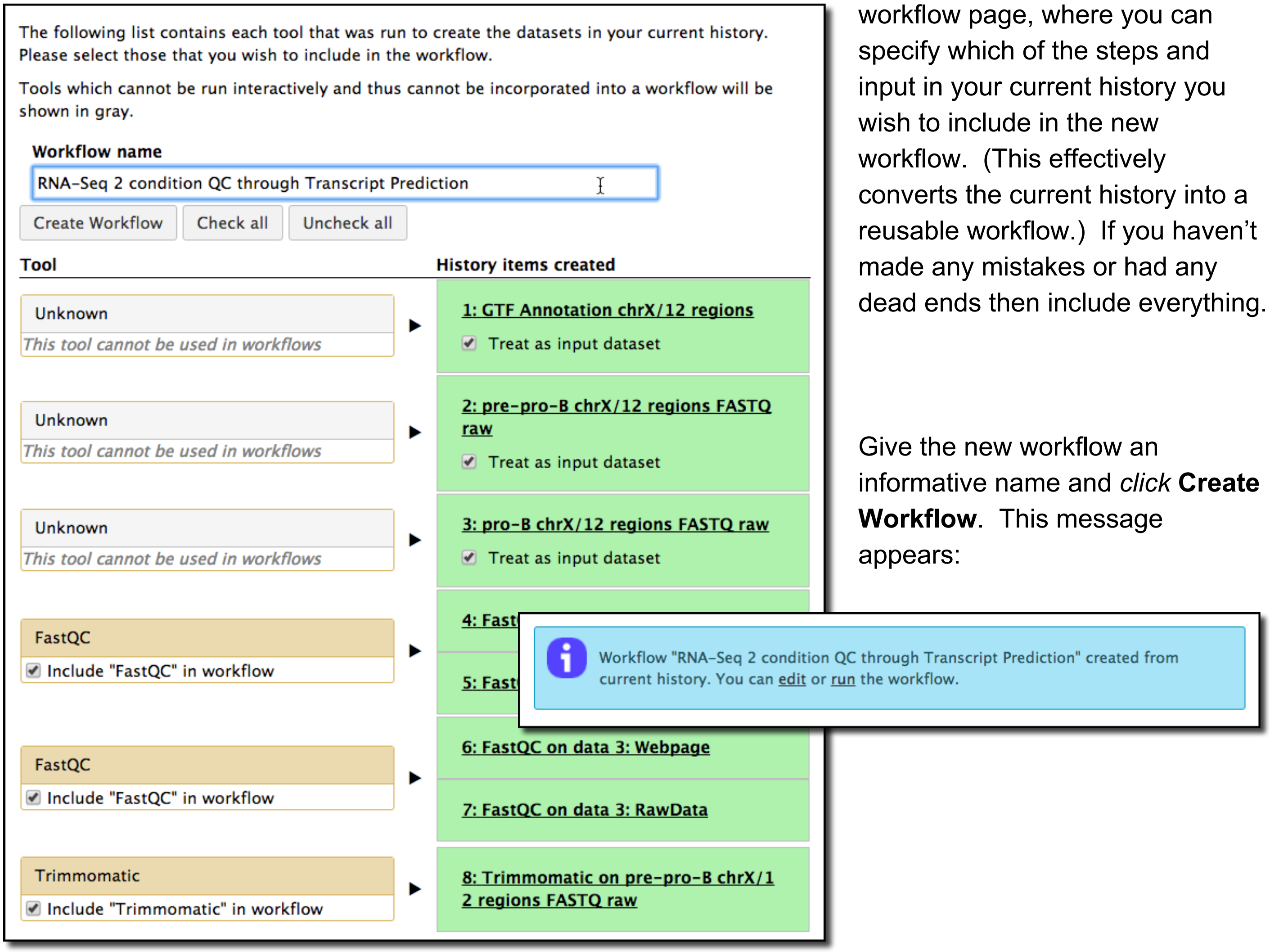


Now that we've taken our test data from raw to clean to mapped to transcript prediction, we'll want to apply the same analysis to our full datasets, and maybe with other similar experiments as well. We could manually rerun each step of the way with the larger/newer datasets. This would be tedious and error-prone, but it would get the job done.

A better solution is to create a workflow - a repeatable recipe for doing an analysis. Workflows can be rerun multiple times with different datasets each time. Galaxy enables you to create workflows *de novo*, using a workflow editor, or to create them from histories, which is what we'll do.
To create a workflow from a history, *click* on the **cog** at the top of the history panel, and then *select* **Extract Workflow** from the pulldown.

This takes you to the create workflow page, where you can specify which of the steps and input in your current history you wish to include in the new workflow. (This effectively converts the current history into a reusable workflow.) If you haven't made any mistakes or had any dead ends then include everything.

Give the new workflow an informative name and *click* **Create Workflow**. This message appears:



The following list contains each tool that was run to create the datasets in your current history. Please select those that you wish to include in the workflow.

Tools which cannot be run interactively and thus cannot be incorporated into a workflow will be shown in gray.

Workflow name

RNA-Seq 2 condition QC through Transcript Prediction

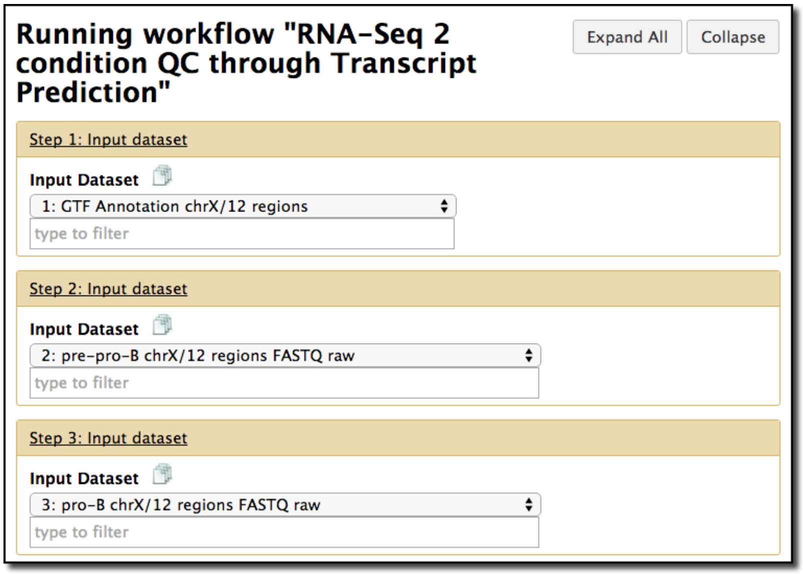
Tool	History items created
Unknown *This tool cannot be used in workflows*	1: GTF Annotation chrX/12 regions ☑ Treat as input dataset
Unknown *This tool cannot be used in workflows*	2: pre-pro-B chrX/12 regions FASTQ raw ☑ Treat as input dataset
Unknown *This tool cannot be used in workflows*	3: pro-B chrX/12 regions FASTQ raw ☑ Treat as input dataset
FastQC ☑ Include "FastQC" in workflow	4: Fast 5: Fast
FastQC ☑ Include "FastQC" in workflow	6: FastQC on data 3: Webpage 7: FastQC on data 3: RawData
Trimmomatic ☑ Include "Trimmomatic" in workflow	8: Trimmomatic on pre-pro-B chrX/12 regions FASTQ raw

Workflow "RNA-Seq 2 condition QC through Transcript Prediction" created from current history. You can edit or run the workflow.

Let's test the workflow on the exact same inputs that we just analyzed manually

Click the **run** link in the message (or you can also get there by clicking Workflow in the top bar). This brings up a form asking you to define inputs to the workflow. *Set* the first three input datasets for this run to the first three datasets from your current history:

Scroll down to the bottom of the form and *check* **Send results to a new history**. This will send the results of our test to a new history, keeping our existing prototype analysis clean. Give the new history an informative name and *click* **Run workflow.**



Running workflow "RNA-Seq 2 condition QC through Transcript Prediction"

Step 1: Input dataset

Input Dataset: 1: GTF Annotation chrX/12 regions

Step 2: Input dataset

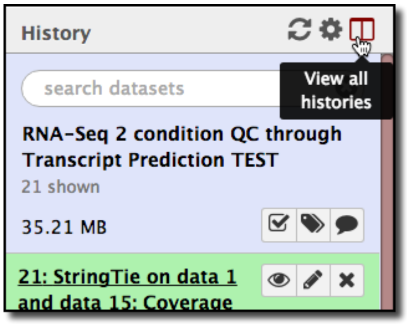
Input Dataset: 2: pre-pro-B chrX/12 regions FASTQ raw

Step 3: Input dataset

Input Dataset: 3: pro-B chrX/12 regions FASTQ raw

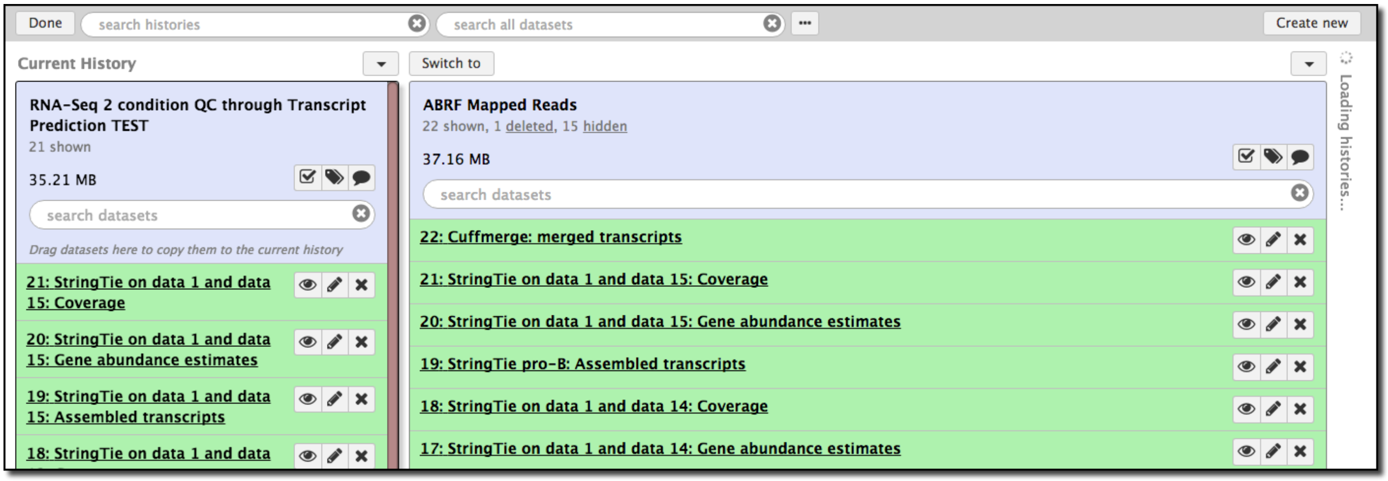
This displays an enormous green box in the middle panel. To get the new history, click on the link to new history, near the top of the box. This takes us to the newly running analysis and a quick check shows that the StringTie coverage and assembled transcript files at least have the same number of entries. It appears (at least at a first glance) that we have successfully created a reusable workflow.

Working with multiple histories



RNA-Seq 2 condition QC through Transcript Prediction TEST
21 shown
35.21 MB
21: StringTie on data 1 and data 15: Coverage

Running the workflow and sending results to a new history means that we now have two histories in this Galaxy instance. There are a couple of ways to see all your existing histories and to switch between them. The most recent and easiest to use is the all histories view. This can be accessed by *clicking* on the **table icon (View all histories)** at the top of the history panel.



RNA-Seq 2 condition QC through Transcript Prediction TEST
21 shown
35.21 MB
Drag datasets here to copy them to the current history
21: StringTie on data 1 and data 15: Coverage
20: StringTie on data 1 and data 15: Gene abundance estimates
19: StringTie on data 1 and data 15: Assembled transcripts
18: StringTie on data 1 and data

ABRF Mapped Reads
22 shown, 1 deleted, 15 hidden
37.16 MB
22: Cuffmerge: merged transcripts
21: StringTie on data 1 and data 15: Coverage
20: StringTie on data 1 and data 15: Gene abundance estimates
19: StringTie pro-B: Assembled transcripts
18: StringTie on data 1 and data 14: Coverage
17: StringTie on data 1 and data 14: Gene abundance estimates

The all histories view presents all your saved histories (we only have two), with the current one pinned at the left, and your other histories listed in reverse chronological order, from left to right. You can make any history the current history by click Switch to at the top of that history. You can also create a new history, and drag and drop datasets from prior histories into the current history. You can search all your histories and datasets for keywords.

Sharing and publishing

Galaxy is also a platform for collaboration and sharing. Galaxy histories, workflows, and visualization can all be shared and published with Galaxy. Sharing in Galaxy means sharing something with someone else either directly with their Galaxy account, or by creating a URL that can be shared.

In addition Galaxy objects can be published, making it easy for anyone to discover the object. Anything that is published will be listed in the appropriate Shared Data section.

Galaxy pages

The datasets and analysis used in the paper (and in this tutorial) are available here:

https://usegalaxy.org/u/thereddylab/p/prediction-of-gene-activity-in-early-b-cell-development-based-on-an-integrative-multi-omics-analysis

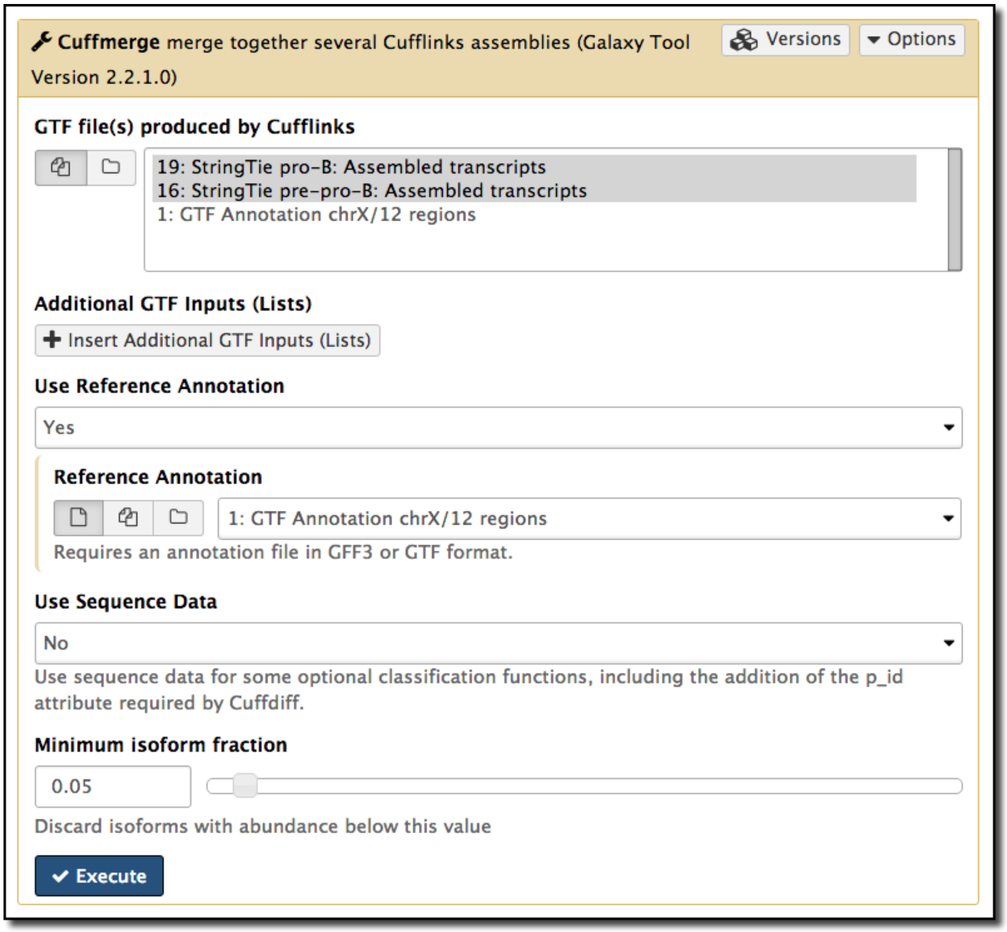
This is a *Galaxy Page*, a document integrated into a Galaxy server that embeds and provides direct links to Galaxy objects such as histories, workflows, and datasets. Galaxy Pages are a way to bundle together all related analysis and to describe the semantics of your analysis. They are most often linked to from the methods sections of papers. As we will see this is just one way that data and analysis can be shared with Galaxy.

Cuffmerge

As a final step, we can unify the reference annotations and the two sets of predicted transcripts into a single, non-redundant set.

Search the tool panel for **Cuffmerge** and *click* on it to launch the Cuffmerge tool form. Note that Cuffmerge has **Multiple datasets** already turned on. Cuffmerge only make sense when merging multiple transcript prediction datasets.

Select both **StringTie Assembled transcripts** files, *change* **Use Reference Annotation** to **Yes**, and *set* the annotation to **1: GTF Annotation chrX/12 regions**. *Click* **Execute**. Finally rename the output dataset to something more meaningful and add it to the existing visualization. You'll note that there has been some, but not much consolidation



Cuffmerge merge together several Cufflinks assemblies (Galaxy Tool Version 2.2.1.0)

GTF file(s) produced by Cufflinks

19: StringTie pro-B: Assembled transcripts
16: StringTie pre-pro-B: Assembled transcripts
1: GTF Annotation chrX/12 regions

Additional GTF Inputs (Lists)

Use Reference Annotation

Yes

Reference Annotation

1: GTF Annotation chrX/12 regions

Requires an annotation file in GFF3 or GTF format.

Use Sequence Data

No

Use sequence data for some optional classification functions, including the addition of the p_id attribute required by Cuffdiff.

Minimum isoform fraction

0.05

Discard isoforms with abundance below this value