EMC Galaxy Training - 1: Introduction to Galaxy

This practical aims to familiarize you with the Galaxy user interface. It will teach you how to perform basic tasks such as uploading data, running tools, working with histories, creating workflows, and sharing data.

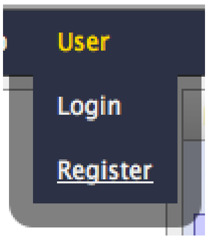
Preparations

1. Open Galaxy
Please open a web browser and navigate to your assigned Galaxy server:

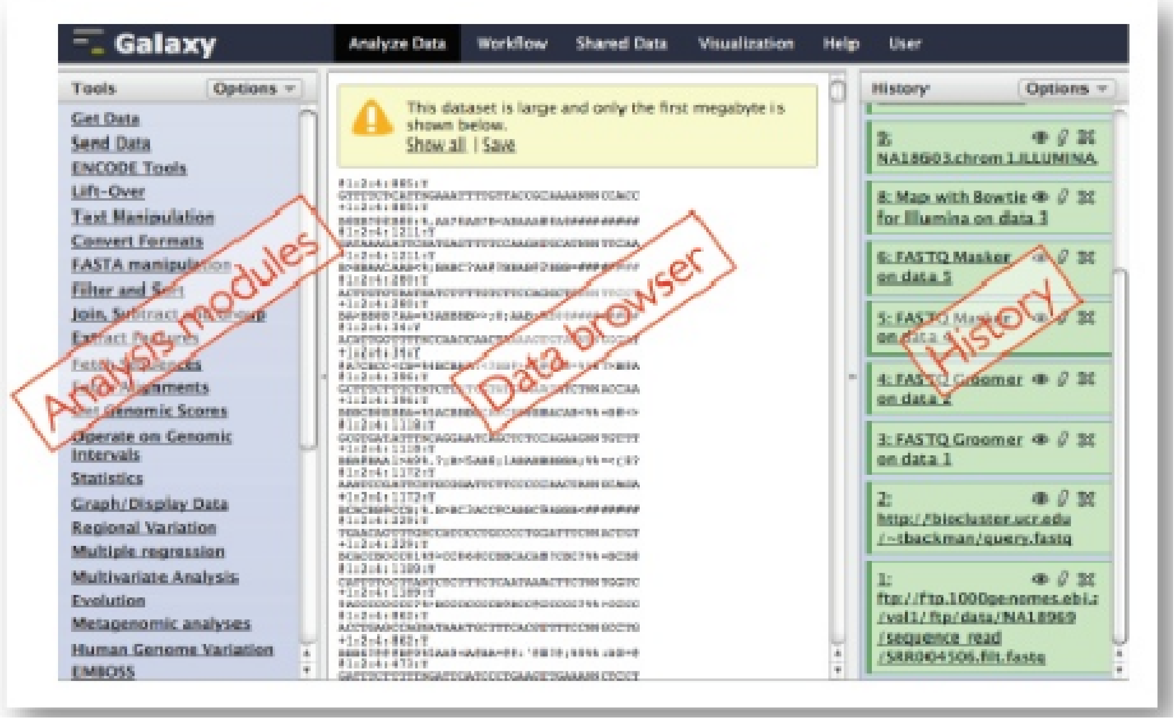
galaxy-training1.trait-ctmm.cloudlet.sara.nl
galaxy-training2.trait-ctmm.cloudlet.sara.nl
galaxy-training3.trait-ctmm.cloudlet.sara.nl
.. etc

It is recommended that you use Firefox or Chrome for this tutorial (not IE)

2. Register for an account
In the top menu bar, go to *User* and then choose *Register*. After registration, click on *Analyze data* in the top menu to return to the main screen.



The main screen consists of three parts, on the left is the list of available tools, on the right side is your *history pane*, showing the analysis you have performed so far, and in the middle you view your tools and data.



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

Follow the steps on the following pages adapted from the *Galaxy 101* course made by the Galaxy Team and one of many tutorials available from the Galaxy Teaching Resources wiki (https://wiki.galaxyproject.org/Teach/Resources).

Galaxy 101: The first thing you should try

In this very simple example we will introduce you to bare basics of Galaxy:

- Getting data from UCSC
- Performing simple data manipulation
- Understanding Galaxy's History system
- Creating and editing workflows
- Applying workflows to your data

You can watch a step-by-step explanation of this entire tutorial here.

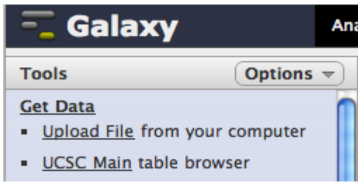
What are we trying to do?

Suppose you get the following question: "Mom (or Dad) ... Which coding exon has the highest number of single nucleotide polymorphisms on chromosome 22?". You think to yourself "Wow! This is a simple question ... I know exactly where the data is (at UCSC) but how do I actually compute this?" The truth is, there is really no straightforward way of answering this question in a time frame comparable to the attention span of a 7-year-old. Well ... actually there is and it is called Galaxy. So let's try it...

1. Getting data from UCSC

1.0. Getting coding exons

First thing we will do is to obtain data from UCSC by clicking "Get Data -> UCSC Main":



You will see Galaxy's middle pane change to looks like this:

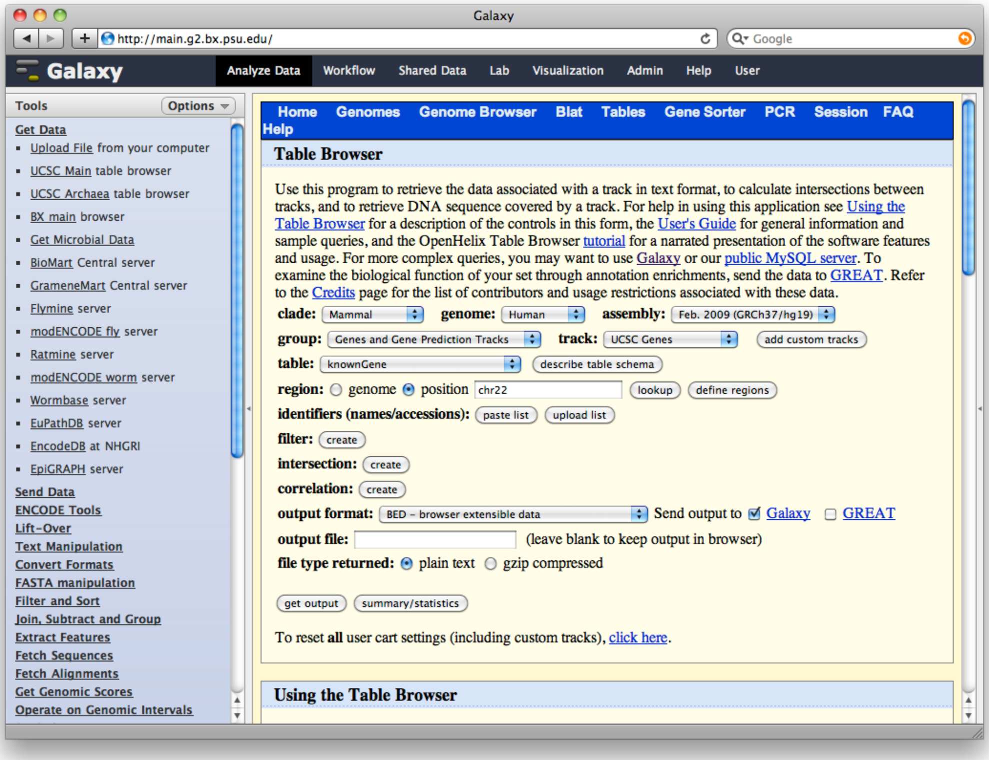


Table Browser

Use this program to retrieve the data associated with a track in text format, to calculate intersections between tracks, and to retrieve DNA sequence covered by a track. For help in using this application see Using the Table Browser for a description of the controls in this form, the User's Guide for general information and sample queries, and the OpenHelix Table Browser tutorial for a narrated presentation of the software features and usage. For more complex queries, you may want to use Galaxy or our public MySQL server. To examine the biological function of your set through annotation enrichments, send the data to GREAT. Refer to the Credits page for the list of contributors and usage restrictions associated with these data.

clade: Mammal **genome:** Human **assembly:** Feb. 2009 (GRCh37/hg19)

group: Genes and Gene Prediction Tracks **track:** UCSC Genes [add custom tracks]

table: knownGene [describe table schema]

region: ○ genome ● position chr22 [lookup] [define regions]

identifiers (names/accessions): [paste list] [upload list]

filter: [create]

intersection: [create]

correlation: [create]

output format: BED - browser extensible data Send output to ☑ Galaxy ☐ GREAT

output file: (leave blank to keep output in browser)

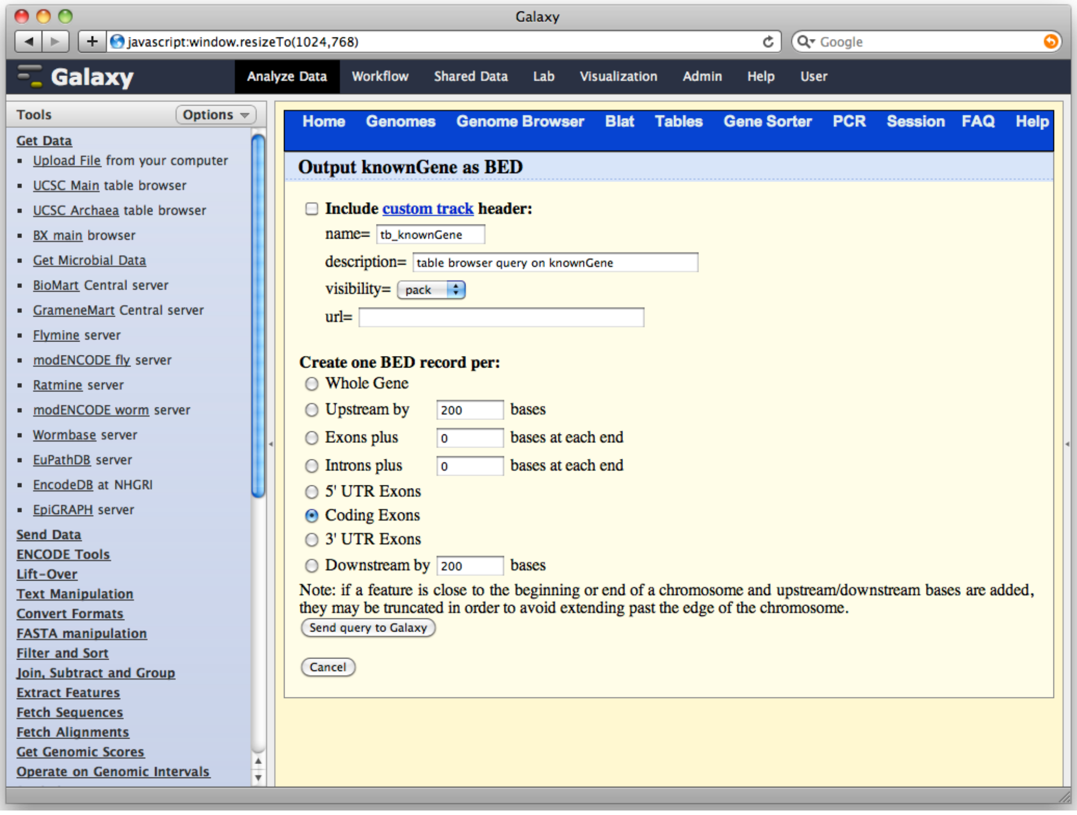
file type returned: ● plain text ○ gzip compressed

[get output] [summary/statistics]

To reset **all** user cart settings (including custom tracks), click here.

Using the Table Browser

Make sure that your settings are exactly the same as shown on the screen (in particular, **position** should be set to "chr22", **output format** should be set to "BED - browser extensible data", and "Galaxy" should be checked by **Send output to** option). Click **get output** and you will see the next screen:



Output knownGene as BED

□ **Include custom track header:**

name= tb_knownGene

description= table browser query on knownGene

visibility= pack

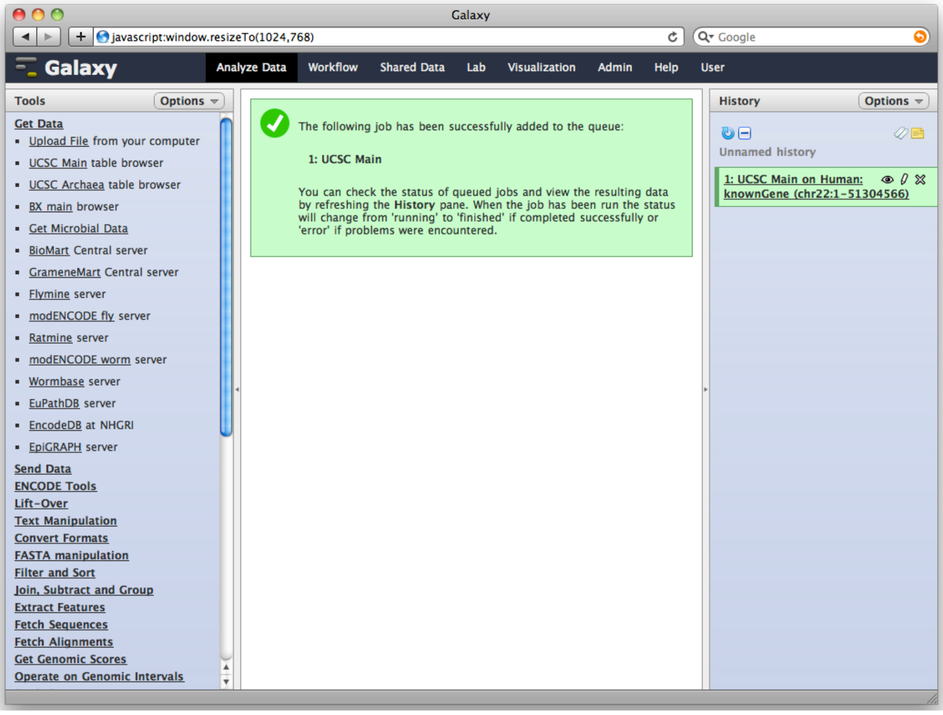
url=

Create one BED record per:

○ Whole Gene
○ Upstream by 200 bases
○ Exons plus 0 bases at each end
○ Introns plus 0 bases at each end
○ 5' UTR Exons
● Coding Exons
○ 3' UTR Exons
○ Downstream by 200 bases

Note: if a feature is close to the beginning or end of a chromosome and upstream/downstream bases are added, they may be truncated in order to avoid extending past the edge of the chromosome.

here make sure **Create one BED record per** is set to "Coding Exons" and click **Send Query to Galaxy**. After this you will see your first History Item in Galaxy's right pane. It will go through gray (preparing) and yellow (running) states to become green:



The following job has been successfully added to the queue:

1: UCSC Main

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.

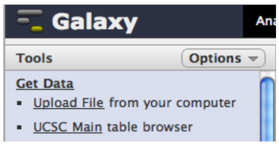
Unnamed history

1: UCSC Main on Human: knownGene (chr22:1-51304566)

To view the contents of the file, click on the eye icon.

1.1. Getting SNPs

Now is the time to obtain SNP data. This is done almost exactly the same way. First thing we will do is to again click on "Get Data -> UCSC Main":



but now change **group** to "Variation" so that the whole page looks like this:

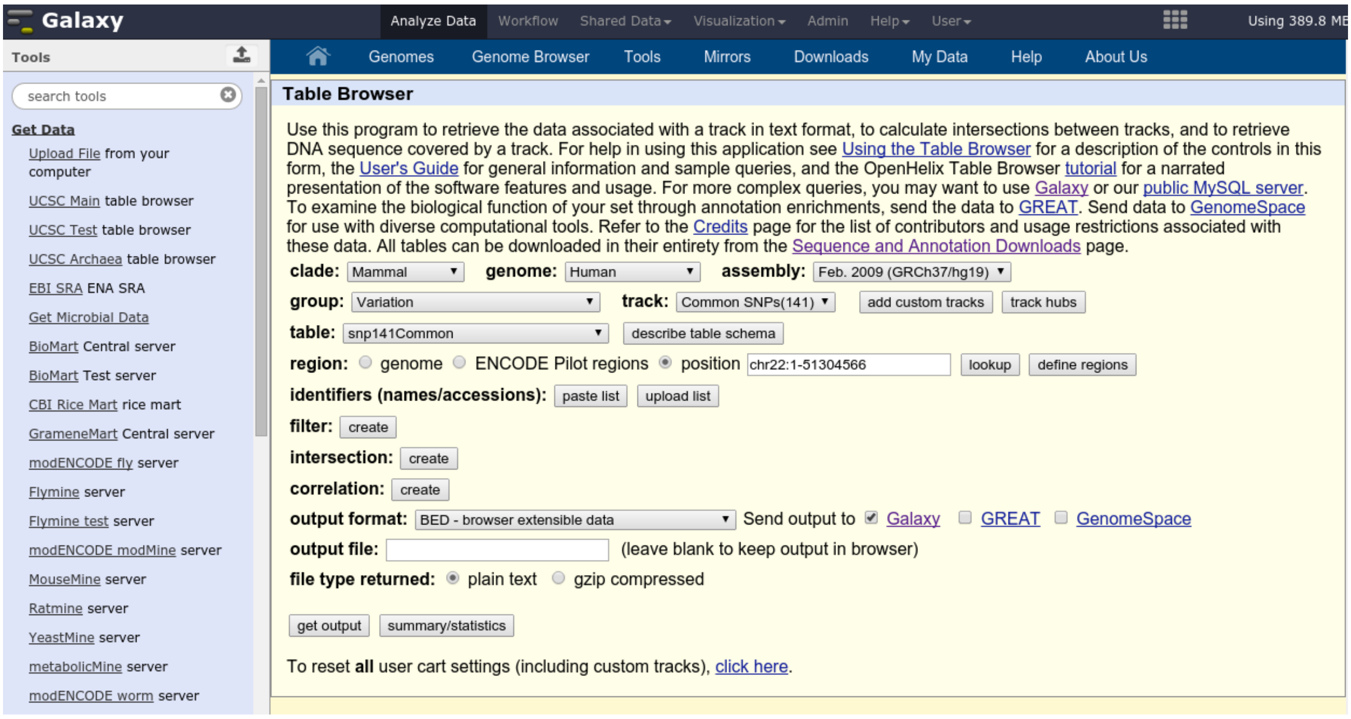


Table Browser

Use this program to retrieve the data associated with a track in text format, to calculate intersections between tracks, and to retrieve DNA sequence covered by a track. For help in using this application see Using the Table Browser for a description of the controls in this form, the User's Guide for general information and sample queries, and the OpenHelix Table Browser tutorial for a narrated presentation of the software features and usage. For more complex queries, you may want to use Galaxy or our public MySQL server. To examine the biological function of your set through annotation enrichments, send the data to GREAT. Send data to GenomeSpace for use with diverse computational tools. Refer to the Credits page for the list of contributors and usage restrictions associated with these data. All tables can be downloaded in their entirety from the Sequence and Annotation Downloads page.

clade: Mammal **genome:** Human **assembly:** Feb. 2009 (GRCh37/hg19)

group: Variation **track:** Common SNPs(141) [add custom tracks] [track hubs]

table: snp141Common [describe table schema]

region: ○ genome ○ ENCODE Pilot regions ● position chr22:1-51304566 [lookup] [define regions]

identifiers (names/accessions): [paste list] [upload list]

filter: [create]

intersection: [create]

correlation: [create]

output format: BED - browser extensible data Send output to ☑ Galaxy □ GREAT □ GenomeSpace

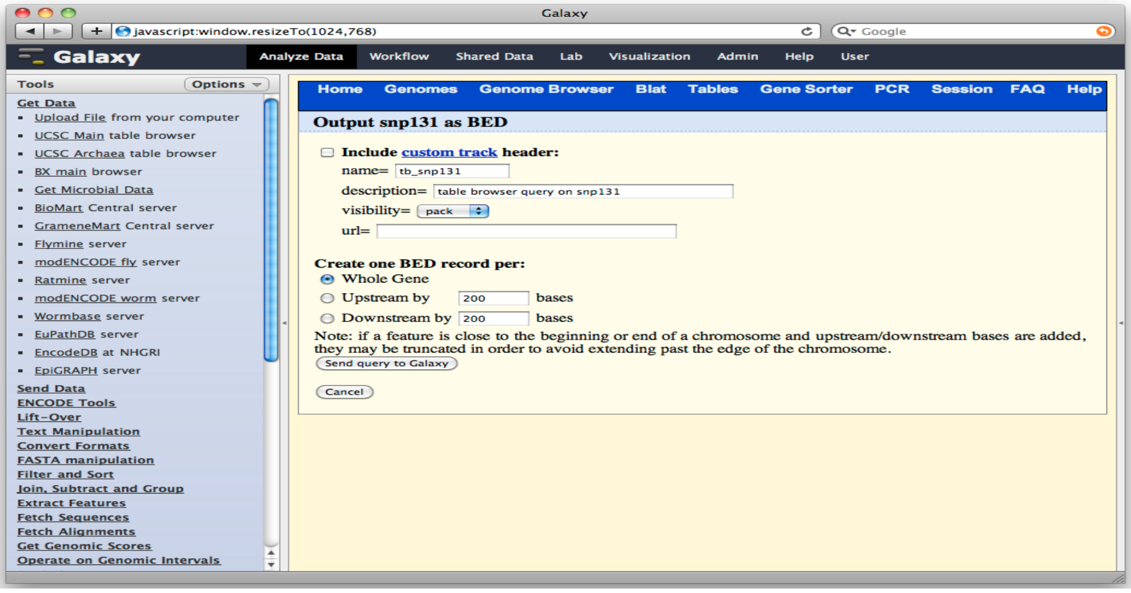
output file: (leave blank to keep output in browser)

file type returned: ● plain text ○ gzip compressed

[get output] [summary/statistics]

To reset **all** user cart settings (including custom tracks), click here.

click **get output** and you should see this:



Output snp131 as BED

□ **Include custom track header:**

name= tb_snp131

description= table browser query on snp131

visibility= pack

url=

Create one BED record per:

● Whole Gene

○ Upstream by 200 bases

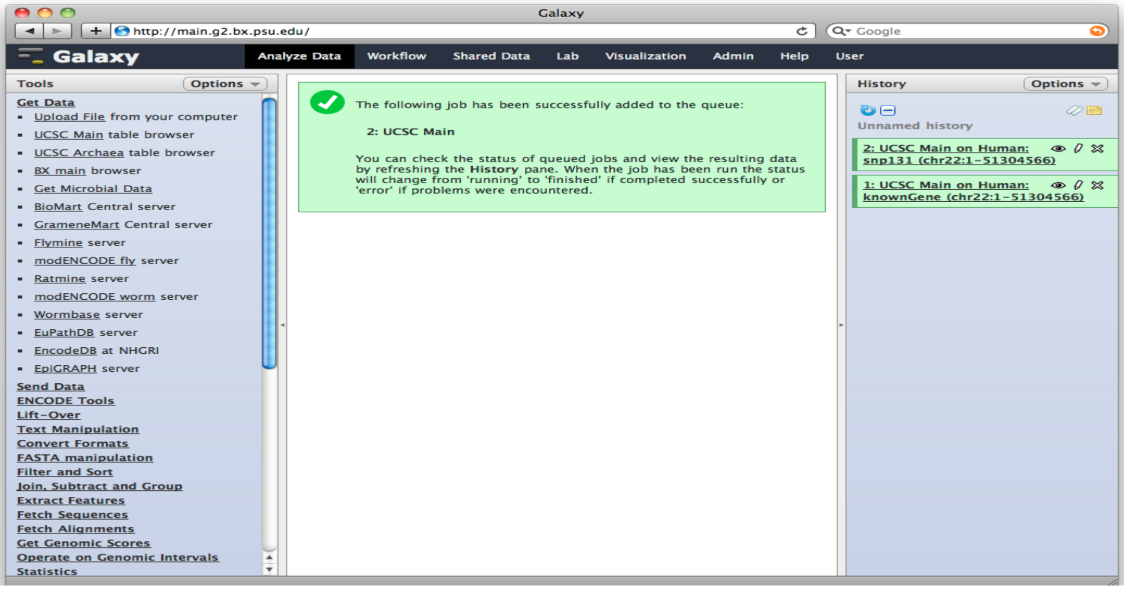
○ Downstream by 200 bases

Note: if a feature is close to the beginning or end of a chromosome and upstream/downstream bases are added, they may be truncated in order to avoid extending past the edge of the chromosome.

[Send query to Galaxy]

[Cancel]

where you need to make sure that **Whole Gene** is selected ("Whole Gene" here really means "Whole Feature") and click **Send Query to Galaxy**. You will get your second item in the history:

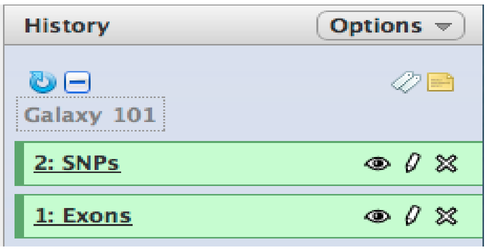


The following job has been successfully added to the queue:

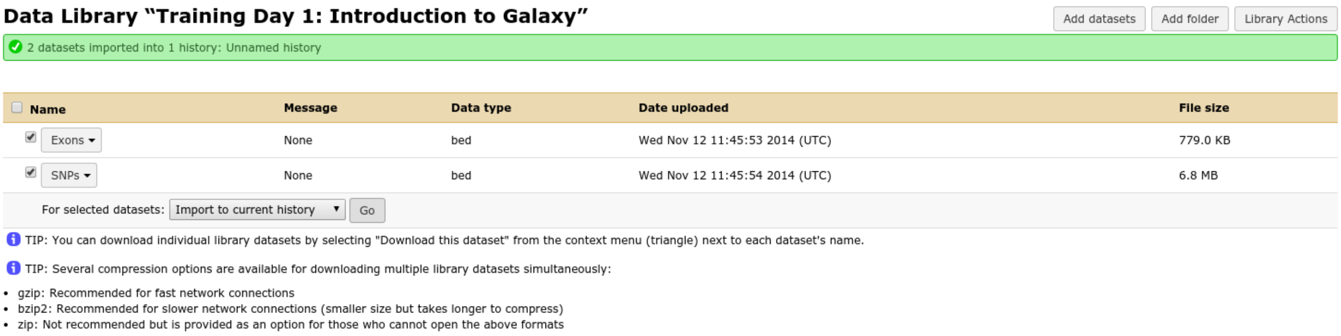
2: UCSC Main

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.

Now we will rename the two history items to "Exons" and "SNPs" by clicking on the Pencil icon adjacent to each item. Also we will rename history to "Galaxy 101" (or whatever you want) by clicking on "Unnamed history" so everything looks like this:



NOTE: If the import from UCSC takes too long, the files can also be found in a shared data library in Galaxy. Browse to “Shared Data → Data Libraries”, select the data library named “Training Day 1: Introduction to Galaxy”, select the files named “SNPs” and “Exons”, choose “Import to current history” and click “go”. Click on “Analyze Data” on the top menu bar to return to your analysis.



Data Library “Training Day 1: Introduction to Galaxy”

✓ 2 datasets imported into 1 history: Unnamed history

□ Name	Message	Data type	Date uploaded	File size
☑ Exons ▾	None	bed	Wed Nov 12 11:45:53 2014 (UTC)	779.0 KB
☑ SNPs ▾	None	bed	Wed Nov 12 11:45:54 2014 (UTC)	6.8 MB

For selected datasets: Import to current history ▾ Go

TIP: You can download individual library datasets by selecting "Download this dataset" from the context menu (triangle) next to each dataset's name.

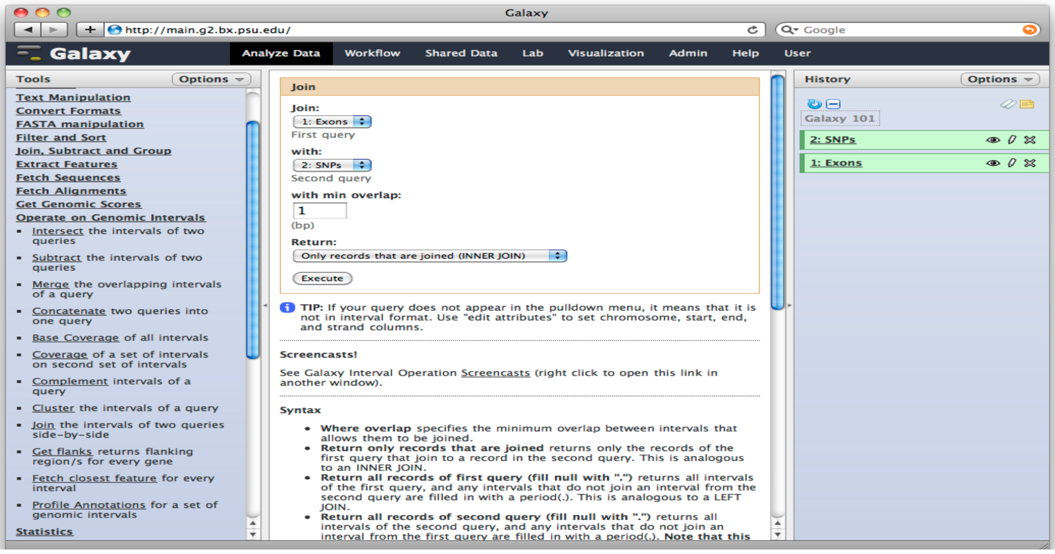
TIP: Several compression options are available for downloading multiple library datasets simultaneously:

- gzip: Recommended for fast network connections
- bzip2: Recommended for slower network connections (smaller size but takes longer to compress)
- zip: Not recommended but is provided as an option for those who cannot open the above formats

2. Finding Exons with the highest number of SNPs

2.0. Joining exons with SNPs

Let's remind ourselves that our objective was to find which exon contains the most SNPs. This first step in answering this question will be joining exons with SNPs (a fancy word for printing exons and SNPs that overlap side by side). This is done using the tool "Join, Subtract and Group -> Join the intervals of two datasets side-by-side"


Galaxy
http://main.g2.bx.psu.edu/
Google
Galaxy
Analyze Data
Workflow
Shared Data
Lab
Visualization
Admin
Help
User
Tools
Options
Text Manipulation
Convert Formats
FASTA manipulation
Filter and Sort
Join, Subtract and Group
Extract Features
Fetch Sequences
Fetch Alignments
Get Genomic Scores
Operate on Genomic Intervals
Intersect the intervals of two queries
Subtract the intervals of two queries
Merge the overlapping intervals of a query
Concatenate two queries into one query
Base Coverage of all intervals
Coverage of a set of intervals on second set of intervals
Complement intervals of a query
Cluster the intervals of a query
Join the intervals of two queries side-by-side
Get flanks returns flanking region/s for every gene
Fetch closest feature for every interval
Profile Annotations for a set of genomic intervals
Statistics
History
Options
Galaxy 101
2: SNPs
1: Exons

Join

Join: 1: Exons
First query

with: 2: SNPs
Second query

with min overlap: 1
(bp)

Return: Only records that are joined (INNER JOIN)

Execute

TIP: If your query does not appear in the pulldown menu, it means that it is not in interval format. Use "edit attributes" to set chromosome, start, end, and strand columns.

Screencasts!

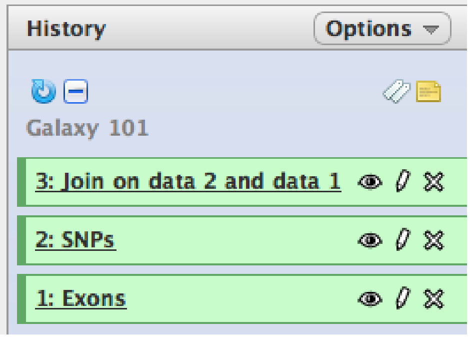
See Galaxy Interval Operation Screencasts (right click to open this link in another window).

Syntax

- **Where overlap** specifies the minimum overlap between intervals that allows them to be joined.
- **Return only records that are joined** returns only the records of the first query that join to a record in the second query. This is analogous to an INNER JOIN.
- **Return all records of first query (fill null with ".")** returns all intervals of the first query, and any intervals that do not join an interval from the second query are filled in with a period(.). This is analogous to a LEFT JOIN.
- **Return all records of second query (fill null with ".")** returns all intervals of the second query, and any intervals that do not join an interval from the first query are filled in with a period(.). **Note that this**

Note that if you scroll down on this page, you will find an explanation about the tool.

make sure your exons are first and SNPs are second and click **Execute**. You will get the third history item:



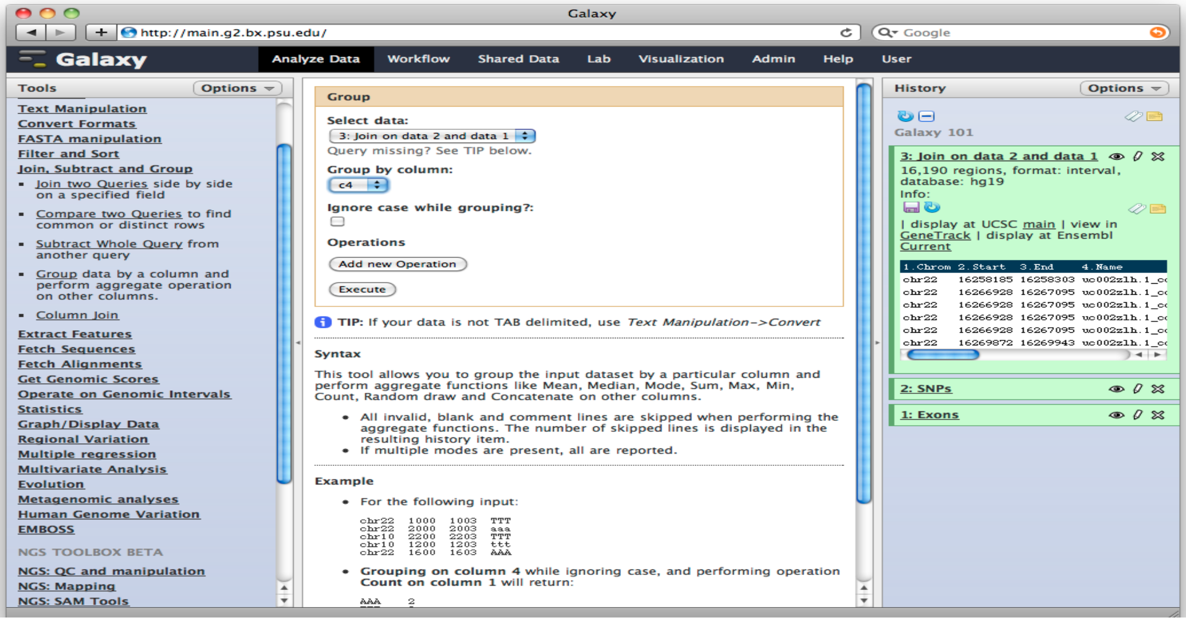
which will contain the following data:

```
…
chr22 16277747 16277885 uc002zlj.1_cds_4_0_chr22_16277748_r 0 - chr22 16277851 16277852 rs200742649 0 +
chr22 16287253 16287390 uc002zlj.1_cds_8_0_chr22_16287254_r 0 - chr22 16287338 16287339 rs199952431 0 +
chr22 16287253 16287390 uc002zlj.1_cds_8_0_chr22_16287254_r 0 - chr22 16287345 16287346 rs200013113 0 +
chr22 16287253 16287390 uc002zlj.1_cds_8_0_chr22_16287254_r 0 - chr22 16287371 16287372 rs201840700 0 +
chr22 16448823 16449804 uc011agd.2_cds_0_0_chr22_16448824_r 0 - chr22 16449404 16449405 rs201956705 0 +
…
```

Let's take a look at this dataset. The first six columns correspond to exons. The last six correspond to SNPs. You can see that exon with ID `uc002zlj.1_cds_8_0_chr22_16287254_r` contains three SNPs with IDs rs199952431, rs200073113, and rs201840700.

2.1. Counting the number of SNPs per exon

Above we've seen that exon `uc002zlj.1_cds_8_0_chr22_16287254_r` is repeated three times in the above dataset. Thus we can easily compute the number of SNPs per exon by simply counting the number of repetitions of name for each exon. This can be easily done with the "Join, Subtract, and Group -> Group" tool:



- Join two Queries side by side on a specified field
- Compare two Queries to find common or distinct rows
- Subtract Whole Query from another query
- Group data by a column and perform aggregate operation on other columns.
- Column Join

Group

Select data: 3: Join on data 2 and data 1

Query missing? See TIP below.

Group by column: c4

Ignore case while grouping?: □

Operations

Add new Operation

Execute

TIP: If your data is not TAB delimited, use *Text Manipulation->Convert*

Syntax

This tool allows you to group the input dataset by a particular column and perform aggregate functions like Mean, Median, Mode, Sum, Max, Min, Count, Random draw and Concatenate on other columns.

- All invalid, blank and comment lines are skipped when performing the aggregate functions. The number of skipped lines is displayed in the resulting history item.
- If multiple modes are present, all are reported.

Example

- For the following input:

```
chr22  1000  1003  TTT
chr22  2000  2003  aaa
chr10  2200  2203  TTT
chr10  1200  1203  ttt
chr22  1600  1603  AAA
```

- **Grouping on column 4** while ignoring case, and performing operation **Count on column 1** will return:

```
AAA    2
```

3: Join on data 2 and data 1

16,190 regions, format: interval, database: hg19

Info:

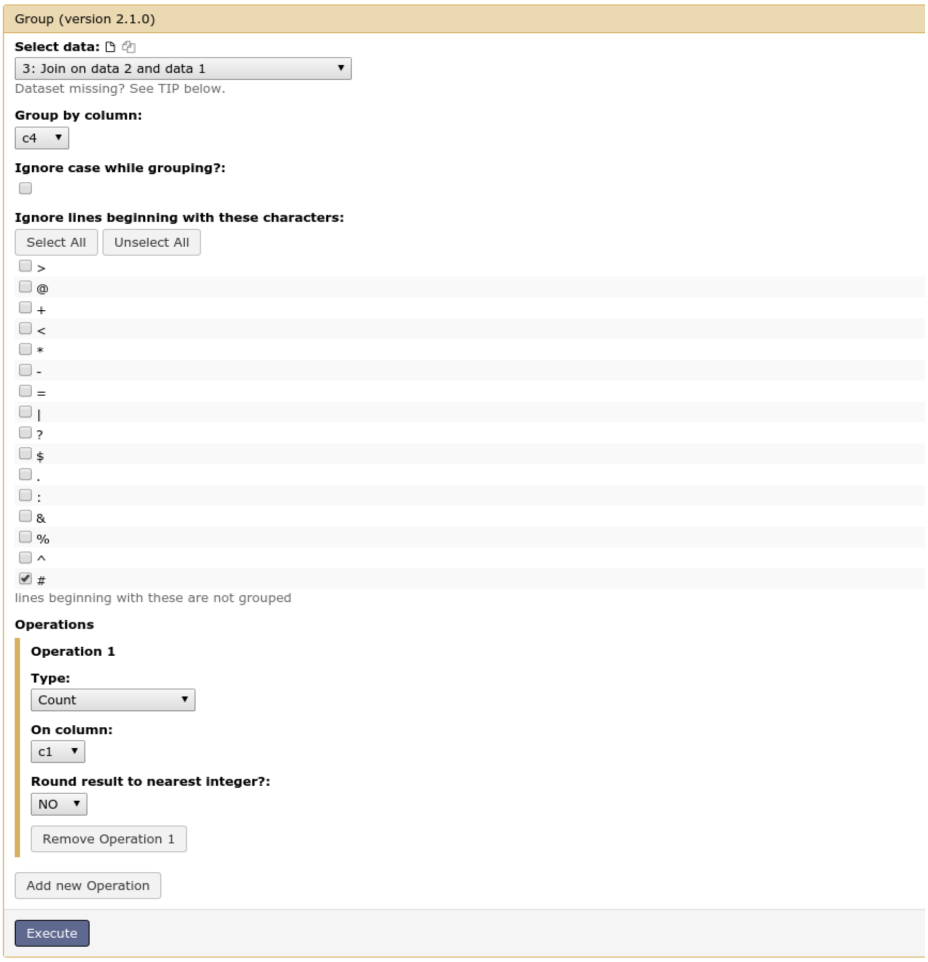
| display at UCSC main | view in GeneTrack | display at Ensembl Current

```
1.Chrom  2.Start   3.End     4.Name
chr22    16258185  16258303  uc002zlh.1_cc
chr22    16266928  16267095  uc002zlh.1_cc
chr22    16266928  16267095  uc002zlh.1_cc
chr22    16266928  16267095  uc002zlh.1_cc
chr22    16266928  16267095  uc002zlh.1_cc
chr22    16269872  16269943  uc002zlh.1_cc
```

2: SNPs

1: Exons

choose column 4 by selecting "c4" in **Group by column**. Then click on **Add new Operation** and select a counting operation on column c4. Set it to ignore lines beginning with #. Make sure the interface looks exactly as shown below:



Group (version 2.1.0)

Select data: 3: Join on data 2 and data 1

Dataset missing? See TIP below.

Group by column: c4

Ignore case while grouping?: □

Ignore lines beginning with these characters:

Select All | Unselect All

□ >
□ @
□ +
□ <
□ *
□ -
□ =
□ |
□ ?
□ $
□ .
□ :
□ &
□ %
□ ^
☑ #

lines beginning with these are not grouped

Operations

Operation 1

Type: Count

On column: c1

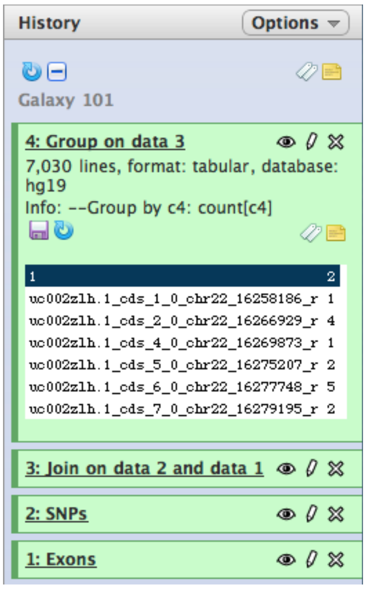
Round result to nearest integer?: NO

Remove Operation 1

Add new Operation

Execute

click **Execute**. Your history will look like this:



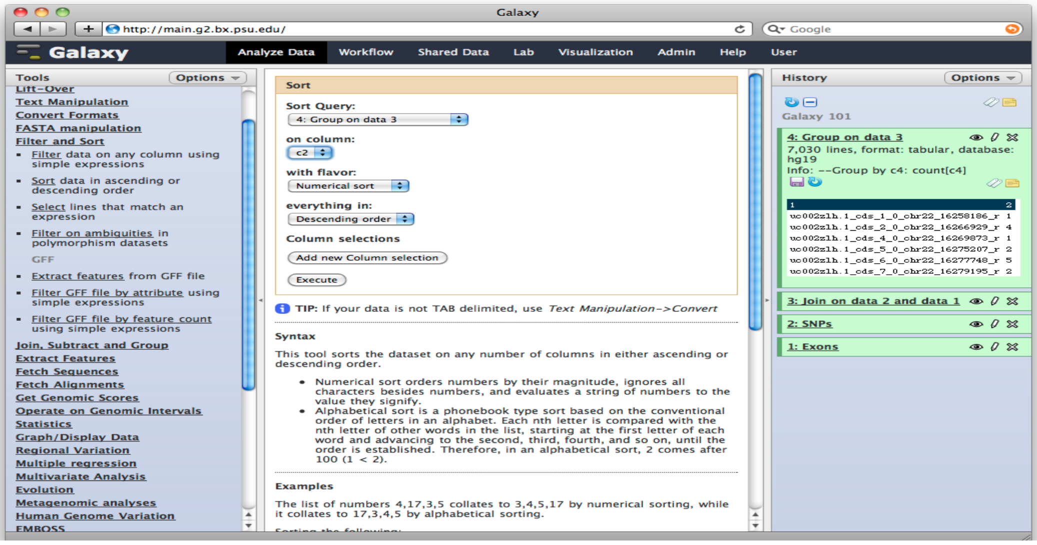
4: Group on data 3
7,030 lines, format: tabular, database: hg19
Info: --Group by c4: count[c4]

```
1                                  2
uc002zlh.1_cds_1_0_chr22_16258186_r 1
uc002zlh.1_cds_2_0_chr22_16266929_r 4
uc002zlh.1_cds_4_0_chr22_16269873_r 1
uc002zlh.1_cds_5_0_chr22_16275207_r 2
uc002zlh.1_cds_6_0_chr22_16277748_r 5
uc002zlh.1_cds_7_0_chr22_16279195_r 2
```

if you look at the above image you will see that the result of grouping (dataset #4) contains two columns. This first contains the exon name while the second shows the number of times this name has been repeated in dataset #3.

2.3. Sorting exons by SNP count

To see which exon has the highest number of SNPs we can simply sort the dataset #4 on the second column in descending order. This is done with "Filter and Sort -> Sort":



Sort

Sort Query: 4: Group on data 3

on column: c2

with flavor: Numerical sort

everything in: Descending order

Column selections

TIP: If your data is not TAB delimited, use *Text Manipulation->Convert*

Syntax

This tool sorts the dataset on any number of columns in either ascending or descending order.

- Numerical sort orders numbers by their magnitude, ignores all characters besides numbers, and evaluates a string of numbers to the value they signify.
- Alphabetical sort is a phonebook type sort based on the conventional order of letters in an alphabet. Each nth letter is compared with the nth letter of other words in the list, starting at the first letter of each word and advancing to the second, third, fourth, and so on, until the order is established. Therefore, in an alphabetical sort, 2 comes after 100 (1 < 2).

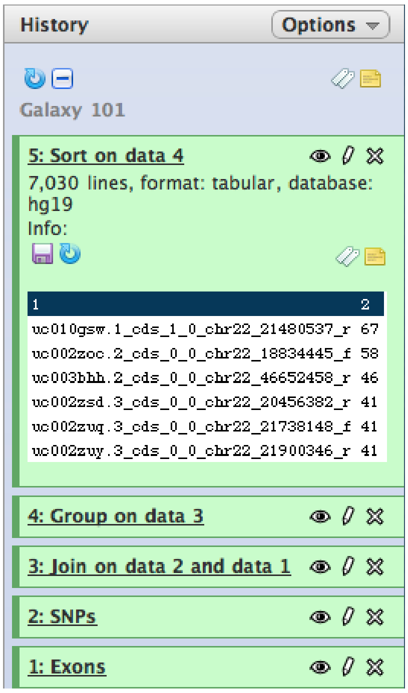
Examples

The list of numbers 4,17,3,5 collates to 3,4,5,17 by numerical sorting, while it collates to 17,3,4,5 by alphabetical sorting.

4: Group on data 3
7,030 lines, format: tabular, database: hg19
Info: --Group by c4: count[c4]

```
1                                  2
uc002zlh.1_cds_1_0_chr22_16258186_r 1
uc002zlh.1_cds_2_0_chr22_16266929_r 4
uc002zlh.1_cds_4_0_chr22_16269873_r 1
uc002zlh.1_cds_5_0_chr22_16275207_r 2
uc002zlh.1_cds_6_0_chr22_16277748_r 5
uc002zlh.1_cds_7_0_chr22_16279195_r 2
```

This will generate the fifth history item:



5: Sort on data 4
7,030 lines, format: tabular, database: hg19
Info:

```
1                                  2
uc010gsw.1_cds_1_0_chr22_21480537_r 67
uc002zoc.2_cds_0_0_chr22_18834445_f 58
uc003bhh.2_cds_0_0_chr22_46652458_r 46
uc002zsd.3_cds_0_0_chr22_20456382_r 41
uc002zuq.3_cds_0_0_chr22_21738148_f 41
uc002zuy.3_cds_0_0_chr22_21900346_r 41
```

4: Group on data 3

3: Join on data 2 and data 1

2: SNPs

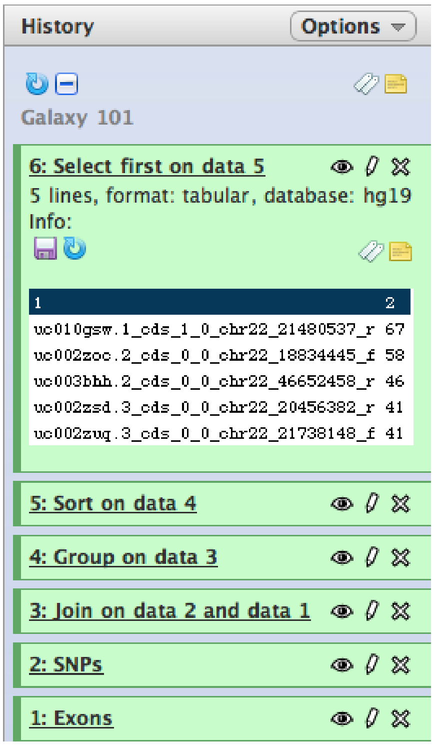
1: Exons

and you can now see that the highest number of SNPs per exon is 30.

2.4. Selecting top five

Now let's select top five exons with the highest number of SNPs. For this we will use "Text Manipulation -> Select First" tool:

Clicking **Execute** will produce the sixth history item that will contain just five lines:



6: Select first on data 5
5 lines, format: tabular, database: hg19
Info:

```
1                                  2
uc010gsw.1_cds_1_0_chr22_21480537_r 67
uc002zoc.2_cds_0_0_chr22_18834445_f 58
uc003bhh.2_cds_0_0_chr22_46652458_r 46
uc002zsd.3_cds_0_0_chr22_20456382_r 41
uc002zuq.3_cds_0_0_chr22_21738148_f 41
```

5: Sort on data 4

4: Group on data 3

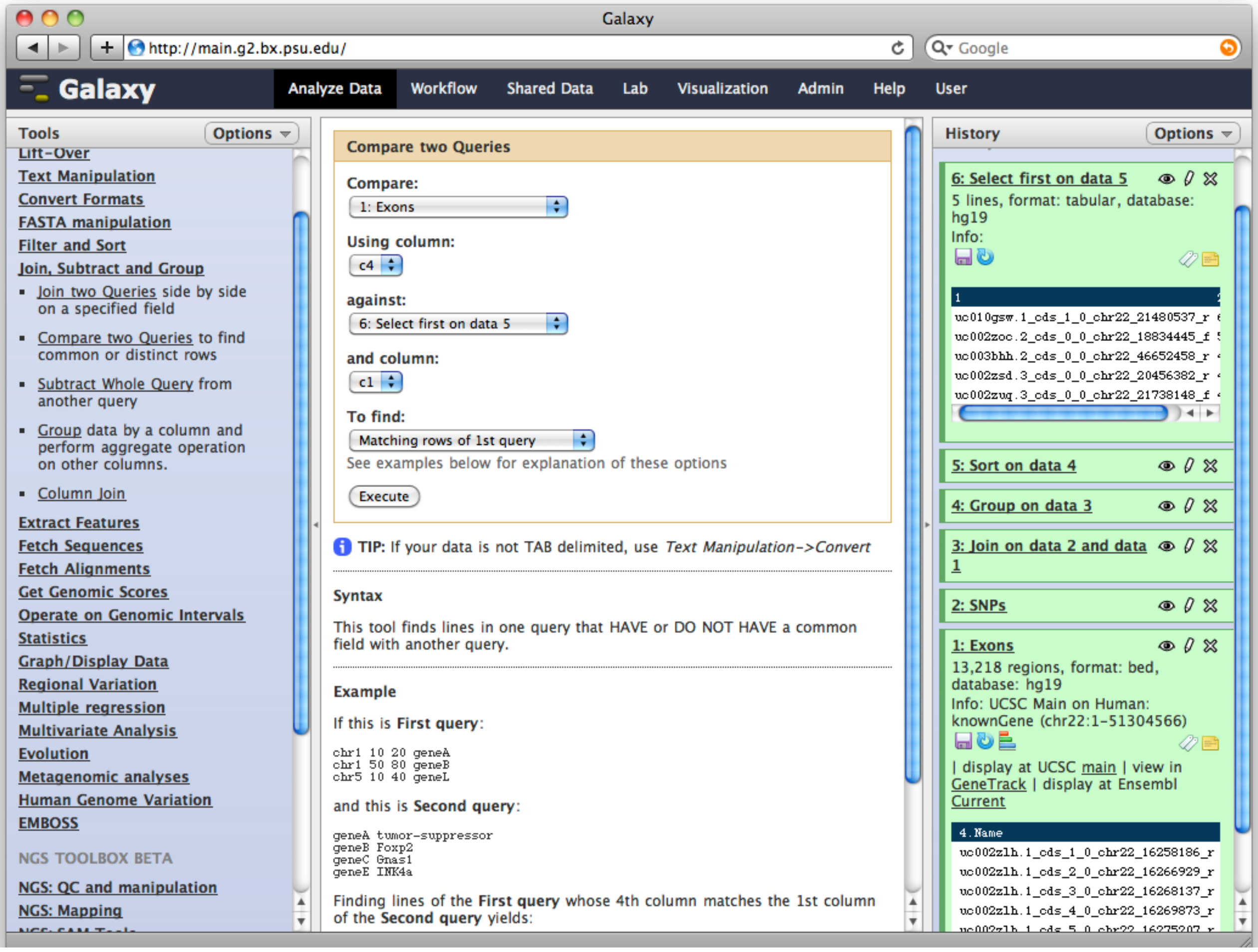
3: Join on data 2 and data 1

2: SNPs

1: Exons

2.5. Recovering exon info and displaying data in genome browsers

Now we know that in this dataset the five top exons contain between 16 and 30 SNPs. But what else can we learn about these? To know more we need to get back the positional information (coordinates) of these exons. This information was lost at the grouping step and now all we have is just two columns. To get coordinates back we will match the names of exons in dataset #6 (column 1) against names of the exons in the original dataset #1 (column 4). This can be done with "Join, Subtract and Group -> Compare two Datasets" tool (note the settings of the tool in the middle pane):



Compare two Queries

Compare: 1: Exons

Using column: c4

against: 6: Select first on data 5

and column: c1

To find: Matching rows of 1st query

See examples below for explanation of these options

TIP: If your data is not TAB delimited, use *Text Manipulation->Convert*

Syntax

This tool finds lines in one query that HAVE or DO NOT HAVE a common field with another query.

Example

If this is **First query**:

```
chr1  10  20  geneA
chr1  50  80  geneB
chr5  10  40  geneL
```

and this is **Second query**:

```
geneA tumor-suppressor
geneB Foxp2
geneC Gnas1
geneE INK4a
```

Finding lines of the **First query** whose 4th column matches the 1st column of the **Second query** yields:

6: Select first on data 5

5 lines, format: tabular, database: hg19

Info:

```
1
uc010gsw.1_cds_1_0_chr22_21480537_r
uc002zoc.2_cds_0_0_chr22_18834445_f
uc003bhh.2_cds_0_0_chr22_46652458_r
uc002zsd.3_cds_0_0_chr22_20456382_r
uc002zuq.3_cds_0_0_chr22_21738148_f
```

5: Sort on data 4

4: Group on data 3

3: Join on data 2 and data 1

2: SNPs

1: Exons

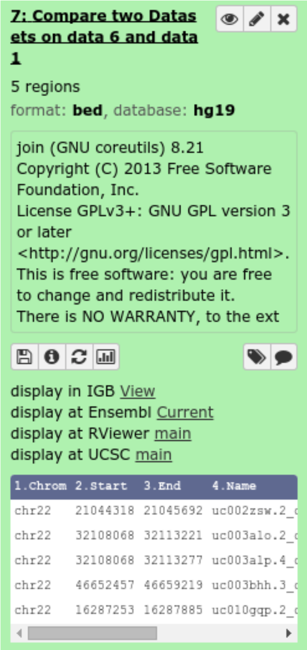
13,218 regions, format: bed, database: hg19

Info: UCSC Main on Human: knownGene (chr22:1-51304566)

| display at UCSC main | view in GeneTrack | display at Ensembl Current

```
4.Name
uc002zlh.1_cds_1_0_chr22_16258186_r
uc002zlh.1_cds_2_0_chr22_16266929_r
uc002zlh.1_cds_3_0_chr22_16268137_r
uc002zlh.1_cds_4_0_chr22_16269873_r
uc002zlh.1_cds_5_0_chr22_16275207_r
```

this adds the seventh dataset to the history:



7: Compare two Datasets on data 6 and data 1

5 regions

format: **bed**, database: **hg19**

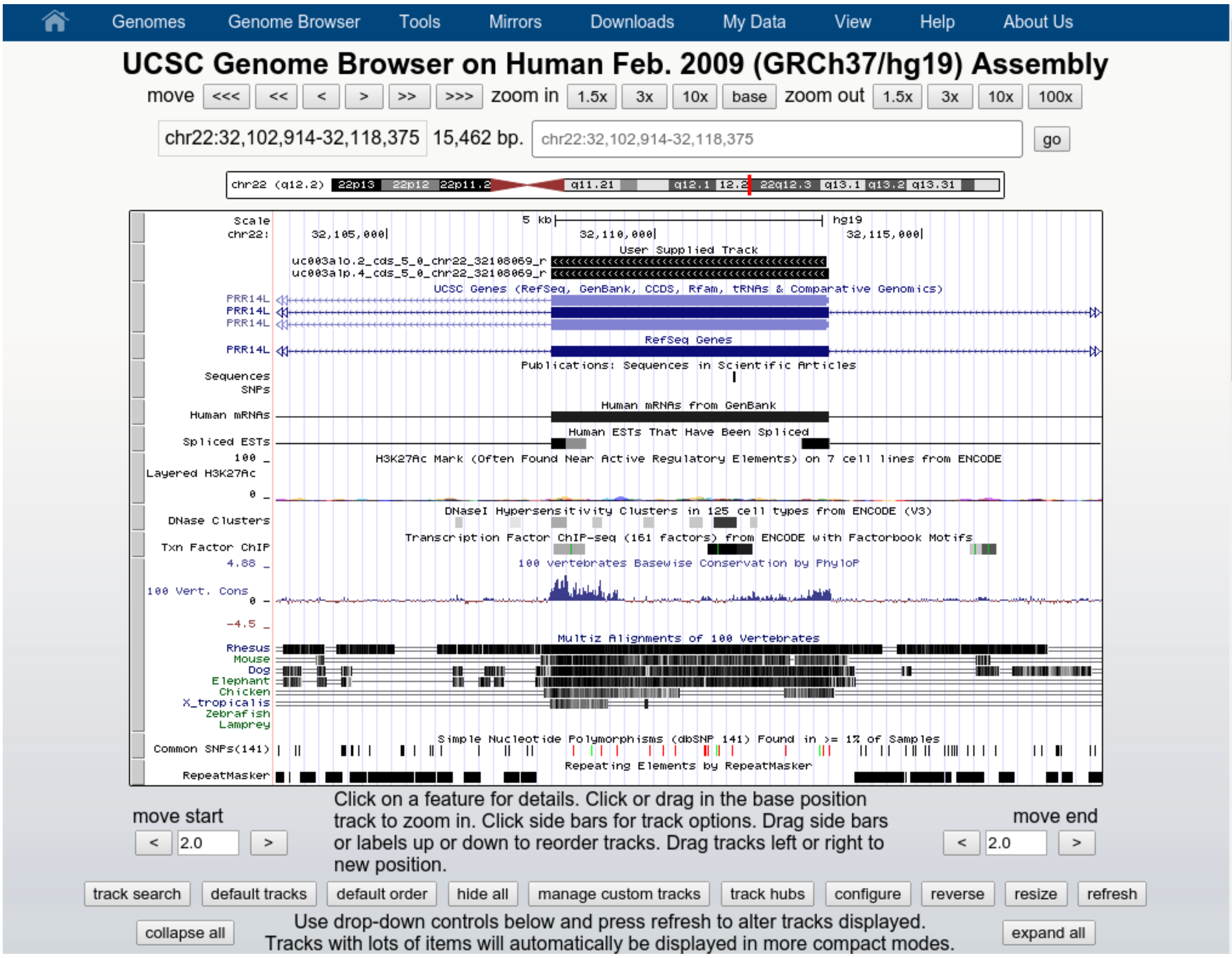
```
join (GNU coreutils) 8.21
Copyright (C) 2013 Free Software
Foundation, Inc.
License GPLv3+: GNU GPL version 3
or later
<http://gnu.org/licenses/gpl.html>.
This is free software: you are free
to change and redistribute it.
There is NO WARRANTY, to the ext
```

display in IGB View
display at Ensembl Current
display at RViewer main
display at UCSC main

1.Chrom	2.Start	3.End	4.Name
chr22	21044318	21045692	uc002zsw.2_c
chr22	32108068	32113221	uc003alo.2_c
chr22	32108068	32113277	uc003alp.4_c
chr22	46652457	46659219	uc003bhh.3_c
chr22	16287253	16287885	uc010gqp.2_c

The best way to learn about these exons is to look at their genomic surrounding. There is really no better way to do this than using genome browsers. For example, clicking on **"display at UCSC main"** will show this (to see your regions look at "User Supplied Track" (track near the top)):

Enter coordinates: **chr22:32,102,914-32,118,375** to get the same view as in the screenshot below. This centers around the second exon in your top-5 list.



UCSC Genome Browser on Human Feb. 2009 (GRCh37/hg19) Assembly

chr22:32,102,914-32,118,375 15,462 bp.

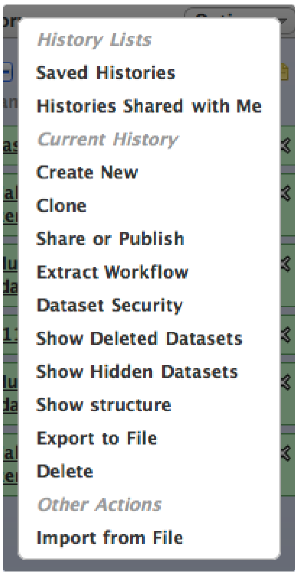
Click on a feature for details. Click or drag in the base position track to zoom in. Click side bars for track options. Drag side bars or labels up or down to reorder tracks. Drag tracks left or right to new position.

Use drop-down controls below and press refresh to alter tracks displayed. Tracks with lots of items will automatically be displayed in more compact modes.

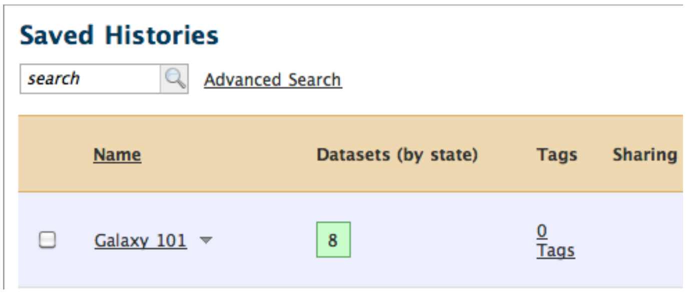
Genomes
Genome Browser
Tools
Mirrors
Downloads
My Data
View
Help
About Us
move <<< << < > >> >>>
zoom in 1.5x 3x 10x base
zoom out 1.5x 3x 10x 100x
go
chr22 (q12.2)
Scale
chr22:
5 kb
hg19
32,105,000
32,110,000
32,115,000
User Supplied Track
uc003alo.2_cds_5_0_chr22_32108069_r
uc003alp.4_cds_5_0_chr22_32108069_r
UCSC Genes (RefSeq, GenBank, CCDS, Rfam, tRNAs & Comparative Genomics)
PRR14L
RefSeq Genes
Publications: Sequences in Scientific Articles
Sequences
SNPs
Human mRNAs from GenBank
Human mRNAs
Human ESTs That Have Been Spliced
Spliced ESTs
H3K27Ac Mark (Often Found Near Active Regulatory Elements) on 7 cell lines from ENCODE
Layered H3K27Ac
DNaseI Hypersensitivity Clusters in 125 cell types from ENCODE (V3)
DNase Clusters
Transcription Factor ChIP-seq (161 factors) from ENCODE with Factorbook Motifs
Txn Factor ChIP
100 vertebrates Basewise Conservation by PhyloP
100 Vert. Cons
Multiz Alignments of 100 Vertebrates
Rhesus
Mouse
Dog
Elephant
Chicken
X_tropicalis
Zebrafish
Lamprey
Simple Nucleotide Polymorphisms (dbSNP 141) Found in >= 1% of Samples
Common SNPs(141)
Repeating Elements by RepeatMasker
RepeatMasker
move start
move end
2.0
track search
default tracks
default order
hide all
manage custom tracks
track hubs
configure
reverse
resize
refresh
collapse all
expand all

3. Understanding histories

In Galaxy your analyses live in histories such as your current one. Histories can be very large, you can have as many histories as you want, and all history behaviour is controlled by the **Options** button on the top of the History pane (gear symbol):



If you create a new history, your current history does not disappear. If you would like to list all of your histories just choose Saved Histories and you will see a list of all your histories in the center pane:



Saved Histories

search Advanced Search

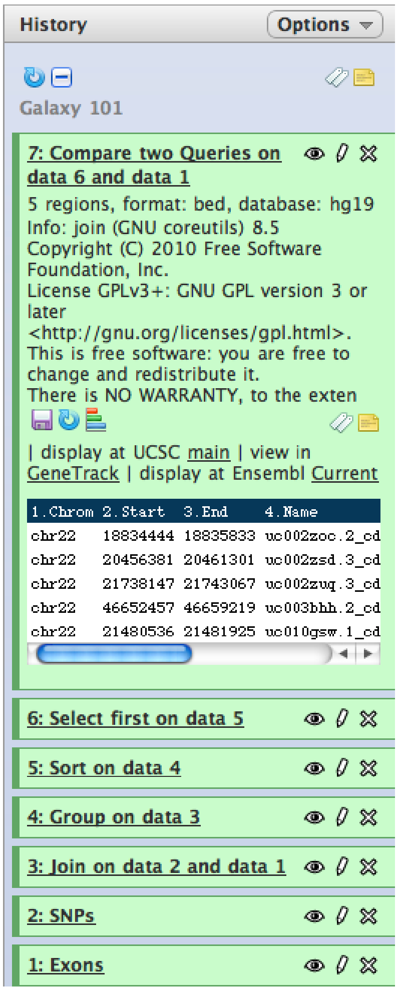
	Name	Datasets (by state)	Tags	Sharing
□	Galaxy 101	8	0 Tags	

4. Converting histories into workflows

One of the history options listed above is very special. It allows you to easily convert existing histories into analysis workflows. Why would you want to create a workflows out of a history? To redo the analysis again with minimal clicking.

4.0. Extracting workfklow

Lets take a look at the history again:



Galaxy 101

7: Compare two Queries on data 6 and data 1

5 regions, format: bed, database: hg19
Info: join (GNU coreutils) 8.5
Copyright (C) 2010 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later
<http://gnu.org/licenses/gpl.html>.
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the exten

| display at UCSC main | view in GeneTrack | display at Ensembl Current

1.Chrom	2.Start	3.End	4.Name
chr22	18834444	18835833	uc002zoc.2_cd
chr22	20456381	20461301	uc002zsd.3_cd
chr22	21738147	21743067	uc002zuq.3_cd
chr22	46652457	46659219	uc003bhh.2_cd
chr22	21480536	21481925	uc010gsw.1_cd

6: Select first on data 5

5: Sort on data 4

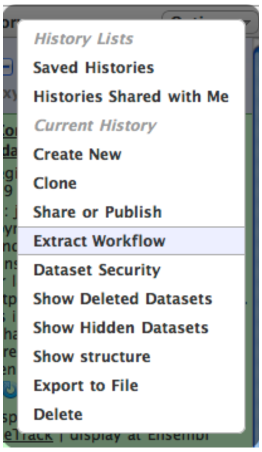
4: Group on data 3

3: Join on data 2 and data 1

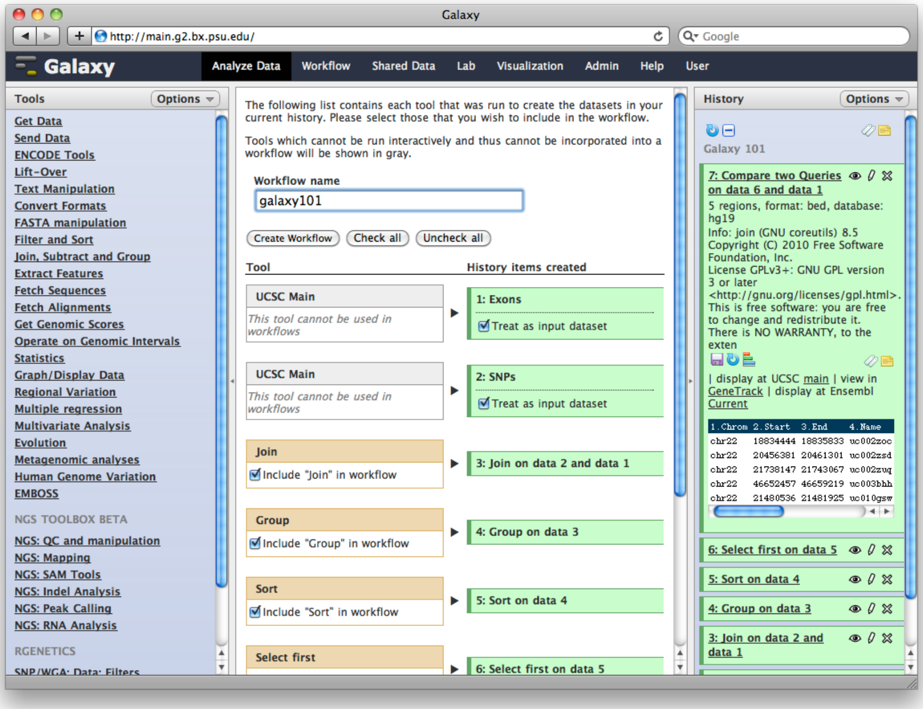
2: SNPs

1: Exons

You can see that this history contains all steps of our analysis. So by building this history we have actually built a complete record of our analysis with Galaxy preserving all parameter settings applied at every step. Wouldn't it be nice to just convert this history into a workflow that we'll be able to execute again and again? This can be done by clicking on **Options** (gear) button and selecting **Extract Workflow** option:



The center pane will change as shown below and you will be able to choose which steps to include/exclude and how to name the newly created workflow. In this case I named it "galaxy101":



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: galaxy101

Create Workflow | Check all | Uncheck all

Tool		History items created
UCSC Main *This tool cannot be used in workflows*	▶	1: Exons ☑ Treat as input dataset
UCSC Main *This tool cannot be used in workflows*	▶	2: SNPs ☑ Treat as input dataset
Join ☑ Include "Join" in workflow	▶	3: Join on data 2 and data 1
Group ☑ Include "Group" in workflow	▶	4: Group on data 3
Sort ☑ Include "Sort" in workflow	▶	5: Sort on data 4
Select first	▶	6: Select first on data 5

History — Options

Galaxy 101

7: Compare two Queries on data 6 and data 1

5 regions, format: bed, database: hg19
Info: join (GNU coreutils) 8.5
Copyright (C) 2010 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>.
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the exten

| display at UCSC main | view in GeneTrack | display at Ensembl Current

1.Chrom	2.Start	3.End	4.Name
chr22	18834444	18835833	uc002zoc
chr22	20456381	20461301	uc002zsd
chr22	21738147	21743067	uc002zuq
chr22	46652457	46659219	uc003bhh
chr22	21480536	21481925	uc010gsw

6: Select first on data 5

5: Sort on data 4

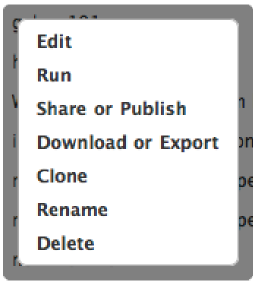
4: Group on data 3

3: Join on data 2 and data 1

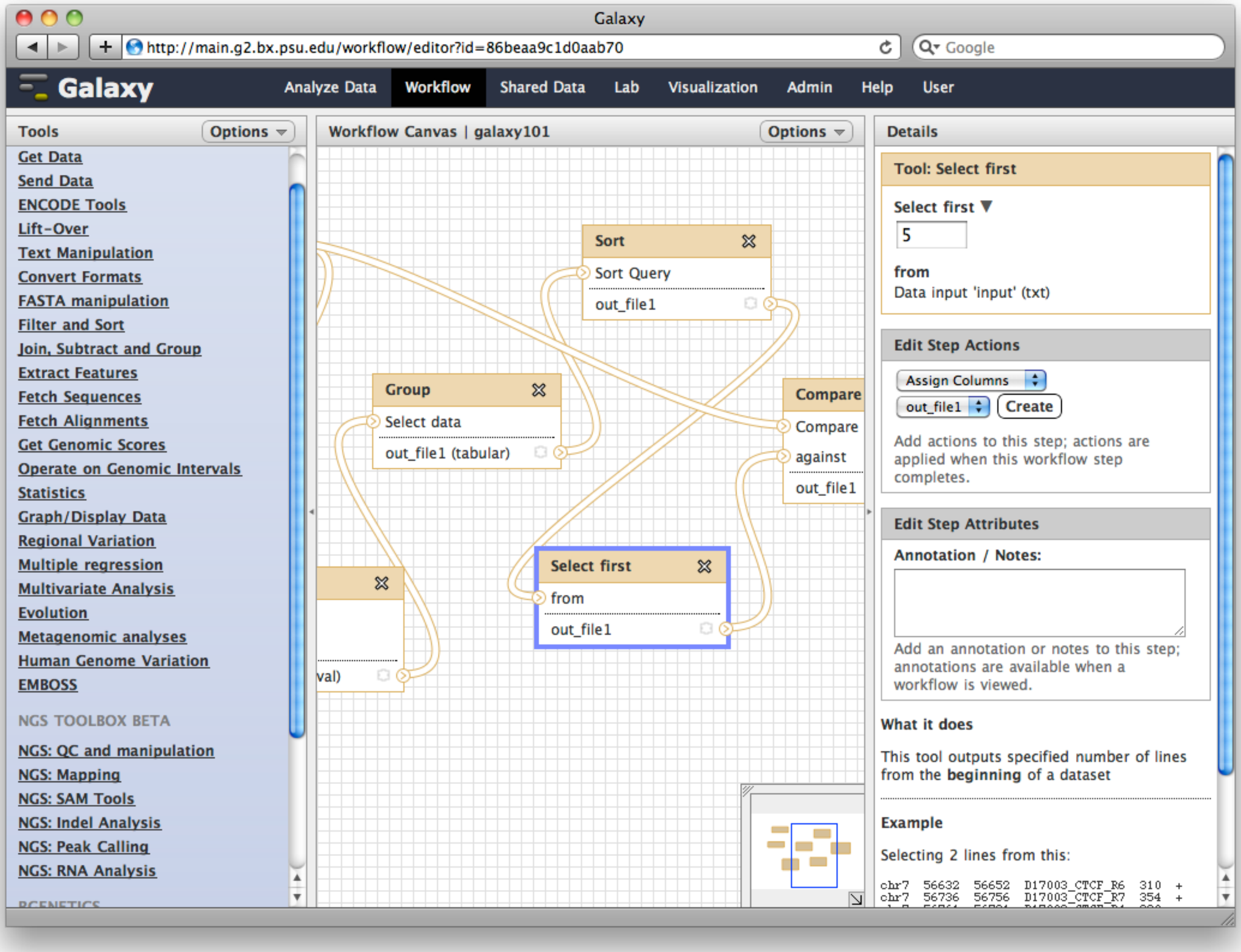
once you click **Create Workflow** you will get the following message: "Workflow 'galaxy101' created from current history". But where did it go? Click on **Workflow** link at the top of Galaxy interface and you will a list of all workflows with "galaxy101" listed at the top:

4.1. Opening workflow editor

If you click on a triangle adjacent to the workflow's name you will see the following dialogue:



Click **Edit** and the workflow editor will launch. It will allow you to examine and change settings of this workflow as shown below. Note that the box corresponding to the "*Select First*" tool is selected (highlighted with the blue border) and you can see parameters of this tool on the right pane. This is how you can view and change parameters of all tools involved in the workflow.



Tool: Select first

Select first ▼ 5

from
Data input 'input' (txt)

Edit Step Actions

Add actions to this step; actions are applied when this workflow step completes.

Edit Step Attributes

Annotation / Notes:

Add an annotation or notes to this step; annotations are available when a workflow is viewed.

What it does

This tool outputs specified number of lines from the **beginning** of a dataset

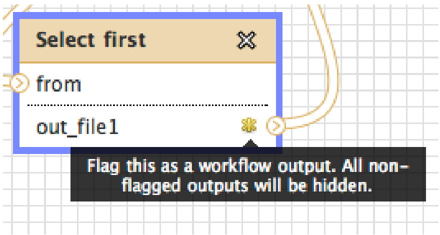
Example

Selecting 2 lines from this:

```
chr7  56632  56652  D17003_CTCF_R6  310  +
chr7  56736  56756  D17003_CTCF_R7  354  +
```

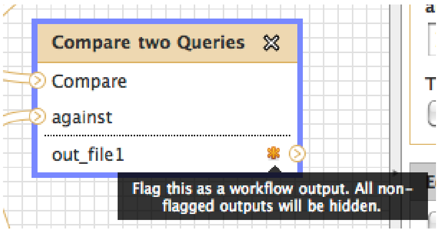
4.2. Hiding intermediate steps

Among multiple things you can do with workflows I will just mention one. When workflow is executed one is usually interested in the final product and not in the intermediate steps. These steps can be hidden by mousing over a small asterisk in the lower right corner of every tool box:



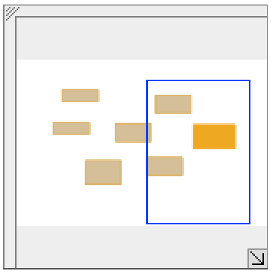
Flag this as a workflow output. All non-flagged outputs will be hidden.

Yet there is a catch. In a newly created workflow all steps are hidden by default and default behavior of Galaxy is that if all steps of a given workflow are hidden, then nothing gets hidden in the history. This may be counterintuitive, but this is done to decrease the amount of clicking if you do want to hide some steps. So in our case if we want to hide all intermediate steps with the exception of the last one we will click that asterisk in last step of the workflow:



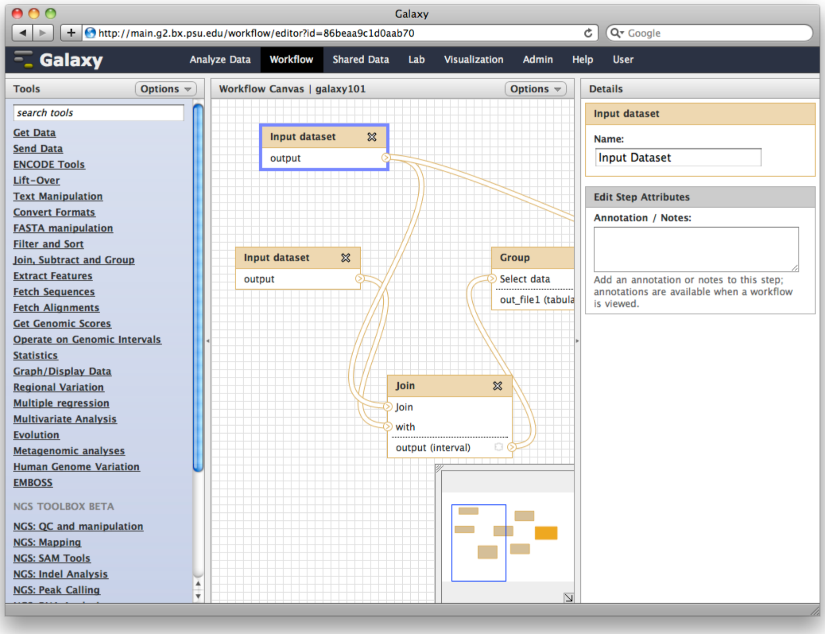
Flag this as a workflow output. All non-flagged outputs will be hidden.

Once you do this the representation of the workflow in the bottom right corner of the editor will change with the last step becoming orange. This means that this is the only step, which will generate a dataset visible in the history:



4.3. Renaming inputs

Right now both inputs to the workflow look exactly the same. This is a problem as will be very confusing which input should be exons and which should be SNPs:

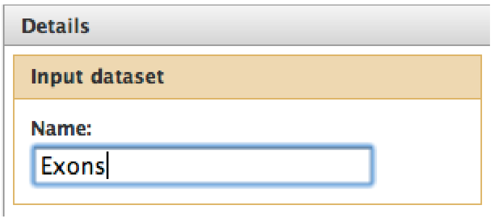


Name: Input Dataset

Annotation / Notes:

Add an annotation or notes to this step; annotations are available when a workflow is viewed.

One the image above you will see that the top input dataset (the one with the blue border) connects to the *Join* tool first, so it must correspond to the exon data. If you click on this box (in the image above it is already clicked on because it is outlined with the blue border) you will be able to rename the dataset in the right pane:

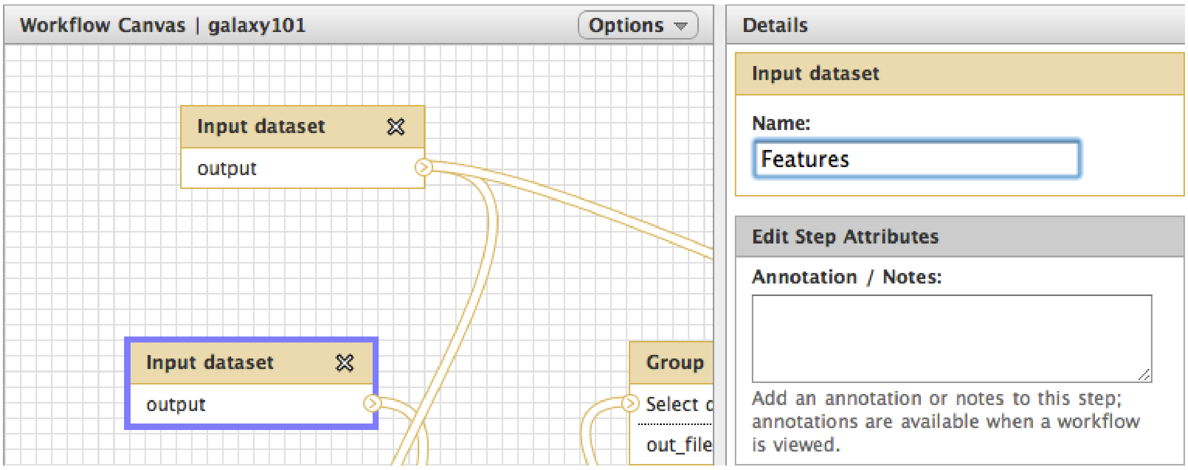


Details

Input dataset

Name: Exons

Then click on the second input dataset and rename it "Features" (this would make this workflow a bit more generic, which will be useful later in this tutorial):



Details

Input dataset

Name: Features

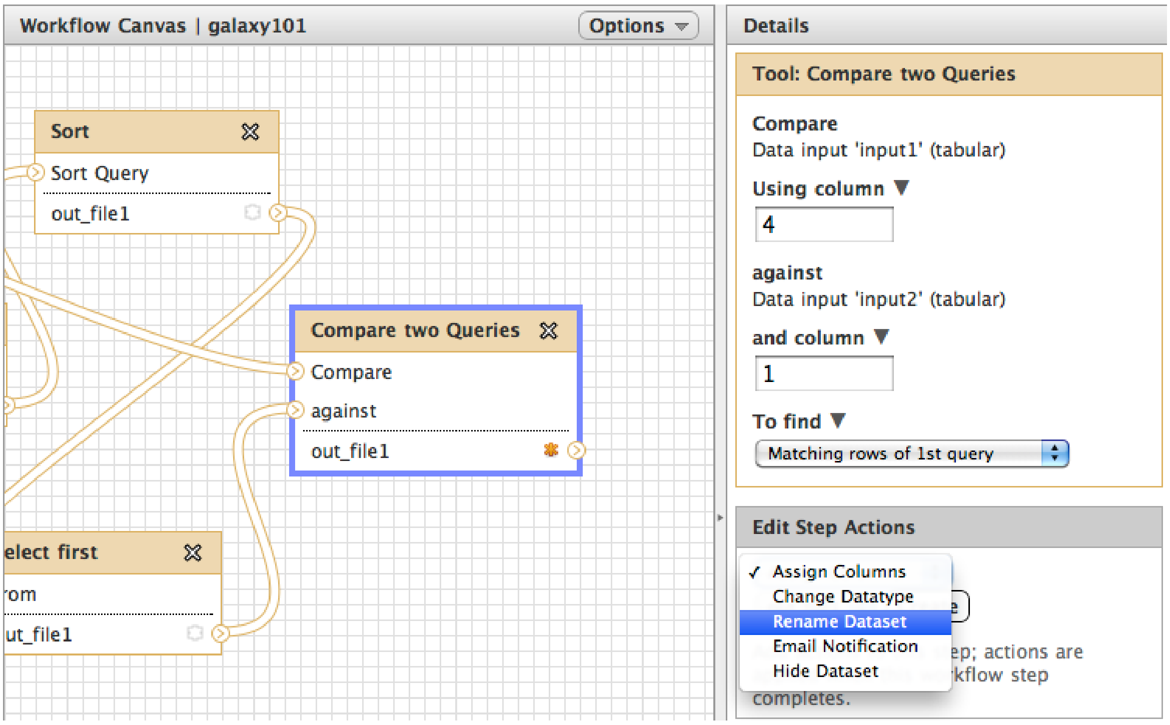
Edit Step Attributes

Annotation / Notes:

Add an annotation or notes to this step; annotations are available when a workflow is viewed.

4.4. Renaming outputs

Finally let's rename the workflow's output. For this click on the last dataset ("Compare two Queries") and in the **Edit Step Actions** dialogue box select "Rename Dataset"



Details

Tool: Compare two Queries

Compare
Data input 'input1' (tabular)

Using column ▼
4

against
Data input 'input2' (tabular)

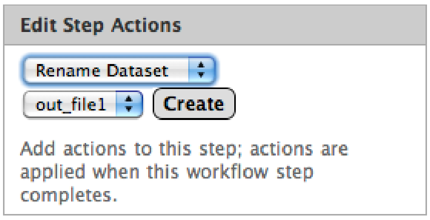
and column ▼
1

To find ▼
Matching rows of 1st query

Edit Step Actions

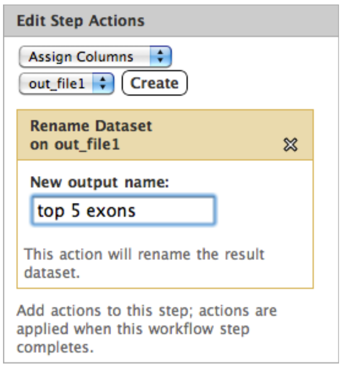
ep; actions are
kflow step
completes.

Click **Create:**



Add actions to this step; actions are applied when this workflow step completes.

and call it something like "top 5 exons":



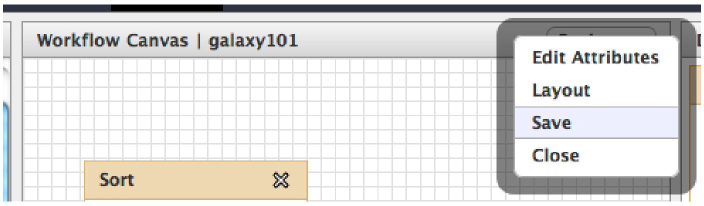
New output name: top 5 exons

This action will rename the result dataset.

Add actions to this step; actions are applied when this workflow step completes.

4.5. Save! It is important...

Now let's save the changes we've made by clicking **Options** (gear symbol) and selecting **Save:**



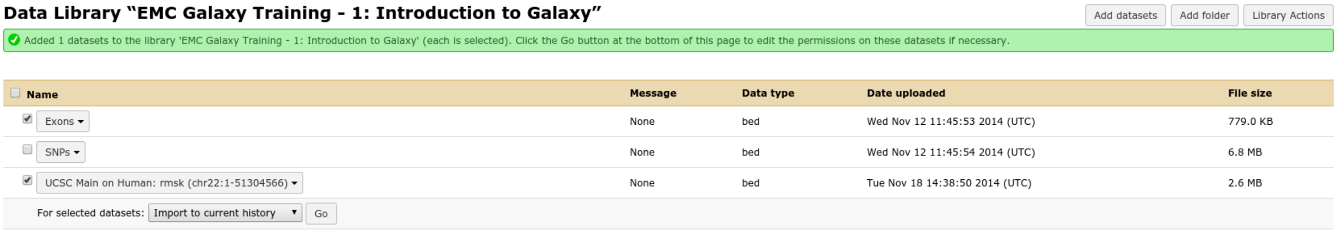
5. Run workflow on different data

Now that we have a workflow, let's use it on some different data. For example, let's find exons with the highest number of repetitive elements.

5.0. Create a new history

Before we start let's create a new history by clicking **Options** and selecting **Create New**.

Now let's get the chr22 exons from the data library again, as well as a list of repetitive elements (which were also obtained from UCSC table browser)



Data Library “EMC Galaxy Training - 1: Introduction to Galaxy”

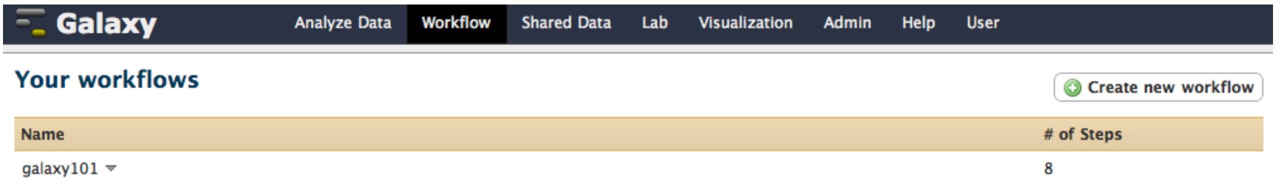
✓ Added 1 datasets to the library 'EMC Galaxy Training - 1: Introduction to Galaxy' (each is selected). Click the Go button at the bottom of this page to edit the permissions on these datasets if necessary.

□ Name	Message	Data type	Date uploaded	File size
☑ Exons ▾	None	bed	Wed Nov 12 11:45:53 2014 (UTC)	779.0 KB
□ SNPs ▾	None	bed	Wed Nov 12 11:45:54 2014 (UTC)	6.8 MB
☑ UCSC Main on Human: rmsk (chr22:1-51304566) ▾	None	bed	Tue Nov 18 14:38:50 2014 (UTC)	2.6 MB

For selected datasets: Import to current history ▾ Go

5.3. Start the Workflow

At this point you will have two items in your history - one with exons and one with repeats. First, click on the **Workflow** link at the top of Galaxy interface, mouse over "galaxy101" (or whatever you named your workflow), and click on the arrow:

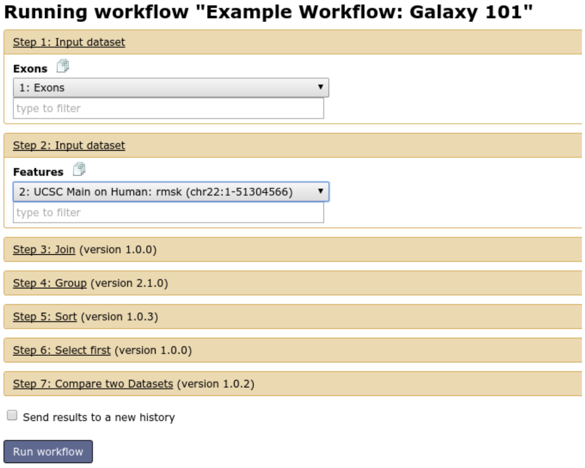


Your workflows

Name	# of Steps
galaxy101 ▾	8

choose **Run.**

Center pane will change to allow you launching the workflow. Select appropriate datasets for Repeats and Exon inputs as shown below, scroll down, and click Run workflow.



Running workflow "Example Workflow: Galaxy 101"

Step 1: Input dataset

Exons: 1: Exons ▾ (type to filter)

Step 2: Input dataset

Features: 2: UCSC Main on Human: rmsk (chr22:1-51304566) ▾ (type to filter)

Step 3: Join (version 1.0.0)

Step 4: Group (version 2.1.0)

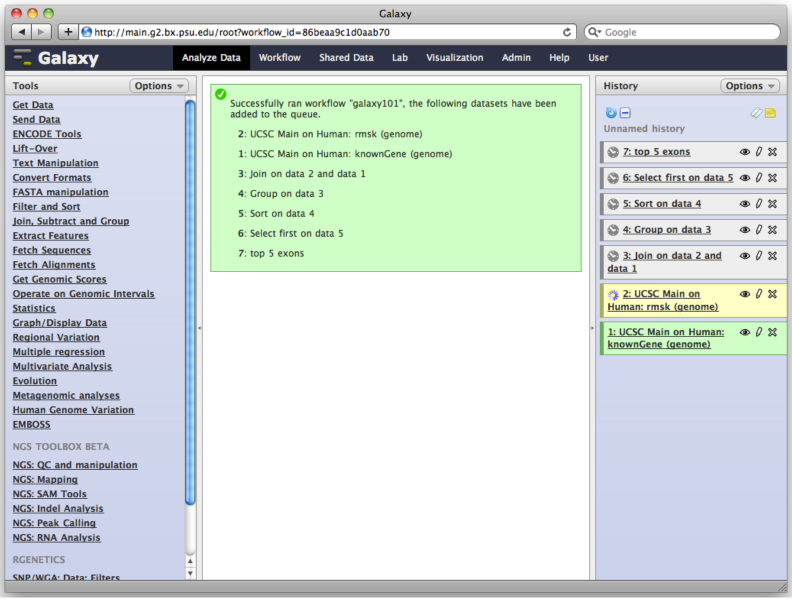
Step 5: Sort (version 1.0.3)

Step 6: Select first (version 1.0.0)

Step 7: Compare two Datasets (version 1.0.2)

□ Send results to a new history

Once the workflow has started you will initially be able to see all its steps.



Successfully ran workflow "galaxy101", the following datasets have been added to the queue.

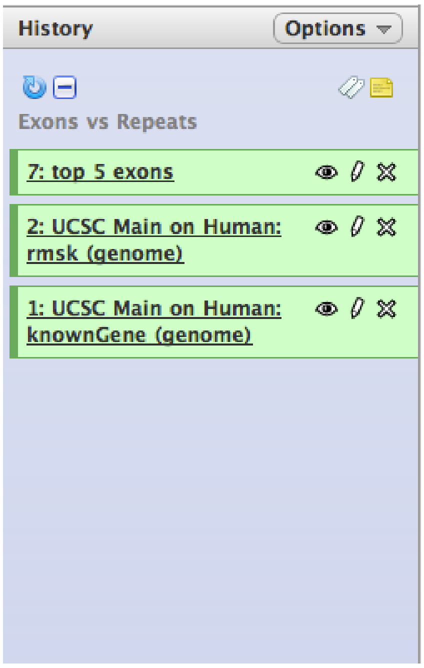
2: UCSC Main on Human: rmsk (genome)
1: UCSC Main on Human: knownGene (genome)
3: Join on data 2 and data 1
4: Group on data 3
5: Sort on data 4
6: Select first on data 5
7: top 5 exons

Unnamed history

7: top 5 exons
6: Select first on data 5
5: Sort on data 4
4: Group on data 3
3: Join on data 2 and data 1
2: UCSC Main on Human: rmsk (genome)
1: UCSC Main on Human: knownGene (genome)

(you may need to click the refresh button at the top of your history if the steps do not show up)

When it is finished, you will see this:


History
Options
Exons vs Repeats
7: top 5 exons
2: UCSC Main on Human: rmsk (genome)
1: UCSC Main on Human: knownGene (genome)

Note that because all intermediate steps of the workflow were hidden, once it is finished you will only see the final dataset #7

If we want to view the intermediate files, we can view the hidden datasets by selecting **"Include Hidden datasets"** from the history options menu.

5.5 Share your work

Often you may want to share a workflow you created, or an analysis history, with other users.

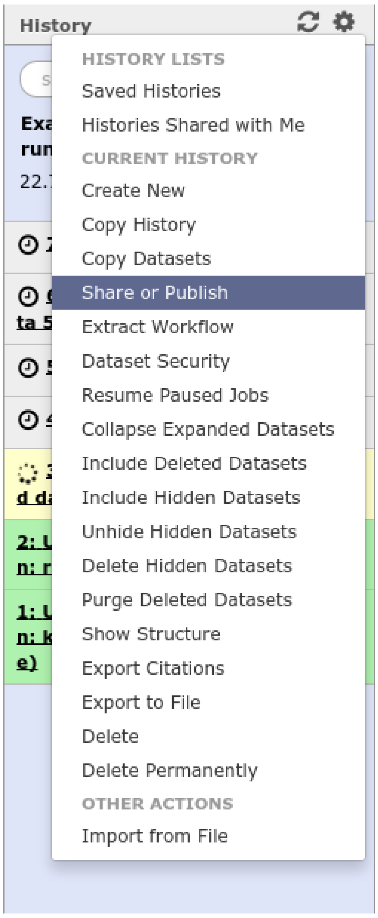
You can share a history by clicking on the gear symbol, and selecting **Share or Publish**.

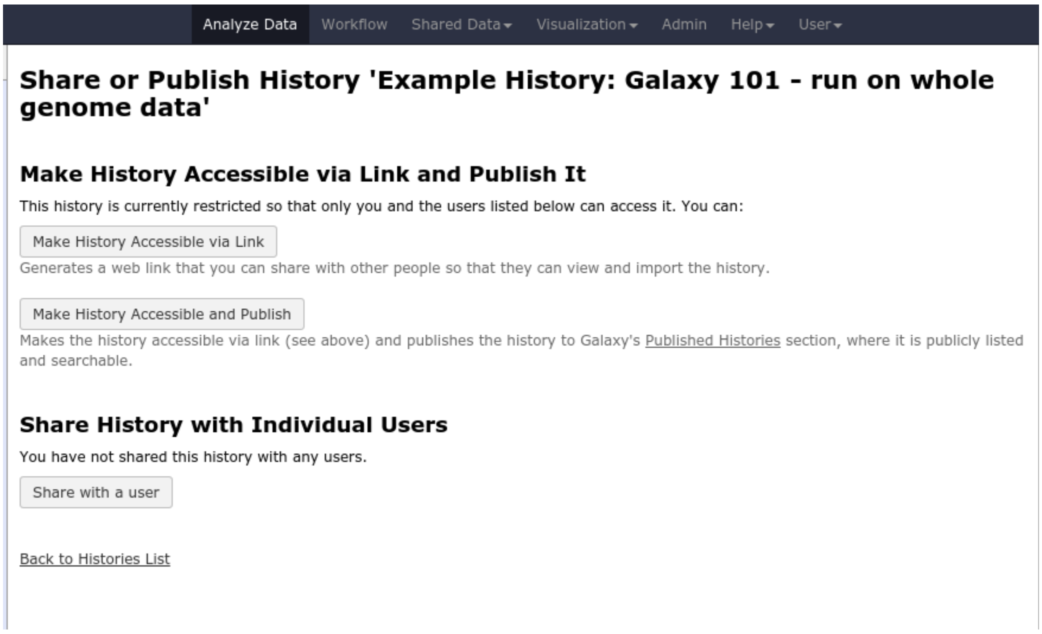
You can do 3 things here:

- **Make accessible via Link**
  This generates a link that you can give out to others. Anybody with this link will be able to view your history (even without a Galaxy account)

- **Publish History**
  This will not only create a link, but will also publish your history. (i.e. Your history will now appear under Shared Data → Published Histories)

- **Share with Individual Users**
  This will share the history with specific users on the Galaxy instance. Enter their email address (which they used to register their account in Galaxy)





Share or Publish History 'Example History: Galaxy 101 - run on whole genome data'

Make History Accessible via Link and Publish It

This history is currently restricted so that only you and the users listed below can access it. You can:

Make History Accessible via Link

Generates a web link that you can share with other people so that they can view and import the history.

Make History Accessible and Publish

Makes the history accessible via link (see above) and publishes the history to Galaxy's Published Histories section, where it is publicly listed and searchable.

Share History with Individual Users

You have not shared this history with any users.

Share with a user

Back to Histories List

- Share one of your histories with your neighbour, or publish it.

- See if you can do the same with your workflow!

THE END!