|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **What for** | **What in** | **What out** | **Example operation, link** |
| **UCSC Main** table browser | **Retrieve and export data from the Genome Browser** annotation track database | What in: **select**  dataset, **define** region of interest (genome | position + identifiers),  **retrieve and display data**  **Send output to Galaxy:** displays results of query in [Galaxy](https://usegalaxy.org/), a framework for interactive genome analysis.  **Send output to GREAT:** displays the functional enrichments of the query results in [GREAT](http://great.stanford.edu/), a tool for analysis of the biological function of cis-regulatory regions.  **get output:** Submits a data query based on the specified parameters and returns the output.  **Summary/statistics:** Displays statistics about the data specified by the parameters. | output formats    file type returned  **plain text** - data is in ASCII format  **\*.gzip** compressed archieve format for Linux|Unix | <https://genome.ucsc.edu/cgi-bin/hgTables?GALAXY_URL=https%3A//usegalaxy.eu/tool_runner&tool_id=ucsc_table_direct1&sendToGalaxy=1&hgta_compressType=none&hgta_outputType=bed>Retrieve list of genes of an animals, viruses, insects for i.e. mice etc. |
| **Select last** lines from a dataset (tail) | **Keep the last X lines in a file**. It is a tool from text processing tools | **Keep last** **lines**| Keep from this line on replace with icon  C:\Users\Ludmilla\Desktop\Galaxy\cheetsheets\peaks and genes\select_last_input_file.png | **C:\Users\Ludmilla\Desktop\Galaxy\cheetsheets\peaks and genes\select_last_output.pngOutput** | https://usegalaxy.eu/root?tool\_id=toolshed.g2.bx.psu.edu/repos/bgruening/text\_processing/tp\_tail\_tool/1.1.0  to compare the two files, to make sure that the chromosome names follow the same format |
| **Replace Text in a specific column** | **Performs find & replace operation** on a specified column in a given file.   For more complex patterns, use the ***awk***tool. | **Parameters**  **In column:** i.e. 5  **Find pattern:** i.e. any symbol or letter expression „hello“  **Replace with:** text, or & (ampersand) and \\1 \\2 \\3  **Insert Replacement** (add new replacement for a new column) | The same text file in a \*.gzip with replacement or replacements | [*https://usegalaxy.eu/root?tool\_id=toolshed.g2.bx.psu.edu/repos/bgruening/text\_processing/tp\_replace\_in\_column/1.1.3*](https://usegalaxy.eu/root?tool_id=toolshed.g2.bx.psu.edu/repos/bgruening/text_processing/tp_replace_in_column/1.1.3)  For i.e. to convert the chromosome names, to change 20 and 21 to X and Y |
| **Get flanks** returns flanking region/s for every gene | **This tool finds the** **upstream and/or downstream flanking region(s)** of all the selected regions in the input file. | Every line should contain at least 3 columns: Chromosome number, Start and Stop co-ordinates.  **Parameters**  **Region:** Whole feature| Around start | Around end  **Location of the flanking region/s:**  **Offset:**  Use positive values to offset co-ordinates in the direction of transcription and negative values to offset in the opposite direction**.**  **Length of the flanking region(s)**  Use non-negative value | **Result**  BED format file with flanking regions for every gene | <https://usegalaxy.eu/root?tool_id=toolshed.g2.bx.psu.edu/repos/devteam/get_flanks/get_flanks1/1.0.0>  Adding promoter regions, i.e. to get regions 2kb bases upstream of the start of the gene to 10kb bases downstream of the start (12kb in length) |
| **Convert** Genomic Intervals To BED | **Converts**  Genomic Intervals To BED | Genomic intervals file | BED file | <https://usegalaxy.eu/root?tool_id=CONVERTER_interval_to_bed_0> |
| **Intersect** | **Intersect the intervals** of two datasets | Use "edit attributes" to set chromosome, start, end, and strand columns to set file to interval format if it doesn’t appear in the pulldown menu  The order of the datasets is important  **Parameters**  **Return:** Overlapping intervals **|** Overlapping pieces of intervals  **Of** first dataset: **[the UCSC file format BED]**  **that intersect** second dataset**:**  **for at least: [1 or more]** | The intersection of two queries that are found. | <https://usegalaxy.eu/root?tool_id=toolshed.g2.bx.psu.edu/repos/devteam/intersect/gops_intersect_1/1.0.0>  i.e. task could be to extract the genes which overlap/intersect with peaks regions in a dataset **OR**  e.g. to find all exons containing repeats **OR**  all regions that are both exonic and repetitive |
| **Group** data by a column and perform aggregate operation on other columns. | This tool allows you to group the input dataset by a particular column and perform aggregate functions: Mean, Median, Mode, Sum, Max, Min, Count, Concatenate, and Randomly pick on any column(s). | **Select Data:** [Data input 'input1' (tabular)]  **Group by column**: [column number]  **Ignore case while grouping:** [on|off]  **Ignore lines beginning with these characters** [character’s list]  **Operation:** insert operation **[type e.g. mean & on column & round &replace non numeric data |** leave empty for no replacements.]  You can add several operations following one by one.  If your data is not TAB delimited, use *Text Manipulation->Convert* | File Group on data with executed grouping operations in TAB format | <https://usegalaxy.eu/root?tool_id=Grouping1> |
| **Compute** an expression on every row | This tool computes an expression for every row of a dataset and appends the result as a new column (field). | Tool needs TAB data  **Parameters**  **Add expression***: [e.g. c2+c5]*  **as a new column to**: [the interval format file]  **Round result?:** No **|** Yes  **Avoid scientific notation:** No **|** Yes  **Input has a header line with column names?** No **|** Yes | TAB file with computation, for e.g. in a new added column | <https://usegalaxy.eu/root?tool_id=toolshed.g2.bx.psu.edu/repos/devteam/column_maker/Add_a_column1/1.6>  To generate a new BED file from the peak file that contains the positions of the peak summits.  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 dataset if c2 and c3 are start and end position |
| **Cut** columns from a table | This tool selects (cuts out) specified columns from the dataset. | 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.**  **Parameters**  **Cut columns** for e.g. c1, c2  **Delimited by** Tab | Whitespace | Dot | Commaetc.  **From** [dataset in csv.] | |  |  |  |  |  | | --- | --- | --- | --- | --- | | **a** | **,** | **is** | **,** | **b** | | **c** | **,** | **is** | **,** | **d** |   The output of this tool is always in tabular format (e.g., if your original delimiters are commas, they will be replaced with tabs).   |  |  |  | | --- | --- | --- | | **a** |  | **b** | | **c** |  | **d** | | <https://usegalaxy.eu/root?tool_id=Cut1>  **Input dataset c1,c2,c3,c4,c5,c6**   |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | **a** | **1** | **#** | **%** | **0** | **+** | | **b** | **2** | **$** | **%** | **0** | **+** |   **cut** on **c1,c4,c6**   |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | **a** | **1** | **#** | **%** | **0** | **+** | | **b** | **2** | **$** | **%** | **0** | **+** |  |  |  |  | | --- | --- | --- | | **a** | **%** | **+** | | **b** | **%** | **+** |   **cut** on columns "**c6,c5,c4,c1**" returns   |  |  |  |  | | --- | --- | --- | --- | | **+** | **0** | **%** | **a** | | **+** | **0** | **%** | **b** |   **cut** on **c8,c7,c4**   |  |  |  | | --- | --- | --- | | **.** | **.** | **%** | | **.** | **.** | **%** | |
| **Join two Datasets side by side on a specified field** | This tool joins lines of two datasets on a common field. | **This tool will force the ouput datatype to tabular.** 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  **Parameters**  **Join [**file as tabular**]**  **using column [**column by its number e.g. 3 refers to third column**]**  **with [**file as tabular**]**  **and column [**column number e.g. 5**]** |  | <https://usegalaxy.eu/root?tool_id=join1>  To add in the end of BED file list of Gene names to RefSeq Gene identifiers in the table -> i.e. join two files  Joining **4th column** of Dataset1   |  |  |  |  | | --- | --- | --- | --- | | **chr1** | **10** | **20** | **geneA** | | **chr1** | **50** | **80** | **geneB** | | **chr5** | **10** | **40** | **geneL** |      |  |  | | --- | --- | | **geneA** | **tumor suppressor** | | **geneB** | **Foxp2** | | **geneC** | **Gnas1** | | **geneE** | **INK4a** |   With the **1st column** of Dataset2  Result **(Keep the header lines –No)**   |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | **chr1** | **10** | **20** | **geneA** | **geneA** | **Tumor**  **suppressor** | | **chr1** | **50** | **80** | **geneB** | **geneB** | **Foxp2** |   **Keep the header lines –Yes**   |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | **chr1** | **10** | **20** | **geneA** | **geneA** | **Tumor**  **suppressor** | | **chr1** | **50** | **80** | **geneB** | **geneB** | **Foxp2** | | **chr5** | **10** | **40** | **geneL** |  |  | |
| **Sort** data in ascending or descending order |  |  |  | <https://usegalaxy.eu/root?tool_id=toolshed.g2.bx.psu.edu/repos/bgruening/text_processing/tp_sort_header_tool/1.1.1>  For listing unique genes that was unsorted. |

Formats from Peaks and Genes

GNU zip (gzip) –compressed archive format for Linux and Unix systems

Txt. – plain text file

**Interval**format is a Galaxy format for representing genomic intervals. It is tab-separated, but has the added requirement that three of the columns must be:

* chromosome ID
* start position (0-based)
* end position (end-exclusive)

An optional strand column can also be specified, and an initial header row can be used to label the columns, which do not have to be in any special order. Unlike BED format (see below) arbitrary additional columns can also be present.

The **BED - Browser Extensible Data** format provides a flexible way to encode gene regions. BED lines have three required fields:

* chromosome ID
* start position (0-based)
* end position (end-exclusive)

There can be up to and nine additional optional fields, but the number of fields per line must be consistent throughout any single set of data.

**UCSC Formats** <https://genome.ucsc.edu/FAQ/FAQformat.html>

**GFF** (General Feature Format) lines are based on the Sanger [**GFF2 specification**](http://www.sanger.ac.uk/resources/software/gff/spec.html). 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.

**TAB** One or more columns of text data separated by tabs.