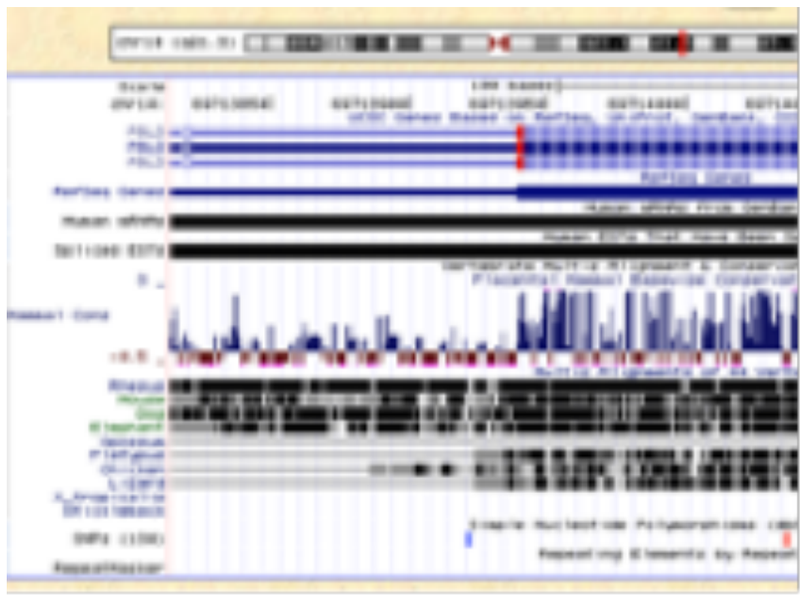
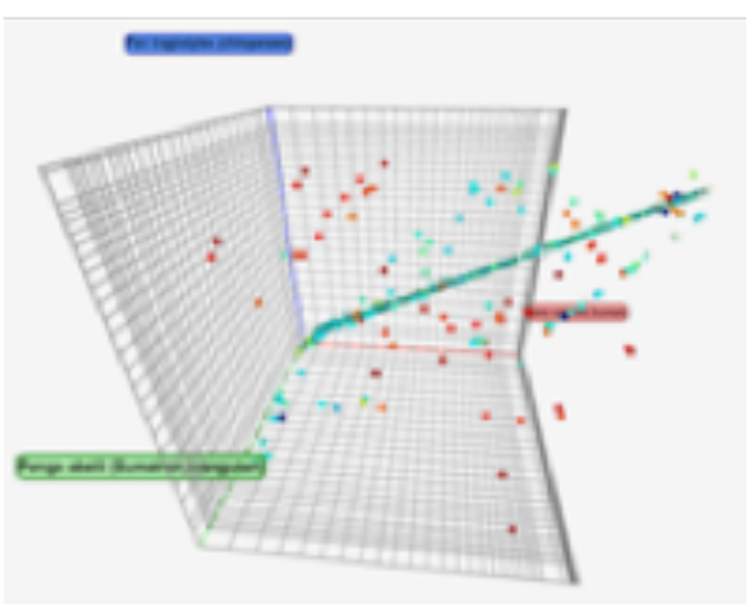
Computational Genomics

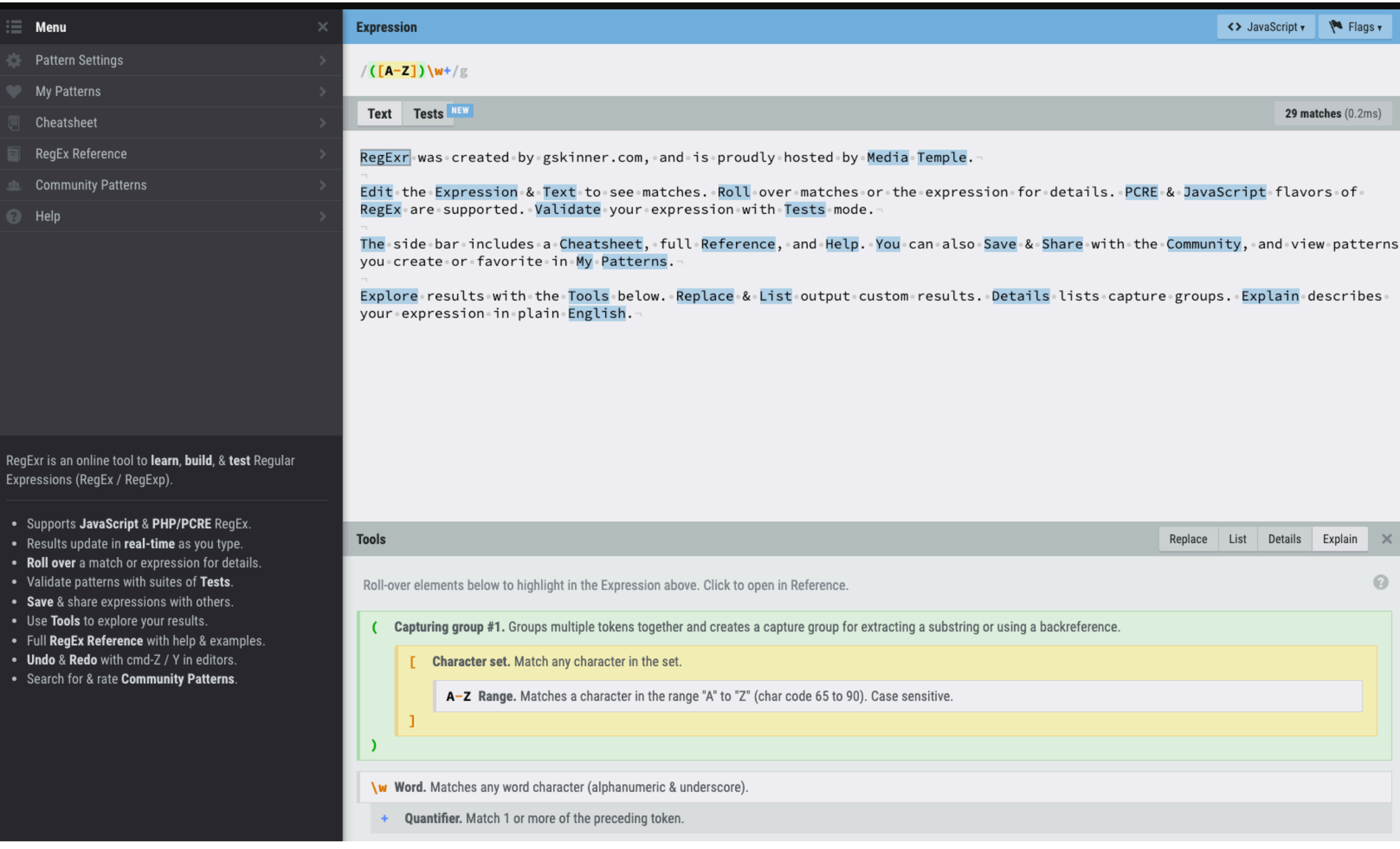
Working with Genome Files in Galaxy





Introduction to Regular Expressions

Go to regexr.com



Introduction to Regular Expressions

Regex: ^(chr.*)\t([0-9]+)\t+([0-9]+)\t+ENST([0-9]+)

Expression PCRE ▾ Flags ▾

```
/^(chr.*)\t([0-9]+)\t+([0-9]+)\t+ENST([0-9]+)/gmx
```

Text **Tests** **12 matches** (0.3ms)

```
chr1→65564→65573→ENST00000641515.2_cds_1_0_chr1_65565_f→0→+¬
chr1→69036→70008→ENST00000641515.2_cds_2_0_chr1_69037_f→0→+¬
chrX→284187→284314→ENST00000429181.6_cds_2_0_chrX_284188_f→0→+¬
chrX→288732→288787→ENST00000429181.6_cds_3_0_chrX_288733_f→0→+¬
chrY→284187→284314→ENST00000429181.6_cds_2_0_chrY_284188_f→0→+¬
chrY→288732→288787→ENST00000429181.6_cds_3_0_chrY_288733_f→0→+¬
chr10→48054→48114→ENST00000562809.1_cds_2_0_chr10_48055_r→0→-¬
chr10→48614→48725→ENST00000562809.1_cds_1_0_chr10_48615_r→0→-¬
chr11→168957→169052→ENST00000410108.5_cds_4_0_chr11_168958_r→0→-¬
chr11→180208→180404→ENST00000410108.5_cds_3_0_chr11_180209_r→0→-¬
chr12→66882→67436→ENST00000538872.6_cds_0_0_chr12_66883_f→0→+¬
chr12→99145→99214→ENST00000538872.6_cds_1_0_chr12_99146_f→0→+¬
```

Introduction to GREP in Galaxy

GREP (Global Regular Expression Print)
Where GREP Came From – *Computerphile*

- **Regular Expressions (regex)**
  - Describes text and text patterns
  - Do not have to contain literal text
  - Comprised of metacharacters
  - Metacharacters are processed by ‘parsing’

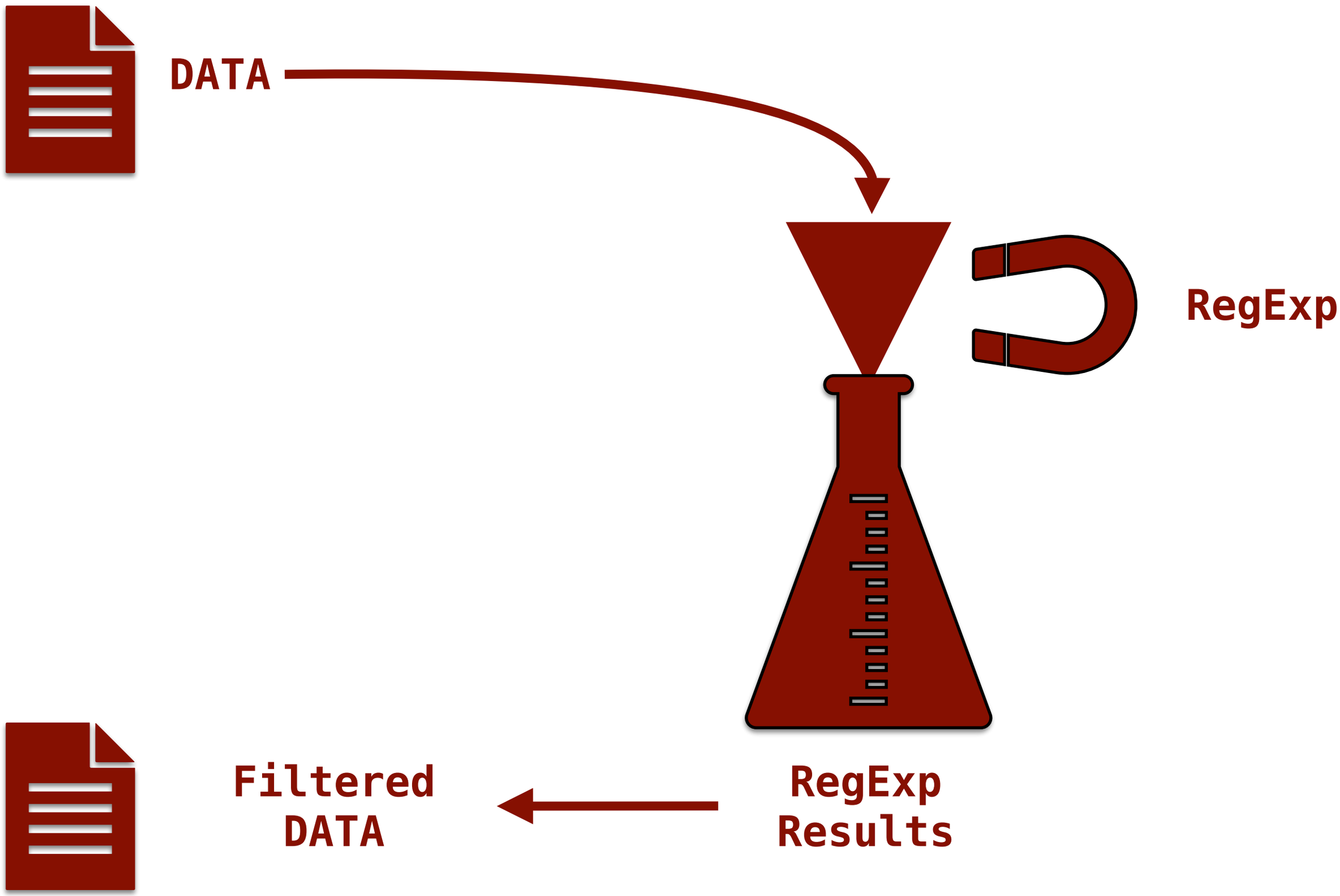
- **Text Searching versus Grep Searching**
  - Text searching is literal, whereas GREP searching is abstract and conditional
  - Text is finite - GREP is flexible
  - Text looks for characters (what) - GREP looks for locations (where and what)
  - History of GREP
  - Also known as Regular Expression Parser
  - Original command-line text search utility
    In sed (stream editor):

g/re/p or global/regular expression/print

  - GREP remembers what it found and can be directed to re-use it
  - GREP searches for patterns and most text can be described as a pattern

Introduction to GREP in Galaxy

GREP (Global Regular Expression Print)

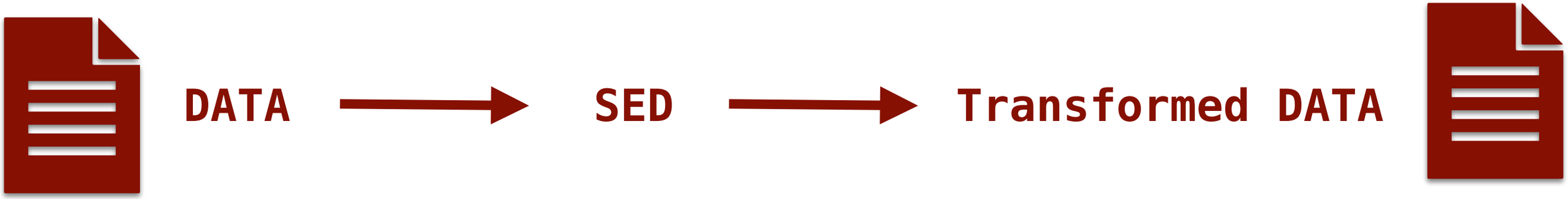


Introduction to SED: Stream Editor

Regular SED

Typical Command: sed 's/a/b/'

s=substitution
a=search string
b=replacement string



To Remove a String:

's/string//'
's/;Parent=gene//'
's/>//'

To Replace a String:

's/string/new_string/'

Working With Genome Files In Galaxy

BED Format

BED format

BED (Browser Extensible Data) format provides a flexible way to define the data lines that are displayed in an annotation track. BED lines have three required fields and nine additional optional fields. The number of fields per line must be consistent throughout any single set of data in an annotation track. The order of the optional fields is binding: lower-numbered fields must always be populated if higher-numbered fields are used.

BED information should not be mixed as explained above (BED3 should not be mixed with BED4), rather additional column information must be filled for consistency, for example with a "." in some circumstances, if the field content is to be empty. BED fields in custom tracks can be whitespace-delimited or tab-delimited. Only some variations of BED types, such as **bedDetail**, require a tab character delimitation for the detail columns.

Please note that only in custom tracks can the first lines of the file consist of header lines, which begin with the word "browser" or "track" to assist the browser in the display and interpretation of the lines of BED data following the headers. Such annotation track header lines are not permissible in downstream utilities such as bedToBigBed, which convert lines of BED text to indexed binary files.

If your data set is BED-like, but it is very large (over 50MB) and you would like to keep it on your own server, you should use the **bigBed** data format. Read a **blog post** for step-by-step instructions.

The first three required BED fields are:

1. **chrom** - The name of the chromosome (e.g. chr3, chrY, chr2_random) or scaffold (e.g. scaffold10671).
2. **chromStart** - The starting position of the feature in the chromosome or scaffold. The first base in a chromosome is numbered 0.
3. **chromEnd** - The ending position of the feature in the chromosome or scaffold. The *chromEnd* base is not included in the display of the feature, however, the number in **position format** will be represented. For example, the first 100 bases of chromosome 1 are defined as *chrom=1, chromStart=0, chromEnd=100*, and span the bases numbered 0-99 in our software (not 0-100), but will represent the position notation chr1:1-100. Read more **here**.
   chromStart and *chromEnd* can be identical, creating a feature of length 0, commonly used for insertions. For example, use *chromStart=0, chromEnd=0* to represent an insertion before the first nucleotide of a chromosome.

The 9 additional optional BED fields are:

4. **name** - Defines the name of the BED line. This label is displayed to the left of the BED line in the Genome Browser window when the track is open to full display mode or directly to the left of the item in pack mode.
5. **score** - A score between 0 and 1000. If the track line *useScore* attribute is set to 1 for this annotation data set, the *score* value will determine the level of gray in which this feature is displayed (higher numbers = darker gray). This table shows the Genome Browser's translation of BED score values into shades of gray:

shade									
score in range	≤ 166	167-277	278-388	389-499	500-611	612-722	723-833	834-944	≥ 945

6. **strand** - Defines the strand. Either "." (=no strand) or "+" or "-".
7. **thickStart** - The starting position at which the feature is drawn thickly (for example, the start codon in gene displays). When there is no thick part, thickStart and thickEnd are usually set to the chromStart position.
8. **thickEnd** - The ending position at which the feature is drawn thickly (for example the stop codon in gene displays).
9. **itemRgb** - An RGB value of the form R,G,B (e.g. 255,0,0). If the track line *itemRgb* attribute is set to "On", this RBG value will determine the display color of the data contained in this BED line. NOTE: It is recommended that a simple color scheme (eight colors or less) be used with this attribute to avoid overwhelming the color resources of the Genome Browser and your Internet browser.
10. **blockCount** - The number of blocks (exons) in the BED line.
11. **blockSizes** - A comma-separated list of the block sizes. The number of items in this list should correspond to *blockCount*.
12. **blockStarts** - A comma-separated list of block starts. All of the *blockStart* positions should be calculated relative to *chromStart*. The number of items in this list should correspond to *blockCount*.

In BED files with block definitions, the first *blockStart* value must be 0, so that the first block begins at *chromStart*. Similarly, the final *blockStart* position plus the final *blockSize* value must equal *chromEnd*. Blocks may not overlap.

```
chr22 1000 5000 cloneA 960 + 1000 5000 0 2 567,488, 0,3512
chr22 2000 6000 cloneB 900 - 2000 6000 0 2 433,399, 0,3601
```

Working With Genome Files In Galaxy

BED Format

BED (Browser Extensible Data) format provides a flexible way to define the data lines that are displayed in an annotation track. BED lines have three required fields and nine additional optional fields. The number of fields per line must be consistent throughout any single set of data in an annotation track. The order of the optional fields is binding: lower-numbered fields must always be populated if higher-numbered fields are used.

The first three required BED fields are:

1. **chrom** - The name of the chromosome (e.g. chr3, chrY, chr2_random) or scaffold (e.g. scaffold10671).
2. **chromStart** - The starting position of the feature in the chromosome or scaffold. The first base in a chromosome is numbered 0.
3. **chromEnd** - The ending position of the feature in the chromosome or scaffold. The *chromEnd* base is not included in the display of the feature, however, the number in **position format** will be represented. For example, the first 100 bases of chromosome 1 are defined as *chrom=1, chromStart=0, chromEnd=100*, and span the bases numbered 0-99 in our software (not 0-100), but will represent the position notation chr1:1-100. Read more **here**.
   chromStart and *chromEnd* can be identical, creating a feature of length 0, commonly used for insertions. For example, use *chromStart=0, chromEnd=0* to represent an insertion before the first nucleotide of a chromosome.

Working With Genome Files In Galaxy

GTF Format

GTF format

GTF (Gene Transfer Format, GTF2.2) is an extension to, and backward compatible with, GFF2. The first eight GTF fields are the same as GFF. The *feature* field is the same as GFF, with the exception that it also includes the following optional values: *5UTR, 3UTR, inter, inter_CNS,* and *intron_CNS*. The *group* field has been expanded into a list of *attributes*. Each attribute consists of a type/value pair. Attributes must end in a semi-colon, and be separated from any following attribute by exactly one space.

The attribute list must begin with the two mandatory attributes:

- **gene_id *value*** - A globally unique identifier for the genomic source of the sequence.
- **transcript_id *value*** - A globally unique identifier for the predicted transcript.

Example:
Here is an example of the ninth field in a GTF data line:

```
gene_id "Em:U62317.C22.6.mRNA"; transcript_id "Em:U62317.C22.6.mRNA"; exon_number 1
```

The Genome Browser groups together GTF lines that have the same *transcript_id* value. It only looks at features of type *exon* and *CDS*.

gene_id "Em:U62317.C22.6.mRNA"; transcript_id "Em:U62317.C22.6.mRNA"; exon_number 1

Working With Genome Files In Galaxy

GTF Format

GTF stands for Gene transfer format. It borrows from GFF, but has additional structure that warrants a separate definition and format name.

Structure is as GFF, so the fields are:
<seqname> <source> <feature> <start> <end> <score> <strand> <frame> [attributes] [comments]

Here is a simple example with 3 translated exons. Order of rows is not important.

```
381 Twinscan  CDS           380   401   .   +   0  gene_id "001"; transcript_id "001.1";
381 Twinscan  CDS           501   650   .   +   2  gene_id "001"; transcript_id "001.1";
381 Twinscan  CDS           700   707   .   +   2  gene_id "001"; transcript_id "001.1";
381 Twinscan  start_codon   380   382   .   +   0  gene_id "001"; transcript_id "001.1";
381 Twinscan  stop_codon    708   710   .   +   0  gene_id "001"; transcript_id "001.1";
```

The whitespace in this example is provided only for readability. In GTF, fields must be separated by a single TAB and no white space.

Working With Genome Files In Galaxy

GTF Format

Here is an example of a gene on the negative strand including UTR regions. Larger coordinates are 5' of smaller coordinates. Thus, the start codon is 3 bp with largest coordinates among all those bp that fall within the CDS regions. Note that the stop codon lies between the 3UTR and the CDS

```
140    Twinscan    inter          5141    8522    .    -    .    gene_id ""; transcript_id "";
140    Twinscan    inter_CNS      8523    9711    .    -    .    gene_id ""; transcript_id "";
140    Twinscan    inter          9712    13182   .    -    .    gene_id ""; transcript_id "";
140    Twinscan    3UTR           65149   65487   .    -    .    gene_id "140.000"; transcript_id "140.000.1";
140    Twinscan    3UTR           66823   66992   .    -    .    gene_id "140.000"; transcript_id "140.000.1";
140    Twinscan    stop_codon     66993   66995   .    -    0    gene_id "140.000"; transcript_id "140.000.1";
140    Twinscan    CDS            66996   66999   .    -    1    gene_id "140.000"; transcript_id "140.000.1";
140    Twinscan    intron_CNS     70103   70151   .    -    .    gene_id "140.000"; transcript_id "140.000.1";
140    Twinscan    CDS            70207   70294   .    -    2    gene_id "140.000"; transcript_id "140.000.1";
140    Twinscan    CDS            71696   71807   .    -    0    gene_id "140.000"; transcript_id "140.000.1";
140    Twinscan    start_codon    71805   71806   .    -    0    gene_id "140.000"; transcript_id "140.000.1";
140    Twinscan    start_codon    73222   73222   .    -    2    gene_id "140.000"; transcript_id "140.000.1";
140    Twinscan    CDS            73222   73222   .    -    0    gene_id "140.000"; transcript_id "140.000.1";
140    Twinscan    5UTR           73223   73504   .    -    .    gene_id "140.000"; transcript_id "140.000.1";
```

Note the frames of the coding exons. For example:

1. The first CDS (from 71807 to 71696) always has frame zero.
2. Frame of the 1st CDS =0, length =112. (3-((length - frame) mod 3)) mod 3 = 2, the frame of the 2nd CDS.
3. Frame of the 2nd CDS=2, length=88. (3-((length - frame) mod 3)) mod 3 = 1, the frame of the terminal CDS.
4. Alternatively, the frame of terminal CDS can be calculated without the rest of the gene. Length of the terminal CDS=4. length mod 3 =1, the frame of the terminal CDS.

Note the split start codon. The second start codon region has a frame of 2, since it is the second base, and has an accompanying CDS feature, since CDS always includes the start codon.

Working With Genome Files In Galaxy

GFF Format

GFF format

GFF (General Feature Format) lines are based on the Sanger **GFF2 specification**. GFF lines have nine required fields that *must* be tab-separated. If the fields are separated by spaces instead of tabs, the track will not display correctly. For more information on GFF format, refer to Sanger's **GFF page**.

Note that there is also a GFF3 specification that is not currently supported by the Browser. All GFF tracks must be formatted according to Sanger's GFF2 specification.

If you would like to obtain browser data in GFF (GTF) format, please refer to **Genes in gtf or gff format** on the Wiki.

Here is a brief description of the GFF fields:

1. **seqname** - The name of the sequence. Must be a chromosome or scaffold.
2. **source** - The program that generated this feature.
3. **feature** - The name of this type of feature. Some examples of standard feature types are "CDS" "start_codon" "stop_codon" and "exon"li>
4. **start** - The starting position of the feature in the sequence. The first base is numbered 1.
5. **end** - The ending position of the feature (inclusive).
6. **score** - A score between 0 and 1000. If the track line *useScore* attribute is set to 1 for this annotation data set, the *score* value will determine the level of gray in which this feature is displayed (higher numbers = darker gray). If there is no score value, enter ".".
7. **strand** - Valid entries include "+", "-", or "." (for don't know/don't care).
8. **frame** - If the feature is a coding exon, *frame* should be a number between 0-2 that represents the reading frame of the first base. If the feature is not a coding exon, the value should be ".".
9. **group** - All lines with the same group are linked together into a single item.

Example:
Here's an example of a GFF-based track. This data format require tabs and some operating systems convert tabs to spaces. If pasting doesn't work, this **example's** contents or the url itself can be pasted into the custom track text box.

```
browser position chr22:10000000-10025000
browser hide all
track name=regulatory description="TeleGene(tm) Regulatory Regions" visibility=2
chr22   TeleGene        enhancer        10000000        10001000        500     +       .       touch1
chr22   TeleGene        promoter        10010000        10010100        900     +       .       touch1
chr22   TeleGene        promoter        10020000        10025000        800     -       .       touch2
```

```
chr22   TeleGene        enhancer        10000000        10001000        500     +       .       touch1
chr22   TeleGene        promoter        10010000        10010100        900     +       .       touch1
chr22   TeleGene        promoter        10020000        10025000        800     -       .       touch2
```

Working With Genome Files In Galaxy

For Gene ENSG00000139618

Which file would you use:?

- **To determine in which Chromosome this gene is located**
- **To extract all the transcripts IDs associated with gene**
- **To extract all the proteins IDs associated with gene**
- **To extract all the Exons IDs associated with gene**
- **To determine if the gene lives in either the Watson or the Crick strand**

Working With Genome Files In Galaxy

For Gene ENSG00000139618

Tools

grep

Upload Data

Show Sections

Search in textfiles (grep)

obigrep Filters sequence file

FASTA Width formatter

Advanced Grep

newcpgreport Report CpG rich areas

cpgreport Reports all CpG rich regions

Text reformatting with awk

WORKFLOWS

All workflows

Search in textfiles (grep) (Galaxy Version 1.1.1)

Select lines from

4: Homo_sapiens.GRCh38.pep.all.fa
3: Hsapiens_BED
2: Homo_sapiens.GRCh38.109.gtf
1: Homo_sapiens.GRCh38.109.gff3

This is a batch mode input field. Separate jobs will be triggered for each dataset selection.

that

Match

Type of regex

Perl

Regular Expression

ENSG00000139618

See below for more details

Match type

case insensitive

(-i)

Show lines preceding the matched line

0

leave it at zero unless you know what you're doing. (-B)

Show lines trailing the matched line

0

leave it at zero unless you know what you're doing. (-A)

Output

text file (for further processing)

Job Resource Parameters

Specify job resource parameters

Memory (GB)

7

Maximum Job Memory

Time (hours)

24

Maximum job time

Execute

History

search datasets

L09-A

4 shown

1.97 GB

4: Homo_sapiens.GRCh38.pep.all.fa

3: Hsapiens_BED

2: Homo_sapiens.GRCh38.109.gtf

1: Homo_sapiens.GRCh38.109.gff3

Working With Genome Files In Galaxy

For Gene ENSG00000139618

GFF3

Seqid	Source	Type	Start	End	Score	Strand	Phase	Attributes
13	ensembl_havana	gene	32315086	32400268	.	+	.	ID=gene:ENSG00000139618;Name=BRCA2;biotype=protein_coding;description
13	havana	mRNA	32315086	32400268	.	+	.	ID=transcript:ENST00000544455;Parent=gene:ENSG00000139618;Name=BRC
13	havana	mRNA	32315505	32333291	.	+	.	ID=transcript:ENST00000530893;Parent=gene:ENSG00000139618;Name=BRC
13	ensembl_havana	mRNA	32315508	32400268	.	+	.	ID=transcript:ENST00000380152;Parent=gene:ENSG00000139618;Name=BRCA
13	havana	lnc_RNA	32315583	32316889	.	+	.	ID=transcript:ENST00000700199;Parent=gene:ENSG00000139618;Name=BRCA
13	havana	lnc_RNA	32315583	32326971	.	+	.	ID=transcript:ENST00000700200;Parent=gene:ENSG00000139618;Name=BRCA
13	havana	mRNA	32315585	32333290	.	+	.	ID=transcript:ENST00000700201;Parent=gene:ENSG00000139618;Name=BRCA
13	havana	mRNA	32316072	32400268	.	+	.	ID=transcript:ENST00000680887;Parent=gene:ENSG00000139618;Name=BRC
13	havana	mRNA	32316461	32400268	.	+	.	ID=transcript:ENST00000614259;Parent=gene:ENSG00000139618;Name=BRCA
13	havana	mRNA	32356428	32398233	.	+	.	ID=transcript:ENST00000665585;Parent=gene:ENSG00000139618;Name=BRC
13	havana	mRNA	32356526	32398770	.	+	.	ID=transcript:ENST00000700202;Parent=gene:ENSG00000139618;Name=BRCA
13	havana	mRNA	32370971	32379495	.	+	.	ID=transcript:ENST00000528762;Parent=gene:ENSG00000139618;Name=BRCA
13	havana	lnc_RNA	32375911	32398918	.	+	.	ID=transcript:ENST00000700203;Parent=gene:ENSG00000139618;Name=BRCA
13	havana	mRNA	32379840	32398272	.	+	.	ID=transcript:ENST00000470094;Parent=gene:ENSG00000139618;Name=BRC
13	havana	mRNA	32380007	32394933	.	+	.	ID=transcript:ENST00000666593;Parent=gene:ENSG00000139618;Name=BRC

Working With Genome Files In Galaxy

For Gene ENSG00000139618

GTF

Seqname	Source	Feature	Start	End	Score	Strand	Frame	Attributes
13	ensembl_havana	gene	32315086	32400268	.	+	.	gene_id "ENSG00000139618"; gene_version "18"; gene_name "BR
13	havana	transcript	32315086	32400268	.	+	.	gene_id "ENSG00000139618"; gene_version "18"; transcript_id "E
13	havana	exon	32315086	32315145	.	+	.	gene_id "ENSG00000139618"; gene_version "18"; transcript_id "E
13	havana	exon	32316422	32316527	.	+	.	gene_id "ENSG00000139618"; gene_version "18"; transcript_id "E
13	havana	CDS	32316461	32316527	.	+	0	gene_id "ENSG00000139618"; gene_version "18"; transcript_id "E
13	havana	start_codon	32316461	32316463	.	+	0	gene_id "ENSG00000139618"; gene_version "18"; transcript_id "E
13	havana	exon	32319077	32319325	.	+	.	gene_id "ENSG00000139618"; gene_version "18"; transcript_id "E
13	havana	CDS	32319077	32319325	.	+	2	gene_id "ENSG00000139618"; gene_version "18"; transcript_id "E
13	havana	exon	32325076	32325184	.	+	.	gene_id "ENSG00000139618"; gene_version "18"; transcript_id "E
13	havana	CDS	32325076	32325184	.	+	2	gene_id "ENSG00000139618"; gene_version "18"; transcript_id "E
13	havana	exon	32326101	32326150	.	+	.	gene_id "ENSG00000139618"; gene_version "18"; transcript_id "E
13	havana	CDS	32326101	32326150	.	+	1	gene_id "ENSG00000139618"; gene_version "18"; transcript_id "E
13	havana	exon	32326242	32326282	.	+	.	gene_id "ENSG00000139618"; gene_version "18"; transcript_id "E
13	havana	CDS	32326242	32326282	.	+	2	gene_id "ENSG00000139618"; gene_version "18"; transcript_id "E
13	havana	exon	32326499	32326613	.	+	.	gene_id "ENSG00000139618"; gene_version "18"; transcript_id "E

Working With Genome Files In Galaxy

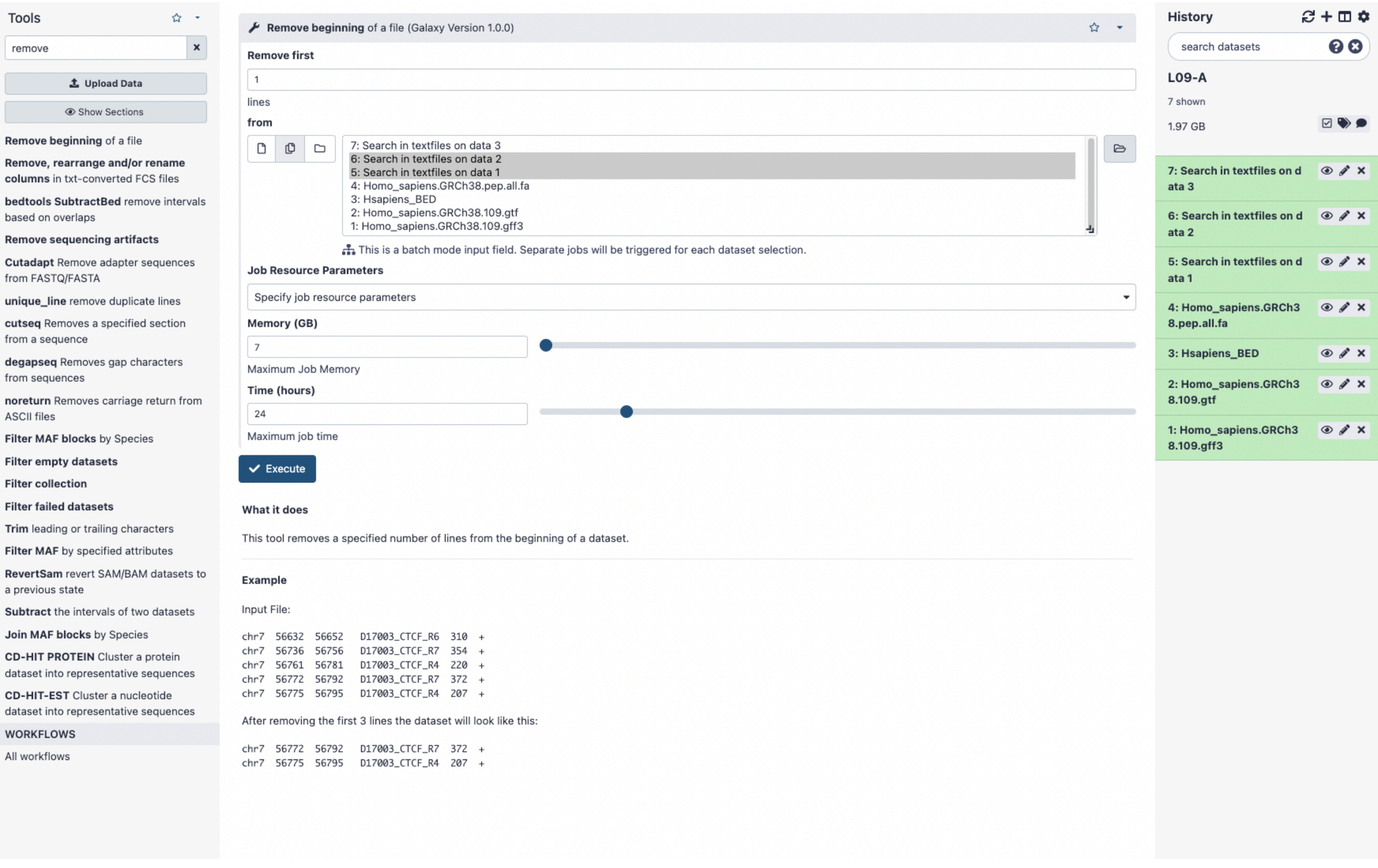
For Gene ENSG00000139618

BED

Chrom	Start	End	Name	Score	Strand	ThickStart	ThickEnd	ItemRGB	BlockCount	BlockSizes	BlockStarts

Working With Genome Files In Galaxy

For Gene ENSG00000139618



Working With Genome Files In Galaxy

For Gene ENSG00000139618

Tools

remove

Upload Data

Show Sections

Remove beginning of a file

Remove, rearrange and/or rename columns in txt-converted FCS files

bedtools SubtractBed remove intervals based on overlaps

Remove sequencing artifacts

Cutadapt Remove adapter sequences from FASTQ/FASTA

unique_line remove duplicate lines

cutseq Removes a specified section from a sequence

degapseq Removes gap characters from sequences

noreturn Removes carriage return from ASCII files

Filter MAF blocks by Species

Filter empty datasets

Filter collection

Filter failed datasets

Trim leading or trailing characters

Filter MAF by specified attributes

RevertSam revert SAM/BAM datasets to a previous state

Subtract the intervals of two datasets

Join MAF blocks by Species

CD-HIT PROTEIN Cluster a protein dataset into representative sequences

CD-HIT-EST Cluster a nucleotide dataset into representative sequences

WORKFLOWS

All workflows

Executed **Remove beginning** and successfully added 2 jobs to the queue.

The tool uses 2 inputs:

- **5: Search in textfiles on data 1**
- **6: Search in textfiles on data 2**

It produces 2 outputs:

- **8: Remove beginning on data 5**
- **9: Remove beginning on data 6**

You can check the status of queued jobs and view the resulting data by refreshing the History panel. 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

L09-A

9 shown

1.97 GB

9: Remove beginning on data 6

8: Remove beginning on data 5

7: Search in textfiles on data 3

6: Search in textfiles on data 2

5: Search in textfiles on data 1

4: Homo_sapiens.GRCh38.pep.all.fa

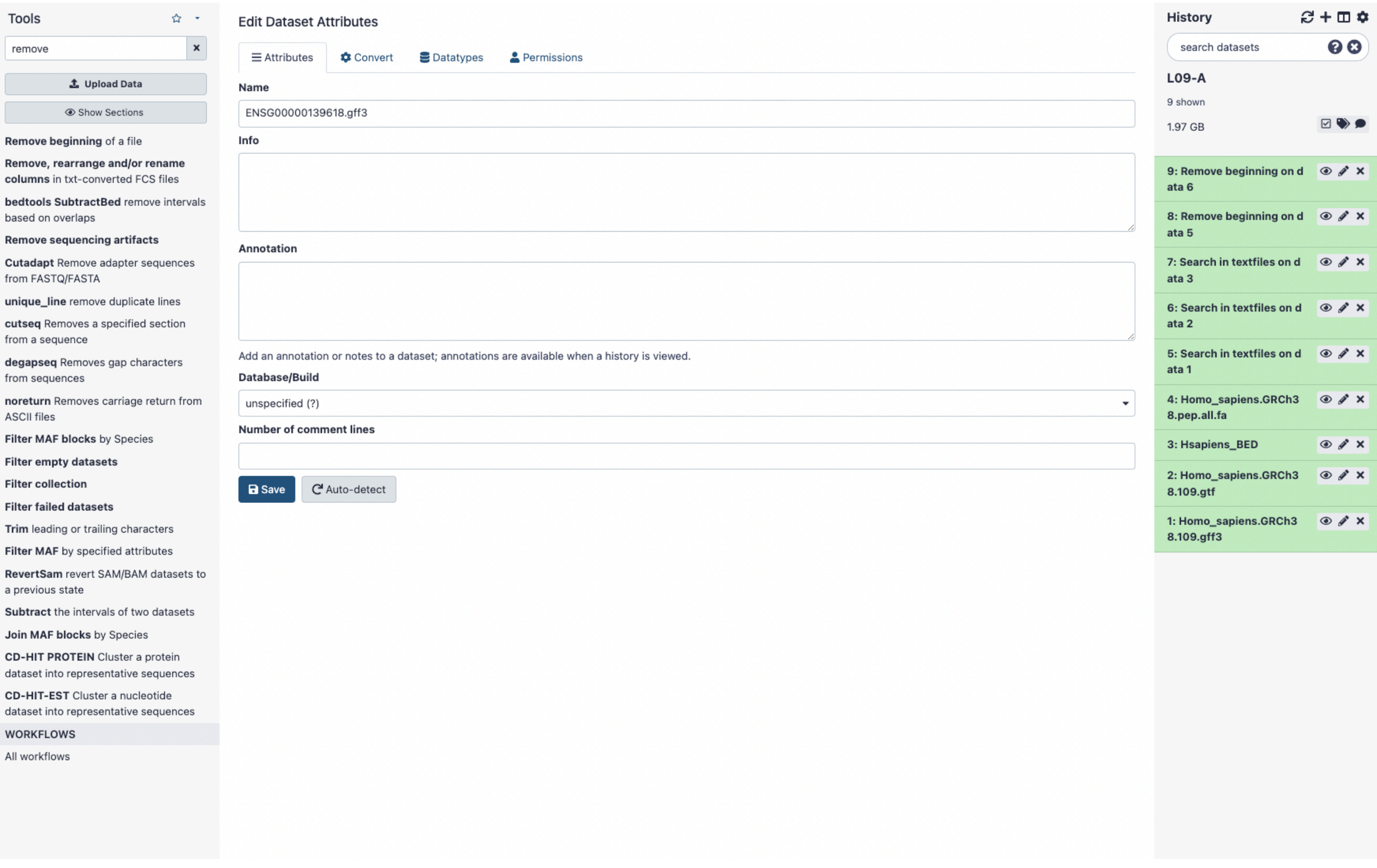
3: Hsapiens_BED

2: Homo_sapiens.GRCh38.109.gtf

1: Homo_sapiens.GRCh38.109.gff3

Working With Genome Files In Galaxy

For Gene ENSG00000139618



Working With Genome Files In Galaxy

For Gene ENSG00000139618

Tools

remove

Upload Data

Show Sections

Remove beginning of a file

Remove, rearrange and/or rename columns in txt-converted FCS files

bedtools SubtractBed remove intervals based on overlaps

Remove sequencing artifacts

Cutadapt Remove adapter sequences from FASTQ/FASTA

unique_line remove duplicate lines

cutseq Removes a specified section from a sequence

degapseq Removes gap characters from sequences

noreturn Removes carriage return from ASCII files

Filter MAF blocks by Species

Filter empty datasets

Filter collection

Filter failed datasets

Trim leading or trailing characters

Filter MAF by specified attributes

RevertSam revert SAM/BAM datasets to a previous state

Subtract the intervals of two datasets

Join MAF blocks by Species

CD-HIT PROTEIN Cluster a protein dataset into representative sequences

CD-HIT-EST Cluster a nucleotide dataset into representative sequences

WORKFLOWS

All workflows

Edit Dataset Attributes

Attributes | Convert | Datatypes | Permissions

Name

ENSG00000139618.gtf

Info

Annotation

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

Database/Build

unspecified (?)

Number of comment lines

Save | Auto-detect

History

search datasets

L09-A

9 shown

1.97 GB

9: Remove beginning on data 6

8: ENSG00000139618.gff3

7: Search in textfiles on data 3

6: Search in textfiles on data 2

5: Search in textfiles on data 1

4: Homo_sapiens.GRCh38.pep.all.fa

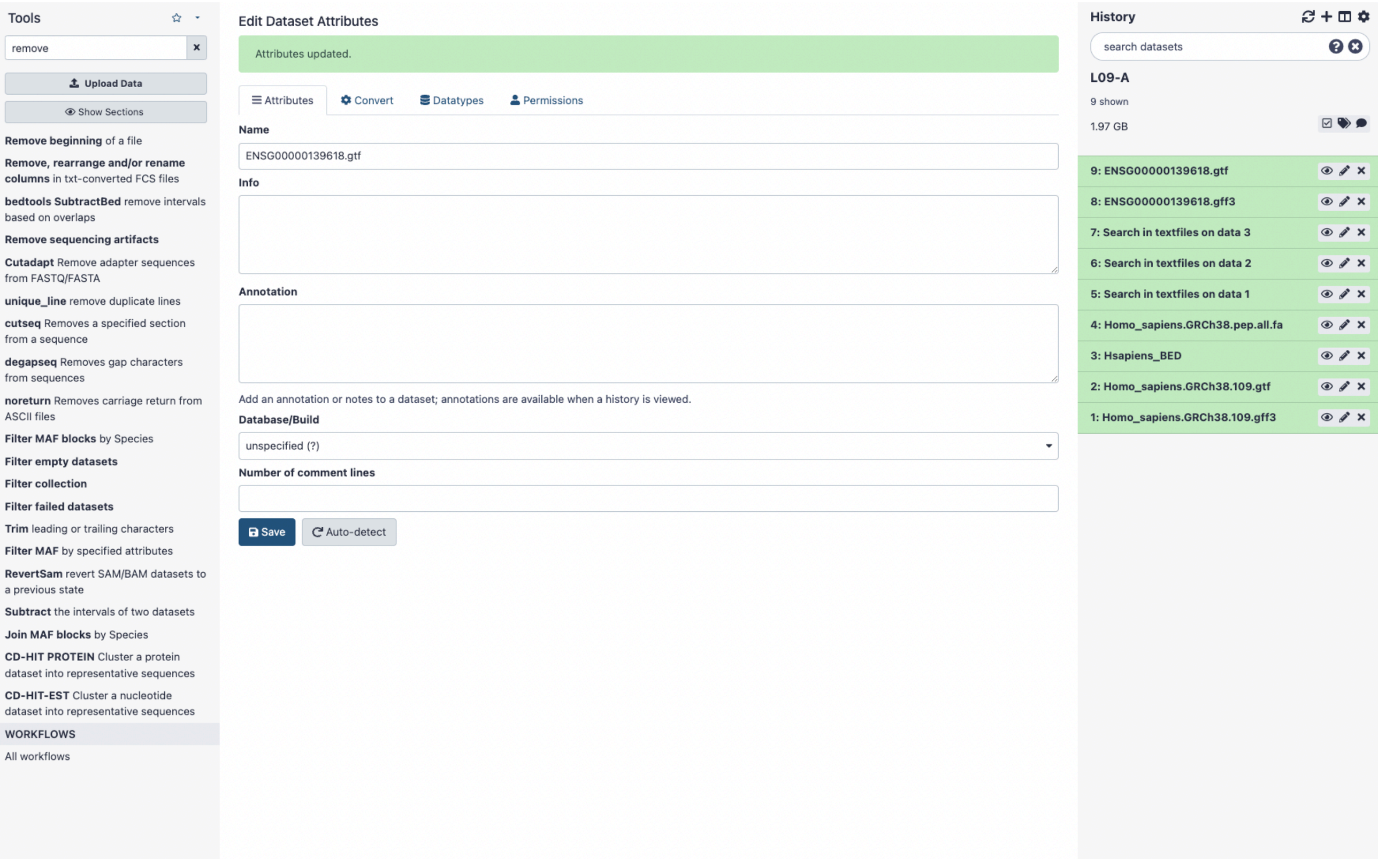
3: Hsapiens_BED

2: Homo_sapiens.GRCh38.109.gtf

1: Homo_sapiens.GRCh38.109.gff3

Working With Genome Files In Galaxy

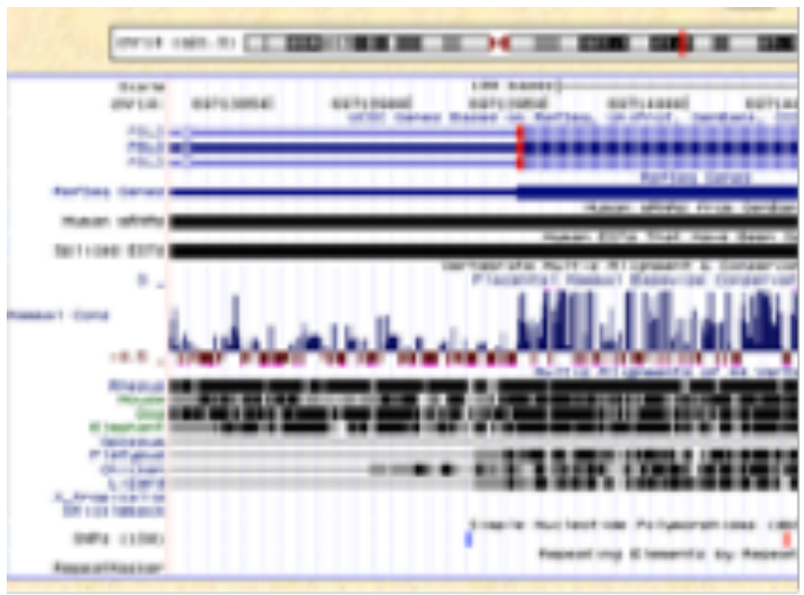
For Gene ENSG00000139618

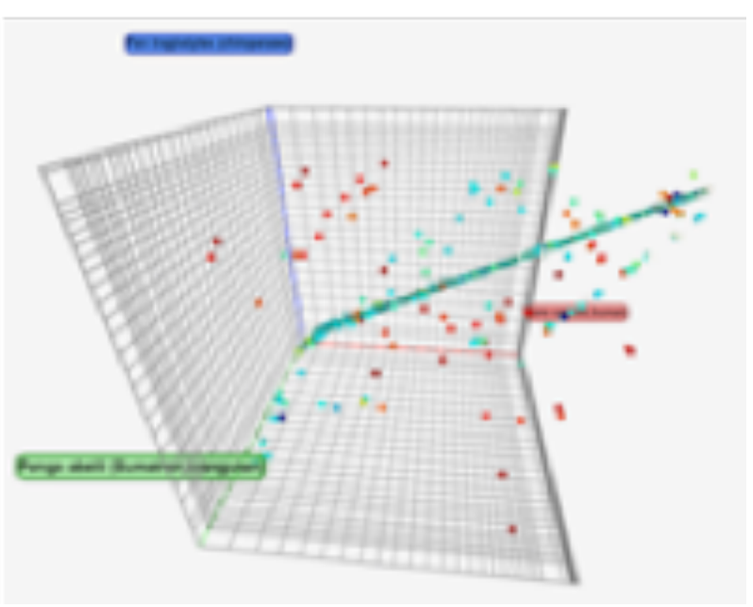


Computational Genomics

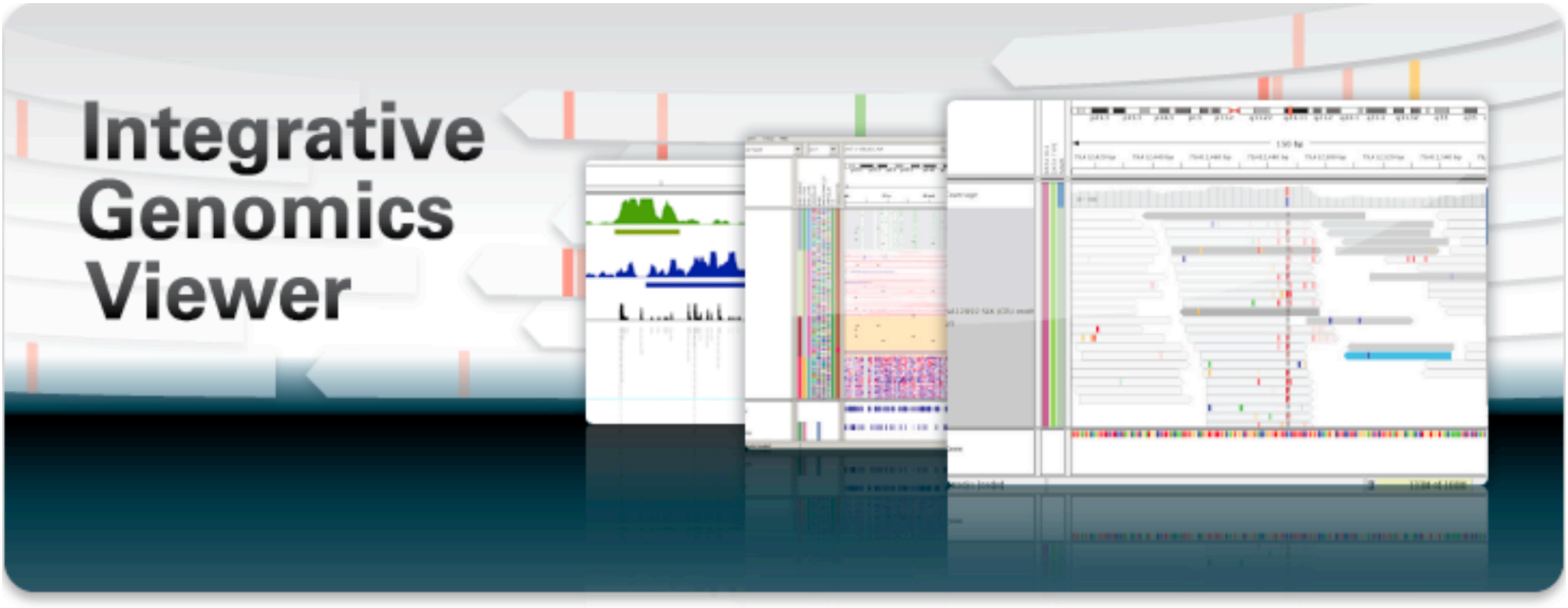
Introduction to Genome Browsers

IGV





Integrative Genome Viewer



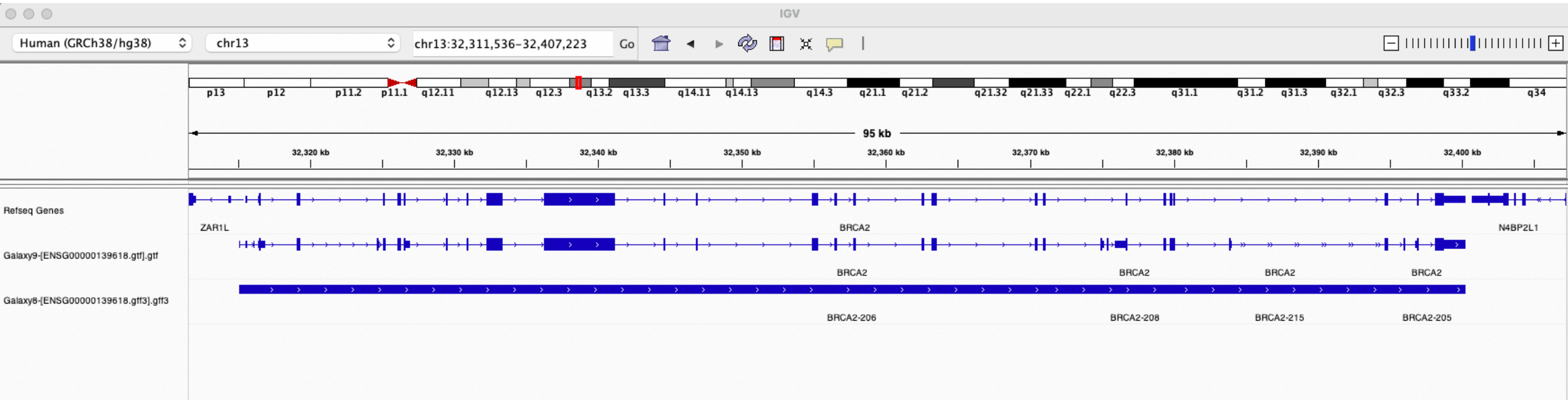
Download IGV

Working With Genome Files In Galaxy

For Gene ENSG00000139618

Working With Genome Files In Galaxy

For Gene ENSG00000139618



Working With Genome Files In Galaxy

For Gene ENSG00000139618

Working With Genome Files In Galaxy

For Gene ENSG00000139618

Tools

cut

Upload Data

Show Sections

Cut columns from a table

Advanced Cut columns from a table (cut)

Cutadapt Remove adapter sequences from FASTQ/FASTA

Filter by quality

Trimmomatic flexible read trimming tool for Illumina NGS data

Trim sequences

cutseq Removes a specified section from a sequence

WORKFLOWS

All workflows

Cut columns from a table (Galaxy Version 1.0.2)

Cut columns

c9

Delimited by

Tab

From

8: ENSG00000139618.gff3

Job Resource Parameters

Specify job resource parameters

Memory (GB)

7

Maximum Job Memory

Time (hours)

24

Maximum job time

Execute

WARNING: This tool breaks column assignments. To re-establish column assignments run the tools and click on the pencil icon in the latest history item.

The output of this tool is always in tabular format (e.g., if your original delimiters are commas, they will be replaced with tabs). For example:

Cutting columns 1 and 3 from:

```
apple,is,good
windows,is,bad
```

will give:

```
apple   good
windows bad
```

What it does

This tool selects (cuts out) specified columns from the dataset.

- Columns are specified as **c1**, **c2**, and so on. Column count begins with **1**
- Columns can be specified in any order (e.g., **c2,c1,c6**)
- If you specify more columns than actually present - empty spaces will be filled with dots

---

Example

Input dataset (six columns: c1, c2, c3, c4, c5, and c6):

```
chr1 10   1000  gene1 0 +
```

History

search datasets

L09-A

9 shown

1.97 GB

9: ENSG00000139618.gtf

8: ENSG00000139618.gff3

7: Search in textfiles on data 3

6: Search in textfiles on data 2

5: Search in textfiles on data 1

4: Homo_sapiens.GRCh38.pep.all.fa

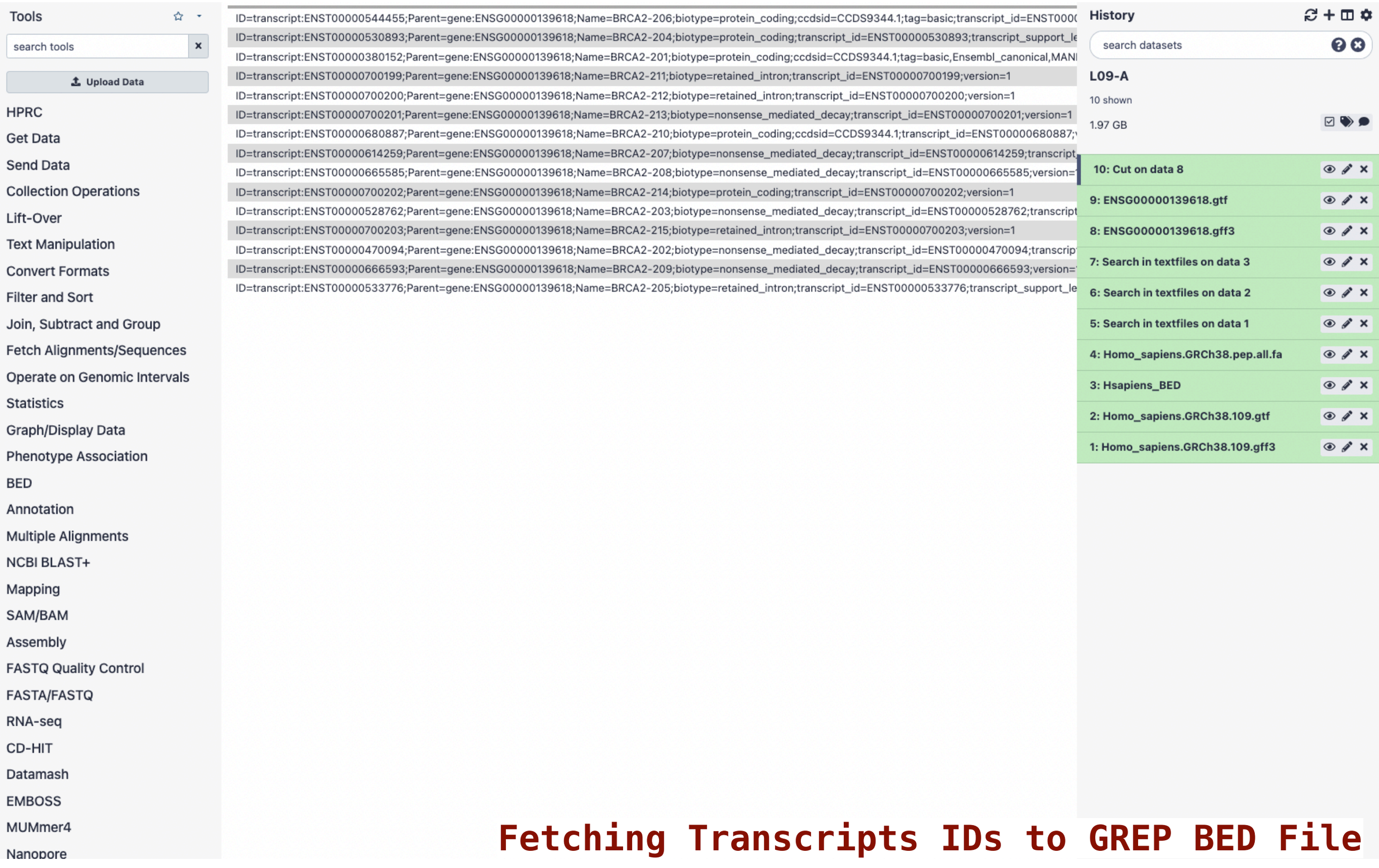
3: Hsapiens_BED

2: Homo_sapiens.GRCh38.109.gtf

1: Homo_sapiens.GRCh38.109.gff3

Working With Genome Files In Galaxy

For Gene ENSG00000139618



Fetching Transcripts IDs to GREP BED File

Working With Genome Files In Galaxy

For Gene ENSG00000139618

Tools

convert

Upload Data

Show Sections

Convert delimiters to TAB

Convert delimiters to TAB

NCBI BLAST+ convert2blastmask Convert masking information in lower-case masked FASTA input to file formats suitable for makeblastdb

Convert genome coordinates between assemblies and genomes

Convert SAM to interval

SFF converter

Quality format converter (ASCII-Numeric)

MAF to FASTA Converts a MAF formatted file to FASTA format

BED-to-bigBed converter

Tabular-to-FASTA converts tabular file to FASTA format

Wig/BedGraph-to-bigWig converter

MAF to Interval Converts a MAF formatted file to the Interval format

GFF-to-BED converter

BED-to-GFF converter

MAF to BED Converts a MAF formatted file to the BED format

FASTA-to-Tabular converter

FASTQ to FASTA converter from FASTX-toolkit

BAM-to-SAM convert BAM to SAM

SAM-to-BAM convert SAM to BAM

WORKFLOWS

All workflows

Convert delimiters to TAB (Galaxy Version 1.0.1)

Convert all

Colons

in Query

10: Cut on data 8

Job Resource Parameters

Specify job resource parameters

Memory (GB)

7

Maximum Job Memory

Time (hours)

24

Maximum job time

Execute

What it does

Converts all delimiters of a specified type into TABs. Consecutive characters are condensed. For example, if columns are separated by 5 spaces they will converted into 1 tab.

Example

- Input file:

```
chrX||151283558|151283724|NM_000808_exon_8_0_chrX_151283559_r|0|-
chrX|151370273|151370486|NM_000808_exon_9_0_chrX_151370274_r|0|-
chrX|151559494|151559583|NM_018558_exon_1_0_chrX_151559495_f|0|+
chrX|151564643|151564711|NM_018558_exon_2_0_chrX_151564644_f||||0|+
```

- Converting all pipe delimiters of the above file to TABs will get:

```
chrX  151283558  151283724  NM_000808_exon_8_0_chrX_151283559_r  0  -
chrX  151370273  151370486  NM_000808_exon_9_0_chrX_151370274_r  0  -
chrX  151559494  151559583  NM_018558_exon_1_0_chrX_151559495_f  0  +
chrX  151564643  151564711  NM_018558_exon_2_0_chrX_151564644_f  0  +
```

Requirements: ?
- python (Version 3.8)

History

search datasets

L09-A

10 shown

1.97 GB

10: Cut on data 8

9: ENSG00000139618.gtf

8: ENSG00000139618.gff3

7: Search in textfiles on data 3

6: Search in textfiles on data 2

5: Search in textfiles on data 1

4: Homo_sapiens.GRCh38.pep.all.fa

3: Hsapiens_BED

2: Homo_sapiens.GRCh38.109.gtf

1: Homo_sapiens.GRCh38.109.gff3

Fetching Transcripts IDs to GREP BED File

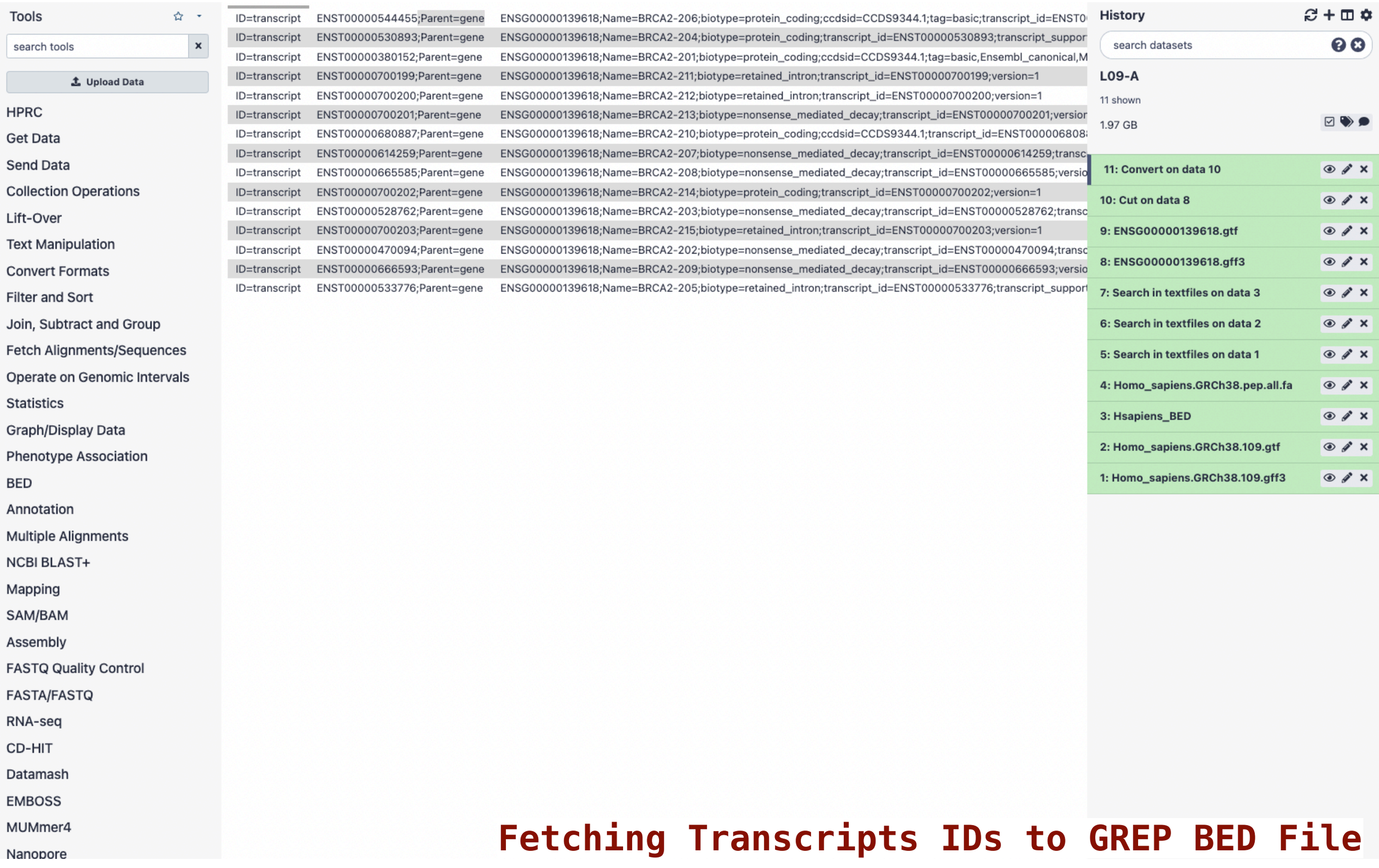
Working With Genome Files In Galaxy

For Gene ENSG00000139618

Fetching Transcripts IDs to GREP BED File

Working With Genome Files In Galaxy

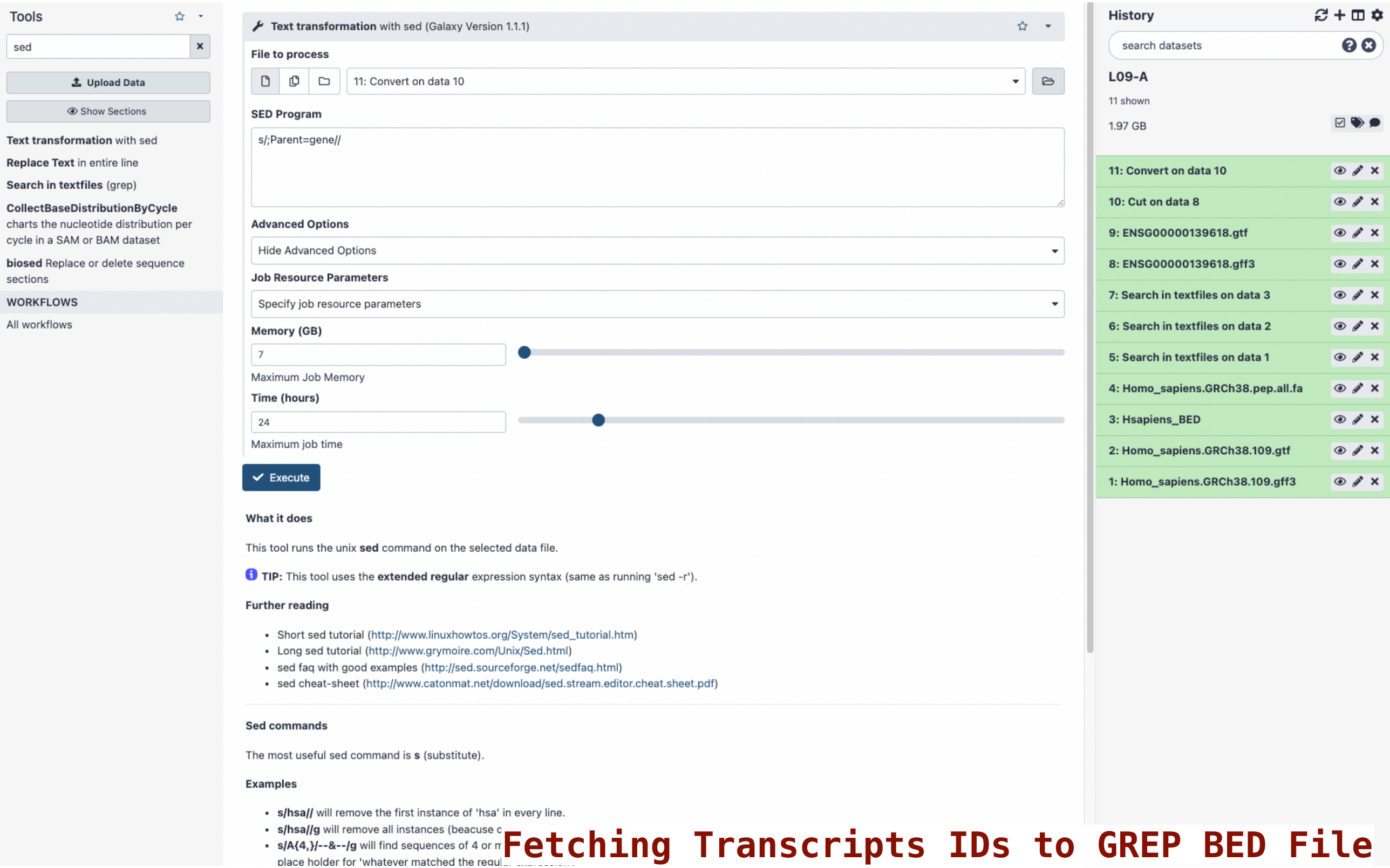
For Gene ENSG00000139618



Fetching Transcripts IDs to GREP BED File

Working With Genome Files In Galaxy

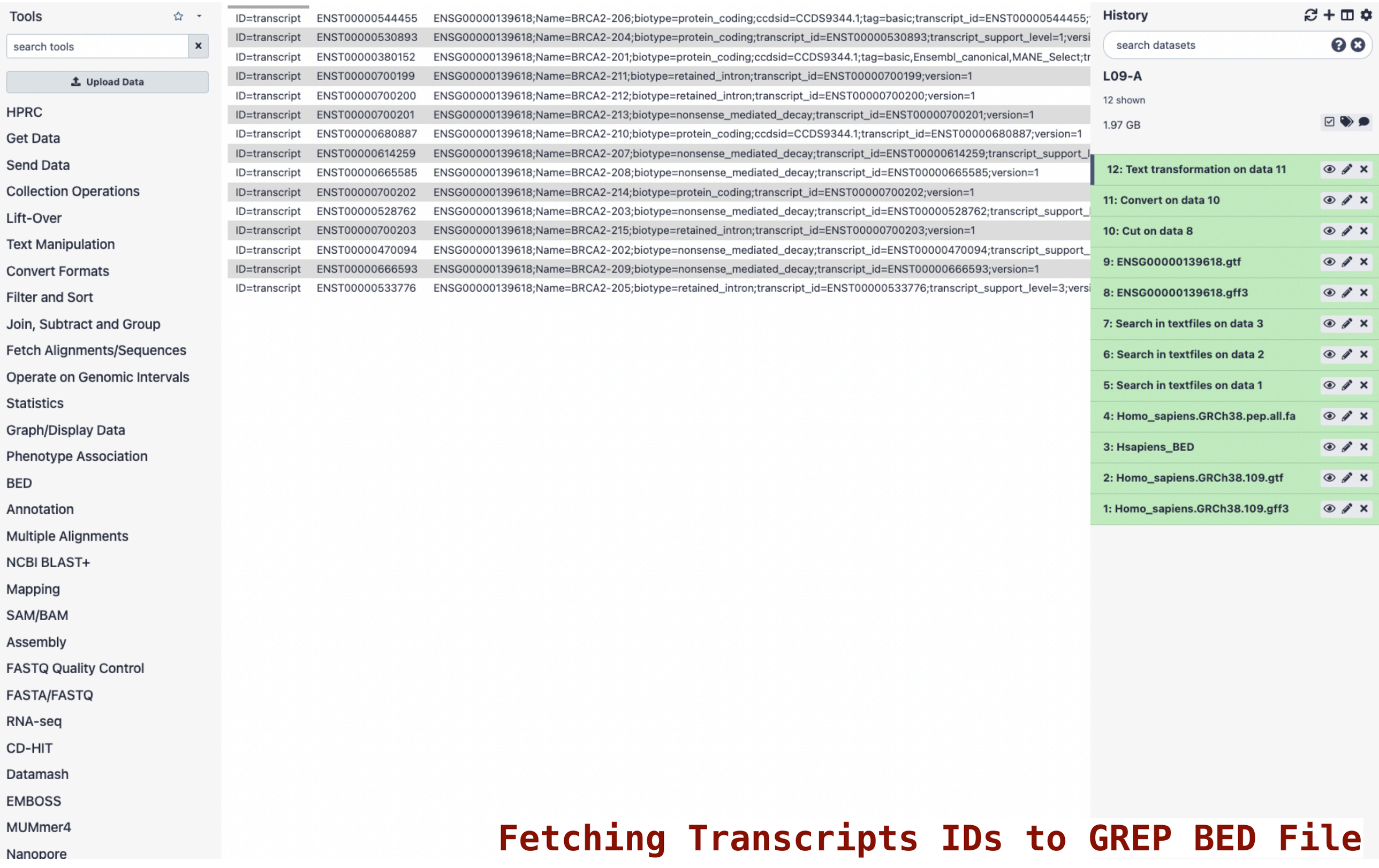
For Gene ENSG00000139618



Fetching Transcripts IDs to GREP BED File

Working With Genome Files In Galaxy

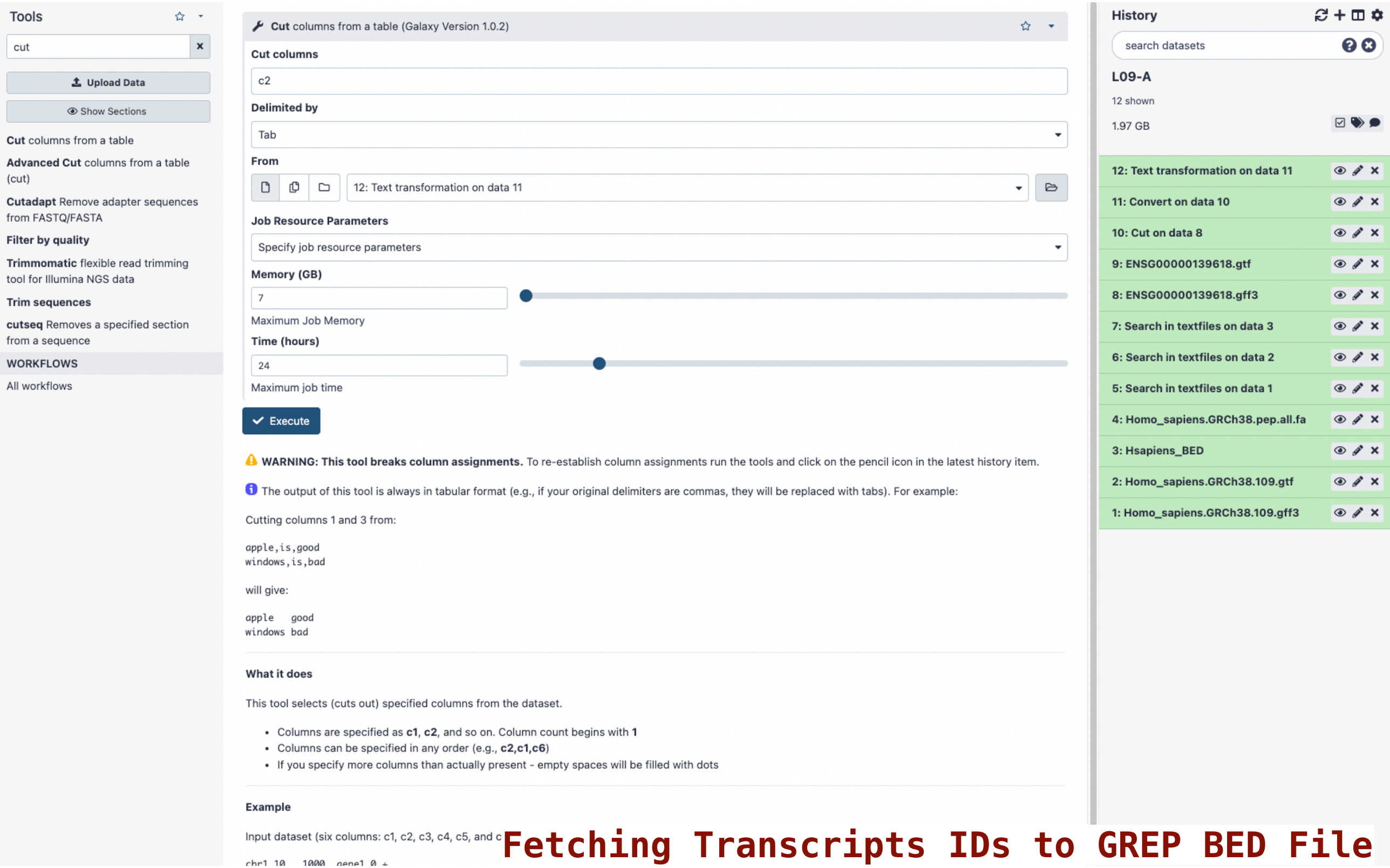
For Gene ENSG00000139618



Fetching Transcripts IDs to GREP BED File

Working With Genome Files In Galaxy

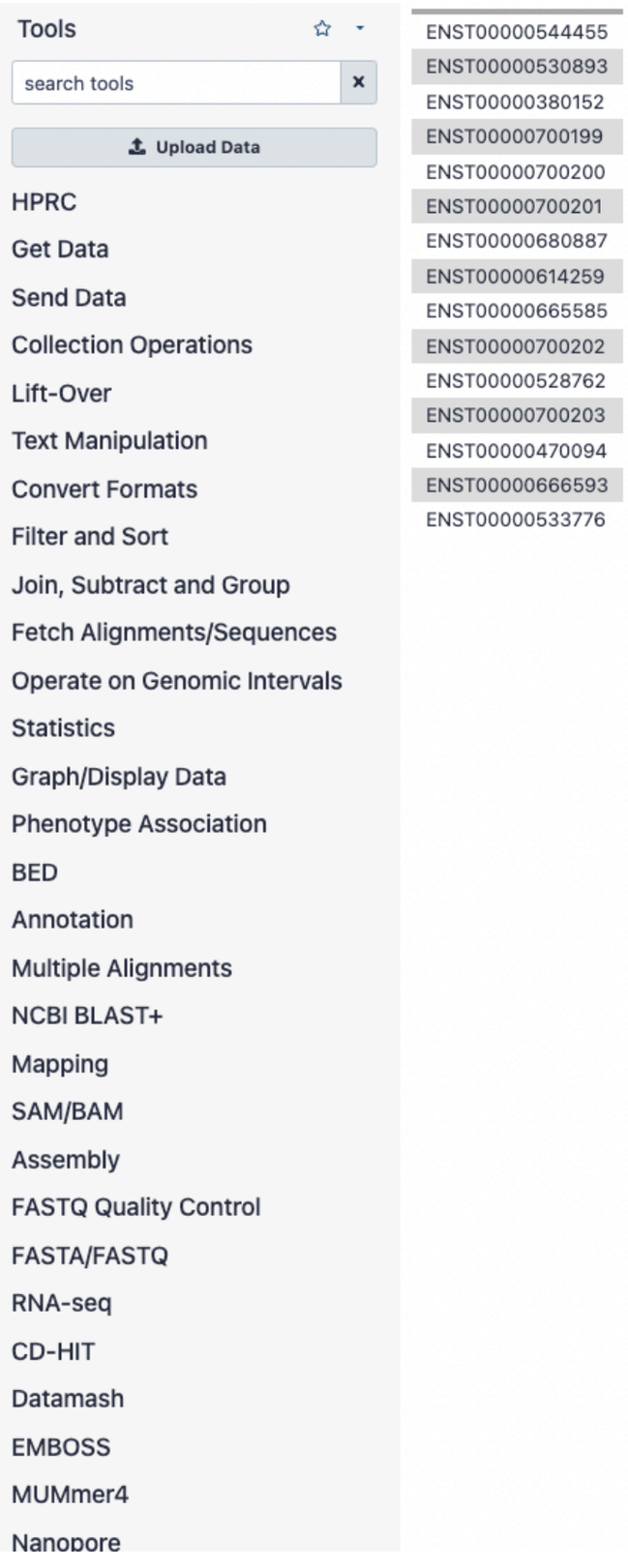
For Gene ENSG00000139618

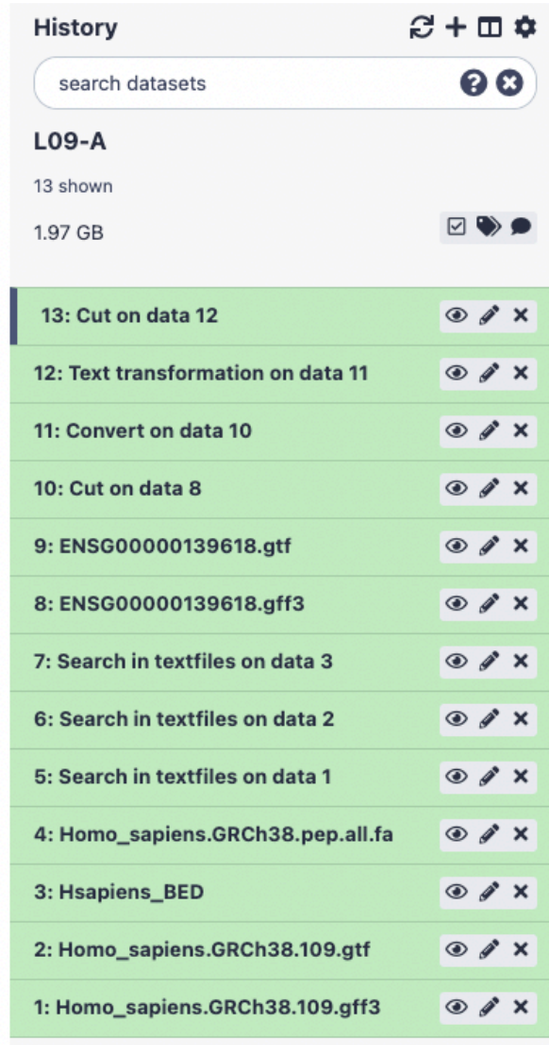


Fetching Transcripts IDs to GREP BED File

Working With Genome Files In Galaxy

For Gene ENSG00000139618

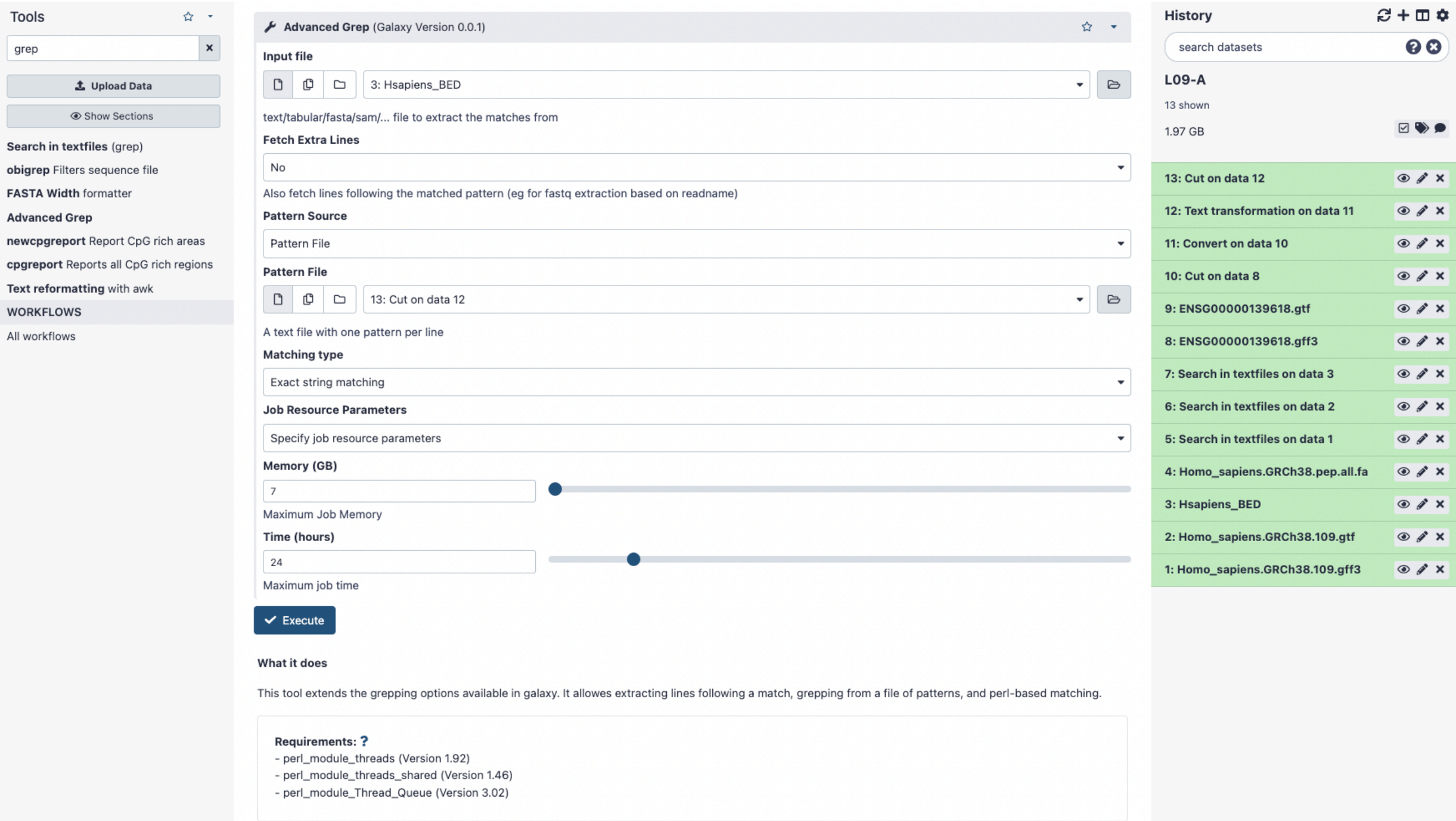




Fetching Transcripts IDs to GREP BED File

Working With Genome Files In Galaxy

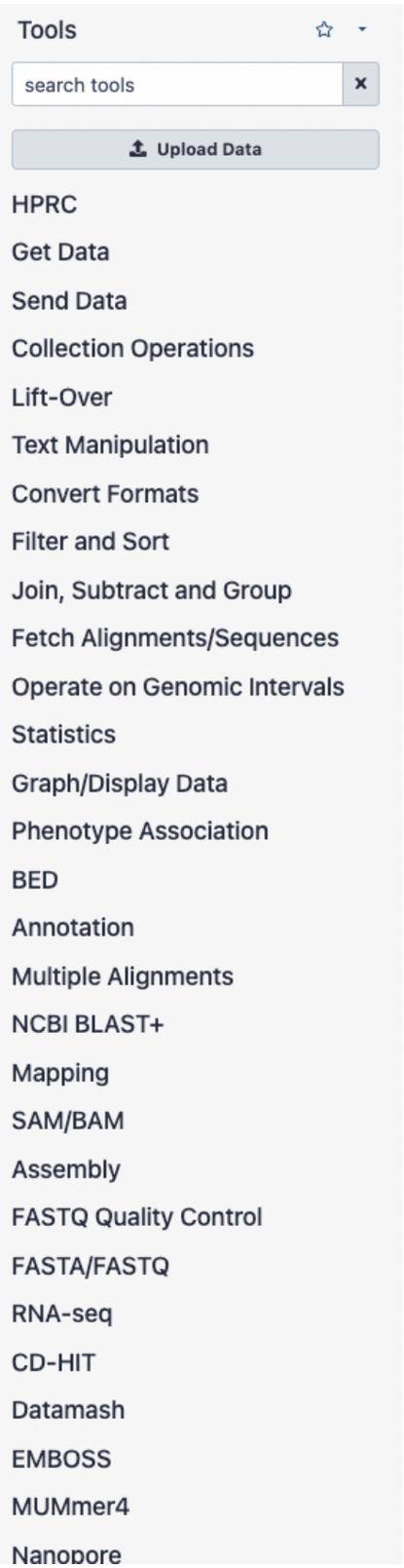
For Gene ENSG00000139618



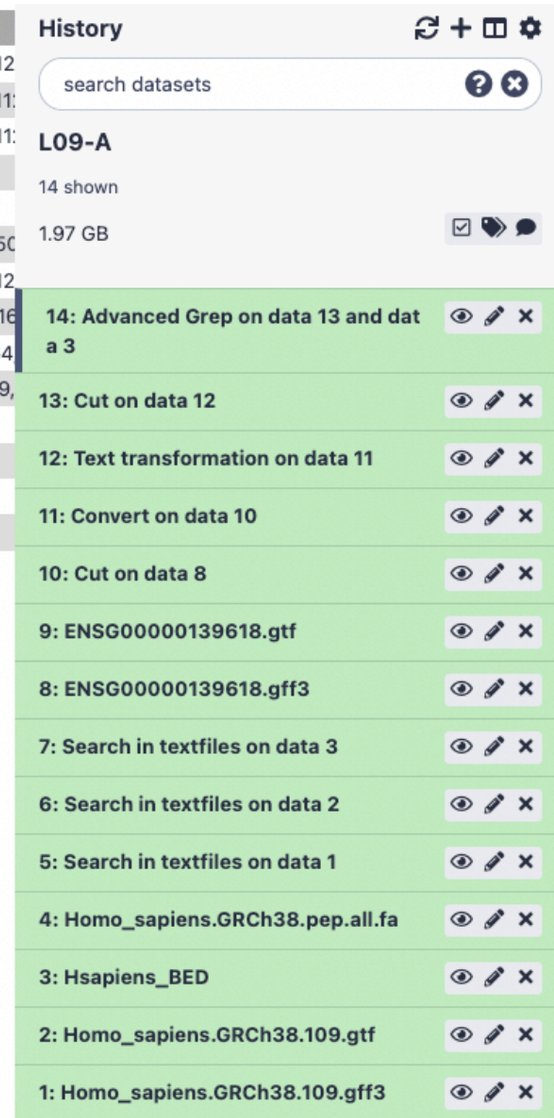
Fetching Transcripts IDs to GREP BED File

Working With Genome Files In Galaxy

For Gene ENSG00000139618



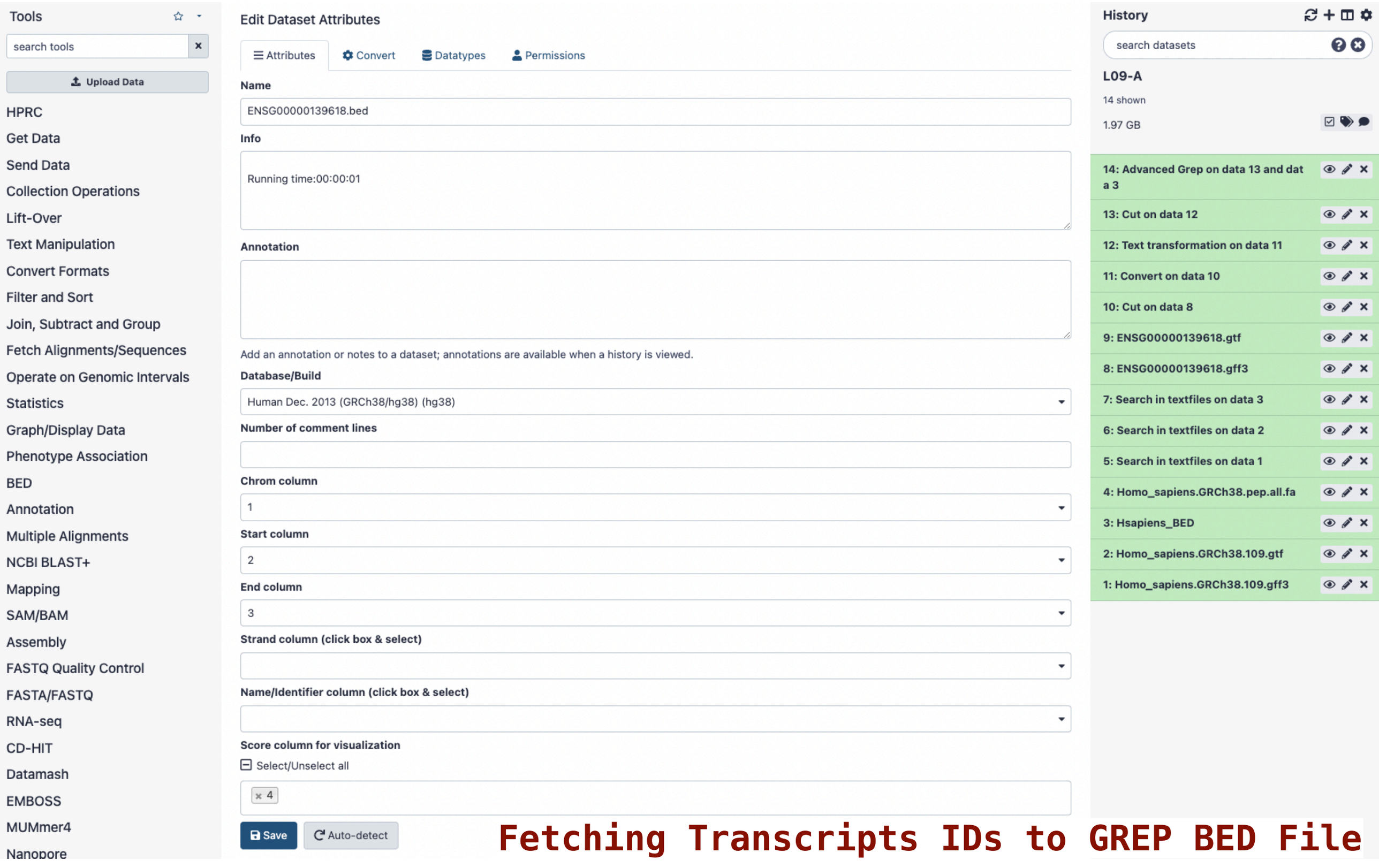
Chrom	Start	End	Name	Score	Strand	ThickStart	ThickEnd	ItemRGB	BlockCount	BlockSizes
chr13	32315085	32400268	ENST00000544455.6	0	+	32316460	32398770	0,0,0	27	60,106,249,109,50,41,115,50,112
chr13	32315504	32333291	ENST00000530893.6	0	+	32325128	32333291	0,0,0	10	163,102,249,109,50,41,115,50,11
chr13	32315507	32400268	ENST00000380152.8	0	+	32316460	32398770	0,0,0	27	160,106,249,109,50,41,115,50,11
chr13	32315582	32316889	ENST00000700199.1	0	+	32315582	32315582	0,0,0	2	85,468,
chr13	32315582	32326971	ENST00000700200.1	0	+	32315582	32315582	0,0,0	6	85,106,105,50,41,473,
chr13	32315584	32333290	ENST00000700201.1	0	+	32316460	32324731	0,0,0	11	83,106,249,169,109,50,41,115,50
chr13	32316071	32400268	ENST00000680887.1	0	+	32316460	32398770	0,0,0	27	86,106,249,109,50,41,115,50,112
chr13	32316460	32400268	ENST00000614259.2	0	+	32316460	32362659	0,0,0	26	67,249,109,50,41,115,50,112,1116
chr13	32356427	32398233	ENST00000665585.1	0	+	32356427	32375080	0,0,0	15	182,188,171,355,156,145,249,64
chr13	32356525	32398770	ENST00000700202.1	0	+	32356525	32398770	0,0,0	13	84,188,171,355,156,145,122,199,
chr13	32370970	32379495	ENST00000528762.1	0	+	32370970	32376670	0,0,0	4	130,64,122,179,
chr13	32375910	32398918	ENST00000700203.1	0	+	32375910	32375910	0,0,0	7	881,199,164,139,245,147,757,
chr13	32379839	32398272	ENST00000470094.1	0	+	32379839	32396092	0,0,0	6	74,139,245,126,147,111,
chr13	32380006	32394933	ENST00000666593.1	0	+	32380006	32383930	0,0,0	3	139,139,245,
chr13	32396808	32398448	ENST00000533776.1	0	+	32396808	32396808	0,0,0	2	236,287,



Fetching Transcripts IDs to GREP BED File

Working With Genome Files In Galaxy

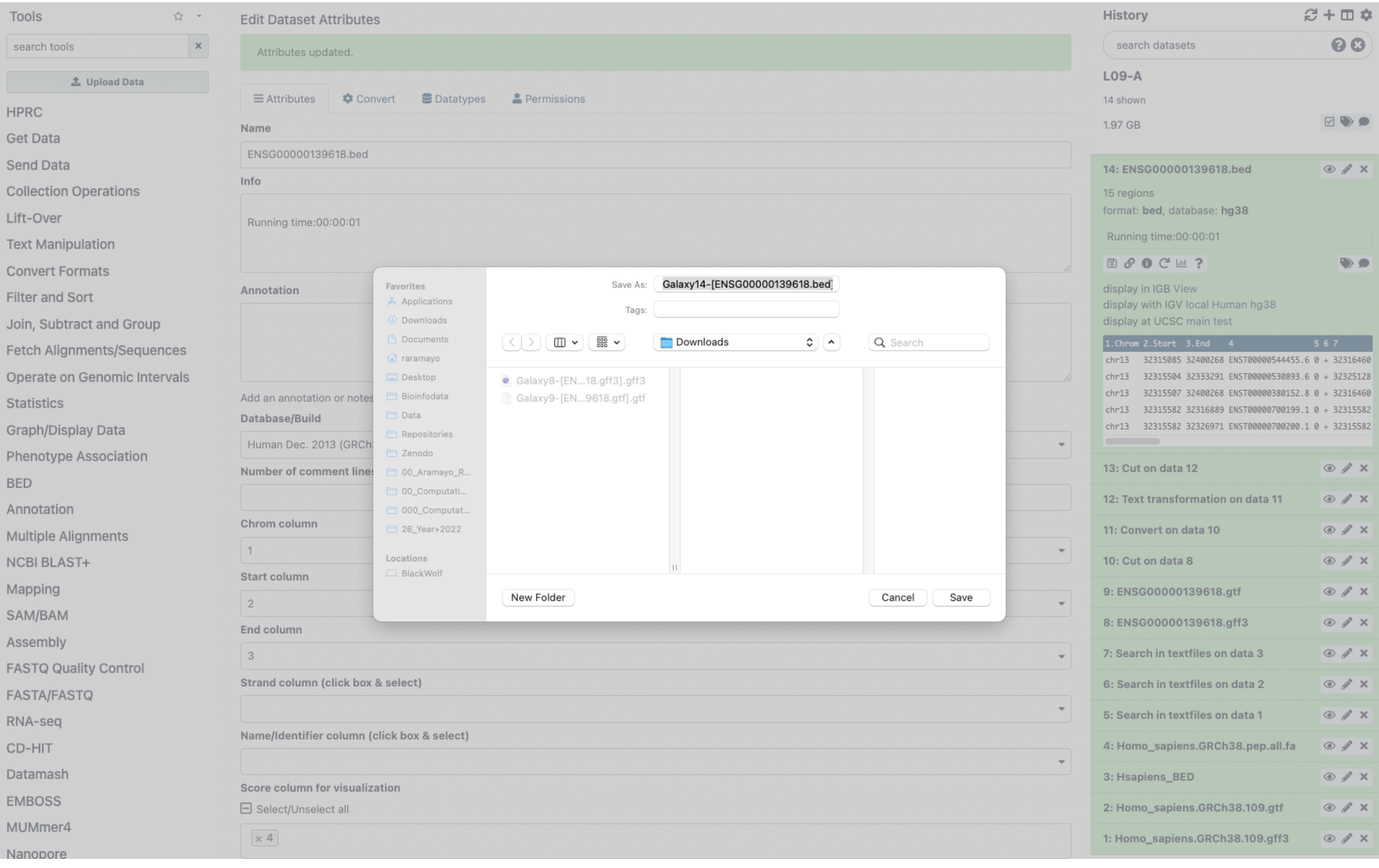
For Gene ENSG00000139618



Fetching Transcripts IDs to GREP BED File

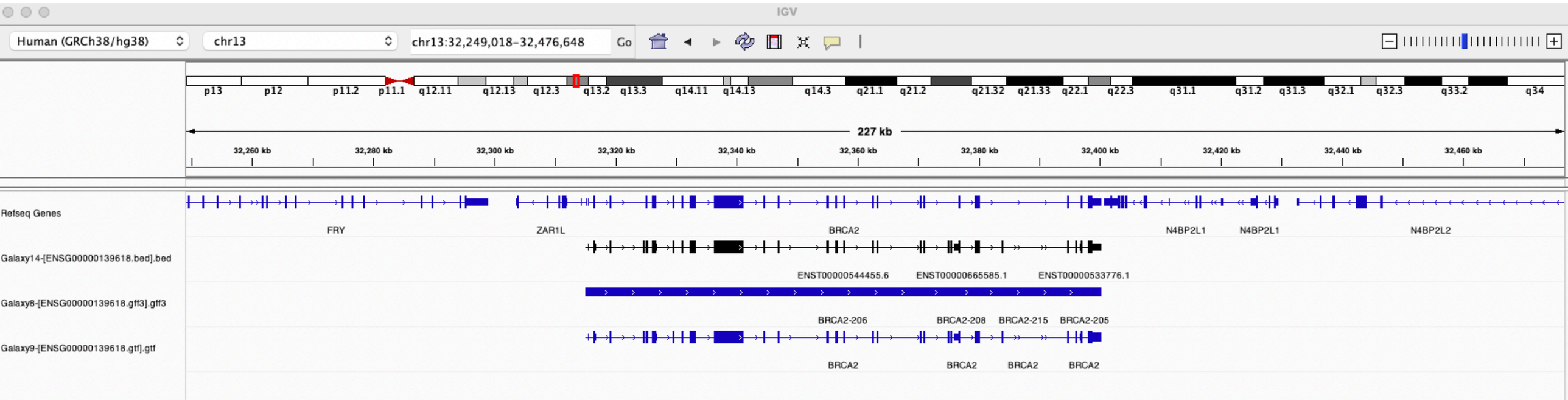
Working With Genome Files In Galaxy

For Gene ENSG00000139618



Working With Genome Files In Galaxy

For Gene ENSG00000139618



Working With Genome Files In Galaxy

Extracting Gene, Transcript and Protein IDs From Fasta Headers

Tools

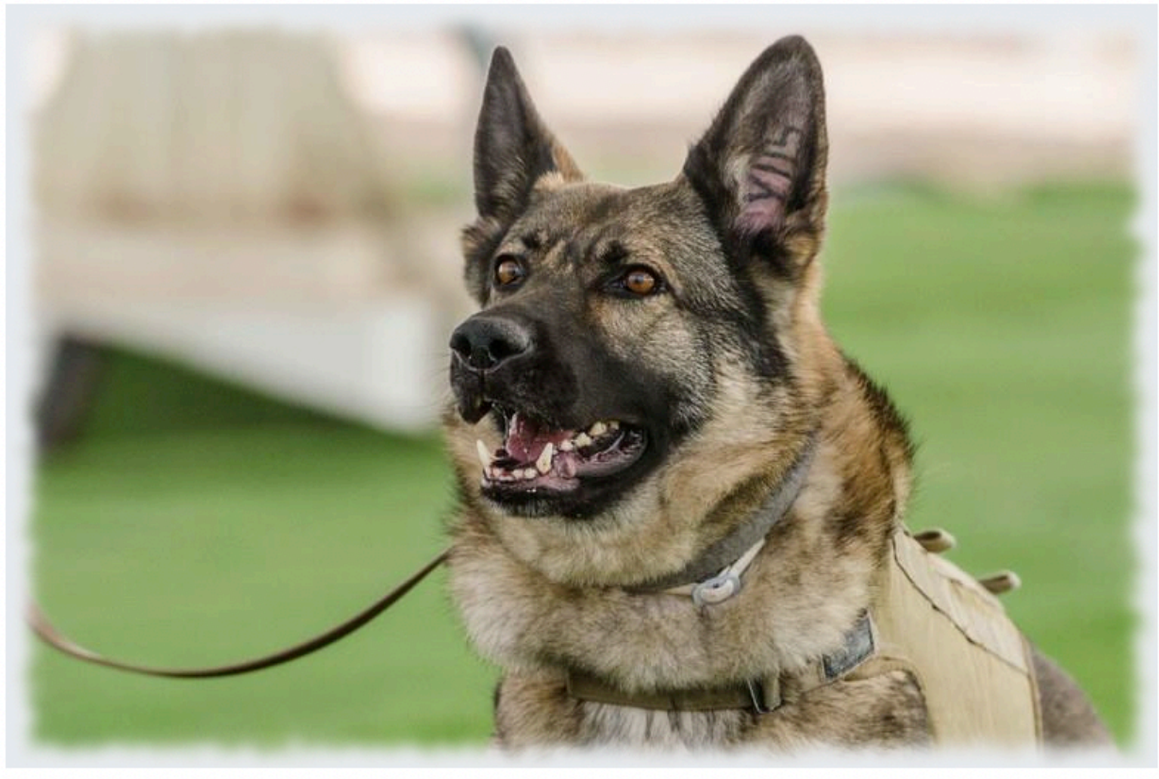
search tools

Upload Data

HPRC
Get Data
Send Data
Collection Operations
Lift-Over
Text Manipulation
Convert Formats
Filter and Sort
Join, Subtract and Group
Fetch Alignments/Sequences
Operate on Genomic Intervals
Statistics
Graph/Display Data
Phenotype Association
BED
Annotation
Multiple Alignments
NCBI BLAST+
Mapping
SAM/BAM
Assembly
FASTQ Quality Control
FASTA/FASTQ
RNA-seq
CD-HIT
Datamash
EMBOSS
MUMmer4
Nanopore

Best Practices for Kaiser Galaxy

- Kaiser Galaxy (docs, slides) is configured for teaching purposes so all users have a file quota of 1TB. How to permanently delete nonessential files.
- Only certain tools that support multi-core processing have the Job Resources Parameters option which allow you to select cores, memory and time.
- The default job resource parameters for all tools is 1 core with 7GB memory for 24 hours (24 SUs).
  - Configuring a job to use 48 cores for 1 hour requires 48 SUs. (48 cores for 168 hours = 7872 SUs).
  - Configuring a job to use 360GB memory for 1 hour requires 48 SUs. (360GB memory for 168 hours = 7872 SUs).
  - If a tool you used failed because it needs the Job Resource Parameters option added, contact the HPRC helpdesk.



Take an interactive tour: Galaxy UI | History | Window Manager | Deferred Datasets

Galaxy is an open platform for supporting data intensive research. Galaxy is developed by The Galaxy Team with the support of many contributors.

The Galaxy Project is supported in part by NHGRI, NSF, The Huck Institutes of the Life Sciences, The Institute for CyberScience at Penn State, and Johns Hopkins University.

History

search datasets

L09-B

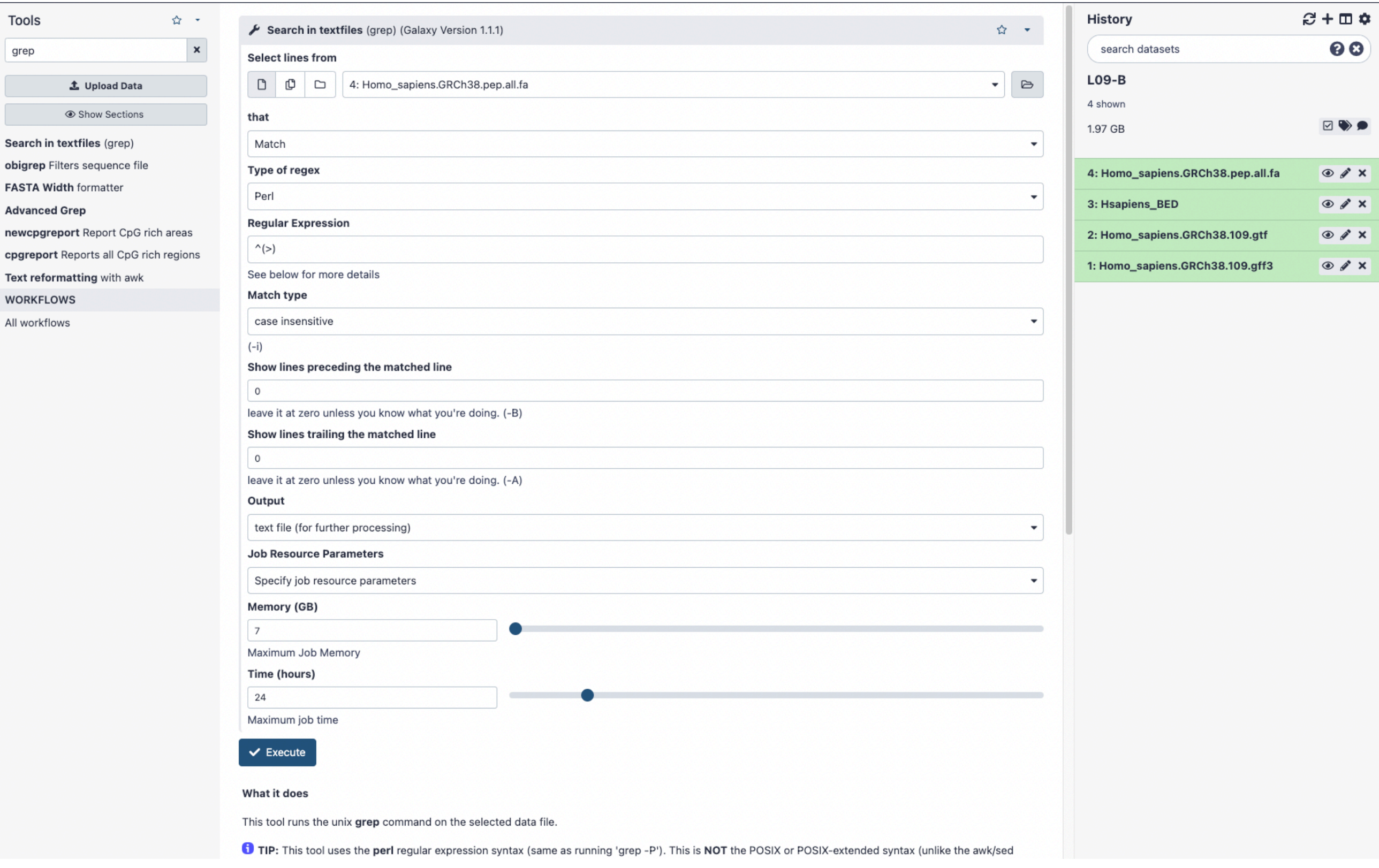
4 shown

1.97 GB

4: Homo_sapiens.GRCh38.pep.all.fa
3: Hsapiens_BED
2: Homo_sapiens.GRCh38.109.gtf
1: Homo_sapiens.GRCh38.109.gff3

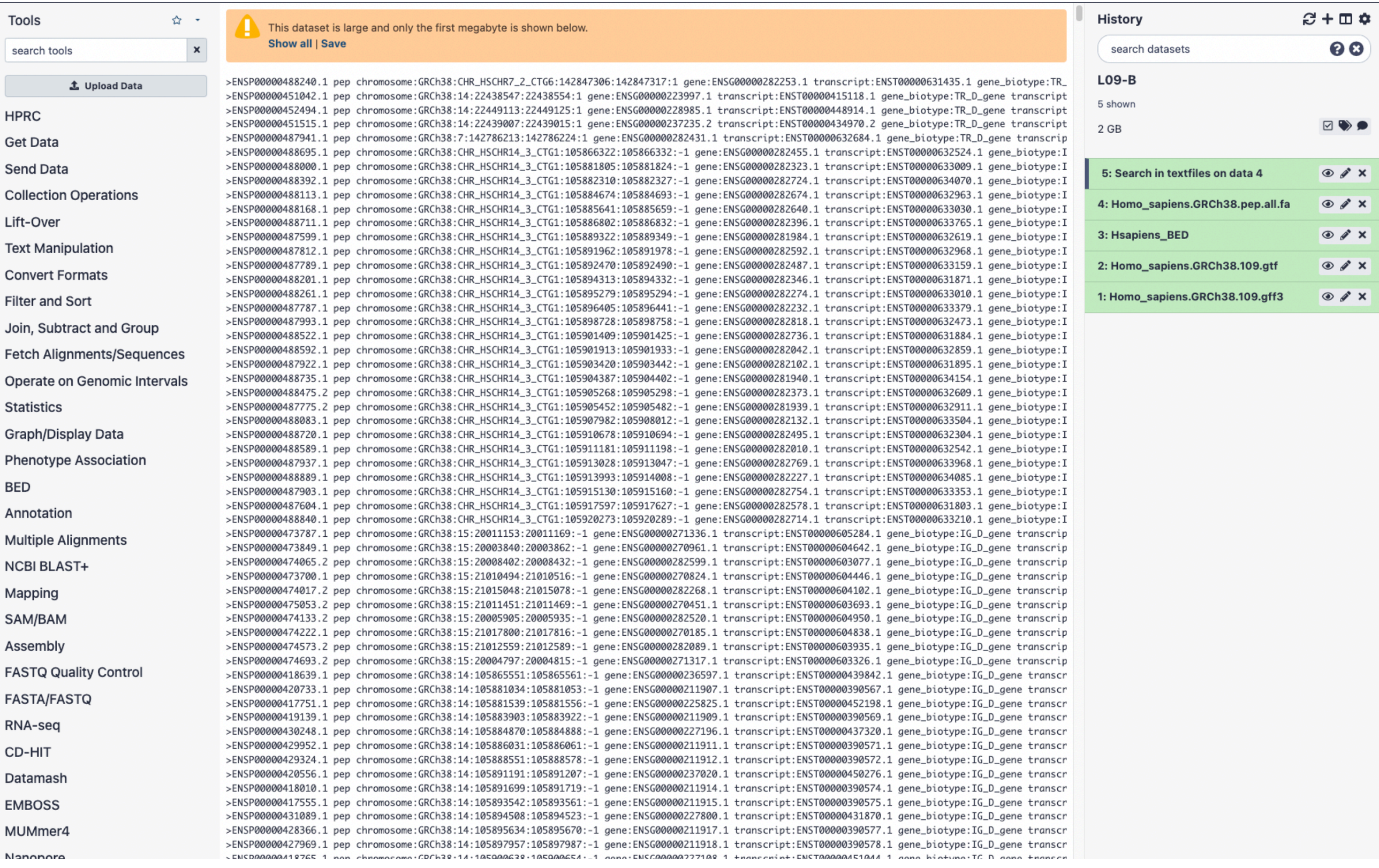
Working With Genome Files In Galaxy

Extracting Gene, Transcript and Protein IDs From Fasta Headers



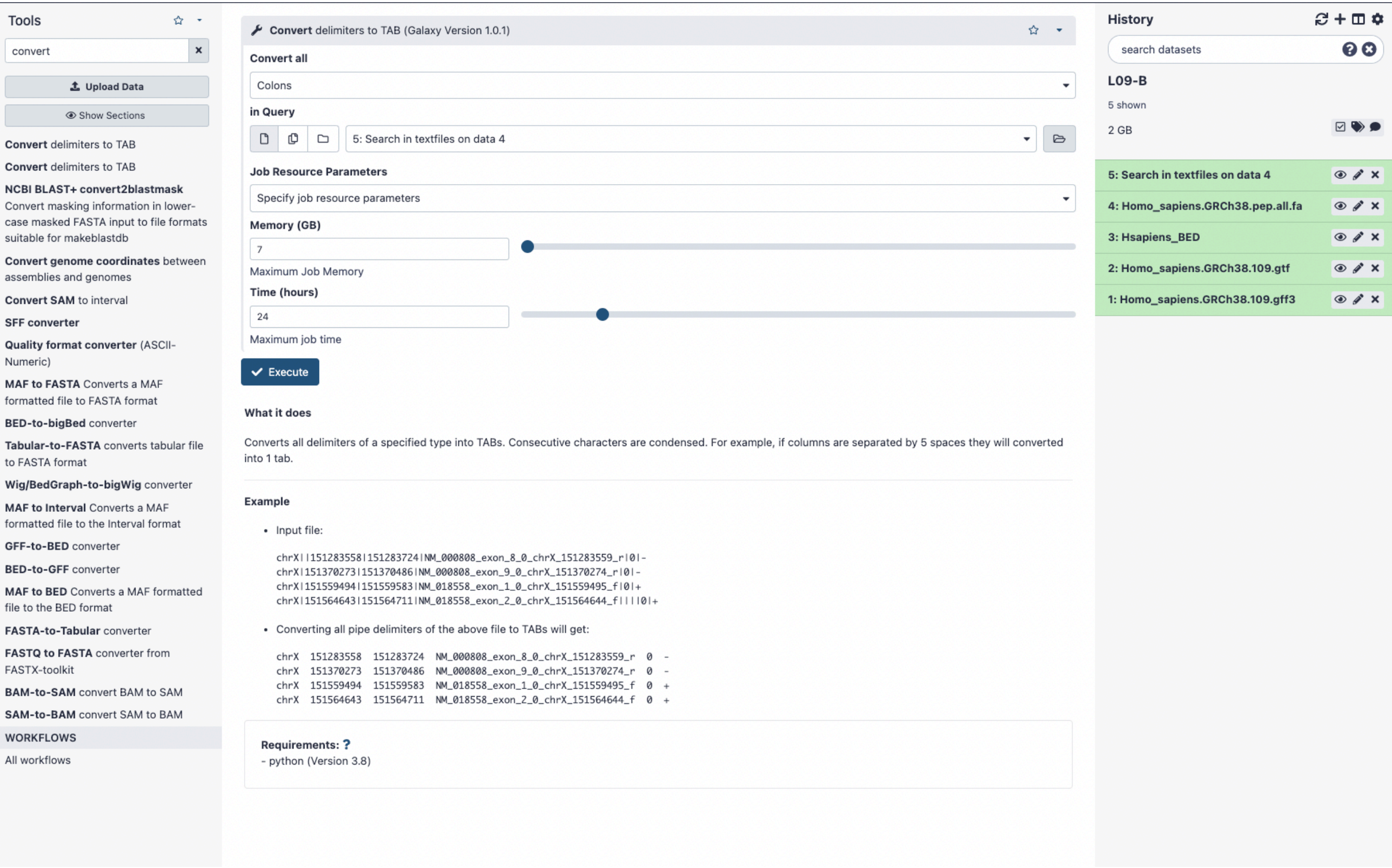
Working With Genome Files In Galaxy

Extracting Gene, Transcript and Protein IDs From Fasta Headers



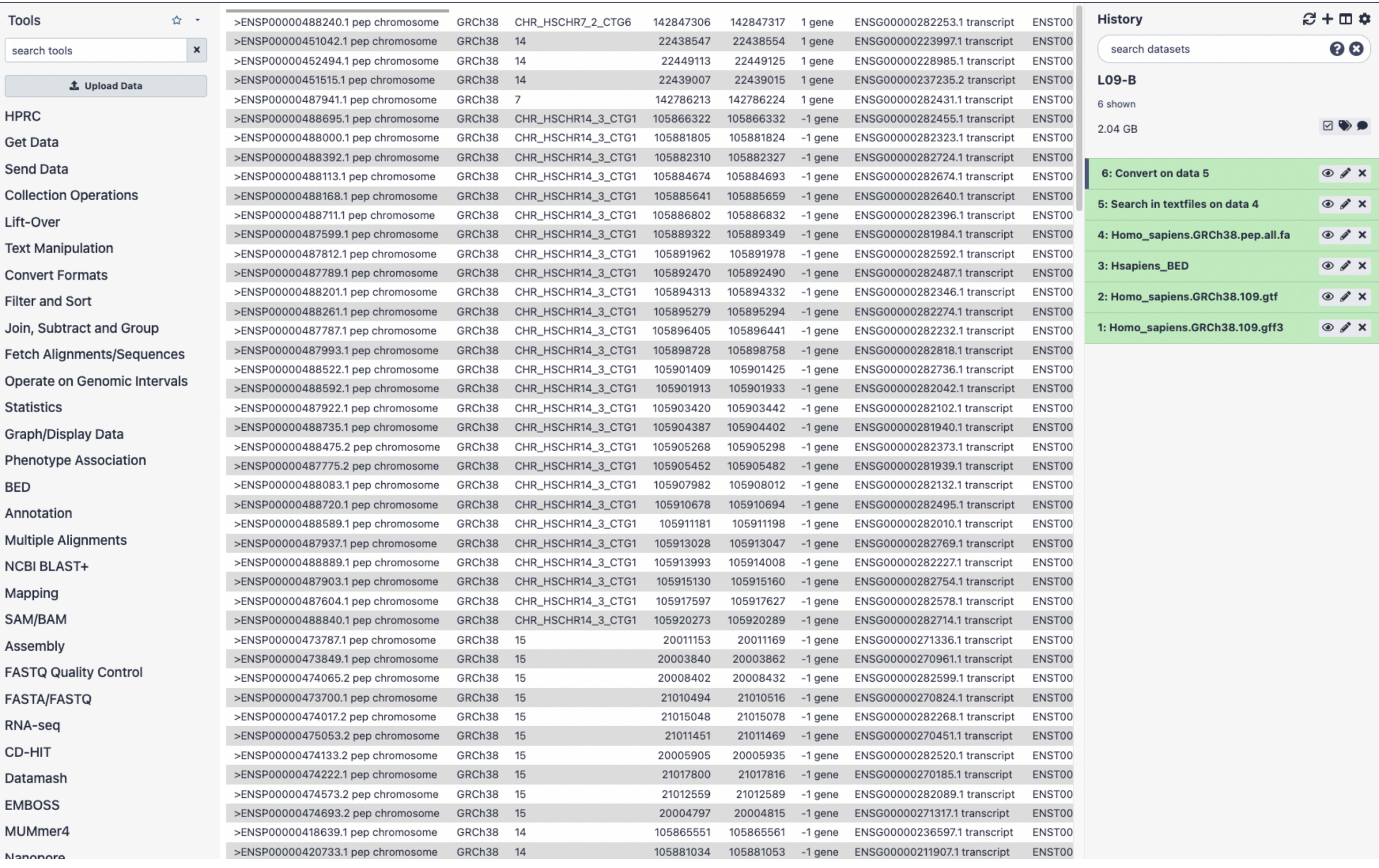
Working With Genome Files In Galaxy

Extracting Gene, Transcript and Protein IDs From Fasta Headers



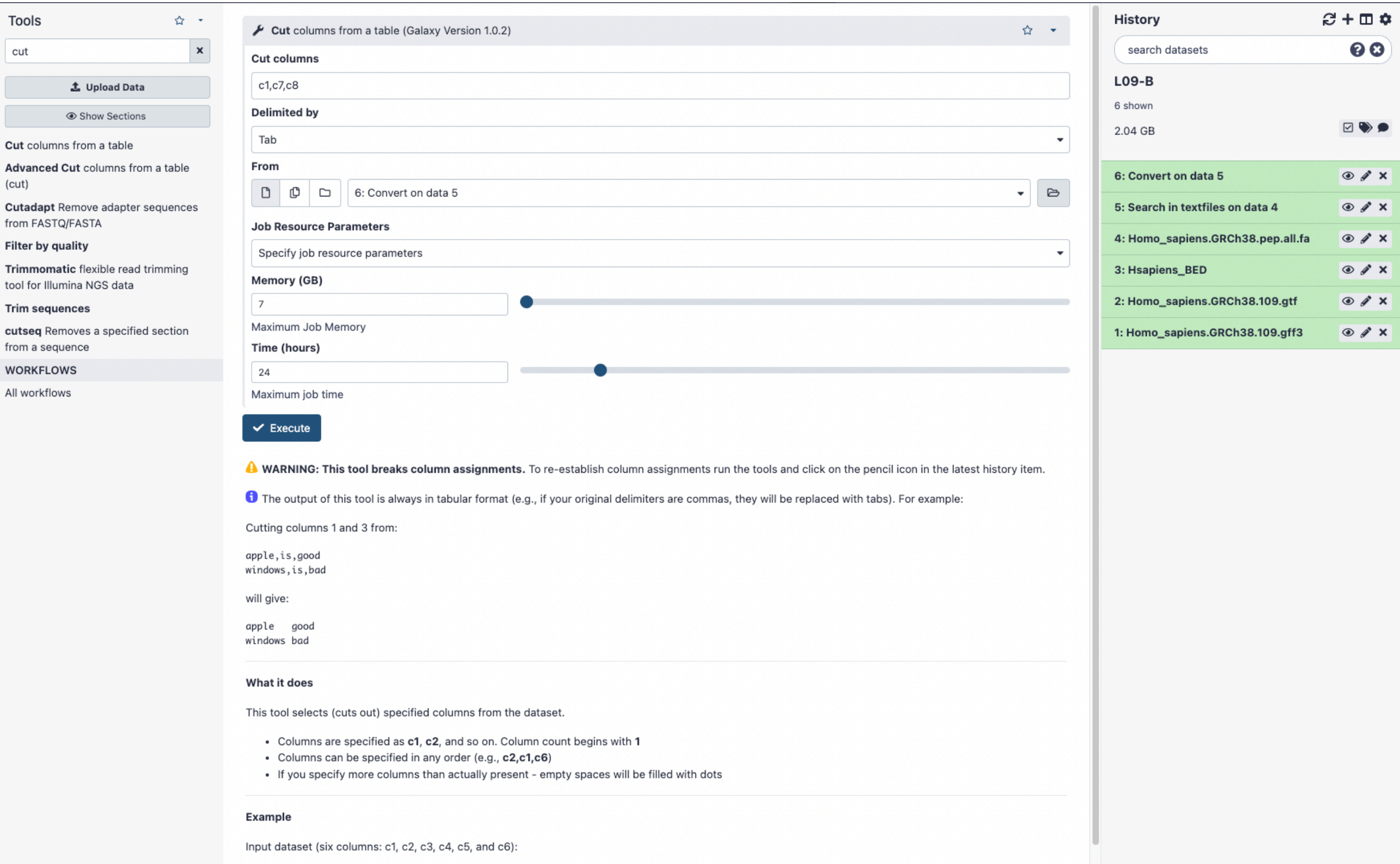
Working With Genome Files In Galaxy

Extracting Gene, Transcript and Protein IDs From Fasta Headers



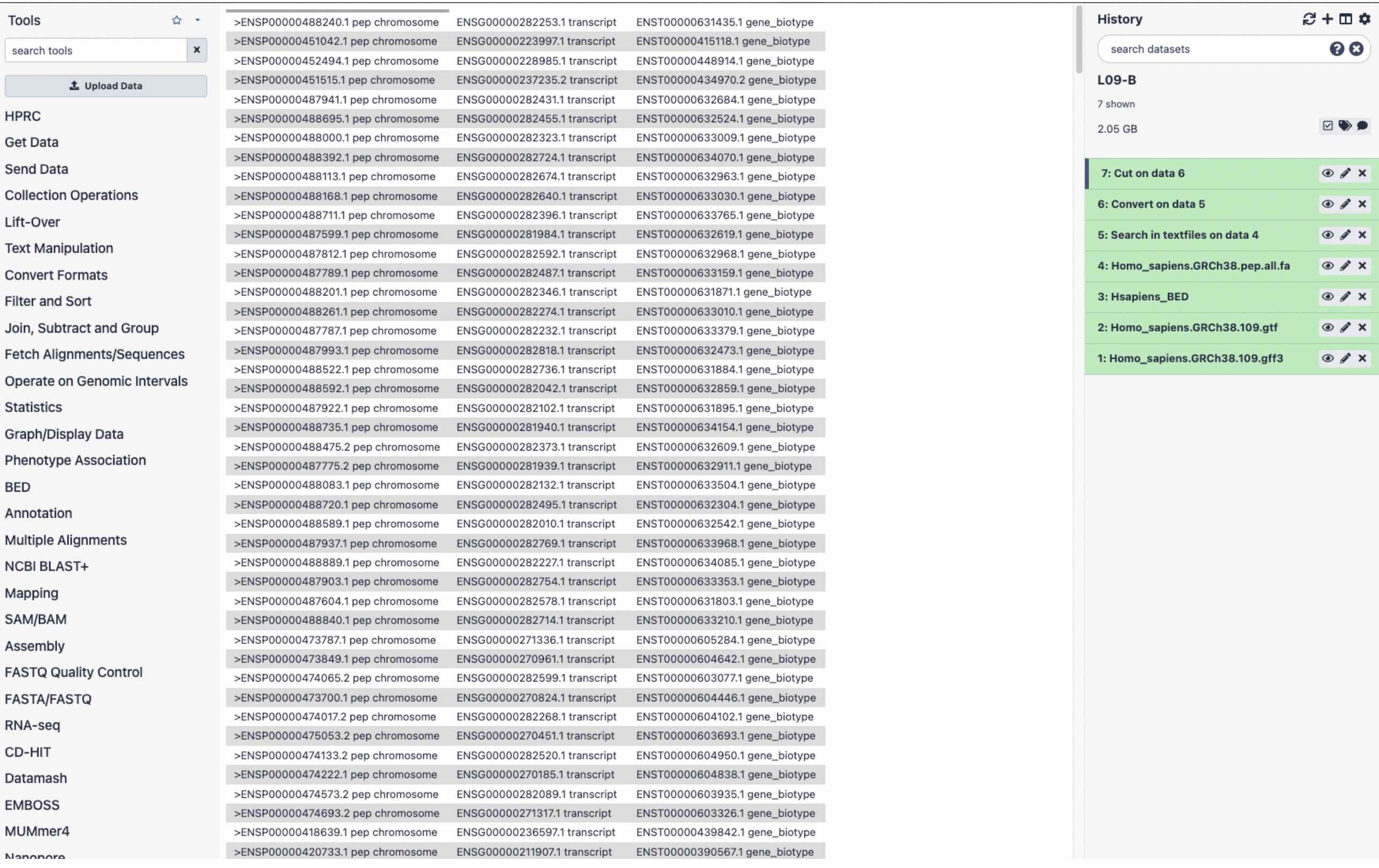
Working With Genome Files In Galaxy

Extracting Gene, Transcript and Protein IDs From Fasta Headers



Working With Genome Files In Galaxy

Extracting Gene, Transcript and Protein IDs From Fasta Headers



Working With Genome Files In Galaxy

Extracting Gene, Transcript and Protein IDs From Fasta Headers

Tools

convert

Upload Data

Show Sections

Convert delimiters to TAB

Convert delimiters to TAB

NCBI BLAST+ convert2blastmask Convert masking information in lower-case masked FASTA input to file formats suitable for makeblastdb

Convert genome coordinates between assemblies and genomes

Convert SAM to interval

SFF converter

Quality format converter (ASCII-Numeric)

MAF to FASTA Converts a MAF formatted file to FASTA format

BED-to-bigBed converter

Tabular-to-FASTA converts tabular file to FASTA format

Wig/BedGraph-to-bigWig converter

MAF to Interval Converts a MAF formatted file to the Interval format

GFF-to-BED converter

BED-to-GFF converter

MAF to BED Converts a MAF formatted file to the BED format

FASTA-to-Tabular converter

FASTQ to FASTA converter from FASTX-toolkit

BAM-to-SAM convert BAM to SAM

SAM-to-BAM convert SAM to BAM

WORKFLOWS

All workflows

Convert delimiters to TAB (Galaxy Version 1.0.1)

Convert all

Whitespaces

in Query

7: Cut on data 6

Job Resource Parameters

Specify job resource parameters

Memory (GB)

7

Maximum Job Memory

Time (hours)

24

Maximum job time

Execute

What it does

Converts all delimiters of a specified type into TABs. Consecutive characters are condensed. For example, if columns are separated by 5 spaces they will converted into 1 tab.

Example

- Input file:

```
chrX||151283558|151283724|NM_000808_exon_8_0_chrX_151283559_r|0|-
chrX|151370273|151370486|NM_000808_exon_9_0_chrX_151370274_r|0|-
chrX|151559494|151559583|NM_018558_exon_1_0_chrX_151559495_f|0|+
chrX|151564643|151564711|NM_018558_exon_2_0_chrX_151564644_f||||0|+
```

- Converting all pipe delimiters of the above file to TABs will get:

```
chrX  151283558  151283724  NM_000808_exon_8_0_chrX_151283559_r  0  -
chrX  151370273  151370486  NM_000808_exon_9_0_chrX_151370274_r  0  -
chrX  151559494  151559583  NM_018558_exon_1_0_chrX_151559495_f  0  +
chrX  151564643  151564711  NM_018558_exon_2_0_chrX_151564644_f  0  +
```

Requirements: ?
- python (Version 3.8)

History

search datasets

L09-B

7 shown

2.05 GB

7: Cut on data 6

6: Convert on data 5

5: Search in textfiles on data 4

4: Homo_sapiens.GRCh38.pep.all.fa

3: Hsapiens_BED

2: Homo_sapiens.GRCh38.109.gtf

1: Homo_sapiens.GRCh38.109.gff3

Working With Genome Files In Galaxy

Extracting Gene, Transcript and Protein IDs From Fasta Headers

Tools

search tools

Upload Data

HPRC
Get Data
Send Data
Collection Operations
Lift-Over
Text Manipulation
Convert Formats
Filter and Sort
Join, Subtract and Group
Fetch Alignments/Sequences
Operate on Genomic Intervals
Statistics
Graph/Display Data
Phenotype Association
BED
Annotation
Multiple Alignments
NCBI BLAST+
Mapping
SAM/BAM
Assembly
FASTQ Quality Control
FASTA/FASTQ
RNA-seq
CD-HIT
Datamash
EMBOSS
MUMmer4
Nanopore

>ENSP00000488240.1	pep	chromosome	ENSG00000282253.1	transcript	ENST00000631435.1	gene_biotype
>ENSP00000451042.1	pep	chromosome	ENSG00000223997.1	transcript	ENST00000415118.1	gene_biotype
>ENSP00000452494.1	pep	chromosome	ENSG00000228985.1	transcript	ENST00000448914.1	gene_biotype
>ENSP00000451515.1	pep	chromosome	ENSG00000237235.2	transcript	ENST00000434970.2	gene_biotype
>ENSP00000487941.1	pep	chromosome	ENSG00000282431.1	transcript	ENST00000632684.1	gene_biotype
>ENSP00000488695.1	pep	chromosome	ENSG00000282455.1	transcript	ENST00000632524.1	gene_biotype
>ENSP00000488000.1	pep	chromosome	ENSG00000282323.1	transcript	ENST00000633009.1	gene_biotype
>ENSP00000488392.1	pep	chromosome	ENSG00000282724.1	transcript	ENST00000634070.1	gene_biotype
>ENSP00000488113.1	pep	chromosome	ENSG00000282674.1	transcript	ENST00000632963.1	gene_biotype
>ENSP00000488168.1	pep	chromosome	ENSG00000282640.1	transcript	ENST00000633030.1	gene_biotype
>ENSP00000488711.1	pep	chromosome	ENSG00000282396.1	transcript	ENST00000633765.1	gene_biotype
>ENSP00000487599.1	pep	chromosome	ENSG00000281984.1	transcript	ENST00000632619.1	gene_biotype
>ENSP00000487812.1	pep	chromosome	ENSG00000282592.1	transcript	ENST00000632968.1	gene_biotype
>ENSP00000487789.1	pep	chromosome	ENSG00000282487.1	transcript	ENST00000633159.1	gene_biotype
>ENSP00000488201.1	pep	chromosome	ENSG00000282346.1	transcript	ENST00000631871.1	gene_biotype
>ENSP00000488261.1	pep	chromosome	ENSG00000282274.1	transcript	ENST00000633010.1	gene_biotype
>ENSP00000487787.1	pep	chromosome	ENSG00000282232.1	transcript	ENST00000633379.1	gene_biotype
>ENSP00000487993.1	pep	chromosome	ENSG00000282818.1	transcript	ENST00000632473.1	gene_biotype
>ENSP00000488522.1	pep	chromosome	ENSG00000282736.1	transcript	ENST00000631884.1	gene_biotype
>ENSP00000488592.1	pep	chromosome	ENSG00000282042.1	transcript	ENST00000632859.1	gene_biotype
>ENSP00000487922.1	pep	chromosome	ENSG00000282102.1	transcript	ENST00000631895.1	gene_biotype
>ENSP00000488735.1	pep	chromosome	ENSG00000281940.1	transcript	ENST00000634154.1	gene_biotype
>ENSP00000488475.2	pep	chromosome	ENSG00000282373.1	transcript	ENST00000632609.1	gene_biotype
>ENSP00000487775.2	pep	chromosome	ENSG00000281939.1	transcript	ENST00000632911.1	gene_biotype
>ENSP00000488083.1	pep	chromosome	ENSG00000282132.1	transcript	ENST00000633504.1	gene_biotype
>ENSP00000488720.1	pep	chromosome	ENSG00000282495.1	transcript	ENST00000632304.1	gene_biotype
>ENSP00000488589.1	pep	chromosome	ENSG00000282010.1	transcript	ENST00000632542.1	gene_biotype
>ENSP00000487937.1	pep	chromosome	ENSG00000282769.1	transcript	ENST00000633968.1	gene_biotype
>ENSP00000488889.1	pep	chromosome	ENSG00000282227.1	transcript	ENST00000634085.1	gene_biotype
>ENSP00000487903.1	pep	chromosome	ENSG00000282754.1	transcript	ENST00000633353.1	gene_biotype
>ENSP00000487604.1	pep	chromosome	ENSG00000282578.1	transcript	ENST00000631803.1	gene_biotype
>ENSP00000488840.1	pep	chromosome	ENSG00000282714.1	transcript	ENST00000633210.1	gene_biotype
>ENSP00000473787.1	pep	chromosome	ENSG00000271336.1	transcript	ENST00000605284.1	gene_biotype
>ENSP00000473849.1	pep	chromosome	ENSG00000270961.1	transcript	ENST00000604642.1	gene_biotype
>ENSP00000474065.2	pep	chromosome	ENSG00000282599.1	transcript	ENST00000603077.1	gene_biotype
>ENSP00000473700.1	pep	chromosome	ENSG00000270824.1	transcript	ENST00000604446.1	gene_biotype
>ENSP00000474017.2	pep	chromosome	ENSG00000282268.1	transcript	ENST00000604102.1	gene_biotype
>ENSP00000475053.2	pep	chromosome	ENSG00000270451.1	transcript	ENST00000603693.1	gene_biotype
>ENSP00000474133.2	pep	chromosome	ENSG00000282520.1	transcript	ENST00000604950.1	gene_biotype
>ENSP00000474222.1	pep	chromosome	ENSG00000270185.1	transcript	ENST00000604838.1	gene_biotype
>ENSP00000474573.2	pep	chromosome	ENSG00000282089.1	transcript	ENST00000603935.1	gene_biotype
>ENSP00000474693.2	pep	chromosome	ENSG00000271317.1	transcript	ENST00000603326.1	gene_biotype
>ENSP00000418639.1	pep	chromosome	ENSG00000236597.1	transcript	ENST00000439842.1	gene_biotype
>ENSP00000420733.1	pep	chromosome	ENSG00000211907.1	transcript	ENST00000390567.1	gene_biotype

History

search datasets

L09-B

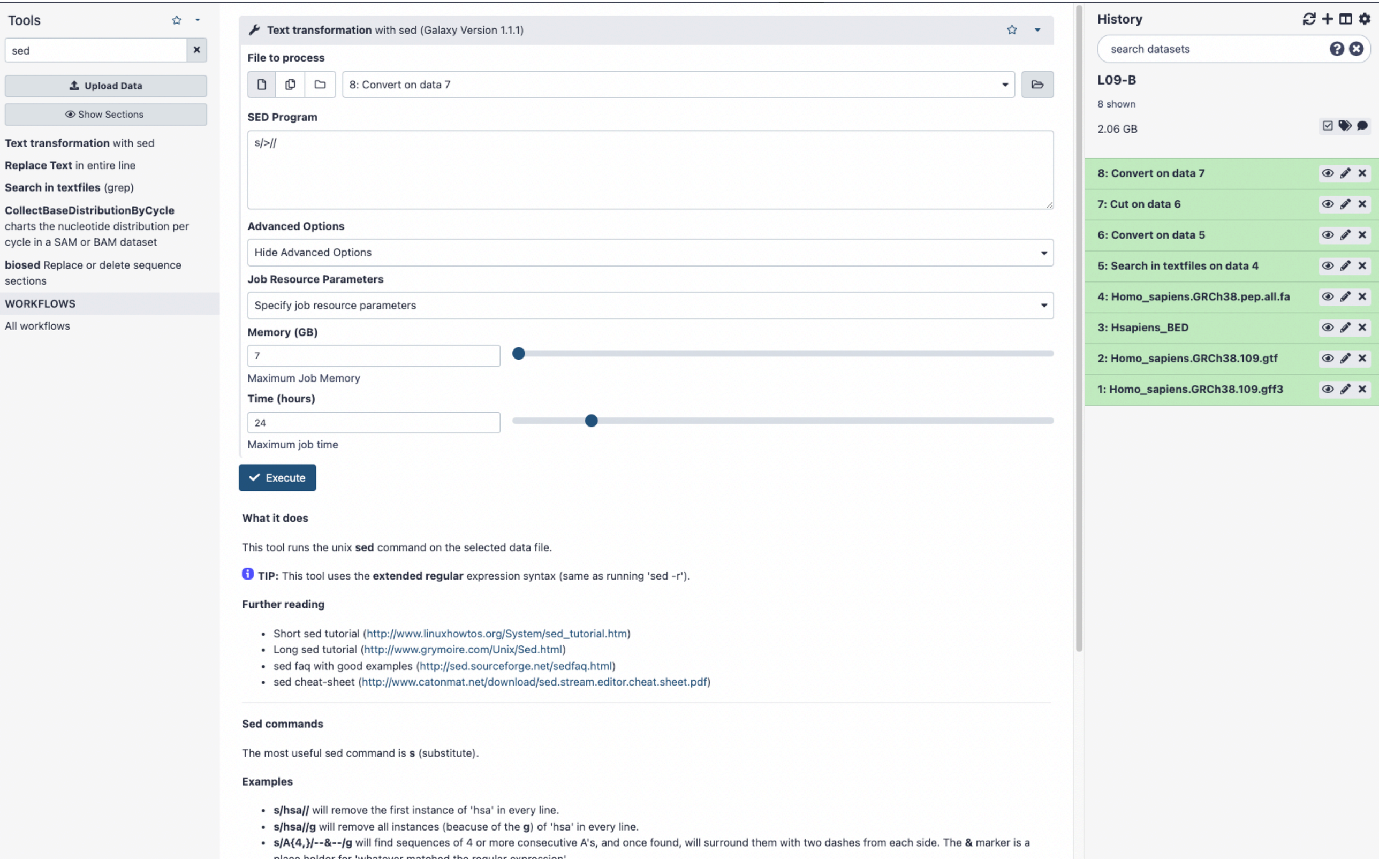
8 shown

2.06 GB

8: Convert on data 7
7: Cut on data 6
6: Convert on data 5
5: Search in textfiles on data 4
4: Homo_sapiens.GRCh38.pep.all.fa
3: Hsapiens_BED
2: Homo_sapiens.GRCh38.109.gtf
1: Homo_sapiens.GRCh38.109.gff3

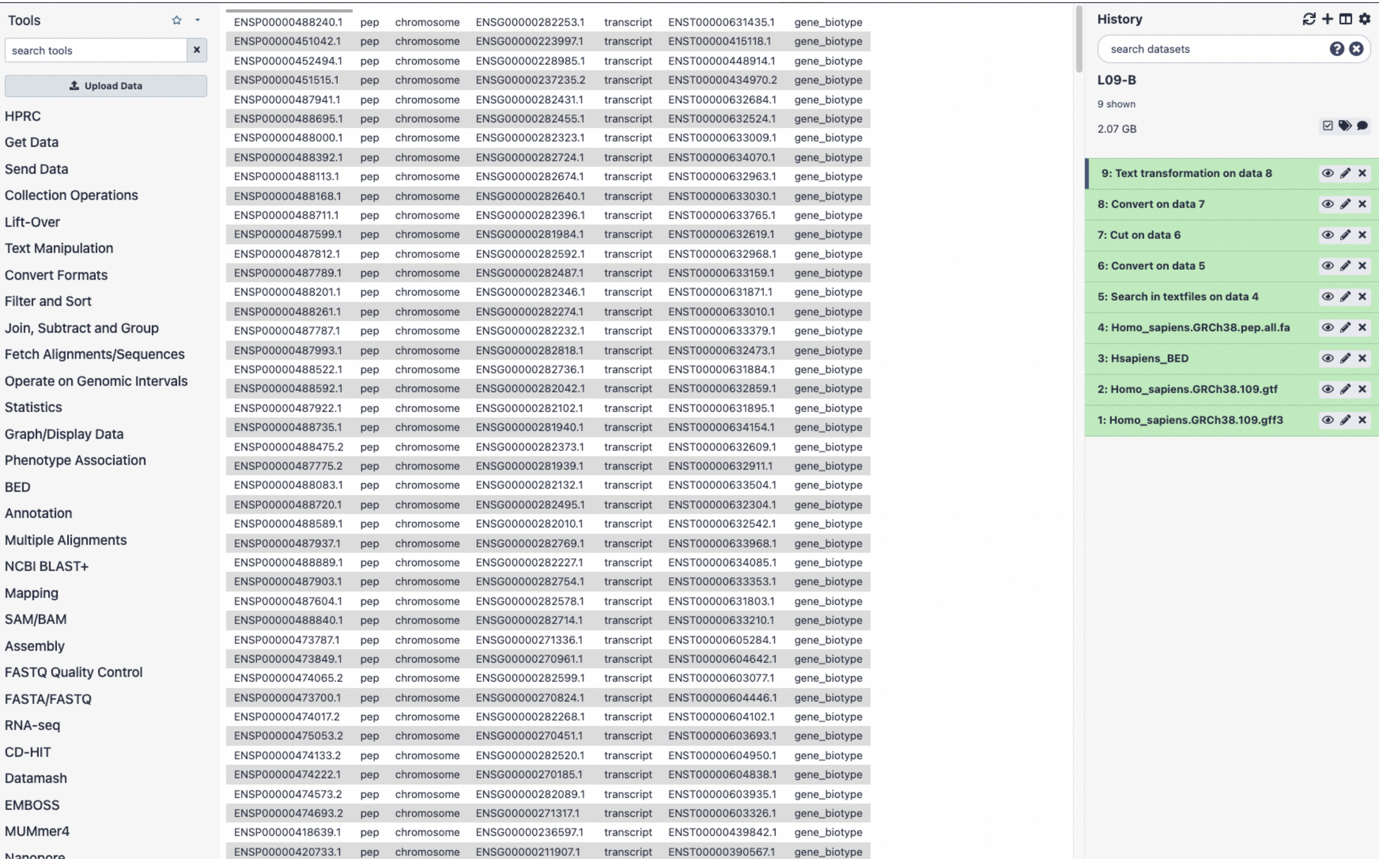
Working With Genome Files In Galaxy

Extracting Gene, Transcript and Protein IDs From Fasta Headers



Working With Genome Files In Galaxy

Extracting Gene, Transcript and Protein IDs From Fasta Headers



Working With Genome Files In Galaxy

Extracting Gene, Transcript and Protein IDs From Fasta Headers

Tools

cut

Upload Data

Show Sections

Cut columns from a table

Advanced Cut columns from a table (cut)

Cutadapt Remove adapter sequences from FASTQ/FASTA

Filter by quality

Trimmomatic flexible read trimming tool for Illumina NGS data

Trim sequences

cutseq Removes a specified section from a sequence

WORKFLOWS

All workflows

Cut columns from a table (Galaxy Version 1.0.2)

Cut columns

c4,c6,c1

Delimited by

Tab

From

9: Text transformation on data 8

Job Resource Parameters

Specify job resource parameters

Memory (GB)

7

Maximum Job Memory

Time (hours)

24

Maximum job time

Execute

WARNING: This tool breaks column assignments. To re-establish column assignments run the tools and click on the pencil icon in the latest history item.

The output of this tool is always in tabular format (e.g., if your original delimiters are commas, they will be replaced with tabs). For example:

Cutting columns 1 and 3 from:

```
apple,is,good
windows,is,bad
```

will give:

```
apple   good
windows bad
```

What it does

This tool selects (cuts out) specified columns from the dataset.

- Columns are specified as **c1**, **c2**, and so on. Column count begins with **1**
- Columns can be specified in any order (e.g., **c2,c1,c6**)
- If you specify more columns than actually present - empty spaces will be filled with dots

---

Example

Input dataset (six columns: c1, c2, c3, c4, c5, and c6):

History

search datasets

L09-B

9 shown

2.07 GB

9: Text transformation on data 8

8: Convert on data 7

7: Cut on data 6

6: Convert on data 5

5: Search in textfiles on data 4

4: Homo_sapiens.GRCh38.pep.all.fa

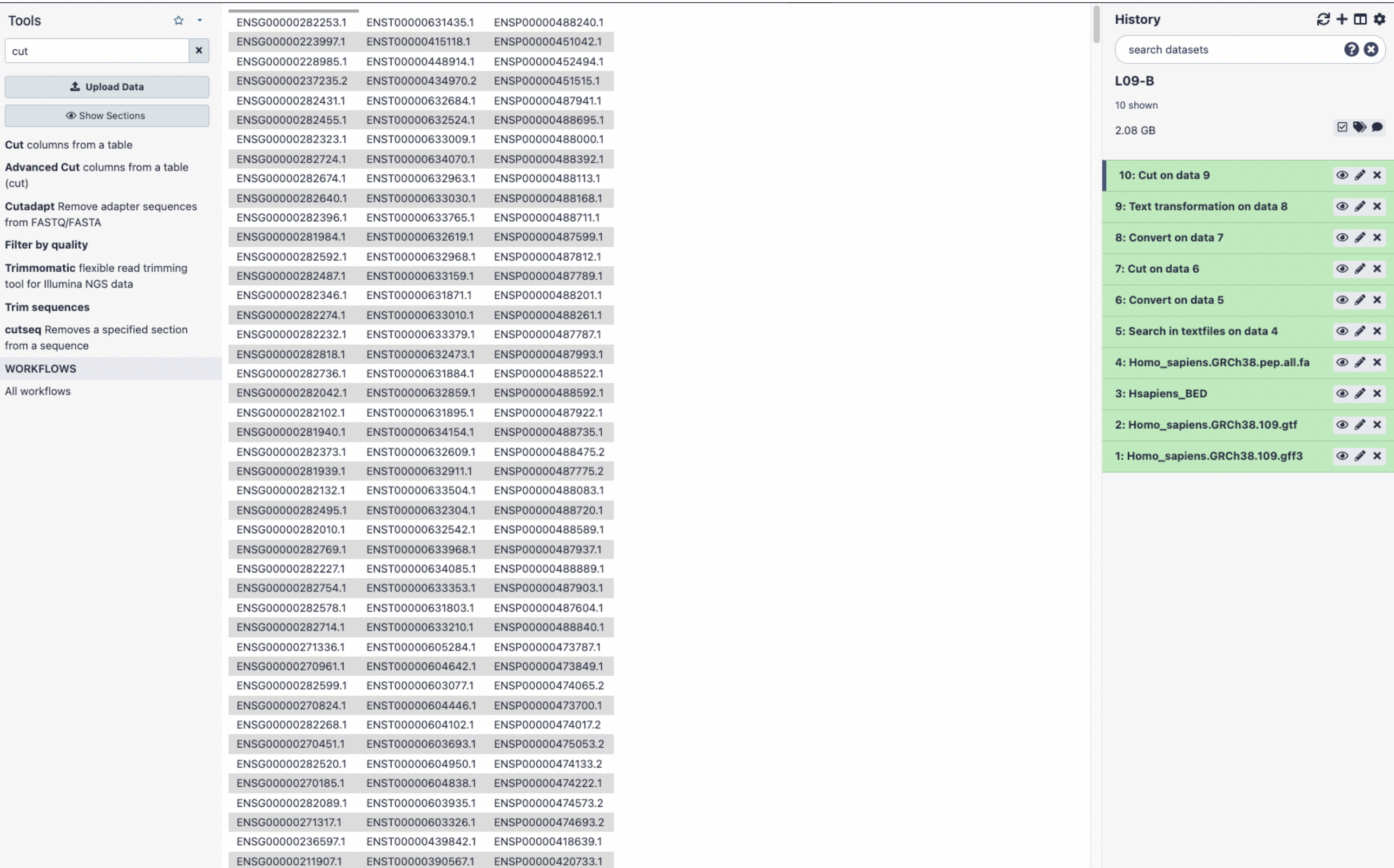
3: Hsapiens_BED

2: Homo_sapiens.GRCh38.109.gtf

1: Homo_sapiens.GRCh38.109.gff3

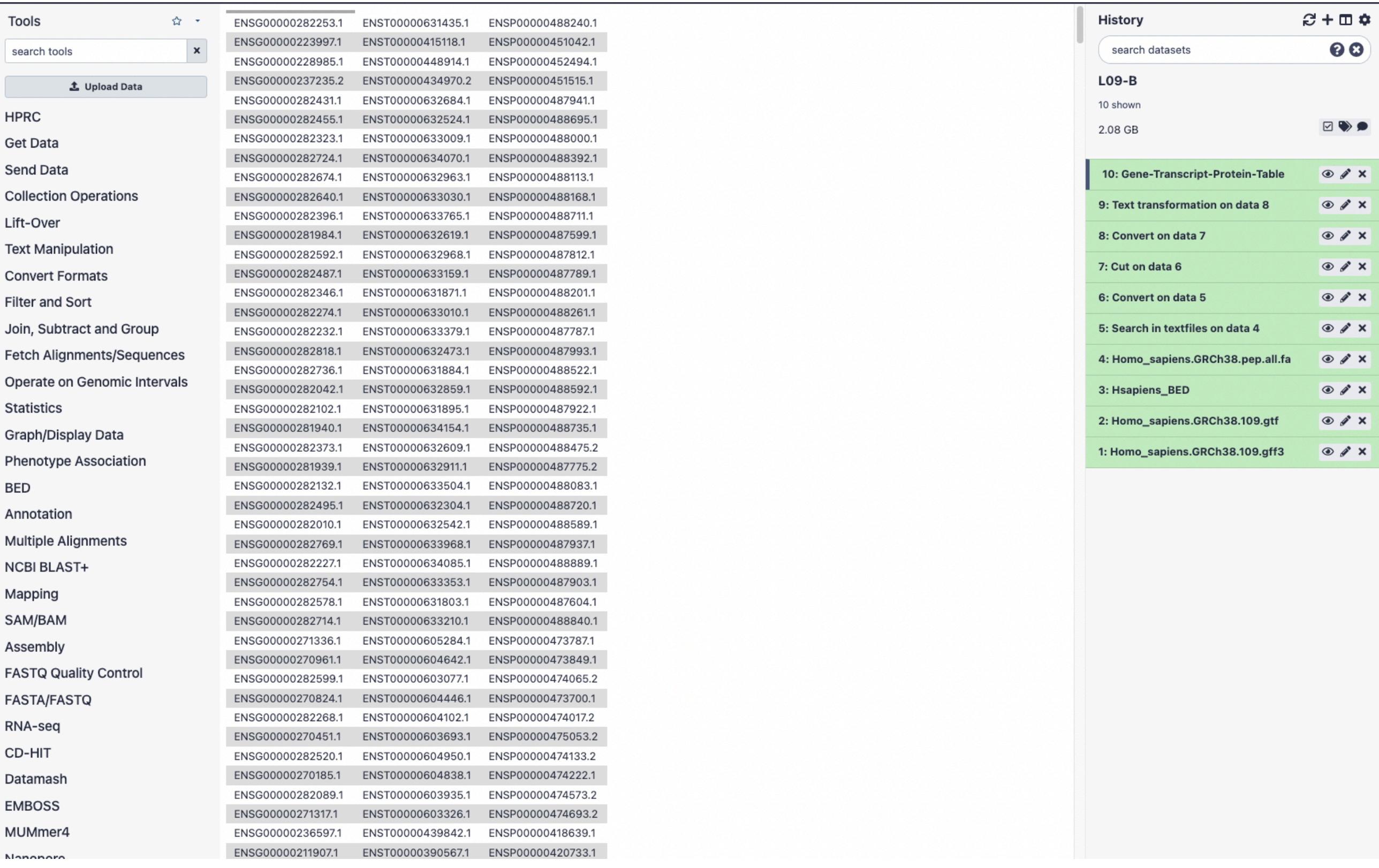
Working With Genome Files In Galaxy

Extracting Gene, Transcript and Protein IDs From Fasta Headers



Working With Genome Files In Galaxy

Extracting Gene, Transcript and Protein IDs From Fasta Headers



Working With Genome Files In Galaxy

For a BED File

- Can we identify Supercontigs?
- Can we identify Chromosomes?
- Can we identify Genes?
- Can we identify transcripts?
- Can we identify Proteins?
- Can we identify/extract All the Features being declared?
- Can we separate them by those present in the Watson and/or Crick DNA strands?

Working With Genome Files In Galaxy

For a GTF File

- Can we identify Supercontigs?
- Can we identify Chromosomes?
- Can we identify Genes?
- Can we identify transcripts?
- Can we identify Proteins?
- Can we identify/extract All the Features being declared?
- Can we separate them by those present in the Watson and/or Crick DNA strands?

Working With Genome Files In Galaxy

For a GFF3 File

- Can we identify Supercontigs?
- Can we identify Chromosomes?
- Can we identify Genes?
- Can we identify transcripts?
- Can we identify Proteins?
- Can we identify/extract All the Features being declared?
- Can we separate them by those present in the Watson and/or Crick DNA strands?