Galaxy

The *Galaxy Server*, more than just a database portal:

Most database providers, like *NCBI* and *Bio-Mart,* offer highly intelligent data accession systems, yet it largely remains up to the user to find ways to parse this data when he wants to mine it for information. In most cases this requires knowledge of scripting languages, or at least the ability to manually pipe the data into other servers.

Galaxy offers a simple solution by providing the user with a host of tools to mine, alter, translate, pipe his data, making bioinformatics tools accessible to a wide range of users.

In this tutorial we will look at an easy way to mine a large data set for (hopefully) useful information, using only *Galaxy* tools

http://main.g2.bx.psu.edu/

Galaxy

We will use *Glaxy* to access two large sets of data.

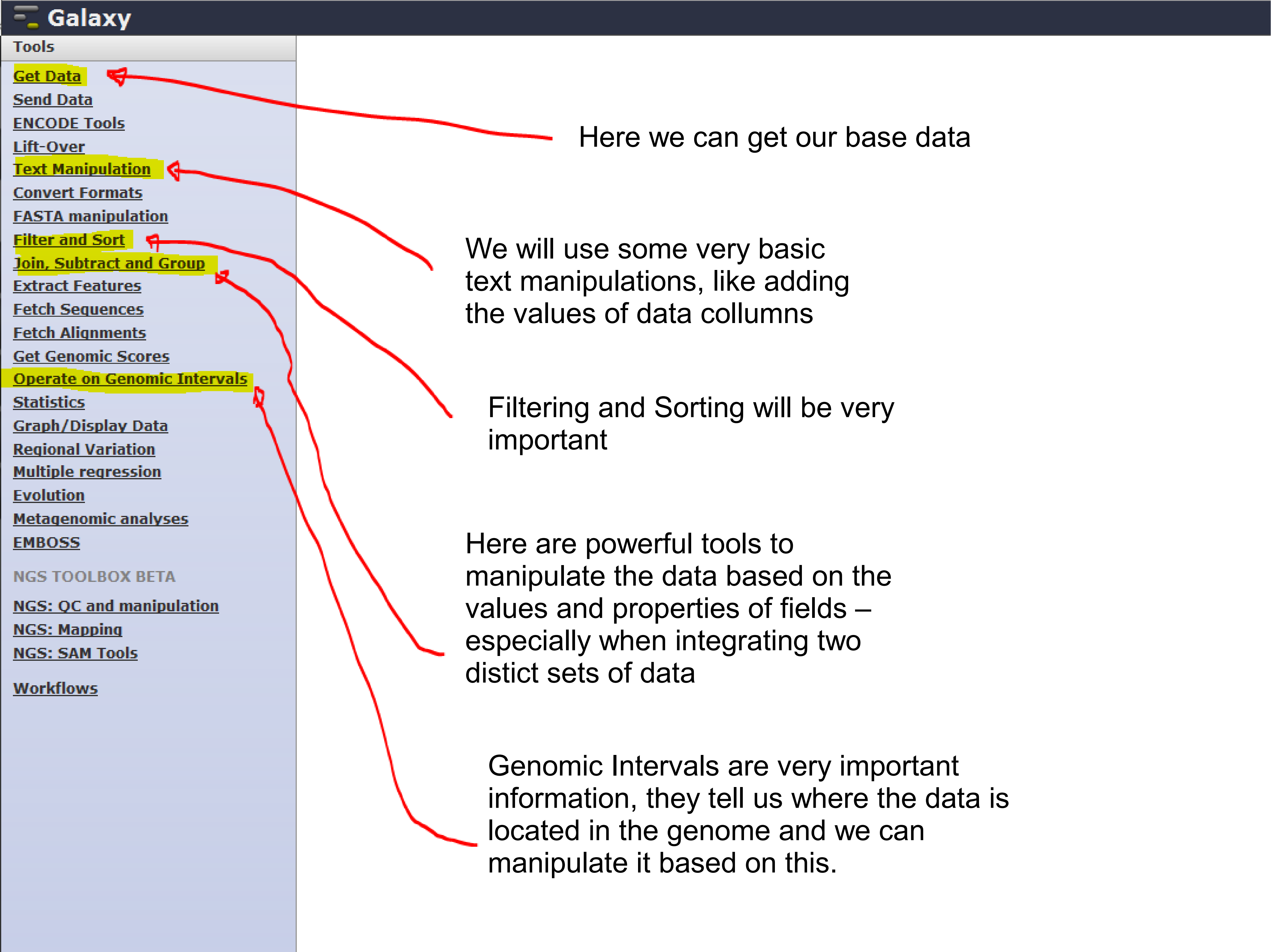
1. All known genes in a human chromosome of our choice.
2. All known SNPs located in this chromome.

Galaxy provides us access to many genomic databases. In this tutorial we will choose to look at genes from the *RefSeq* database because it is a well annotated dataset. We will want to use these annotations later to judge the quality of our filtering.

So, what can we do with these datasets? Well, we know that SNP's are not distributed stochastically even over the genome... so maybe we can gain some information by looking at just those genes which are exceptionally „rich“ or „poor“ in SNPs.

Our goal thus is to mine the data for a suitable small set of genes, which we can then base testable predictions on. Or shorter - to do science - all without ever leaving the site.

Questions you should try to answer to understand the logic of this tutorial are printed in *italics*.



Here we can get our base data

We will use some very basic text manipulations, like adding the values of data collumns

Filtering and Sorting will be very important

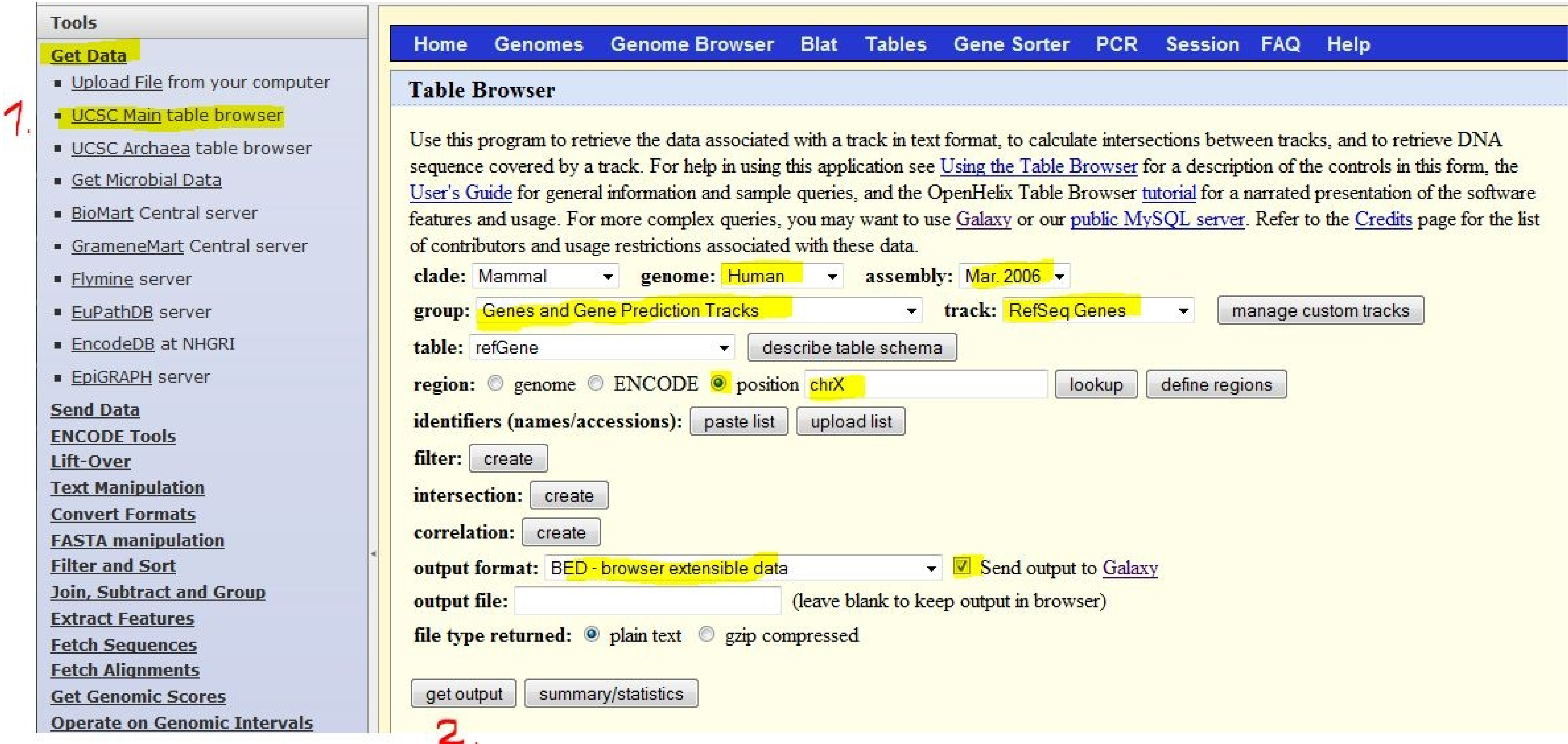
Here are powerful tools to manipulate the data based on the values and properties of fields – especially when integrating two distict sets of data

Genomic Intervals are very important information, they tell us where the data is located in the genome and we can manipulate it based on this.

We will load the data from the UCSC Main table browser.

There are many, many choices to be made, so look around and try to find out what some of the main options are.

The specific data we will use in this example can be accessed like this:



Tools

Get Data

- Upload File from your computer
- 1. UCSC Main table browser
- UCSC Archaea table browser
- Get Microbial Data
- BioMart Central server
- GrameneMart Central server
- Flymine server
- EuPathDB server
- EncodeDB at NHGRI
- EpiGRAPH server

Send Data
ENCODE Tools
Lift-Over
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

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. Refer to the Credits page for the list of contributors and usage restrictions associated with these data.

clade: Mammal genome: Human assembly: Mar. 2006

group: Genes and Gene Prediction Tracks track: RefSeq Genes

table: refGene

region: ○ genome ○ ENCODE ● position chrX

identifiers (names/accessions):

filter:

intersection:

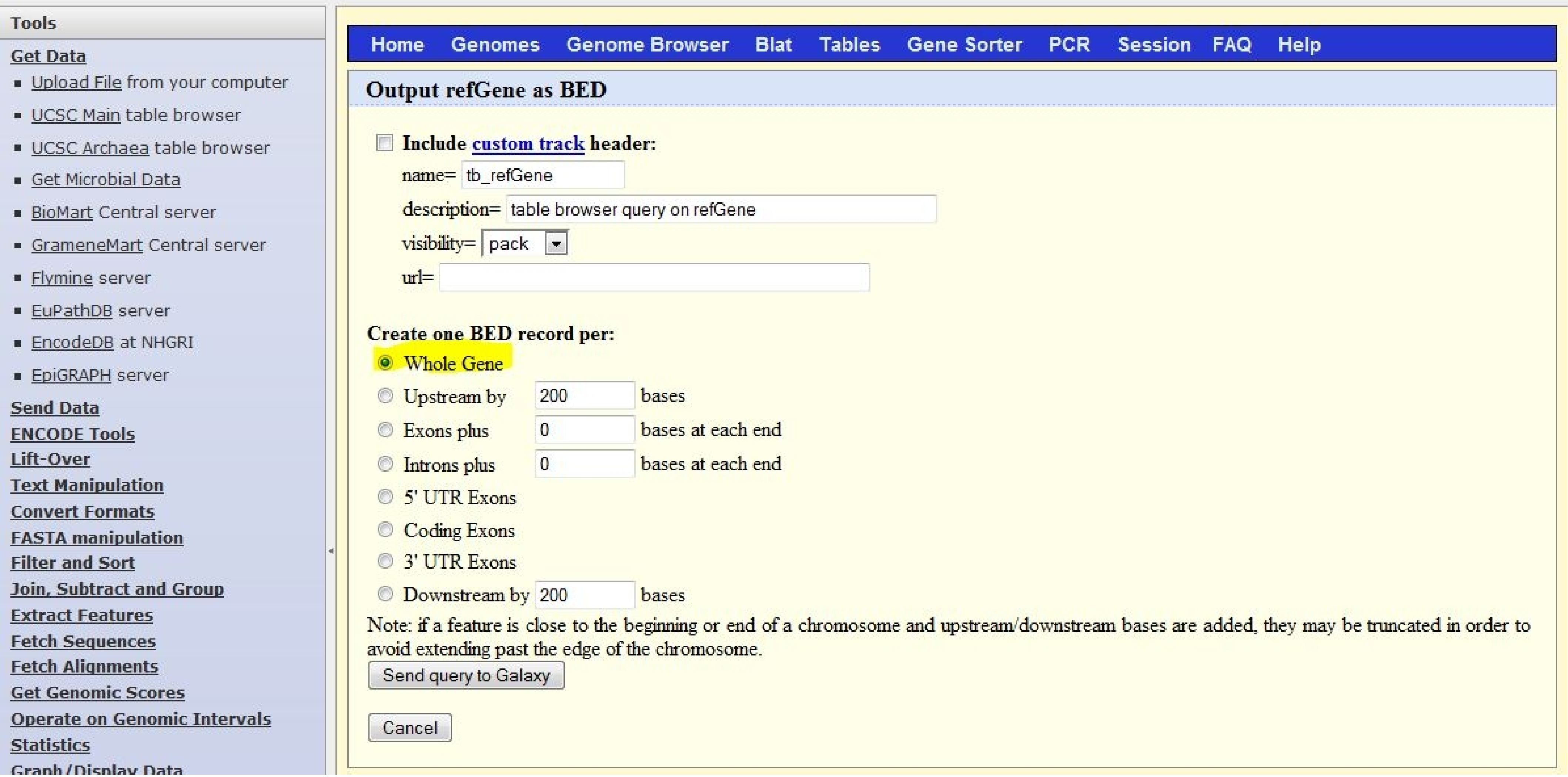
correlation:

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

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

file type returned: ● plain text ○ gzip compressed

We will look at the whole genes. If we were just interested in the coding exons, or perhaps the upstream (promotor) regions of the genes, we could specify this here:



Output refGene as BED

☐ **Include custom track header:**

name= tb_refGene

description= table browser query on refGene

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

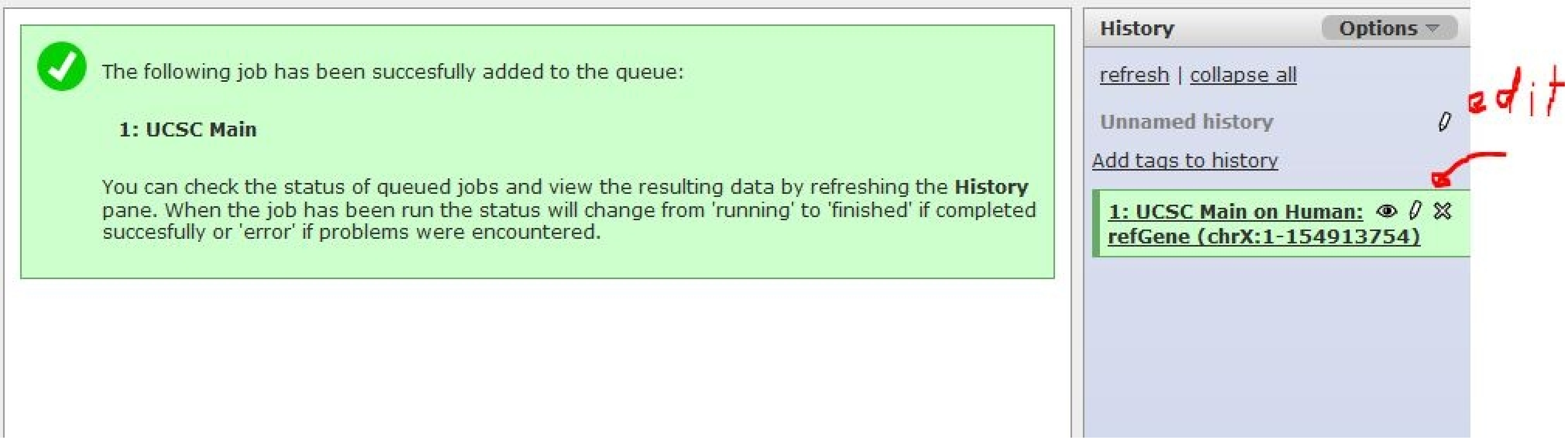
○ Coding Exons

○ 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.

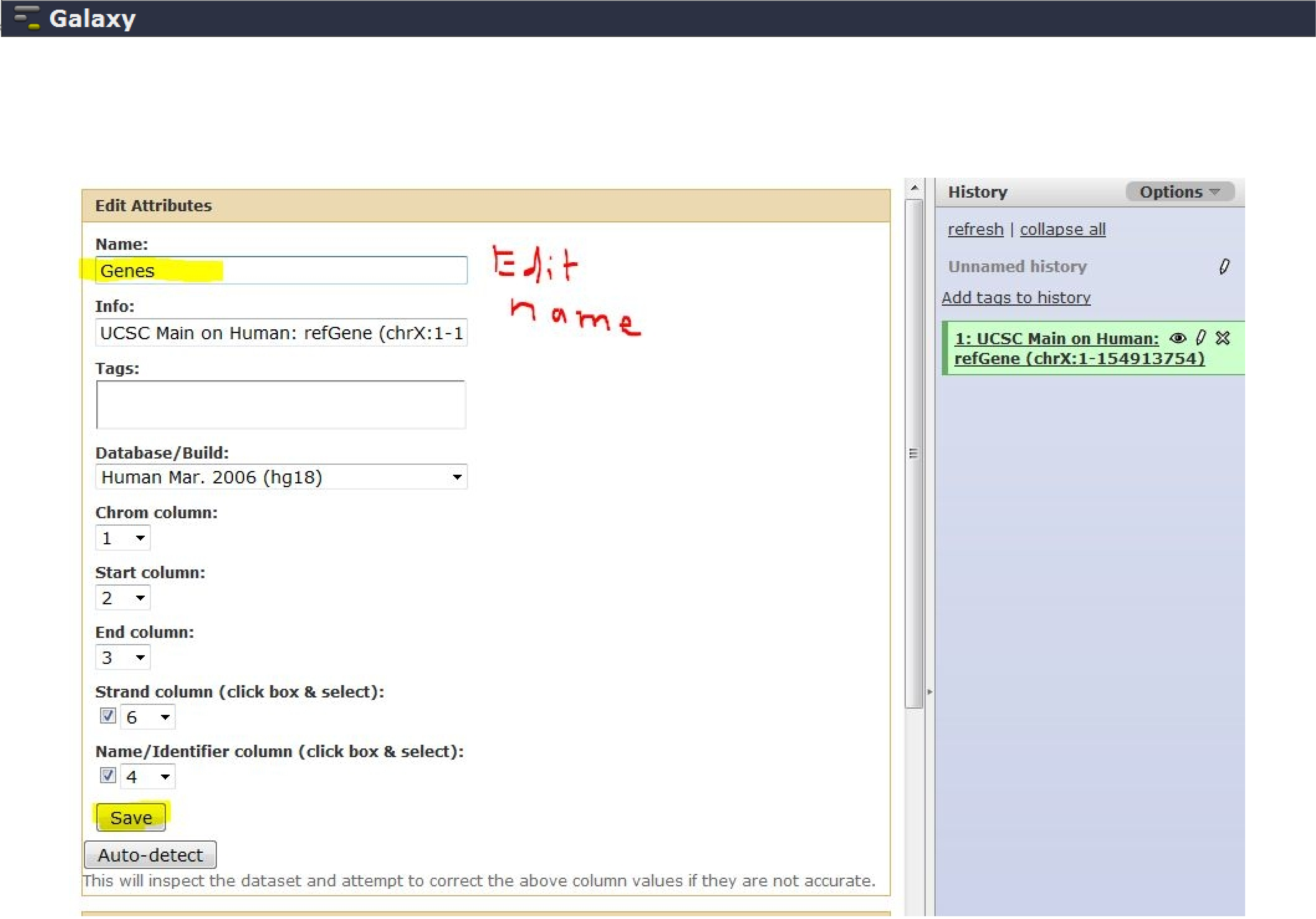
The data is now in our history, but the name of the history entry is a bit unweildy, so we will edit the name to something more pithy



The following job has been succesfully 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 succesfully or 'error' if problems were encountered.



Edit Attributes

Name: Genes

Edit name

Info: UCSC Main on Human: refGene (chrX:1-1

Tags:

Database/Build: Human Mar. 2006 (hg18)

Chrom column: 1

Start column: 2

End column: 3

Strand column (click box & select): ☑ 6

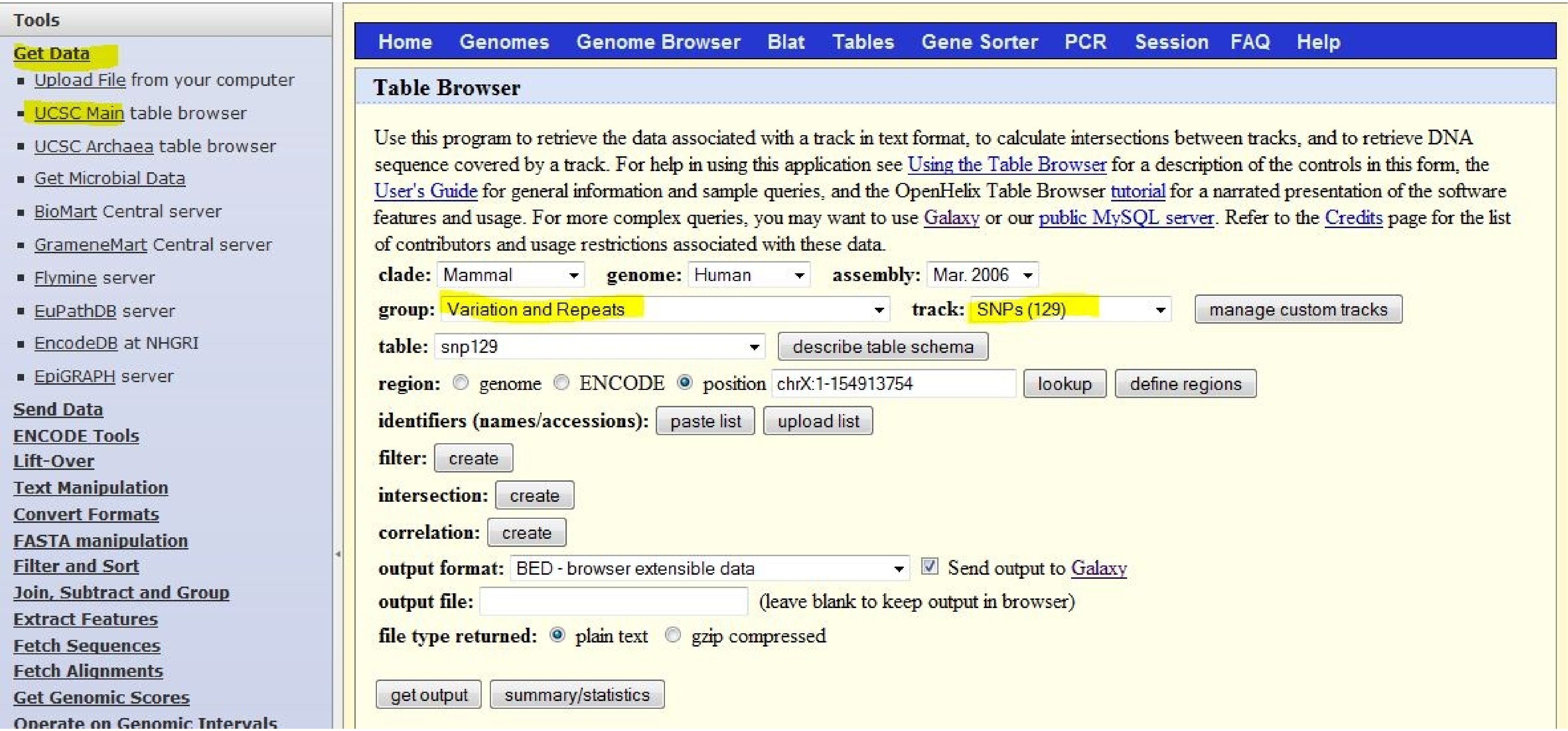
Name/Identifier column (click box & select): ☑ 4

This will inspect the dataset and attempt to correct the above column values if they are not accurate.

Unnamed history

1: UCSC Main on Human: refGene (chrX:1-154913754)

We repeat the whole process for the SNPs data. This time we choose our data from the Variations and Repeats group.



Tools

Get Data

- Upload File from your computer
- UCSC Main table browser
- UCSC Archaea table browser
- Get Microbial Data
- BioMart Central server
- GrameneMart Central server
- Flymine server
- EuPathDB server
- EncodeDB at NHGRI
- EpiGRAPH server

Send Data
ENCODE Tools
Lift-Over
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

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. Refer to the Credits page for the list of contributors and usage restrictions associated with these data.

clade: Mammal **genome:** Human **assembly:** Mar. 2006

group: Variation and Repeats **track:** SNPs (129) [manage custom tracks]

table: snp129 [describe table schema]

region: ○ genome ○ ENCODE ● position chrX:1-154913754 [lookup] [define regions]

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

filter: [create]

intersection: [create]

correlation: [create]

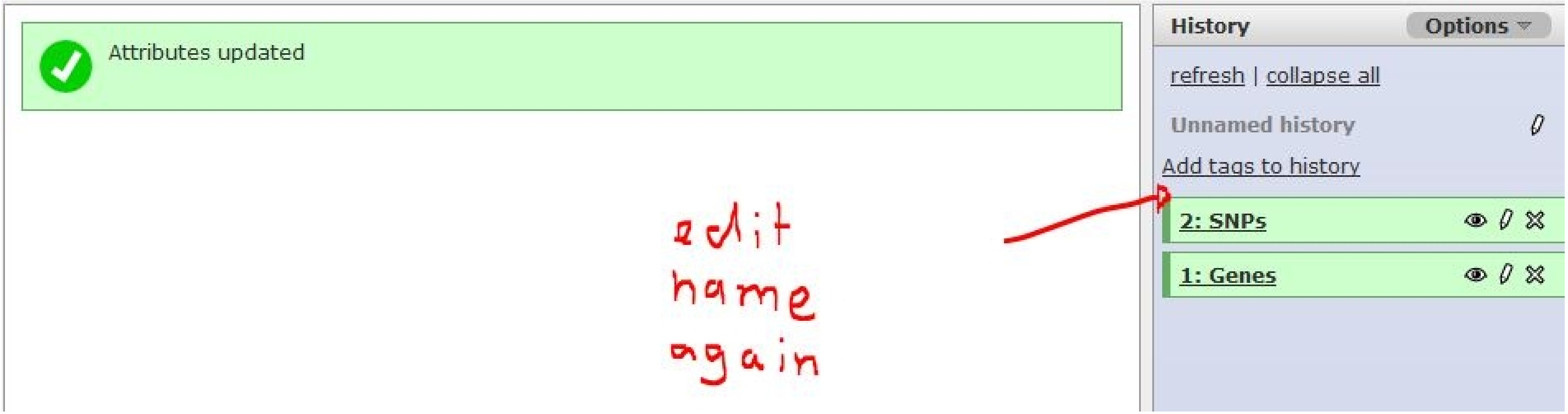
output format: BED - browser extensible data ☑ Send output to Galaxy

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

file type returned: ● plain text ○ gzip compressed

[get output] [summary/statistics]

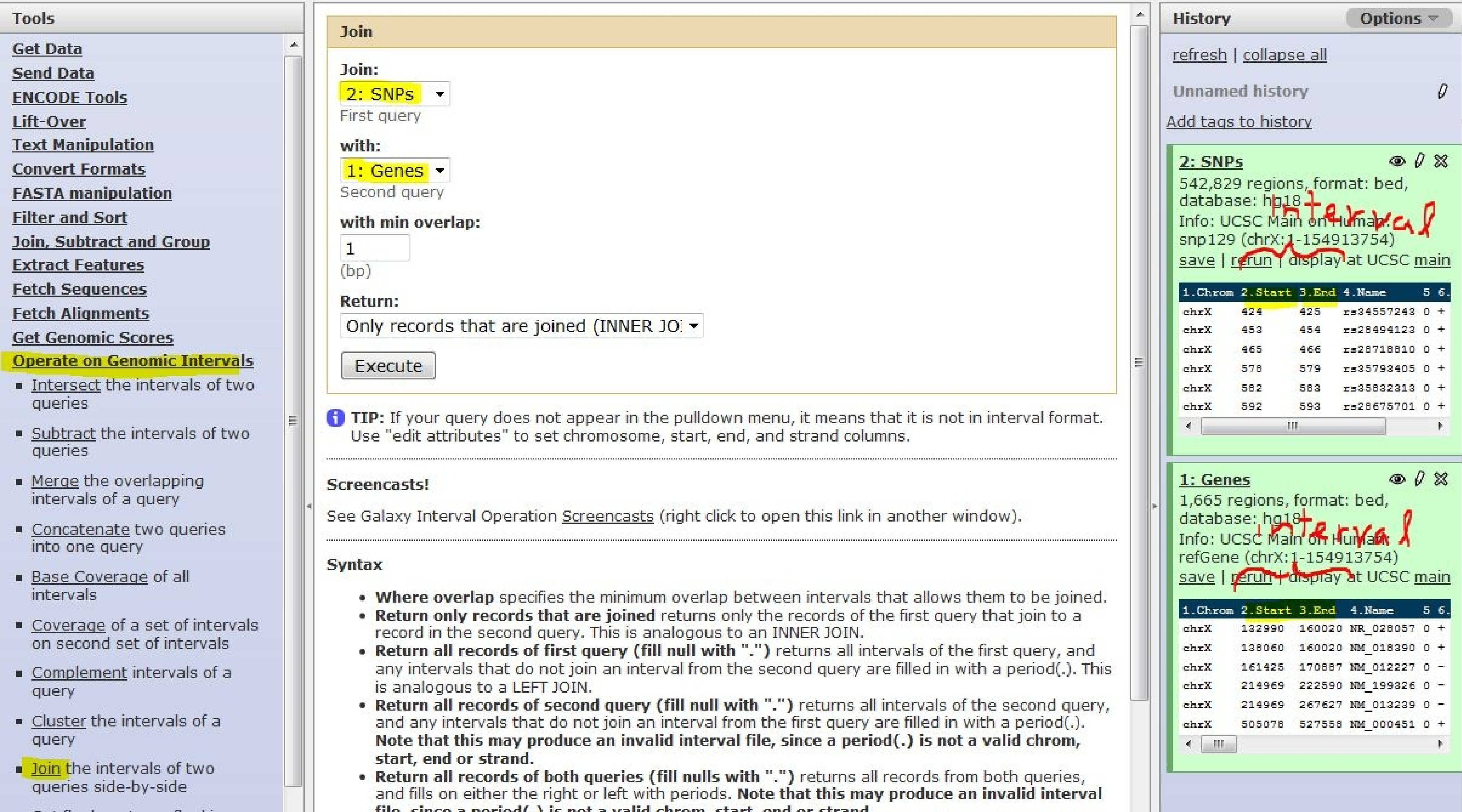
It's useful to edit the names of all history entries, so we don't lose track of them as the history fills.



Attributes updated

Galaxy

Now we can join SNPs and Genes by their genomic locations. By doing this we associate every SNP that is located within any of the genes to the specific gene engulfing it.



- 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

Join

Join: 2: SNPs
First query

with: 1: Genes
Second query

with min overlap: 1
(bp)

Return: Only records that are joined (INNER JO

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 may produce an invalid interval file, since a period(.) is not a valid chrom, start, end or strand.**
- **Return all records of both queries (fill nulls with ".")** returns all records from both queries, and fills on either the right or left with periods. **Note that this may produce an invalid interval file, since a period(.) is not a valid chrom, start, end or strand**

2: SNPs
542,829 regions, format: bed, database: hg18
Info: UCSC Main on Human: snp129 (chrX:1-154913754)
save | rerun | display at UCSC main

Interval

1.Chrom	2.Start	3.End	4.Name	5	6.
chrX	424	425	rs34557243	0	+
chrX	453	454	rs28494123	0	+
chrX	465	466	rs28718810	0	+
chrX	578	579	rs35793405	0	+
chrX	582	583	rs35832313	0	+
chrX	592	593	rs28675701	0	+

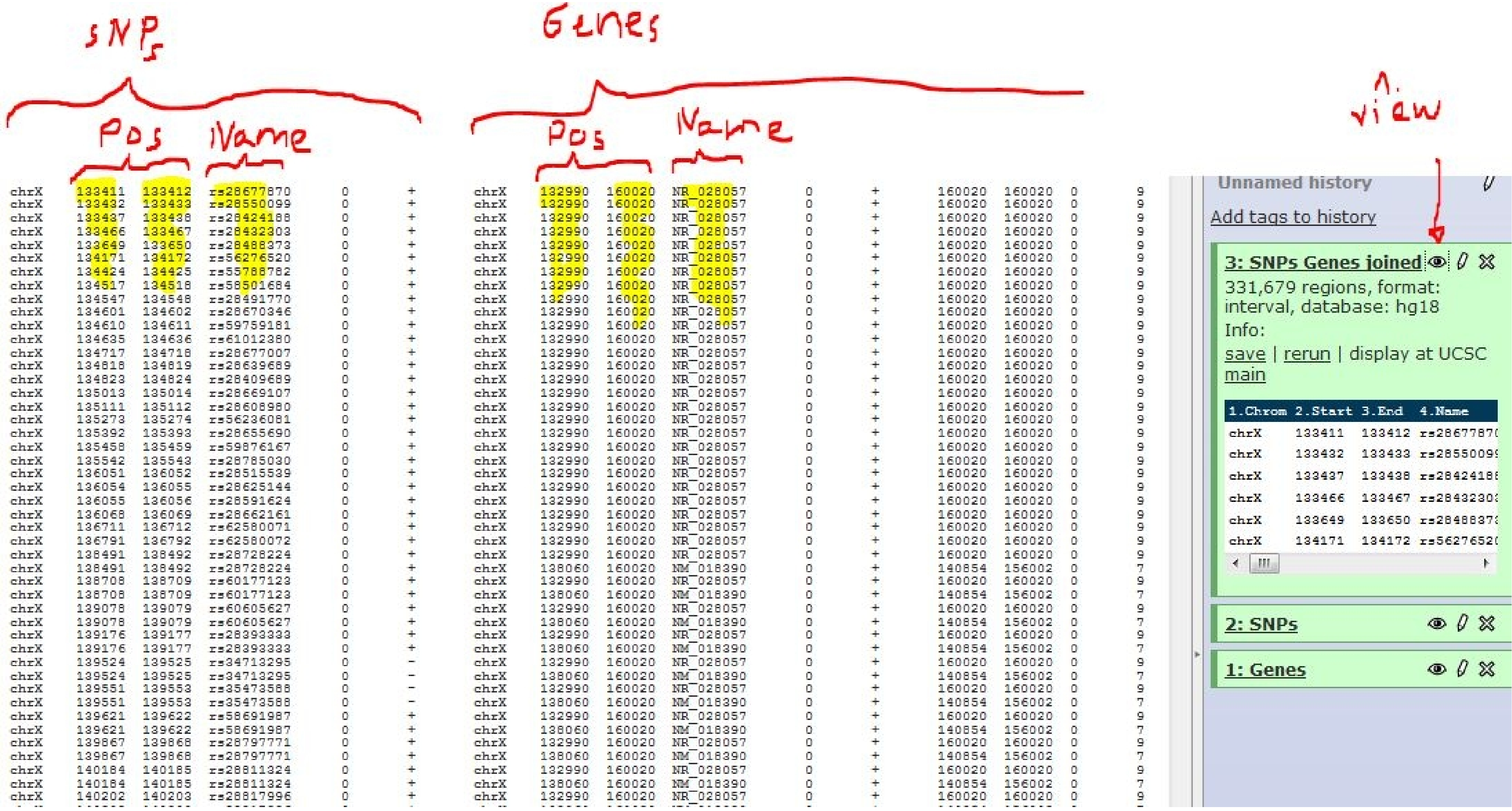
1: Genes
1,665 regions, format: bed, database: hg18
Info: UCSC Main on Human: refGene (chrX:1-154913754)
save | rerun | display at UCSC main

interval

1.Chrom	2.Start	3.End	4.Name	5	6.
chrX	132990	160020	NR_028057	0	+
chrX	138060	160020	NM_018390	0	+
chrX	161425	170887	NM_012227	0	-
chrX	214969	222590	NM_199326	0	-
chrX	214969	267627	NM_013239	0	-
chrX	505078	527558	NM_000451	0	+

Galaxy

Why do individual genes appear multiple times in this joined data, but each SNP only once?



```
chrX   133411  133412  rs28677870   0   +   chrX   132990  160020  NR_028057   0   +   160020  160020  0   9
chrX   133432  133433  rs28550099   0   +   chrX   132990  160020  NR_028057   0   +   160020  160020  0   9
chrX   133437  133438  rs28424188   0   +   chrX   132990  160020  NR_028057   0   +   160020  160020  0   9
chrX   133466  133467  rs28432303   0   +   chrX   132990  160020  NR_028057   0   +   160020  160020  0   9
chrX   133649  133650  rs28488373   0   +   chrX   132990  160020  NR_028057   0   +   160020  160020  0   9
chrX   134171  134172  rs56276520   0   +   chrX   132990  160020  NR_028057   0   +   160020  160020  0   9
chrX   134424  134425  rs55788782   0   +   chrX   132990  160020  NR_028057   0   +   160020  160020  0   9
chrX   134517  134518  rs58501684   0   +   chrX   132990  160020  NR_028057   0   +   160020  160020  0   9
chrX   134547  134548  rs28491770   0   +   chrX   132990  160020  NR_028057   0   +   160020  160020  0   9
chrX   134601  134602  rs28670346   0   +   chrX   132990  160020  NR_028057   0   +   160020  160020  0   9
chrX   134610  134611  rs59759181   0   +   chrX   132990  160020  NR_028057   0   +   160020  160020  0   9
chrX   134635  134636  rs61012380   0   +   chrX   132990  160020  NR_028057   0   +   160020  160020  0   9
chrX   134717  134718  rs28677007   0   +   chrX   132990  160020  NR_028057   0   +   160020  160020  0   9
chrX   134818  134819  rs28639689   0   +   chrX   132990  160020  NR_028057   0   +   160020  160020  0   9
chrX   134823  134824  rs28409689   0   +   chrX   132990  160020  NR_028057   0   +   160020  160020  0   9
chrX   135013  135014  rs28669107   0   +   chrX   132990  160020  NR_028057   0   +   160020  160020  0   9
chrX   135111  135112  rs28608980   0   +   chrX   132990  160020  NR_028057   0   +   160020  160020  0   9
chrX   135273  135274  rs56236081   0   +   chrX   132990  160020  NR_028057   0   +   160020  160020  0   9
chrX   135392  135393  rs28655690   0   +   chrX   132990  160020  NR_028057   0   +   160020  160020  0   9
chrX   135458  135459  rs59876167   0   +   chrX   132990  160020  NR_028057   0   +   160020  160020  0   9
chrX   135542  135543  rs28785030   0   +   chrX   132990  160020  NR_028057   0   +   160020  160020  0   9
chrX   136051  136052  rs28515539   0   +   chrX   132990  160020  NR_028057   0   +   160020  160020  0   9
chrX   136054  136055  rs28625144   0   +   chrX   132990  160020  NR_028057   0   +   160020  160020  0   9
chrX   136055  136056  rs28591624   0   +   chrX   132990  160020  NR_028057   0   +   160020  160020  0   9
chrX   136068  136069  rs28662161   0   +   chrX   132990  160020  NR_028057   0   +   160020  160020  0   9
chrX   136711  136712  rs62580071   0   +   chrX   132990  160020  NR_028057   0   +   160020  160020  0   9
chrX   136791  136792  rs62580072   0   +   chrX   132990  160020  NR_028057   0   +   160020  160020  0   9
chrX   138491  138492  rs28728224   0   +   chrX   138060  160020  NM_018390   0   +   140854  156002  0   7
chrX   138491  138492  rs28728224   0   +   chrX   132990  160020  NR_028057   0   +   160020  160020  0   9
chrX   138708  138709  rs60177123   0   +   chrX   138060  160020  NM_018390   0   +   140854  156002  0   7
chrX   138708  138709  rs60177123   0   +   chrX   132990  160020  NR_028057   0   +   160020  160020  0   9
chrX   139078  139079  rs60605627   0   +   chrX   138060  160020  NM_018390   0   +   140854  156002  0   7
chrX   139078  139079  rs60605627   0   +   chrX   132990  160020  NR_028057   0   +   160020  160020  0   9
chrX   139176  139177  rs28393333   0   +   chrX   138060  160020  NM_018390   0   +   140854  156002  0   7
chrX   139176  139177  rs28393333   0   +   chrX   132990  160020  NR_028057   0   +   160020  160020  0   9
chrX   139524  139525  rs34713295   0   -   chrX   138060  160020  NM_018390   0   +   140854  156002  0   7
chrX   139524  139525  rs34713295   0   -   chrX   132990  160020  NR_028057   0   +   160020  160020  0   9
chrX   139551  139553  rs35473588   0   -   chrX   138060  160020  NM_018390   0   +   140854  156002  0   7
chrX   139551  139553  rs35473588   0   -   chrX   132990  160020  NR_028057   0   +   160020  160020  0   9
chrX   139621  139622  rs58691987   0   +   chrX   138060  160020  NM_018390   0   +   140854  156002  0   7
chrX   139621  139622  rs58691987   0   +   chrX   132990  160020  NR_028057   0   +   160020  160020  0   9
chrX   139867  139868  rs28797771   0   +   chrX   138060  160020  NM_018390   0   +   140854  156002  0   7
chrX   139867  139868  rs28797771   0   +   chrX   132990  160020  NR_028057   0   +   160020  160020  0   9
chrX   140184  140185  rs28811324   0   +   chrX   138060  160020  NM_018390   0   +   140854  156002  0   7
chrX   140184  140185  rs28811324   0   +   chrX   132990  160020  NR_028057   0   +   160020  160020  0   9
chrX   140202  140203  rs28817996   0   +   chrX   132990  160020  NR_028057   0   +   160020  160020  0   9
```

Unnamed history

Add tags to history

3: SNPs Genes joined

331,679 regions, format: interval, database: hg18

Info:

save | rerun | display at UCSC main

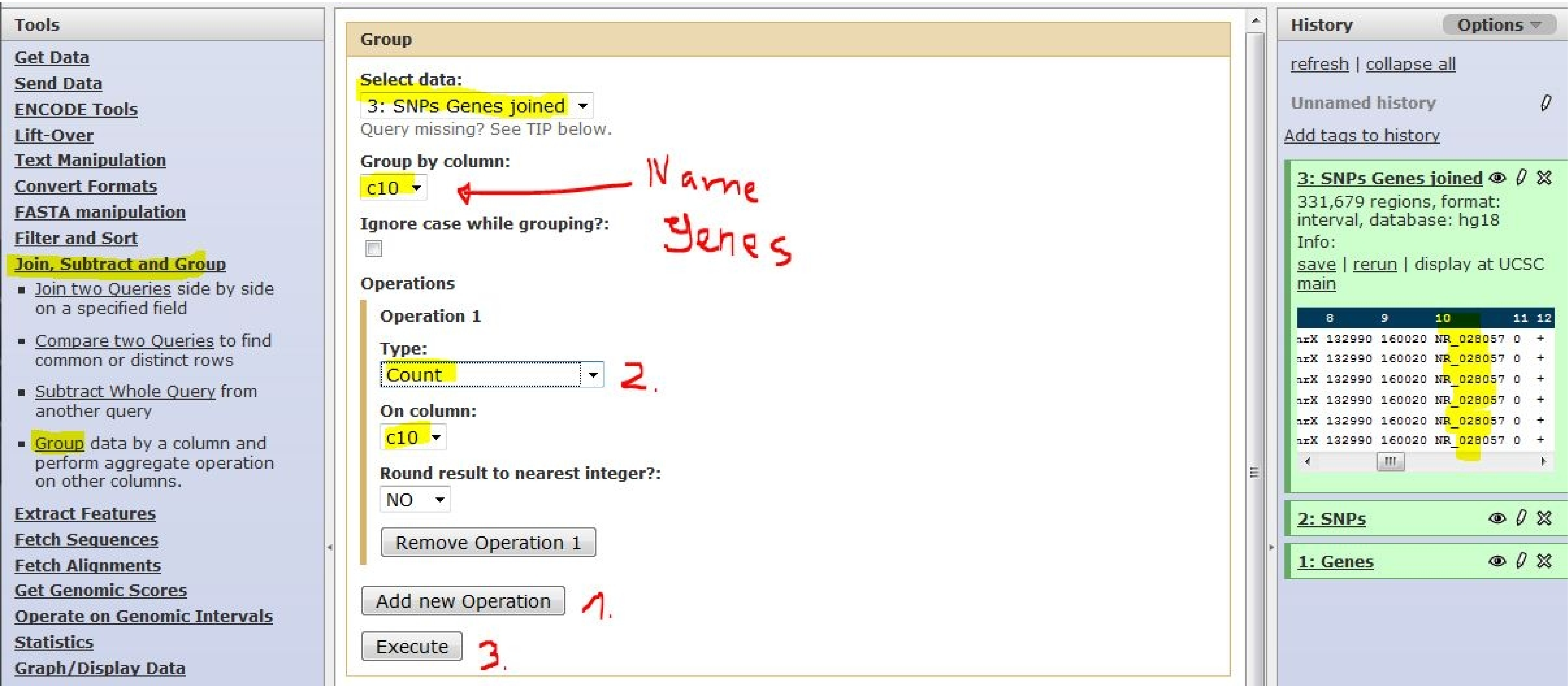
1.Chrom	2.Start	3.End	4.Name
chrX	133411	133412	rs2867787
chrX	133432	133433	rs2855009
chrX	133437	133438	rs2842418
chrX	133466	133467	rs2843230
chrX	133649	133650	rs2848837
chrX	134171	134172	rs5627652

2: SNPs

1: Genes

By grouping this list we can indirectly determine the number of SNPs within every gene. We simply count the number of instances of each distinct gene name in our joined list.

Why can we say that this count is quivalent to the number of SNPs per gene?



- 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.

Group

Select data: 3: SNPs Genes joined

Query missing? See TIP below.

Group by column: c10

Name genes

Ignore case while grouping?: □

Operations

Operation 1

Type: Count

2.

On column: c10

Round result to nearest integer?: NO

1.

3.

3: SNPs Genes joined

331,679 regions, format: interval, database: hg18

Info:

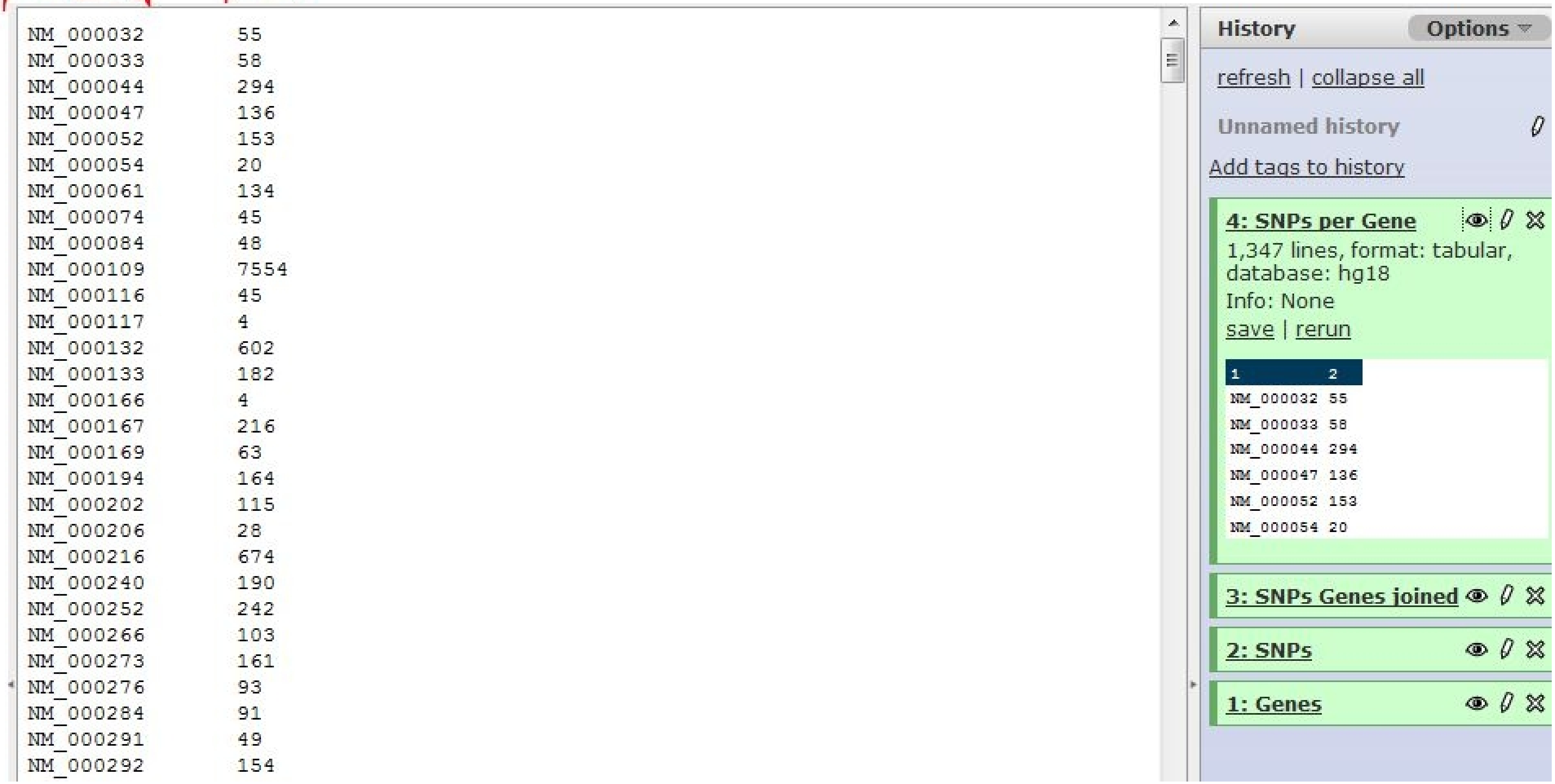
	8	9	10	11	12
rX	132990	160020	NR_028057	0	+
rX	132990	160020	NR_028057	0	+
rX	132990	160020	NR_028057	0	+
rX	132990	160020	NR_028057	0	+
rX	132990	160020	NR_028057	0	+
rX	132990	160020	NR_028057	0	+

Galaxy

This then gives us a list of distinct genes and the number of SNPs within them.

name - genes

SNPs



```
NM_000032        55
NM_000033        58
NM_000044        294
NM_000047        136
NM_000052        153
NM_000054        20
NM_000061        134
NM_000074        45
NM_000084        48
NM_000109        7554
NM_000116        45
NM_000117        4
NM_000132        602
NM_000133        182
NM_000166        4
NM_000167        216
NM_000169        63
NM_000194        164
NM_000202        115
NM_000206        28
NM_000216        674
NM_000240        190
NM_000252        242
NM_000266        103
NM_000273        161
NM_000276        93
NM_000284        91
NM_000291        49
NM_000292        154
```

4: SNPs per Gene

1,347 lines, format: tabular, database: hg18

Info: None

save | rerun

1	2
NM_000032	55
NM_000033	58
NM_000044	294
NM_000047	136
NM_000052	153
NM_000054	20

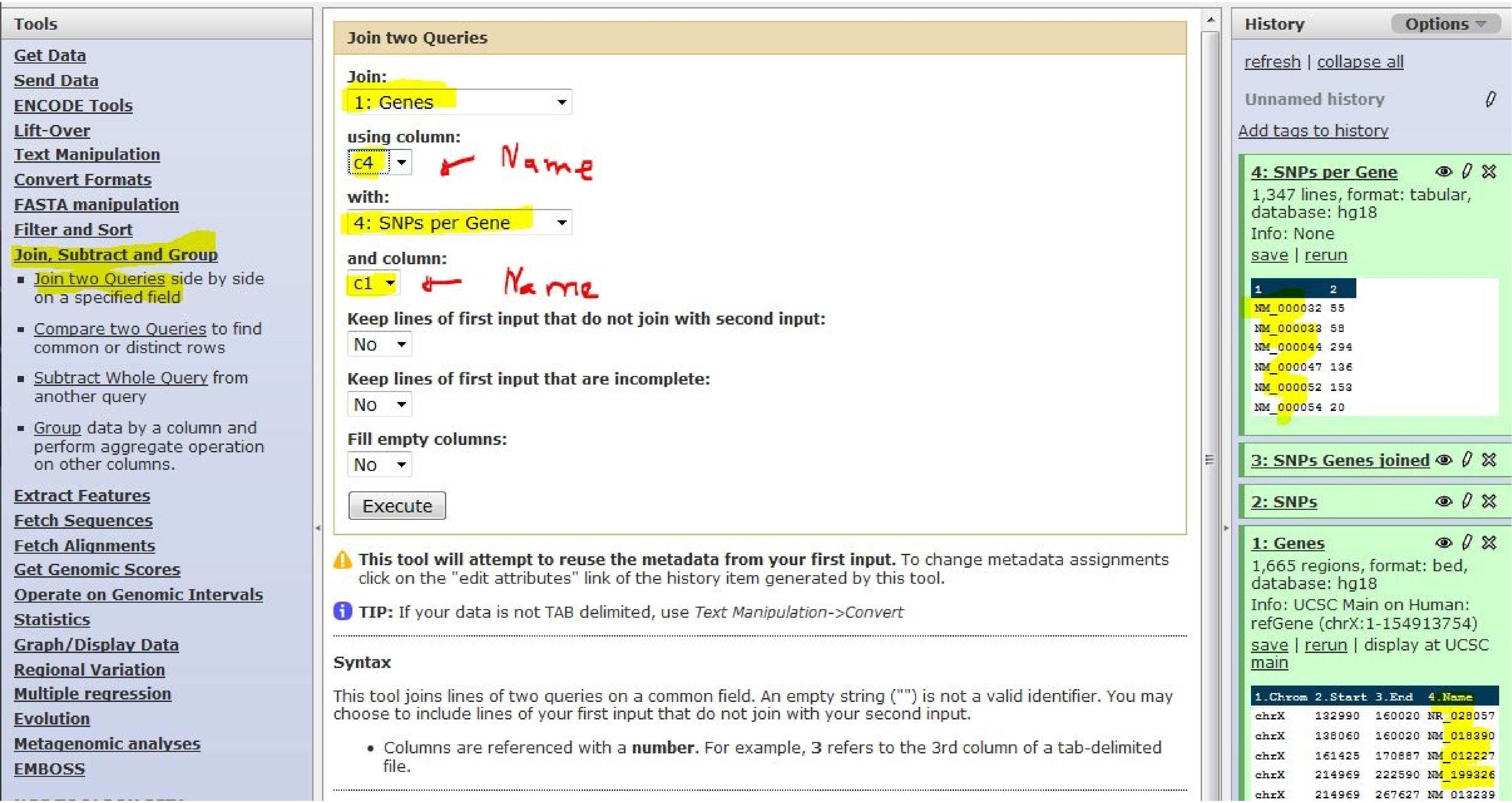
3: SNPs Genes joined

2: SNPs

1: Genes

Galaxy

Unfortunately, we lost crucial information like the gene location, Luckily, getting it back is easy: we simply append the SNPs count to our main gene data



- 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.

Join two Queries

Join: 1: Genes

using column: c4 ← Name

with: 4: SNPs per Gene

and column: c1 ← Name

Keep lines of first input that do not join with second input: No

Keep lines of first input that are incomplete: No

Fill empty columns: No

This tool will attempt to reuse the metadata from your first input. To change metadata assignments click on the "edit attributes" link of the history item generated by this tool.

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

Syntax

This tool joins lines of two queries on a common field. An empty string ("") is not a valid identifier. You may choose to include lines of your first input that do not join with your second input.

- Columns are referenced with a **number**. For example, 3 refers to the 3rd column of a tab-delimited file.

4: SNPs per Gene
1,347 lines, format: tabular, database: hg18
Info: None
save | rerun

1	2
NM_000032	55
NM_000033	58
NM_000044	294
NM_000047	136
NM_000052	153
NM_000054	20

3: SNPs Genes joined

2: SNPs

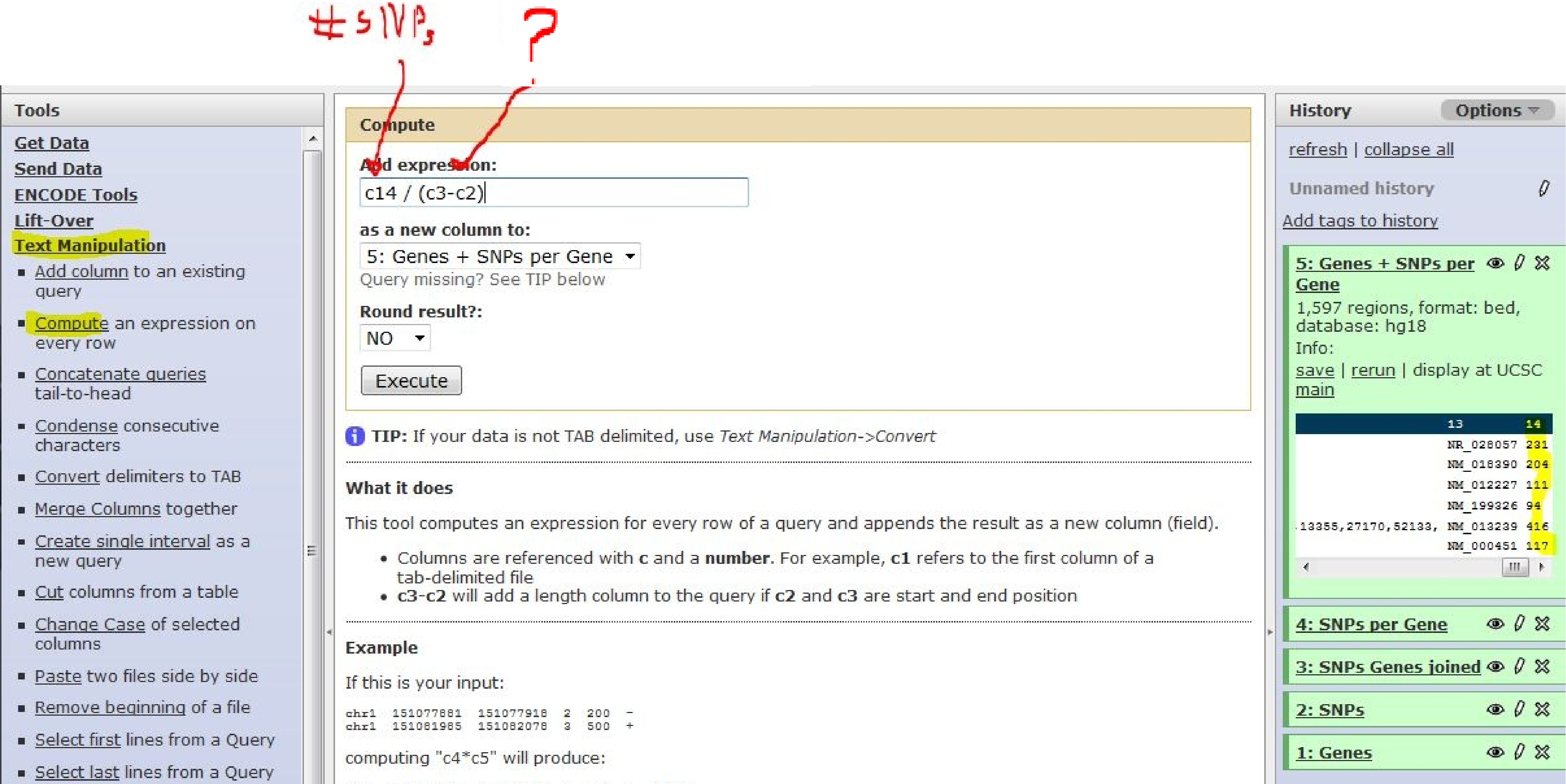
1: Genes
1,665 regions, format: bed, database: hg18
Info: UCSC Main on Human: refGene (chrX:1-154913754)
save | rerun | display at UCSC main

1.Chrom	2.Start	3.End	4.Name
chrX	132990	160020	NR_028057
chrX	138060	160020	NM_018390
chrX	161425	170887	NM_012227
chrX	214969	222590	NM_199326
chrX	214969	267627	NM_013239

Galaxy

In oder to account for variable gene size we could calculate the SNP frequency within the gene.

What does (c3 – c2) and the number this calculation yields mean?


#SNPs
?
Tools
Get Data
Send Data
ENCODE Tools
Lift-Over
Text Manipulation
Add column to an existing query
Compute an expression on every row
Concatenate queries tail-to-head
Condense consecutive characters
Convert delimiters to TAB
Merge Columns together
Create single interval as a new query
Cut columns from a table
Change Case of selected columns
Paste two files side by side
Remove beginning of a file
Select first lines from a Query
Select last lines from a Query
History
Options
refresh | collapse all
Unnamed history
Add tags to history
save | rerun | display at UCSC main
4: SNPs per Gene
3: SNPs Genes joined
2: SNPs
1: Genes

Compute

Add expression: c14 / (c3-c2)

as a new column to: 5: Genes + SNPs per Gene

Query missing? See TIP below

Round result?: NO

Execute

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

What it does

This tool computes an expression for every row of a query and appends the result as a new column (field).

- Columns are referenced with **c** and a **number**. For example, **c1** refers to the first column of a tab-delimited file
- **c3-c2** will add a length column to the query if **c2** and **c3** are start and end position

Example

If this is your input:

```
chr1   151077881   151077918   2   200   -
chr1   151081985   151082078   3   500   +
```

computing "c4*c5" will produce:

5: Genes + SNPs per Gene

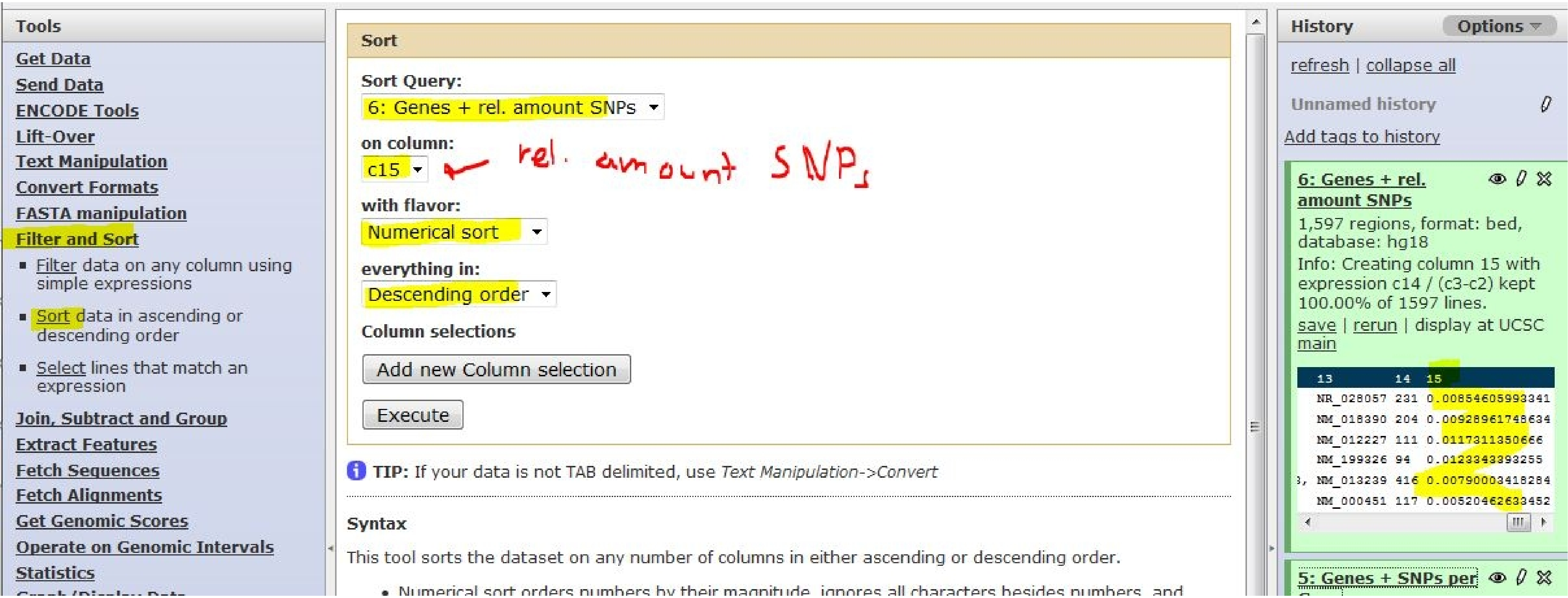
1,597 regions, format: bed, database: hg18

Info:

	13	14
	NR_028057	231
	NM_018390	204
	NM_012227	111
	NM_199326	94
.13355,27170,52133,	NM_013239	416
	NM_000451	117

Galaxy

And finaly we sort the data by the SNP frequency we estimated in the prior step



Tools

- Get Data
- Send Data
- ENCODE Tools
- Lift-Over
- Text Manipulation
- Convert Formats
- FASTA manipulation
- Filter and Sort
  - Filter data on any column using simple expressions
  - Sort data in ascending or descending order
  - Select lines that match an expression
- Join, Subtract and Group
- Extract Features
- Fetch Sequences
- Fetch Alignments
- Get Genomic Scores
- Operate on Genomic Intervals
- Statistics

Sort

Sort Query: 6: Genes + rel. amount SNPs

on column: c15 ← rel. amount SNPs

with flavor: Numerical sort

everything in: Descending order

Column selections

Add new Column selection

Execute

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

History Options

refresh | collapse all

Unnamed history

Add tags to history

6: Genes + rel. amount SNPs

1,597 regions, format: bed, database: hg18

Info: Creating column 15 with expression c14 / (c3-c2) kept 100.00% of 1597 lines.

save | rerun | display at UCSC main

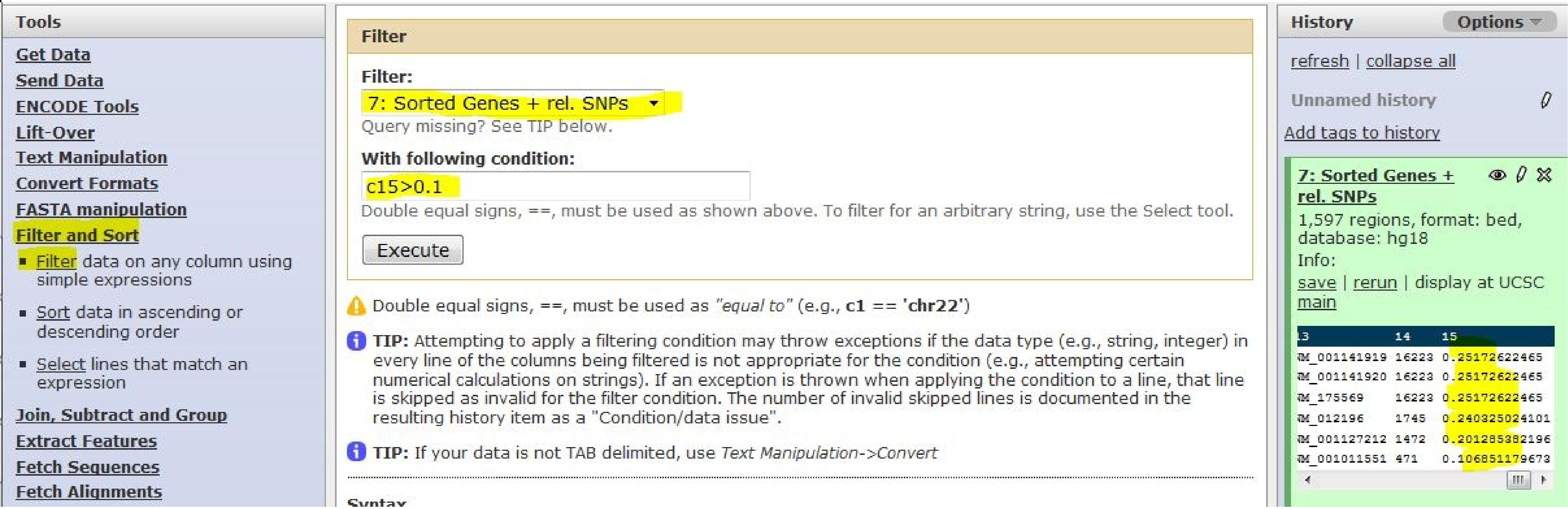
```
   13          14  15
   NR_028057   231 0.00854605993341
   NM_018390   204 0.00928961748634
   NM_012227   111 0.0117311350666
   NM_199326   94  0.0123343393255
3, NM_013239   416 0.00790003418284
   NM_000451   117 0.00520462633452
```

5: Genes + SNPs per

Galaxy

To make the list a little shorter we can cut it off below an arbitrary threshold. This is primarily necessary if we want to visualize the data.

Try to determine for yourself what a suitable threshold would be.



- Filter data on any column using simple expressions
- Sort data in ascending or descending order
- Select lines that match an expression

Filter

Filter: 7: Sorted Genes + rel. SNPs

Query missing? See TIP below.

With following condition: c15>0.1

Double equal signs, ==, must be used as shown above. To filter for an arbitrary string, use the Select tool.

⚠ Double equal signs, ==, must be used as *"equal to"* (e.g., **c1** == **'chr22'**)

ℹ **TIP:** Attempting to apply a filtering condition may throw exceptions if the data type (e.g., string, integer) in every line of the columns being filtered is not appropriate for the condition (e.g., attempting certain numerical calculations on strings). If an exception is thrown when applying the condition to a line, that line is skipped as invalid for the filter condition. The number of invalid skipped lines is documented in the resulting history item as a "Condition/data issue".

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

Syntax

7: Sorted Genes + rel. SNPs

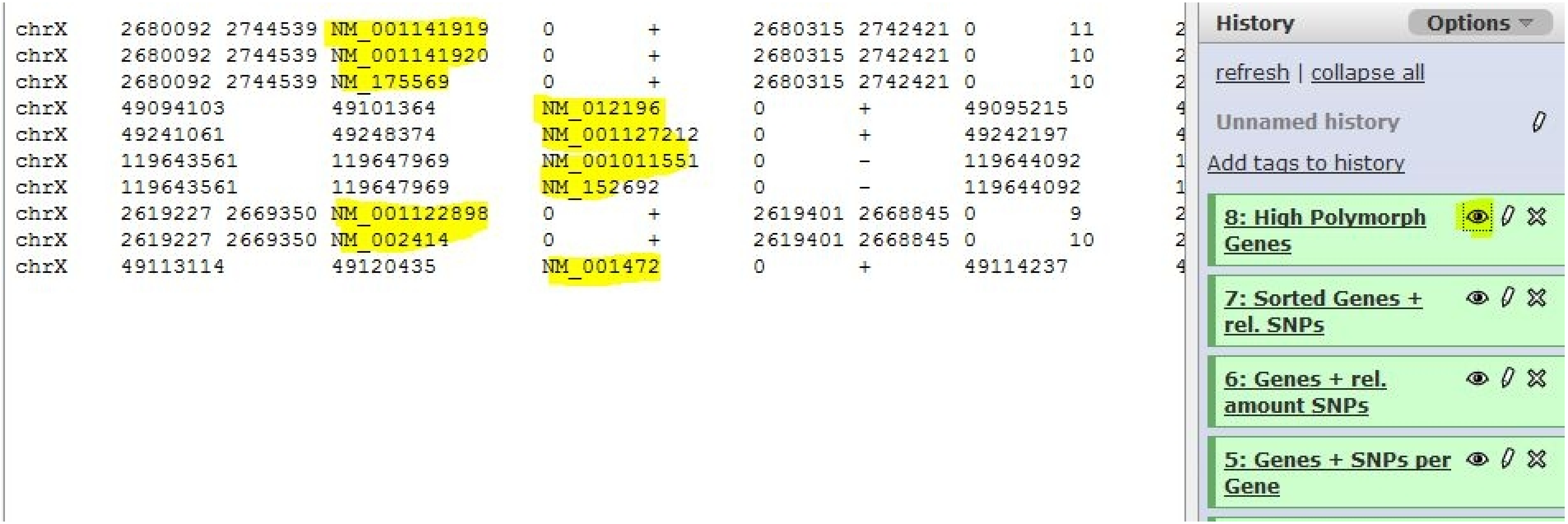
1,597 regions, format: bed, database: hg18

Info:

13	14	15
M_001141919	16223	0.25172622465
M_001141920	16223	0.25172622465
M_175569	16223	0.25172622465
M_012196	1745	0.240325024101
M_001127212	1472	0.201285382196
M_001011551	471	0.106851179673

Now the data looks nice and compact

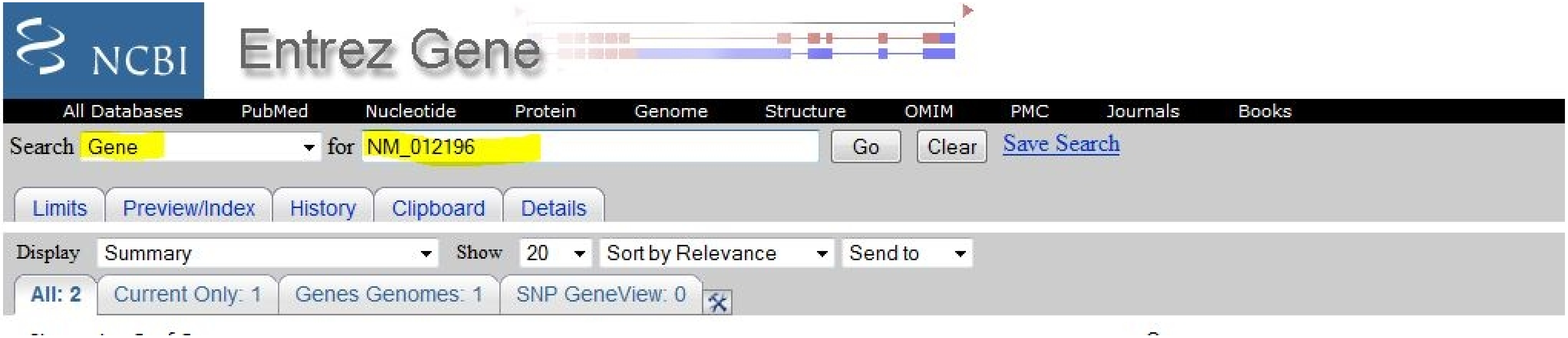
… but does this filter make sense? Is it coherent / predictive in any way? What would we expect from genes harboring so many SNPs?



```
chrX     2680092 2744539 NM_001141919     0        +        2680315 2742421 0        11        2
chrX     2680092 2744539 NM_001141920     0        +        2680315 2742421 0        10        2
chrX     2680092 2744539 NM_175569        0        +        2680315 2742421 0        10        2
chrX     49094103        49101364         NM_012196        0        +        49095215          4
chrX     49241061        49248374         NM_001127212     0        +        49242197          4
chrX     119643561       119647969        NM_001011551     0        -        119644092         1
chrX     119643561       119647969        NM_152692        0        -        119644092         1
chrX     2619227 2669350 NM_001122898     0        +        2619401 2668845 0        9         2
chrX     2619227 2669350 NM_002414        0        +        2619401 2668845 0        10        2
chrX     49113114        49120435         NM_001472        0        +        49114237          4
```

We can test predictions against existing information since we used only well annotated *RefSeq* genes.

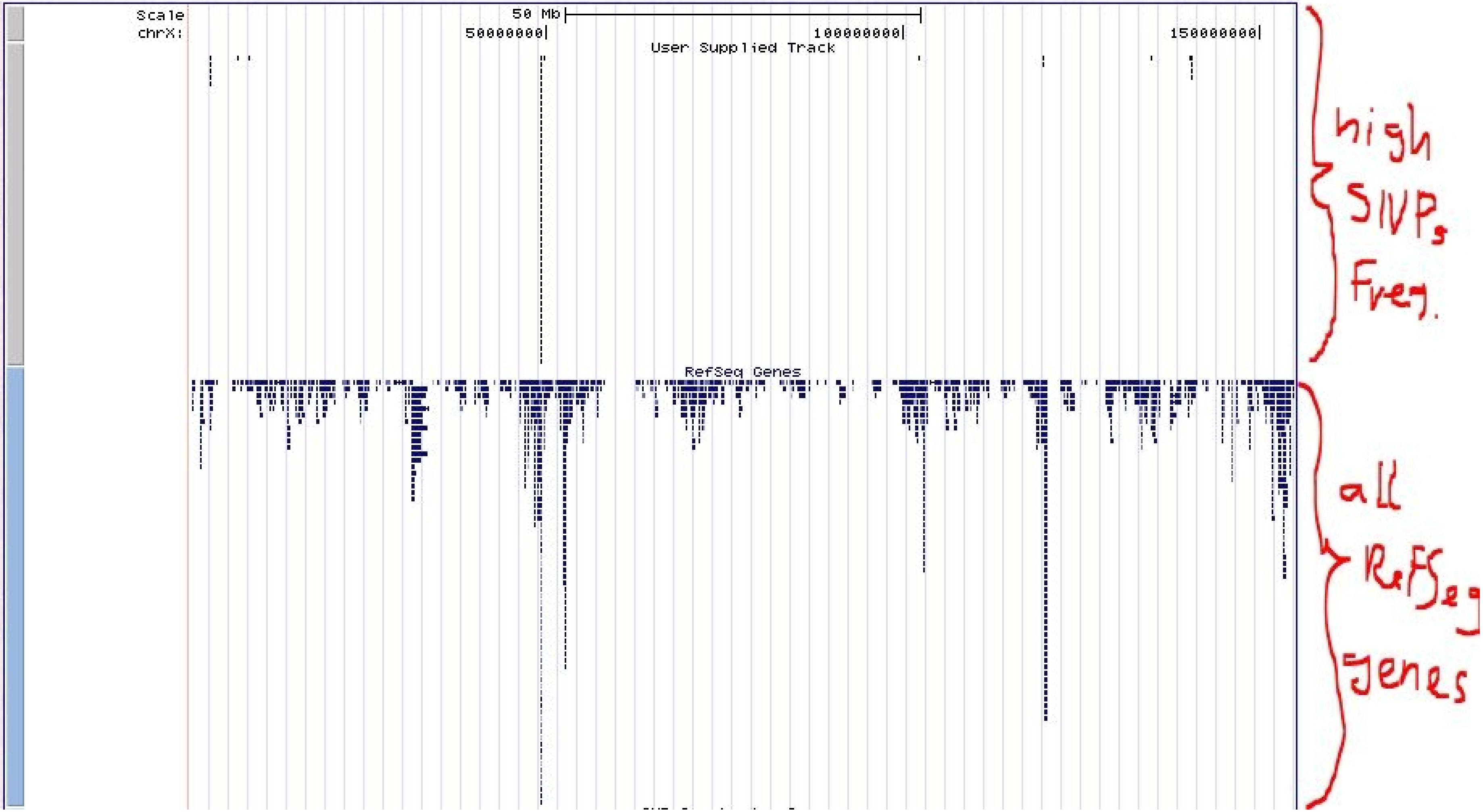
Looking at the top ten on the list of genes you whittled out … are the common characteristics, do they meet your expectations? Try to name the two most common „types“ of genes in this list.



Search Gene for NM_012196

Display Summary Show 20 Sort by Relevance

There are many other tools we could use to glean additional information from our sorted list. For example a a visualization of high SNPs frequency genes in *UCSC*. Is it clustering or chance?



Another very powerful tool within *Galaxy* are Workflows. They enable the user to save a history as though it were a machine. The user can then look at his machine, add new elements, fine tune it, see what happens if he feeds it other input data and let it work over night without his supervision.

To learn about this tool and the many others, check out the screen casts on the *Galaxy* main site.