

An introduction to Galaxy

Sebastian Schmeier

*Institute of Natural and Mathematical Sciences
Massey University Auckland, New Zealand*
<http://sebscientific.org>
s.schmeier@gmail.com

2015-08-11

Contents

Galaxy Introduction	3
1.0 Preface	3
1.1 Overview	3
1.2 How to get access to Galaxy	4
1.3 The user interface	4
1.3.1 Basics	4
1.3.2 User accounts	5
1.4 A word on tools	6
1.5 The task at hand	6
1.6 Loading your own data	7
1.7 Loading data from the web	11
1.8 Loading shared data	12
1.9 Working with data	14
1.9.1 Renaming files	15
1.9.2 Removing unwanted information	16
1.9.3 Creating flanking regions	17
1.9.4 Filter data	18
1.9.5 Joining/intersecting data sets	19
2.0 Visualising data sets	21
2.1 Another word on the history	23
2.1.1 Saved histories	23
2.1.2 Sharing a history	24
2.2 Workflows	25
2.2.1 Creating workflows	25
2.2.2 Editing workflows	27
2.2.3 Applying workflows to your data	31
2.3 References	32
2.4 Web links	32

Galaxy Introduction

1.0 Preface

In this brief tutorial we will learn how to use the excellent tool [Galaxy](http://galaxyproject.org/) (<http://galaxyproject.org/>) to analyze biological data. It is part of a series of introductory tutorials that can be found at <http://sschmeier.github.io/bioinf-workshop/>.

A PDF-version of this tutorial can be downloaded [here](#) or at http://sschmeier.github.io/bioinf-workshop/galaxy-intro/doc/GalaxyIntro_sschatmeier.pdf

An accompanying lecture for this tutorial is available at [figshare](#) ().

1.1 Overview

In this brief tutorial we will learn how to use the excellent tool [Galaxy](#) to analyze biological data. We will see how it [Galaxy](#) allows you to make use of a number of tools in a simple to use graphical interface (more on that in a moment). A user is thus not required to use any of the tools on the command-line (even though many of the integrated tools were developed for the command-line in the first place) but can fully use and control the integrated tools with the mouse pointer. In addition, it also allows developers of tools to easily integrate them into a graphical user interface system that is already known to many scientists and thus make the tools available for the research community.

Another big advantage of [Galaxy](#) is that every step of the analysis is monitored and accessible via a history. This makes reproducible research not only a possibility but also easy to facilitate. Steps from the history can be packaged into work-flows, which can be reused with different data or shared with other scientists.



Figure 1: Galaxy Community Conference 2015

[Galaxy](#) enjoys a large and growing user and developer base, which is evident by its own yearly [conference](#) (see *Figure 1*, <http://gcc2015.tsl.ac.uk/>) and participation in [Google Summer of Code](#). It is relatively easy to find help should one need it, e.g. through their [mailing list](#) or [wiki](#) (<http://wiki.galaxyproject.org/>). Also, many commercial companies that provide next-generation sequencing services, provide Galaxy instances to analyze your data (e.g. we at [New Zealand Genomics Limited](#) (<http://nzgenomics.co.nz>) have a full fledged installation on our infrastructure ready for scientist to be used).

1.2 How to get access to Galaxy

There many option available to either give [Galaxy](#) a test run or do a full analysis with it. There is a ever growing list of public servers [available](#), some of which might have certain restrictions, e.g. maximum data-file size, etc. The standard server is accessible at: <https://usegalaxy.org/>

You can start your own [Galaxy](#) instances on [Cloud](#) infrastructure, e.g. [Amazon Cloud Services](#), should you have bigger analysis needs that you want to perform in the cloud.

You can [download](#) and install [Galaxy](#) on you own machine or server, even integrating a computer cluster on the back-end.

You can install [BioLinux](#) on you own machine or run [BioLinux](#) as a virtual machine and you are set as well, as [Galaxy](#) comes pre-installed on [BioLinux 8](#).

1.3 The user interface

1.3.1 Basics

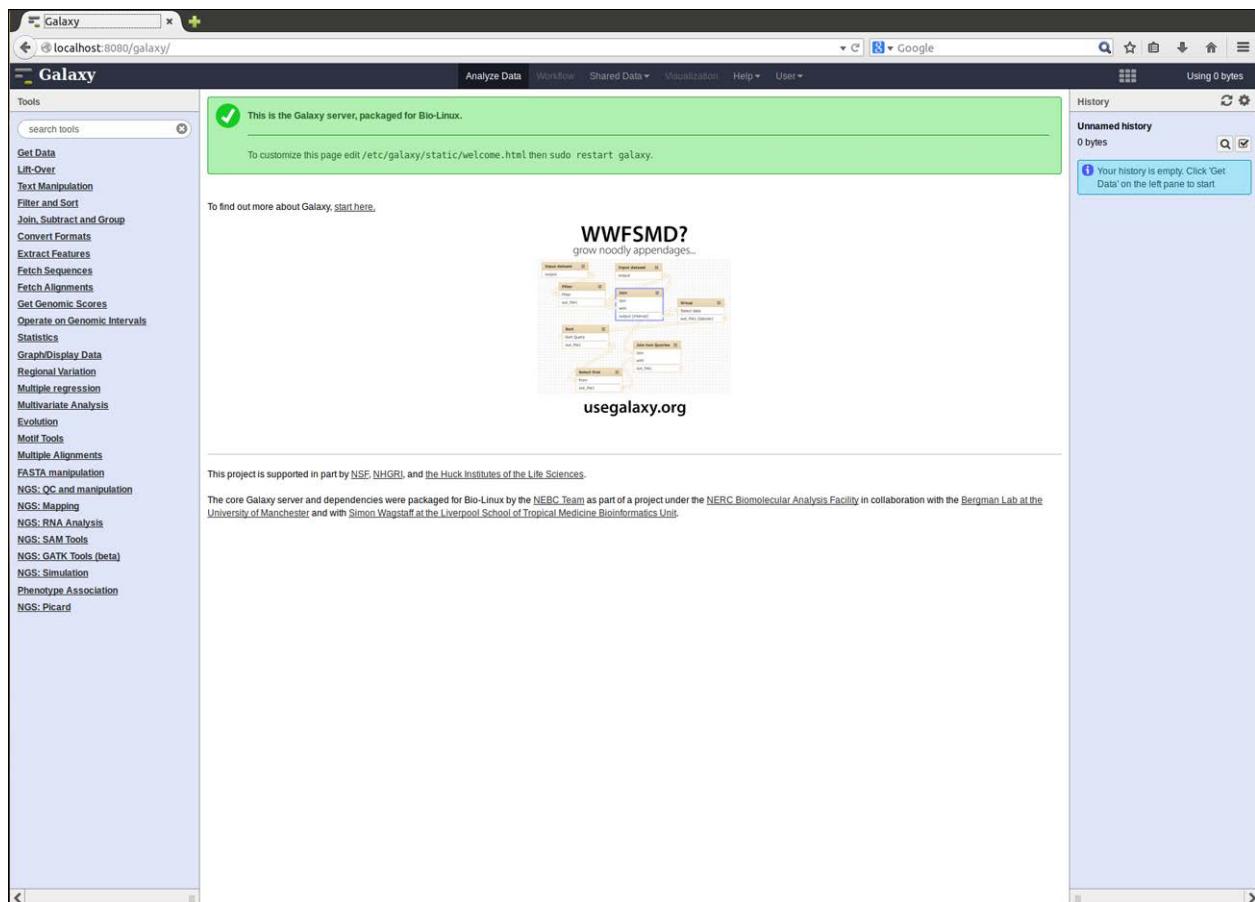


Figure 2: Galaxy overview.

There are 3 areas of interest for now (see *Figure 3*):

1. The links to the tools that the [Galaxy](#) installation contains (this can very from [Galaxy](#) instance to instance).
2. The working area, where we can change parameters of the tools that we want to use for some of our data.
3. The history panel that contains all the data and steps we performed on the data.

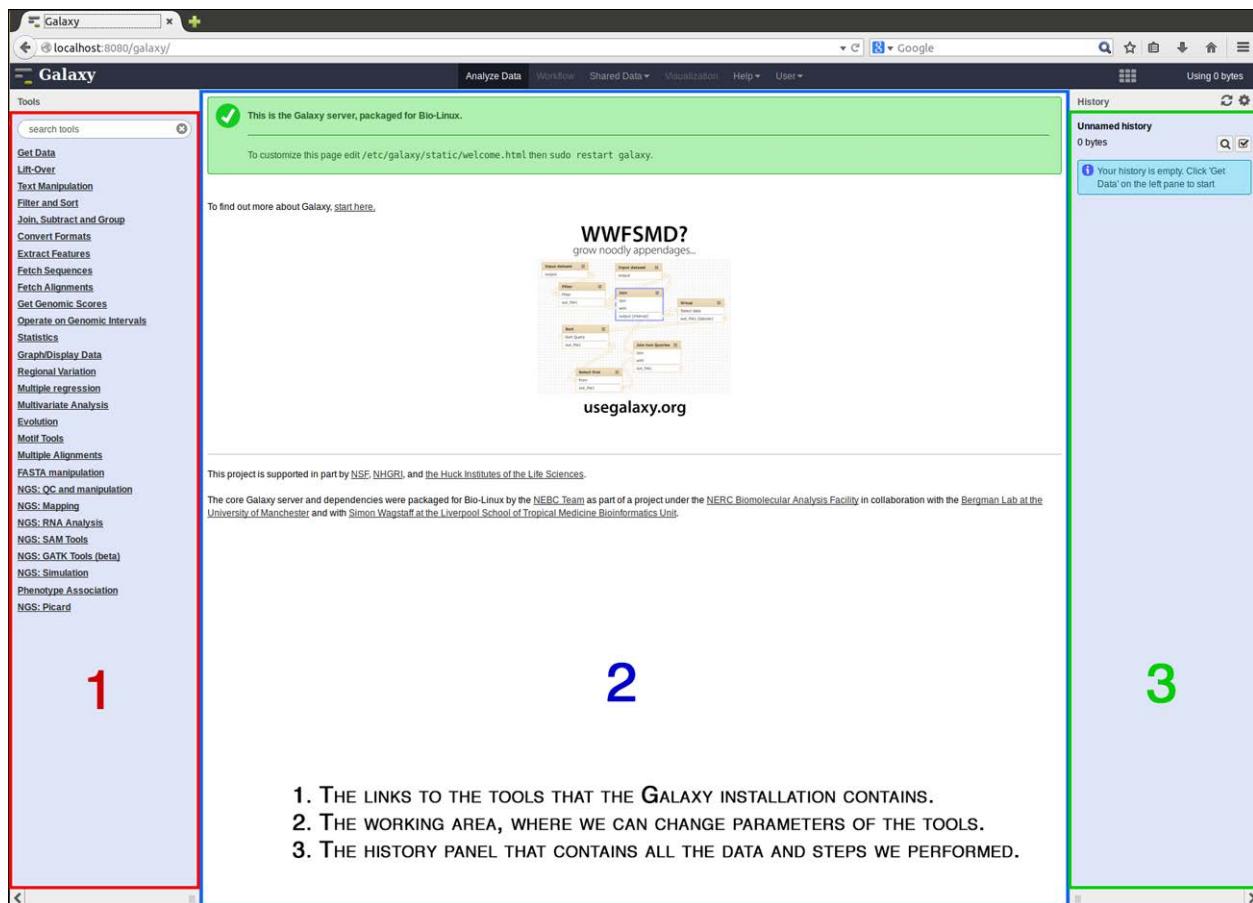


Figure 3: Galaxy main window areas.

1.3.2 User accounts

If you plan to use the public available [Galaxy](https://usegalaxy.org/) instance at <https://usegalaxy.org/>, it is a good idea to create a user account. This is relatively straight forward, just click on **User** in the top panel (see *Figure 4*) and then **Register** (1). This will allow you, amongst other things, to save histories, but more on this in later ([2.1](#)).

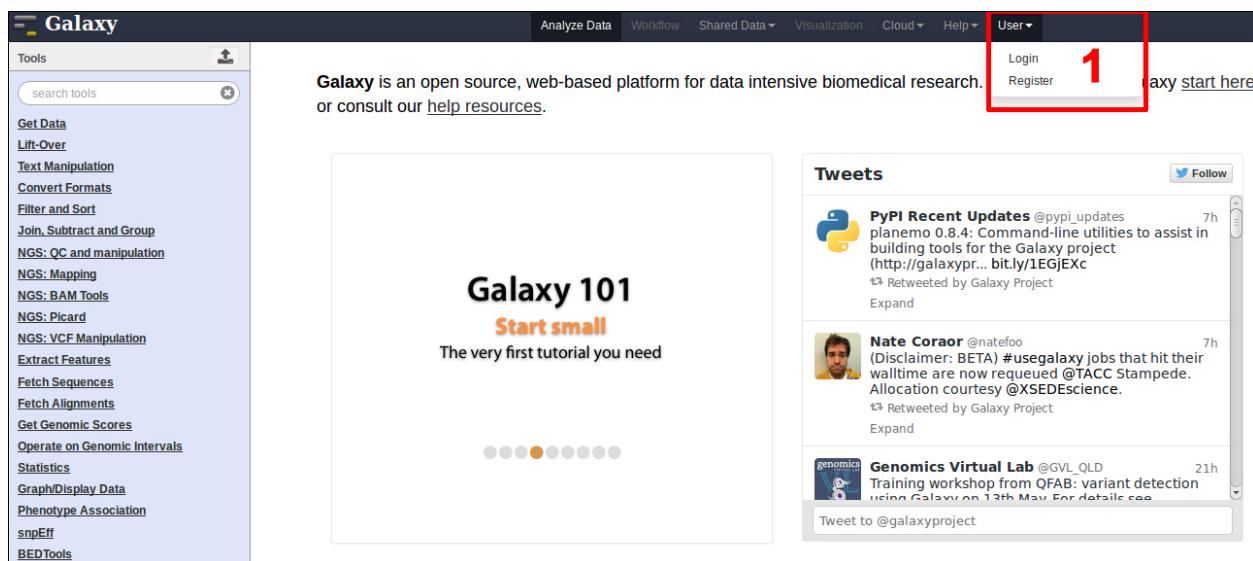


Figure 4: Galaxy user login.

1.4 A word on tools

The tools that you find in the tools area of the [Galaxy](#) instance are nothing else than programs that were originally written for the command-line. As long as you have/write a program that expects a input-file and out-put-file as command-line arguments, it is quite easy to [integrate a tool](#) into an local [Galaxy](#) installation.

Attention! The tools that you find in your [Galaxy](#) instance might differ depending on where you access the particular [Galaxy](#) installation/instance., e.g. you might find a different toolset at the standard online [Galaxy](#) instance at <https://usegalaxy.org/>, than on your local installation.

1.5 The task at hand

The overall purpose in this tutorial is to:

- Understand the [Galaxy](#) system
- Understand how to get your data of interest into the system
- Understand how to do simple data manipulation tasks
- Understand how the [Galaxy](#) History system works
- Understand how to set up a workflow and run your data through it (*advanced*)

In order to develop an understanding of the points above, you are required to solve the following problem (see [Figure 6](#)):

“Using Galaxy, find the mouse chromosome X genes that have single nucleotide polymorphisms in their upstream region”

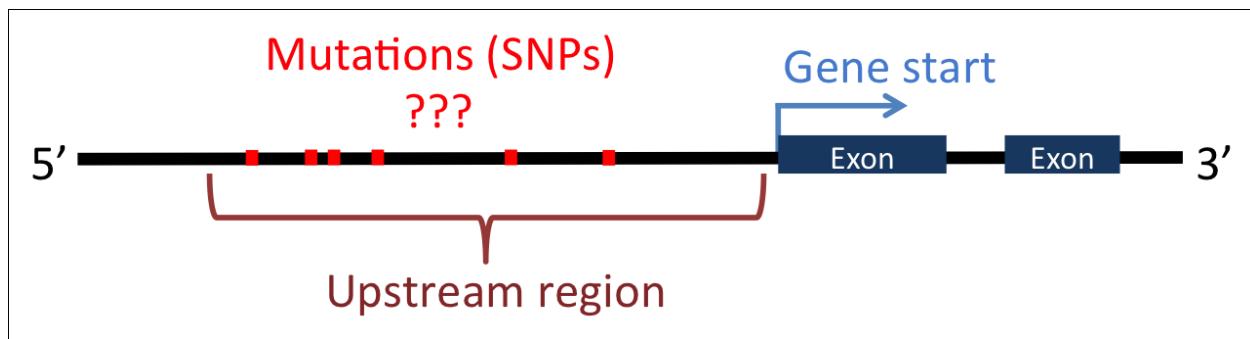


Figure 6: Are there any SNPs?

The individual steps required to find those mutations are:

1. Get single nucleotide polymorphism (SNP) data for chromosome X
2. Get all gene locations on chromosome X
3. Get upstream regions of the genes
4. Overlap the SNPs with the genic upstream regions
5. Visualise results in a genome browser

The deliverables are:

1. The list of genes that have SNPs in their upstream region.
2. The list of SNPs that are located in the upstream regions.
3. A screenshot of one of the genes with SNPs upstream (**other** than gene *ENSMUST00000105020* from Figure 29).

1.6 Loading your own data

Download the following file to your computer: [mm9_chrX_SNP128_set.bed](http://sschmeier.github.io/bioinf-workshop/galaxy-intro/data/mm9_chrX_SNP128_set.bed) or at http://sschmeier.github.io/bioinf-workshop/galaxy-intro/data/mm9_chrX_SNP128_set.bed.

The file is in **bed-format**, a simple tab-separated format containing 6 columns: **chromosome, start, stop, name, score, strand**.

Hint! Bed-format files can have more or less columns. However, the first three columns are the bare minimum.

1. On your **Galaxy** window go to the upper left in the tools area and click on **Get Data**. A subsection of **Get Data** will open and show available option for you to get data into the **Galaxy** system (see Figure 5).
2. Choose **Upload File from your computer**.

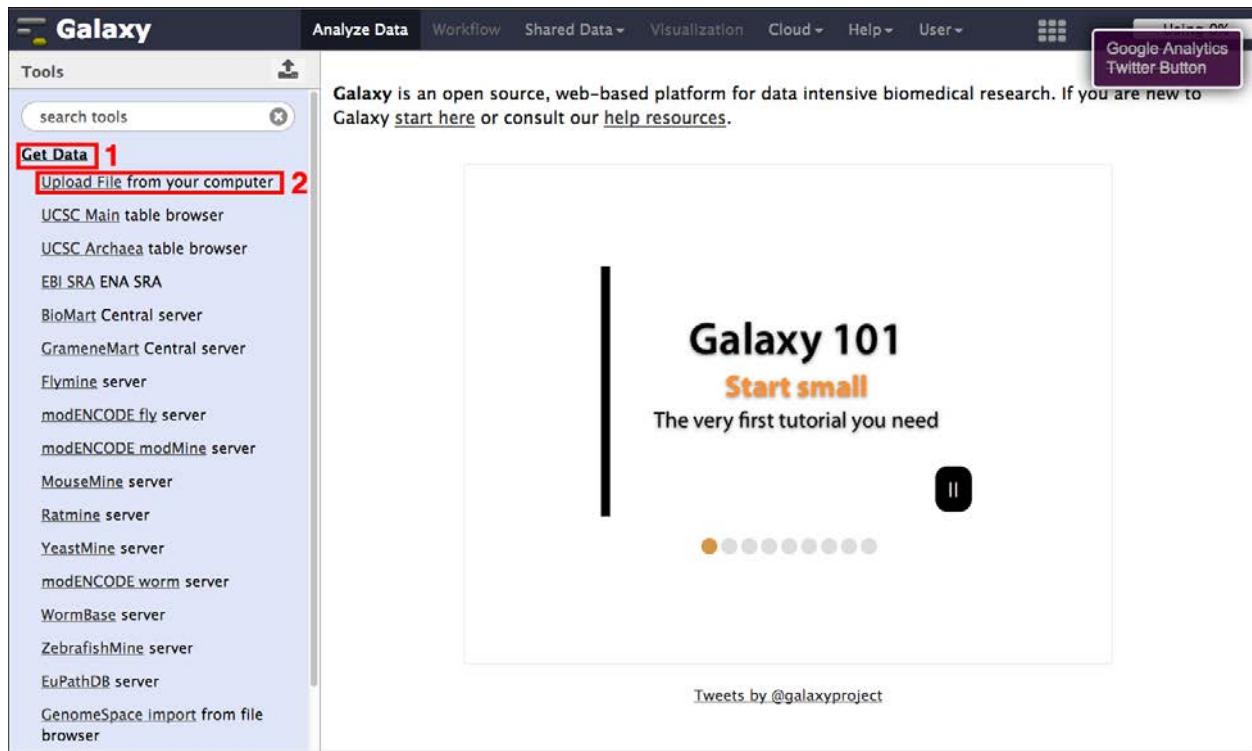


Figure 5: Get data tools.

1. An additional window should open that allows you to select the your file (see *Figure 7*).
2. You can specify the species, given that we are looking at mouse data from mm9 set it to the same.

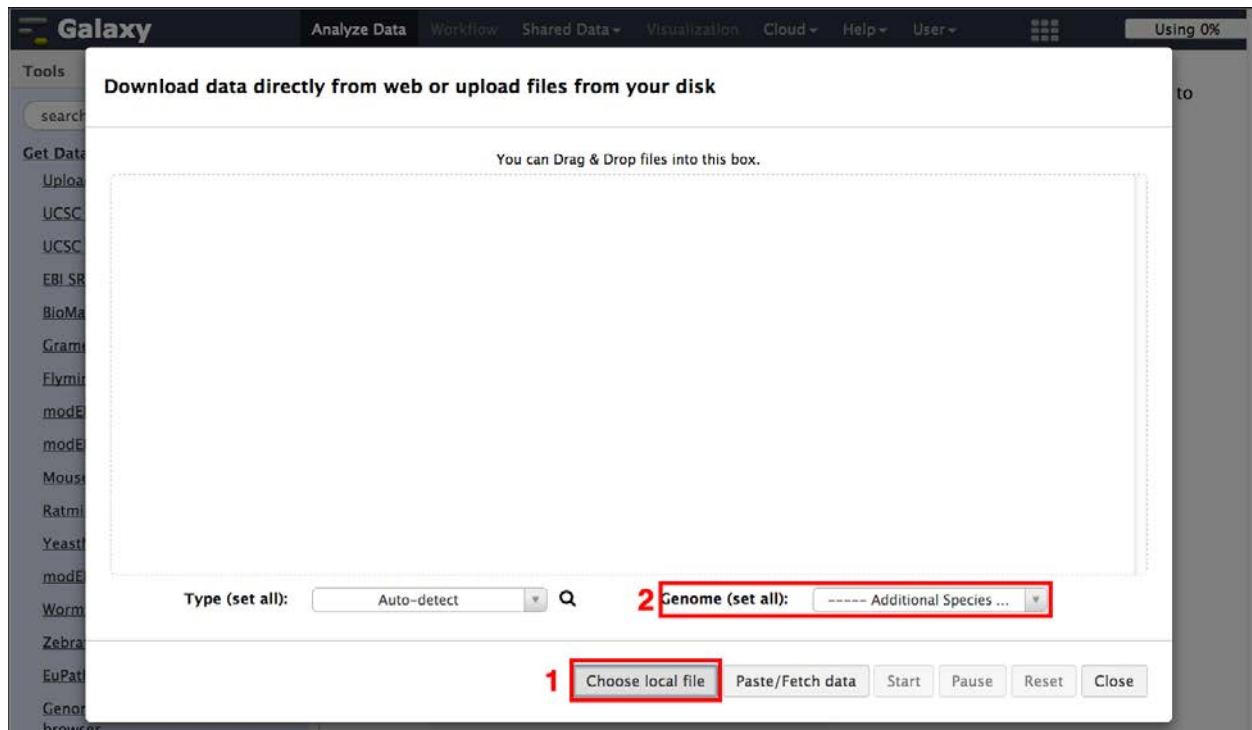


Figure 7: Data upload interface.

Once you hit the **Start** button, your data/analysis will be uploaded. In your history your data goes through three stages indicated by three different colours (see *Figure 8*):

1. Grey: Scheduled for uploading/running
2. Yellow: Currently running
3. Green: Dataset/analysis is ready

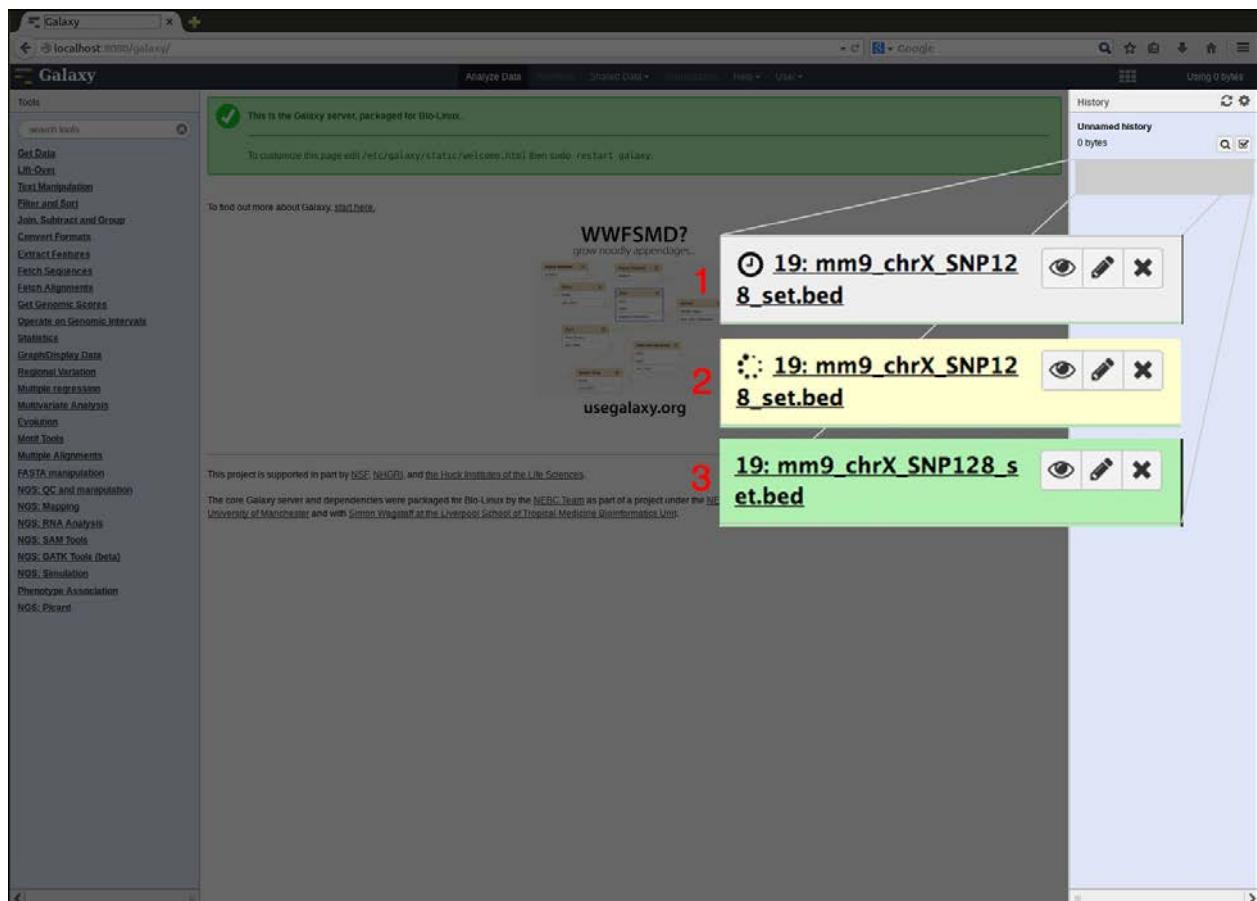


Figure 8: The history panel: Different color codes.

1. Click on the filename and you get some information about the data (see *Figure 9*).
2. Here you will see information like how many regions (lines) are in the file, the format and genome
3. Here you can download the data, get even more information about the data and run the job again (here it would reload the data)

Galaxy is an open source, web-based platform for data intensive biomedical research. If you are new to Galaxy [start here](#) or consult our [help resources](#).

Galaxy 101

Start small

The very first tutorial you need

1. Chrom	2. Start	3. End	4. Name	5.	6.	7.
chrX	3242568	3242569	rs51257154	0	-	1
chrX	3242572	3242573	rs49693543	0	-	1
chrX	3242573	3242574	rs45795462	0	-	1
chrX	3749157	3749158	rs45795462	0	+	1
chrX	3749158	3749159	rs49693543	0	+	1
chrX	3749162	3749163	rs51257154	0	+	1
chrX	3907318	3907319	rs48647149	0	+	1
chrX	3907321	3907322	rs48584752	0	+	1
chrX	3907739	3907740	rs45858970	0	+	1
chrX	3907803	3907804	rs48529475	0	+	1
chrX	3907824	3907825	rs46088235	0	+	1

Figure 9: The history panel: Investigating datasets.

Within the history panel and your data set there are several buttons of importance. The first one which looks like an eye will display you data in the working area (see Figure 10).

1	2	3	4	5	6	7
chrX	3242568	3242569	rs51257154	0	-	1
chrX	3242572	3242573	rs49693543	0	-	1
chrX	3242573	3242574	rs45795462	0	-	1
chrX	3749157	3749158	rs45795462	0	+	1
chrX	3749158	3749159	rs49693543	0	+	1
chrX	3749162	3749163	rs51257154	0	+	1
chrX	3907318	3907319	rs48647149	0	+	1
chrX	3907321	3907322	rs48584752	0	+	1
chrX	3907739	3907740	rs45858970	0	+	1
chrX	3907803	3907804	rs48529475	0	+	1
chrX	3907824	3907825	rs46088235	0	+	1

Figure 10: The history panel: The eye button.

1. The second button will allow you to edit your data (see Figure 11).
2. You can change the file-name.
3. Change the assignment of column numbers to particular properties.
4. Save your changes.

The screenshot shows the Galaxy web interface. On the left, there's a sidebar with various tools like 'Get Data', 'UCSC Main table browser', and 'Convert Formats'. The main area has tabs for 'Attributes', 'Convert Format', 'Datatype', and 'Permissions'. The 'Attributes' tab is active, showing a form to edit dataset attributes. The 'Name' field contains 'mm9_chrX_SNP128_set.bed' (labeled 2). The 'Chrom column' dropdown is set to '1' (labeled 3). The 'Strand column' dropdown is set to '6' (labeled 3). The 'Score column for visualization' dropdown has '4' selected (labeled 3). A 'Save' button is at the bottom (labeled 4). To the right is the 'History' panel, which lists datasets. One dataset, '19: mm9_chrX_SNP128_set.bed', is highlighted in green and has a red box around its delete icon (labeled 1).

Figure 11: The history panel: Data editing interface.

The last button can delete your data/analysis again from the history panel (see Figure 12).

This screenshot is similar to Figure 11 but focuses on the history panel. It shows a dataset named '19: mm9_chrX_SNP128_set.bed' in the list. The delete icon for this dataset is highlighted with a red box (labeled 1).

Figure 12: The history panel: The delete button.

1.7 Loading data from the web

Now we are focusing on getting some data from the [UCSC table browser](#). Many people UCSC were quite busy integrating lots of data and there is plenty of data available especially for mammalian model systems.

1. On your [Galaxy](#) window go to the upper left in the tools area and click on **Get Data** (see Figure 13). A subsection of **Get Data** will open and show available option for you to get data into the [Galaxy](#) system.
2. Click on [UCSC Main table browser](#). This will open the [UCSC table browser](#) in your [Galaxy](#) working area.

3. Here you can choose the genome that you want the data from, we will choose mm9
4. Here you can choose the kind of data that you which to download from the particular genome, we will choose here the **Genes and Gene Prediction group** and the **UCSC Genes** as well as the **knownGene** table. The **describe table schema** button will get you to another webpage that describes the data within the **knownGene** table. Feel free to explore.
5. Here you can chose if you which to download data from the whole genome or a subportion of it. We will choose here only data from **chrX** type this in the field and hit **lookup** button which will complete the start and stop coordinates of the genome.
6. Here we can specify the output-format. It is important here to make sure that the **Send output to Galaxy** choice is selected . Also, we want BED-format again.
7. After we are finsihed we can hit the **get output** button, after which our requested data will be loaded into the **Galaxy** interface.

The screenshot shows the Galaxy Table Browser interface. On the left, there's a sidebar with a 'Get Data' menu. The main area is titled 'Table Browser' and contains various configuration options. Several fields and buttons are highlighted with red boxes and numbered 1 through 7:

- 1. 'Get Data' menu item.
- 2. 'UCSC Main table browser' option in the 'Get Data' menu.
- 3. 'clade: Mammal', 'genome: Mouse', 'assembly: July 2007 (NCBI37/mm9)' dropdowns.
- 4. 'group: Genes and Gene Predictions', 'track: UCSC Genes', 'table: knownGene' dropdowns, and 'describe table schema' button.
- 5. 'region: genome position chrX:1-166650296', 'lookup' button, and 'define regions' link.
- 6. 'output format: BED - browser extensible data', 'Send output to Galaxy' checked, and 'GREAT' checkbox.
- 7. 'get output' button.

Figure 13: Loading UCSC data into Galaxy.

Finally, your data should appear in the right hand side history panel.

1.8 Loading shared data

Another way of loading data into your history panel is by loading data that was shared with you through Galaxy. On the upper panel click on **Shared Data** and then on **Data Libraries** (see *Figure 14*).

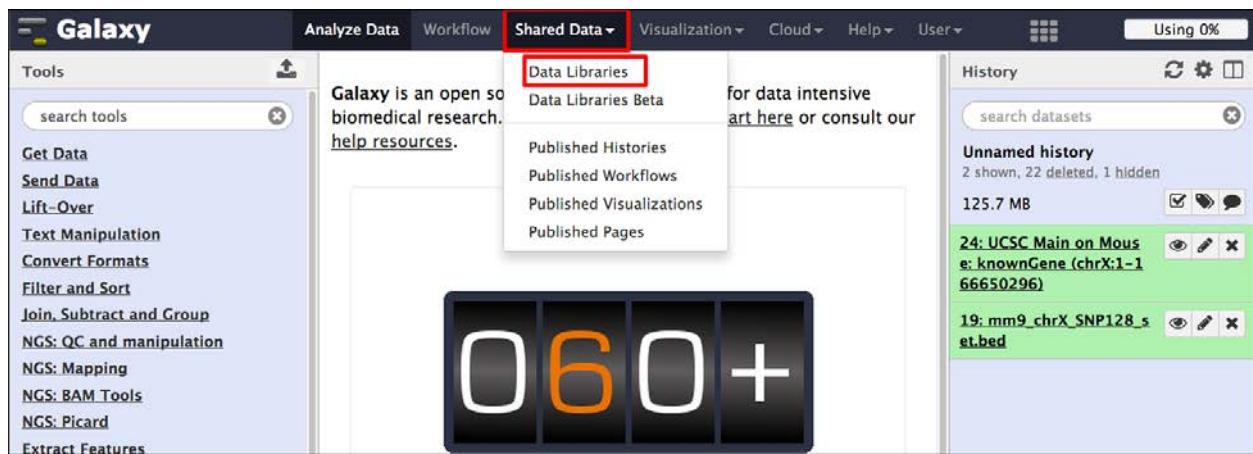


Figure 14: Loading shared data.

Here you will find a search field to search for available datasets (see Figure 15). Search for mouse because currently we are working with mouse data.

Data library name	Data library description
1000 Genomes	Data from the 1000 Genomes Project FTP site
AC-exome	Data for two papers about the Khoisan and other populations.
Bushman	
ChIP-Seq Example Data	

Figure 15: The shared data search bar.

Choose the **ChIP-Seq Mouse Example** dataset from the ENCODE project. This is data of chromatin immunoprecipitation followed by sequencing to find regions in the genome where transcription factors bind.

Data library name	Data library description
ChIP-Seq Mouse Example	Data used in examples that demonstrate analysis of ChIP-Seq data

Figure 16: Details about shared data.

Here you see an overview of the datasets available (see Figure 16). You can choose the dataset, select **Import to current history**, and hit **Go** (see Figure 17).

Data Library “ChIP-Seq Mouse Example”

Use this data to test out and learn Galaxy's ChIP-Seq capabilities. It has been scaled down to relatively small sizes.

These files are from this mouse ChIP-SEQ experiment in the ENCODE project. These data were generated and analyzed by the labs of Michael Snyder at Stanford University and Sherman Weissman at Yale University.

The original files from ENCODE were too large to use in teaching examples, so they have been reduced to contain only data that corresponds to chromosome 19 (the shortest).

These files were created by, well, cheating. We first processed the entire dataset, mapping it to MM9. When went back and extracted from the original datasets only those records that eventually mapped to chromosome 19.

Name	Message	Data type	Date uploaded	File size
Mouse ChIP-Seq example Control Data, chr19, mm9	Control file for mouse ChIP-Seq example. An ungroomed Illumina FASTQ file, it contains only reads that map to chr19, mm9.	fastq	Mon Sep 19 20:01:54 2011 (UTC)	84.1 MB
Mouse ChIP-Seq Example Experimental Data, chr19, mm9	Experimental results for mouse ChIP-Seq example. An ungroomed Illumina FASTQ file that contains only reads that map to chr19, mm9.	fastq	Mon Sep 19 20:07:43 2011 (UTC)	47.4 MB

For selected datasets: Import to current history Go

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

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

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

Figure 17: Selecting shared data for loading into Galaxy.

Once the data is loaded in your history [Galaxy](#) will inform you (see *Figure 18*). You can get back to your working area by clicking on **Analyze Data**.

Data Library “ChIP-Seq Mouse Example”

1 dataset imported into 1 history: Unnamed history

Use this data to test out and learn Galaxy's ChIP-Seq capabilities. It has been scaled down to relatively small sizes.

These files are from this mouse ChIP-SEQ experiment in the ENCODE project. These data were generated and analyzed by the labs of Michael Snyder at Stanford University and Sherman Weissman at Yale University.

The original files from ENCODE were too large to use in teaching examples, so they have been reduced to contain only data that corresponds to chromosome 19 (the shortest).

These files were created by, well, cheating. We first processed the entire dataset, mapping it to MM9. When went back and extracted from the original datasets only those records that eventually mapped to chromosome 19.

Name	Message	Data type	Date uploaded	File size
Mouse ChIP-Seq example Control Data, chr19, mm9	Control file for mouse ChIP-Seq example. An ungroomed Illumina FASTQ file, it contains only reads that map to chr19, mm9.	fastq	Mon Sep 19 20:01:54 2011 (UTC)	84.1 MB
Mouse ChIP-Seq Example Experimental Data, chr19, mm9	Experimental results for mouse ChIP-Seq example. An ungroomed Illumina FASTQ file that contains only reads that map to chr19, mm9.	fastq	Mon Sep 19 20:07:43 2011 (UTC)	47.4 MB

For selected datasets: Import to current history Go

Figure 18: Successful loaded shared data.

You can get rid of the dataset again in your history as it will not be used anymore in theis

1.9 Working with data

The aim here is to get understand how [Galaxy](#) can help you to prepare your data to be able to analyze it further. We will perform some easy tasks like removing redundant information, renaming new datasets, sub-selecting regions of interest, extending our genomic regions to look at promoters upstream of genes, finding the SNPs from our set that overlap the promoter regions.

1.9.1 Renaming files

You should aim at naming your files in a manner that they are easily recognizable. This is especially important once we manipulate them and create new files. You should make it a habit of renaming a file after it was created to keep track of what they are.

1. Click on the **edit icon** of the file you wish to change (see *Figure 19*).
2. Type a new filename in the **Name** field.
3. Click on the **Save** button

The screenshot shows the Galaxy web interface. On the left, a sidebar lists various tools and data types. In the center, a modal window titled 'Edit Attributes' is open, showing a form with a 'Name' field containing 'mm9_knownGene_chrX'. A red box labeled '2' highlights this field. To the right, the 'History' panel displays a list of datasets. One dataset, '24: UCSC Main on Mouse knownGene (chrX:1-1 66650296)', is selected and its details are shown in a preview pane. A red box labeled '1' highlights this dataset.

Figure 19: Renaming datasets.

Attention! I also renamed the data ***Mouse ChIP-Seq example Control Data, chr19, mm9*** to ***-> mm9_ChIP_chr19_control*** and the data ***mm9_chrX_SNP128_set.bed*** to ***-> mm9_chrX_SNP128***.

The screenshot shows the Galaxy web interface. The main page features a large banner with the text 'Running Your Own'. The 'History' panel on the right shows three datasets: '26: mm9_ChIP_chr19_control', '24: mm9_knownGene_chrX', and '19: mm9_chrX_SNP128'. All three datasets are highlighted with red boxes.

Figure 20: Overview of available datasets.

Attention! The numbering of the datasets here might be different from yours depending on how many datasets you have been working on before. *Figure 20* above shows **24: mm9_knownGene_chrX**, however, this may vary for you (and might vary in what follows here as I might have done this tutorial in multiple sessions.). This is one reason why it is a good idea to rename the dataset.

1.9.2 Removing unwanted information

Our gene BED-file that we retrieved from [UCSC table browser](#) is in BED 12 format, e.g. it contains 12 columns, but only the first 6 are necessary for our purposes. Thus, we aim at removing the extra columns to make the file more readable. Let's do this by

1. Clicking on the **Text manipulation** tools section (see *Figure 21*)
2. Selecting the **Cut** tool.
3. Insert the columns you want to retain. We want the first 6 columns.
4. Choose the right file to do the manipulation on
5. **Execute** the tool

The screenshot shows the Galaxy web interface. On the left, the 'Tools' sidebar is open, showing various bioinformatics tools. The 'Text Manipulation' section is highlighted with a red box and labeled '1'. Below it, the 'Cut columns from a table' tool is selected and highlighted with a red box and labeled '2'. The main workspace shows the 'Cut columns from a table' tool configuration. The 'Cut columns' field contains 'c1,c2,c3,c4,c5,c6' (3). The 'Delimited by' dropdown is set to 'Tab' (4). The 'From' dropdown shows '24: mm9_knownGene_chrX' (4). The 'Execute' button is highlighted with a red box and labeled '5'. To the right, the 'History' panel shows a list of datasets, including 'Bioinf-course1' (3 shown, 23 deleted, 1 hidden), '26: mm9_ChIP_chr1_9_control', '24: mm9_knownGene_chrX', '19: mm9_chrX_SNP1', and '28'. The 'Using 0%' status bar is at the top right.

Figure 21: Cutting columns.

You should see a new file in the history. Here it is being scheduled for execution and should be green once the job is finished (see *Figure 22*). Please rename the resulting dataset to -> ***mm9_knownGene_chrX_short***.

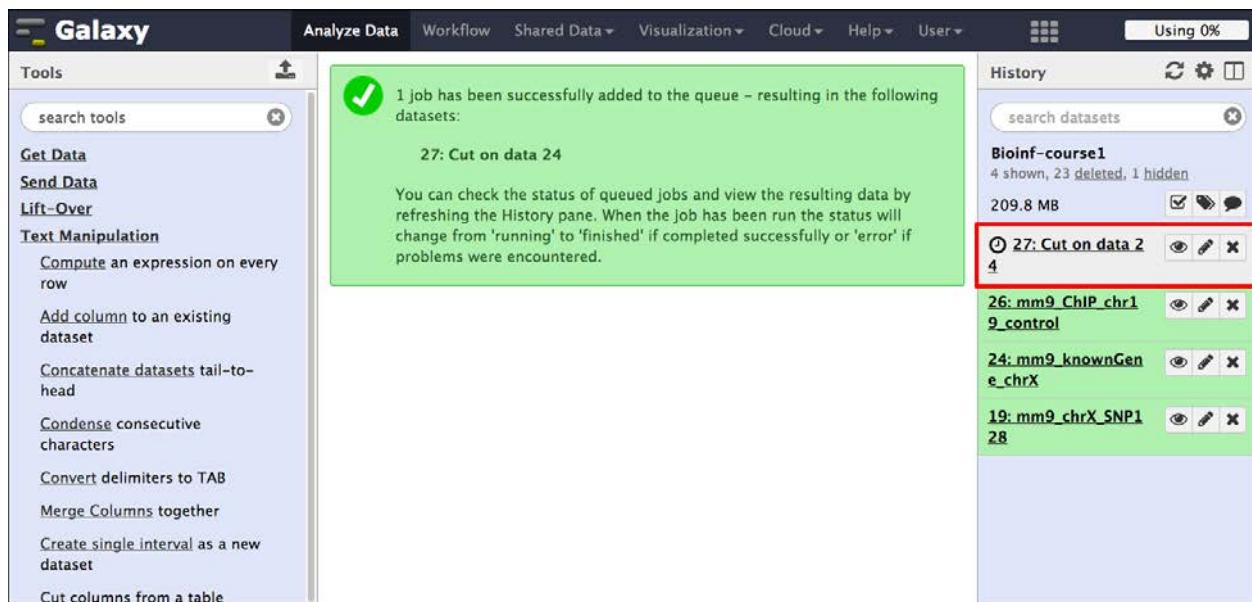


Figure 22: Successful cut on a dataset.

1.9.3 Creating flanking regions

Because we are interested to look in the promoter regions of our genes we need to extract those. We here define the promoter as upstream regions from the transcription start site.

1. Find the **Operate on Genomic Intervals** sections (see *Figure 23*).
2. Select the **Get flanks** tool
3. Choose the right dataset: **mm9_knownGene_chrX_short**
4. The region we are interested in is **Around Start**
5. We want the **Upstream** region
6. We want **5000** bases upstream
7. **Execute**

The screenshot shows the Galaxy web interface with the 'Get flanks' tool selected. The left sidebar contains a list of tools under the 'Operate on Genomic Intervals' section, with 'Get flanks returns flanking region/s for every gene' highlighted. The main panel shows the tool configuration:

- Region:** 27: mm9_knownGene_chrX_short (highlighted by red box 3)
- Location of the flanking region/s:** Upstream (highlighted by red box 5)
- Length of the flanking region(s):** 5000 (highlighted by red box 6)
- Offset:** 0
- Execute:** button (highlighted by red box 7)

The note below the form states: "This tool finds the upstream and/or downstream flanking region(s) of all the selected regions in the input file. Note: Every line should contain at least 3 columns: Chromosome number, Start and Stop co-ordinates. If any of these columns is missing or if start and stop co-ordinates are not numerical, the tool may encounter exceptions and such lines are skipped as invalid. The number of invalid skipped lines is documented in the resulting history item as a 'Data issue'."

The right panel shows the Galaxy history:

- Bioinf-course1 (4 shown, 27 deleted, 1 hidden)
- 210.0 MB
- 27: mm9_knownGene_chrX_short (2,021 regions, format: interval, database: mm9)
- display at Ensembl Current, display at UCSC main
- 1. Chrom 2. Start 3. End 4. Name 5
- chrX 3241669 3243629 uc009skj.1 0
- chrX 3410667 3412627 uc009skk.1 0
- chrX 3461360 3463320 uc009skl.1 0
- chrX 3546313 3547091 uc012hdv.1 0
- chrX 3665477 3667437 uc009skm.1 0
- chrX 3748193 3749684 uc009skn.1 0

Other datasets in the history include 26: mm9_ChIP_chr19_ntrol, 24: mm9_knownGene_ch_rX, and 19: mm9_chrX_SNP128.

Figure 23: Getting flanking regions of intervals.

Attention! I renamed the resulting dataset -> **mm9_chrX_promoter**

1.9.4 Filter data

Filtering data can be done in many different ways, however, here we use the **filter** tool (see Figure 24).

1. Find the **Filter and Sort** tool section (see Figure 24)
2. Select the **Filter** tool
3. Select our promoter dataset: **mm9_chrX_promoter**
4. We only want promoter within the first **8000000** bases, the start position of genes is specified in the second column (**c2**)
5. **Execute**

The screenshot shows the Galaxy web interface with the 'Filter' tool selected. The left sidebar contains a list of tools under the 'Text Manipulation' category, with 'Filter and Sort' highlighted (1). A red box highlights the 'Filter data on any column using simple expressions' link (2). The main panel shows the 'Filter' configuration:

- Filter:** dropdown set to '15: mm9_chrX_promoter' (3).
- With following condition:** input field containing 'c2<8000000' (4).
- Number of header lines to skip:** input field containing '0'.
- Execute:** button (5).

A TIP message states: "Double equal signs, ==, must be used as "equal to" (e.g., c1 == 'chr22')". Another TIP message states: "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".

The right panel shows the 'History' section with several datasets listed:

- Bioinf-course 1 (5 shown, 11 deleted)
- 203.4 MB
- 15: mm9_chrX_promoter** (2,021 regions, format: interval, database: mm9)
- Location: Upstream, Region: start, Flank-length: 5000, Offset: 0
- display at Ensembl Current, display at UCSC main
- 1. Chrom 2. Start 3. End 4. Name 5. 6. Str
- chrX 3243629 3248629 uc009skj.1 0 -
- chrX 3405667 3410667 uc009skk.1 0 +
- chrX 3463320 3468320 uc009skl.1 0 -
- chrX 3547091 3552091 uc012hdv.1 0 -
- chrX 3667437 3672437 uc009skm.1 0 -
- chrX 3743193 3748193 uc009skn.1 0 +

Other datasets listed in the history include:

- 14: mm9_knownGene_chrX_s
- 13: mm9_ChIP_chr19_control
- 11: mm9_knownGene_chrX
- 10: mm9_chrX_SNP128

Figure 24: Filter data.

Attention! I renamed the resulting dataset → **mm9_chrX_promoter_8000000**

Hint! If you click on the dataset name it will also tell you how many lines where extracted from the original dataset.

1.9.5 Joining/intersecting data sets

Lets find those mutations that overlap our promoter subset (see Figure 25).

1. Find the **Operate on genomic Intervals** tool section
2. Select the **Join** tool
3. Select our SNP data **mm9_chrX_SNP128** and the promoter dataset **mm9_chrX_promoter_8000000**
4. **INNER JOIN**
5. **Execute**

Join the intervals of two datasets side-by-side (Galaxy Tool Version 1.0.0)

Join

First dataset: 10: mm9_chrX_SNP128 **3**

with: 19: mm9_chrX_promoter_8000000 **4**

Second dataset:

with min overlap: 1 (bp)

Return: Only records that are joined (INNER JOIN) **4**

Execute **5**

TIP: If your dataset 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 dataset that join to a record in the second dataset. This is analogous to an INNER JOIN.

Return **all records of first dataset (fill null with ".")** returns all intervals of the first dataset, and any intervals that do not join an interval from the second dataset are filled in with a period(.). This is analogous to a LEFT JOIN.

History

search datasets

Bioinf-course 1
6 shown, 14 deleted
203.6 MB

19: mm9_chrX_promote_r_8000000
140 regions
format: interval, database: mm9
Filtering with c2<8000000, kept 6.93% of 2021 valid lines (2021 total lines).

display at Ensembl Current
display at UCSC main

1.Chrom	2.Start	3.End	4.Name	5
chrX	3243629	3248629	uc009skj.1	0 -
chrX	3405667	3410667	uc009skk.1	0 -
chrX	3463320	3468320	uc009skl.1	0 -
chrX	3547091	3552091	uc012hdv.1	0 -
chrX	3667437	3672437	uc009skm.1	0 -
chrX	3743193	3748193	uc009skn.1	0 -

18: mm9_chrX_promoter

14: mm9_knownGene_chrX_short

13: mm9_ChIP_chr19_controls

11: mm9_knownGene_chrX

10: mm9_chrX_SNP128

Figure 25: Joining datasets.

Attention! I renamed the resulting dataset → **SNPs_at_promoter**.

If you temporarily close the history tab we can have a closer look at the resulting dataset.

1	2	3	4	5	6	7	8	9	10	11	12
chrX	3243722	3243723	rs52395861	0	+	chrX	3243629	3248629	uc009skj.1	0	-
chrX	3244443	3244444	rs46254379	0	-	chrX	3243629	3248629	uc009skj.1	0	-
chrX	3244471	3244472	rs50874688	0	-	chrX	3243629	3248629	uc009skj.1	0	-
chrX	3244489	3244490	rs33264252	0	-	chrX	3243629	3248629	uc009skj.1	0	-
chrX	3244489	3244490	rs46879180	0	-	chrX	3243629	3248629	uc009skj.1	0	-
chrX	3244555	3244556	rs48315292	0	-	chrX	3243629	3248629	uc009skj.1	0	-
chrX	3244566	3244567	rs46735700	0	-	chrX	3243629	3248629	uc009skj.1	0	-
chrX	3245984	3245985	rs51980349	0	-	chrX	3243629	3248629	uc009skj.1	0	-
chrX	3246069	3246070	rs50772248	0	-	chrX	3243629	3248629	uc009skj.1	0	-
chrX	3248007	3248008	rs51574865	0	-	chrX	3243629	3248629	uc009skj.1	0	-
chrX	3248008	3248009	rs47066278	0	-	chrX	3243629	3248629	uc009skj.1	0	-
chrX	3248012	3248013	rs49511429	0	-	chrX	3243629	3248629	uc009skj.1	0	-
chrX	3248026	3248027	rs50458919	0	+	chrX	3243629	3248629	uc009skj.1	0	-
chrX	3248050	3248051	rs51752810	0	-	chrX	3243629	3248629	uc009skj.1	0	-
chrX	3248122	3248123	rs45894481	0	-	chrX	3243629	3248629	uc009skj.1	0	-
chrX	3248431	3248432	rs51916321	0	+	chrX	3243629	3248629	uc009skj.1	0	-
chrX	3248433	3248434	rs47111988	0	+	chrX	3243629	3248629	uc009skj.1	0	-
chrX	3248500	3248501	rs50501061	0	+	chrX	3243629	3248629	uc009skj.1	0	-

Figure 26: Join results.

We see that we have 2,218 SNPs overlapping promoter regions in the genes in the first 8,000,000 base pairs. The **Join** tool put the overlapping elements right next to each other.

Note! that for one particular promoter we can have several SNPs (1 in *Figure 26*).

2.0 Visualising data sets

Now that we basically have what we are looking for we want to visualise our found SNPs and the promoter that have mutations in an intuitive manner. Here, Genome Browsers come in that are helpful in getting an overview. In this section we prepare the data we would like to visualise and prepare a custom track for the [UCSC Genome Browser](#). First, what data do we want to visualise:

1. All SNPs
2. The SNPs that overlap our promoter regions
3. The promoter regions

To create a new track that we can visualise in USCS, do the following:

1. Find the **Graph/Display Data** tool section (see *Figure 27*)
2. Select the **Build custom track** tool
3. Click on insert track and select our promoter data ***mm9_chrX_promtoer_8000000***.
4. Give it a unique name
5. Insert more tracks for data like ***SNPs_at_promoter*** and ***mm9_chrX_SNP128***.
6. **Execute**

Attention! Make sure to use **unique names** for each track, because if you use the same name twice the last track overwrites the one from before.

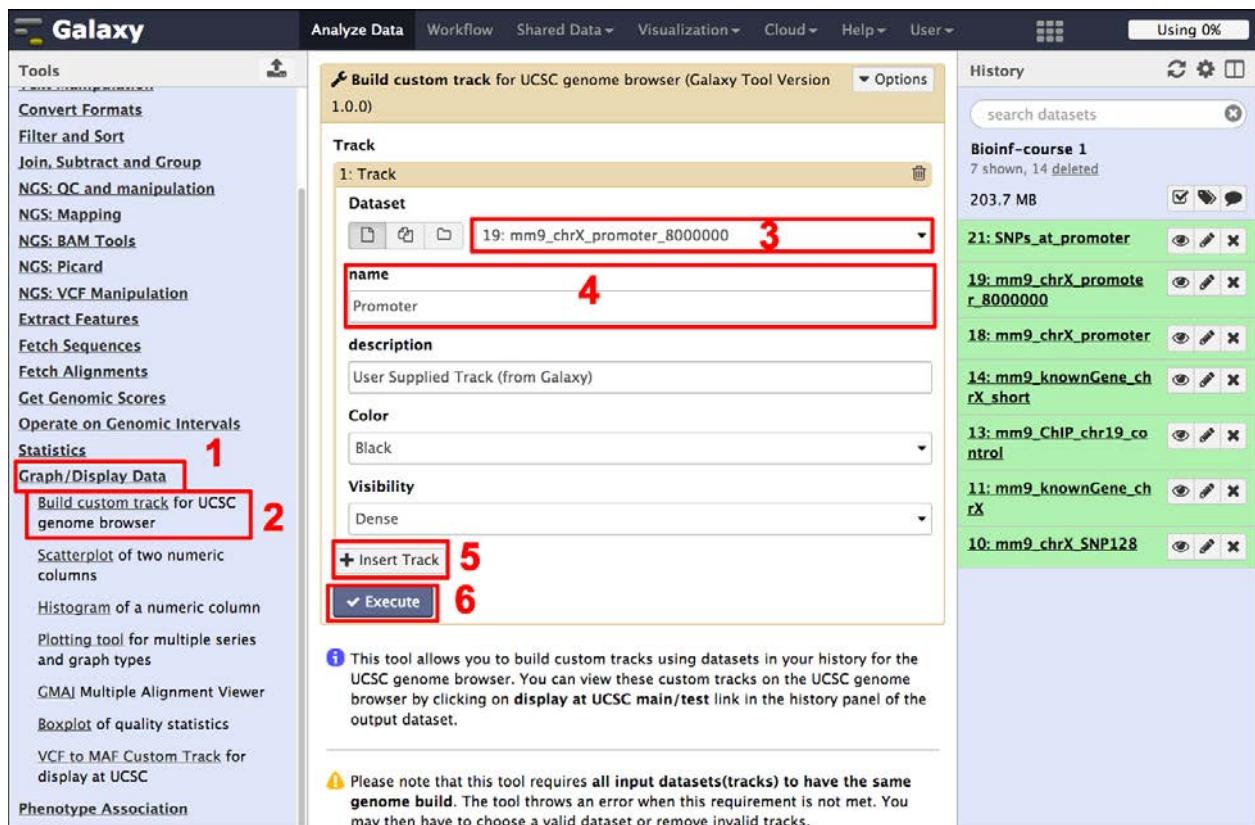


Figure 27: Building a custom UCSC track.

Once you hit the **Execute** button you should have a new track created which is visible in the history panel (1 in *Figure 28*). Click on the name of that track and click **display at UCSC main** (2 in *Figure 28*).

1	2	3	4
chrX	3243629	3248629	0
chrX	3405667	3410667	1
chrX	3463320	3468320	2
chrX	3547091	3552091	3
chrX	3667437	3672437	4
chrX	3743193	3748193	5
chrX	3902010	3907010	6
chrX	3995573	4000573	7
chrX	4069963	4074963	8
chrX	4441526	4446526	9
chrX	5046383	5051383	10
chrX	5241184	5246184	11
chrX	5619624	5624624	12
chrX	5660538	5665538	13
chrX	5660538	5665538	14
chrX	5750109	5755109	15
chrX	5897841	5902841	16
chrX	5972262	5977262	17
chrX	6149403	6154403	18
chrX	6351431	6356431	19
chrX	6618745	6623745	20
chrX	6618745	6623745	21

Figure 28: Visualising the track.

If you do so, a new window at the UCSC Genome browser will open. Put **chrX:3,237,911-3,249,163** in the search bar (1 in *Figure 29*) and you will see a position that shows what is going on. Right on top should be your three tracks located (2 in *Figure 29*). You can scroll left and right, zoom in and out to get to other promoter regions. You can also change the resolution at which your features will be shown. Many other tracks from UCSC are also shown automatically and at the bottom of the page you can chose to show or hide other tracks of interest.

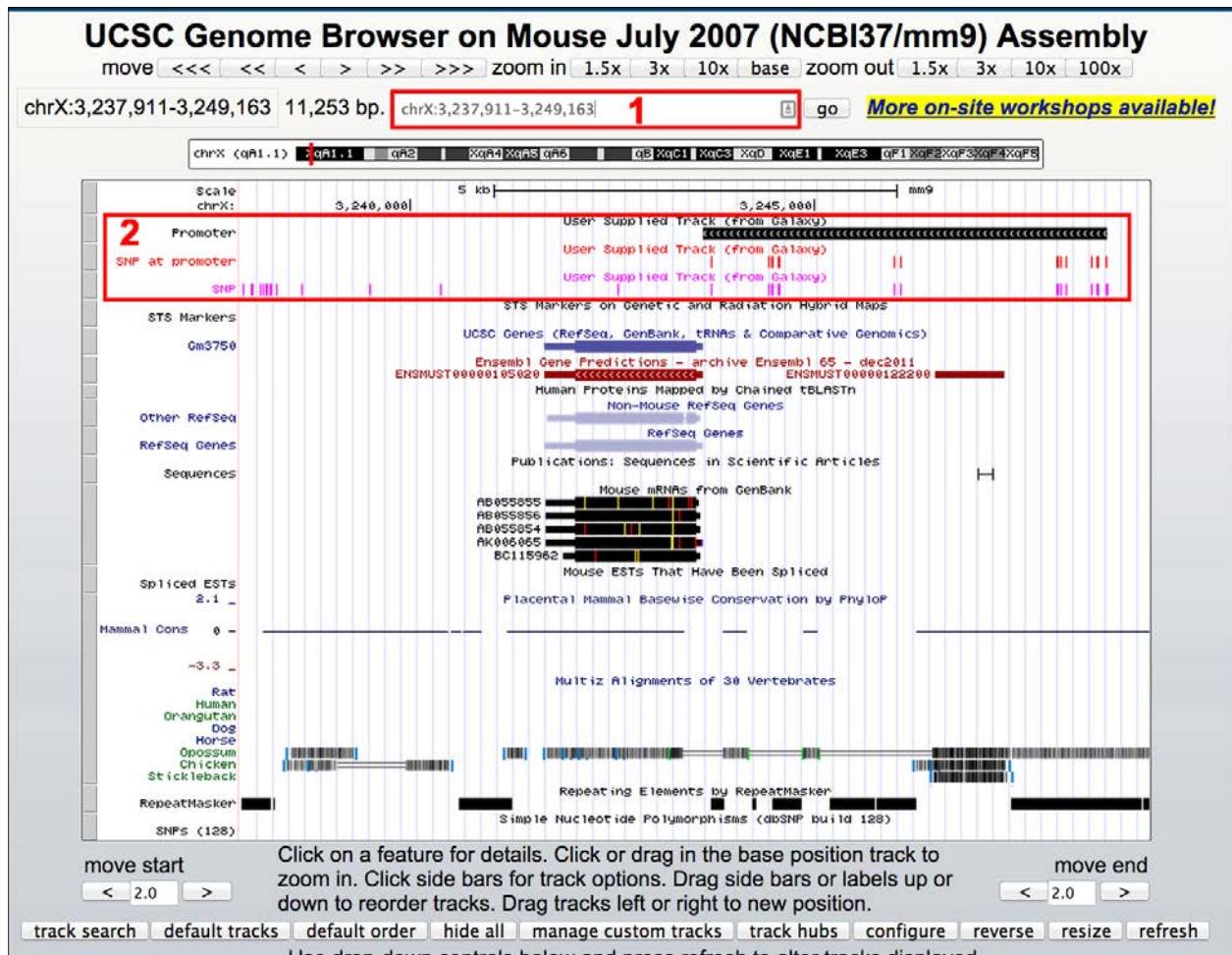


Figure 29: The custom track at UCSC.

2.1 Another word on the history

2.1.1 Saved histories

You are able to create an account on the public Galaxy [web-server](#). Once done, you will be able to save histories and fetch your old histories back. In this manner you are also able to save whole work-flows but more on that later.

For now you can look at your **Saved Histories** by clicking the config button in the upper right (see *Figure 30*).

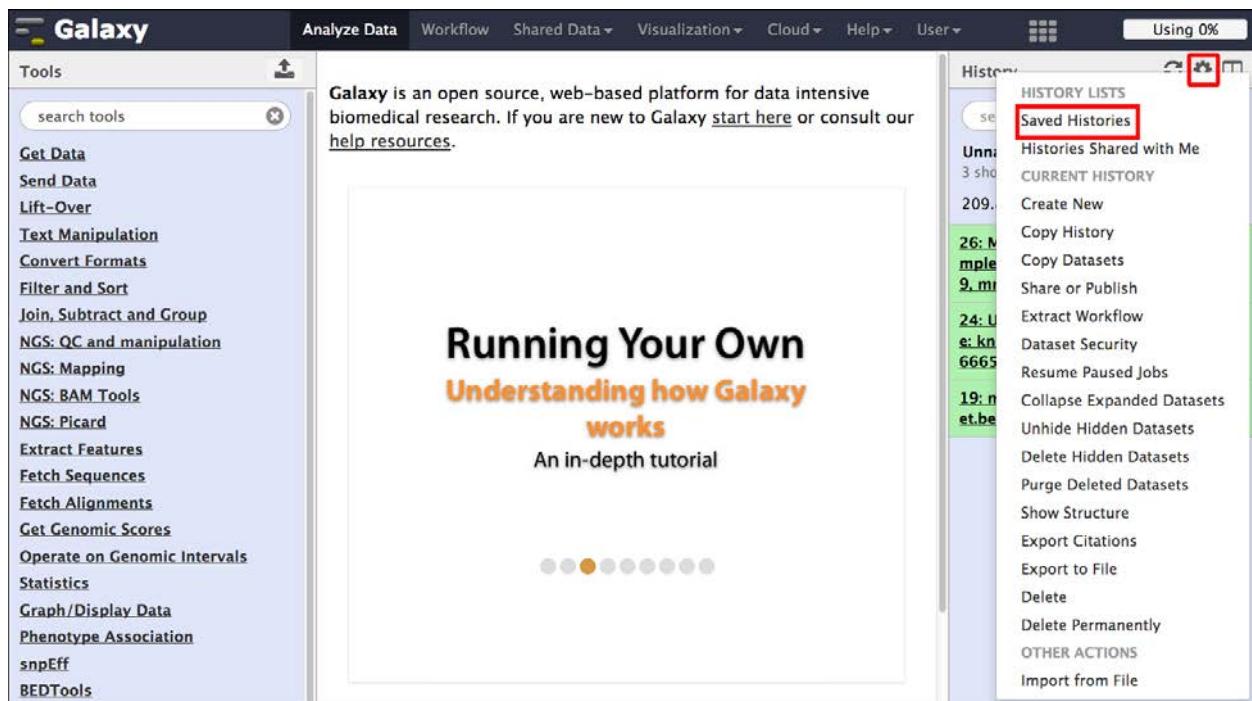


Figure 30: Saving histories.

You will see only one history the one we are currently working on. You can rename the history by clicking the name in the history panel or by doing a rename in the working area (see *Figure 31*).



Figure 31: Renaming a history.

2.1.2 Sharing a history

It is easy to share a saved history with colleagues or make them public (1 in *Figure 32*). Several options are available (see *Figure 33*).

The screenshot shows the Galaxy web interface with the title bar 'Galaxy' and various navigation links like 'Analyze Data', 'Workflow', 'Shared Data', etc. On the left, there's a 'Tools' sidebar with a search bar and a list of tool categories. The main area is titled 'Saved Histories' with a search bar. A table lists histories, including 'Bioinf-course 1'. A context menu is open over 'Bioinf-course 1', with 'Share or Publish' highlighted by a red box and a red number '1'.

Figure 32: History sharing.

This screenshot shows the 'Share or Publish History' dialog for 'Bioinf-course 1'. It includes sections for 'Make History Accessible via Link and Publish It', 'Share History with Individual Users', and a 'Back to Histories List' button.

Figure 33: Sharing options.

2.2 Workflows

2.2.1 Creating workflows

It is possible to create workflows out of histories to analyse similar type of data again with the same procedure and minimal costs. If you look into the history you can see that we still have all the steps present that were needed to come to our final result. Thus, you can convert this history into a workflow by clicking the history **Options** button (1 in Figure 34) and choosing the **Extract Workflow** option (2 in Figure 34)

The screenshot shows the Galaxy web interface. On the left, the 'Tools' panel is open, displaying a list of bioinformatics tools categorized under 'Convert Formats', 'Filter and Sort', 'Join, Subtract and Group', 'NGS: QC and manipulation', 'NGS: Mapping', 'NGS: BAM Tools', 'NGS: Picard', 'NGS: VCF Manipulation', 'Extract Features', 'Fetch Sequences', 'Fetch Alignments', 'Get Genomic Scores', 'Operate on Genomic Intervals', 'Statistics', 'Graph/Display Data', 'Phenotype Association', 'snpeff', 'BEDTools', 'Genome Diversity', 'EMBOSS', 'Regional Variation', 'FASTA manipulation', 'Evolution', and 'Multiple Alignments'. In the center, there is a table with columns labeled 1 through 10, showing genomic data for chromosomes X. On the right, the 'History' panel is visible, showing a list of datasets. A context menu is open over a dataset named 'chrX_promote' in the history list. The menu items include: HISTORY LISTS, Saved Histories, Histories Shared with Me, CURRENT HISTORY, Create New, Copy History, Copy Datasets, Share or Publish, Extract Workflow (which is highlighted with a red box and a red number '2'), Dataset Security, Resume Paused Jobs, Collapse Expanded Datasets, Unhide Hidden Datasets, Delete Hidden Datasets, Purge Deleted Datasets, Show Structure, Export Citations, Export to File, Delete, Delete Permanently, OTHER ACTIONS, and Import from File.

Figure 34: Creating a workflow.

We focus on the center pane in the next screenshot (see Figure 35). Here, we are able to choose which steps to include/exclude and how to name the newly created workflow. Do not focus on the naming of the individual datasets, we need to edit this afterwards in any case. The importance is that all of the analysis steps are included, we can shuffle them around later.

1. You want to give the workflow a proper name
2. We need to realize that the data upload can unfortunately not be part of the workflow, the workflow can only on datasets already in our history. However, we only need two datasets, so deselect the third.
3. We do not include the filter step as we are really interested in finding all SNPs in **all** promoter regions not only in the first 8,000,000 base pairs.
4. Once this is done we can click **Create Workflow**.

Figure 35: Workflow options.

2.2.2 Editing workflows

Now we can see that [Galaxy](#) created our workflow. Click on the **Workflow** button in the top pane (1) to get to the workflow overview page (see *Figure 36*).

Figure 36: Accessing a workflow.

On the workflow overview page click on the workflow and on the **Edit** option (1 in *Figure 37*).

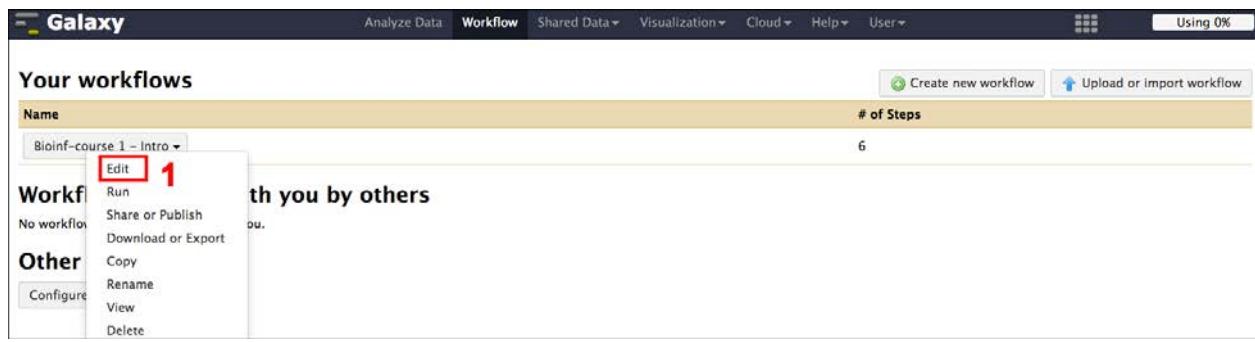


Figure 37: Editing a workflow.

The next window will show you the workflow editor. You will see two areas that are of importance: (1) is the graphical representation of our workflow in form of a flow-diagram, and (2) is the area where we can see/change attributes of individual steps (see *Figure 38*).

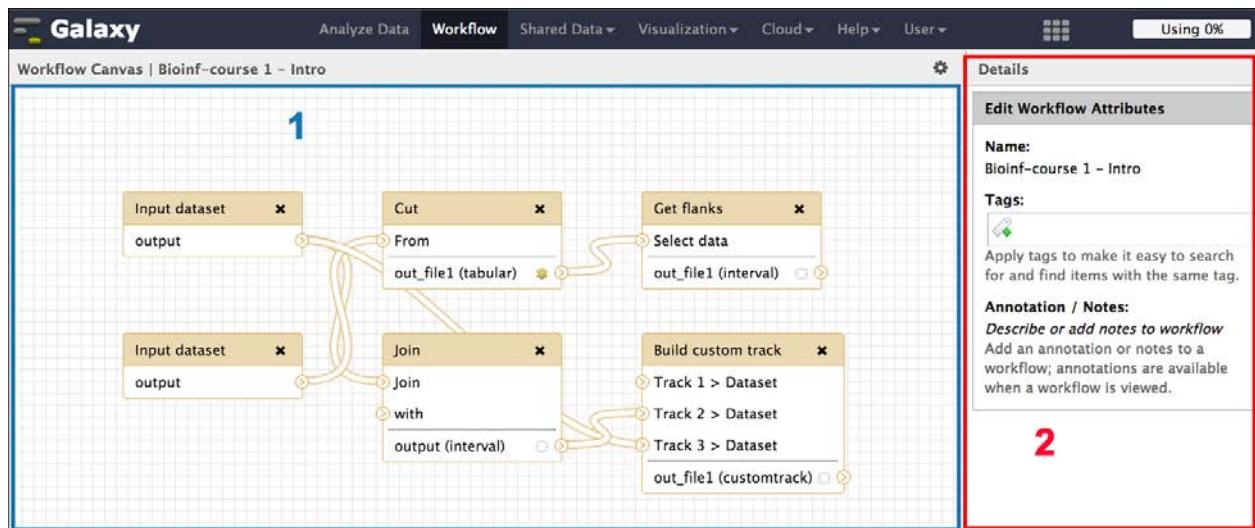


Figure 38: The workflow editor pane.

In the next picture I pulled apart the two input data fields to disentangle the view a bit (see *Figure 39*). We recognise that our workflow is a bit messed up and we need to fix it, e.g. the two input datasets are not connected at the **Join** tool.

In 1 we find the **knownGenes** input dataset as we remember it needs to be **cut** and we need to extract flanking regions for the genes (**Get flanks**). The first thing to do is to rename this dataset to **knownGenes UCSC** (2), so that we later know what this dataset is. We realise that the results of the flanking regions from (3) (**out_file1 (interval)**) is not joined to the SNP data in (4) (see *Figure 39*).

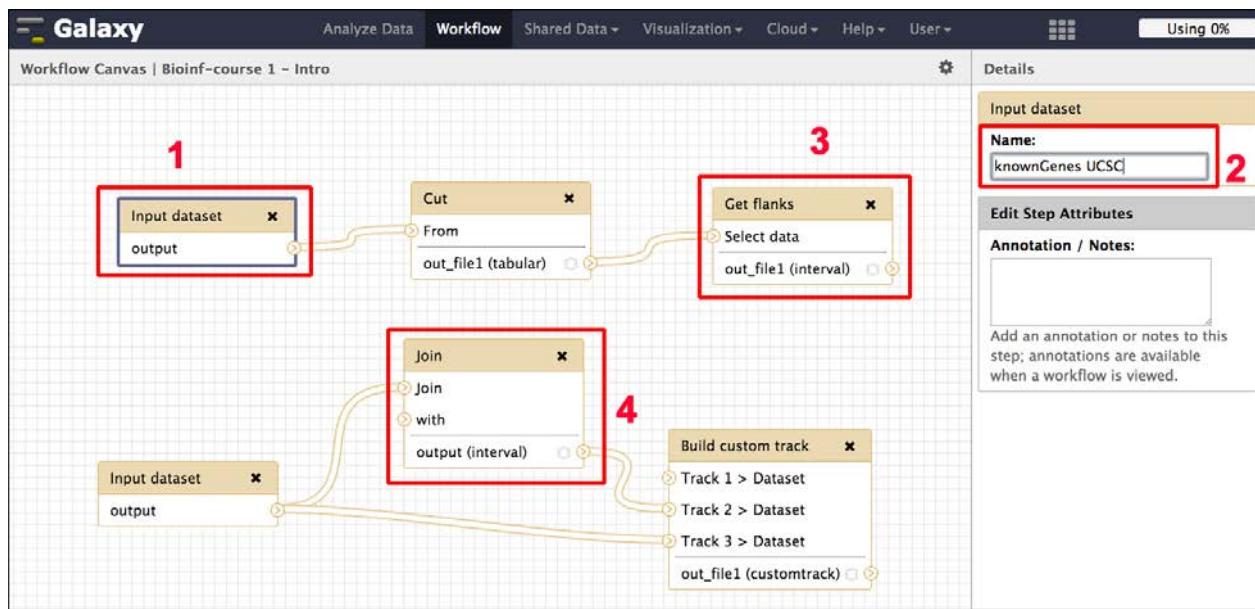


Figure 39: Individual workflow steps.

We connect the output of **Get flanks (out_file1 (interval))** (1) to the input of the **Join** tool (2) (see Figure 40).

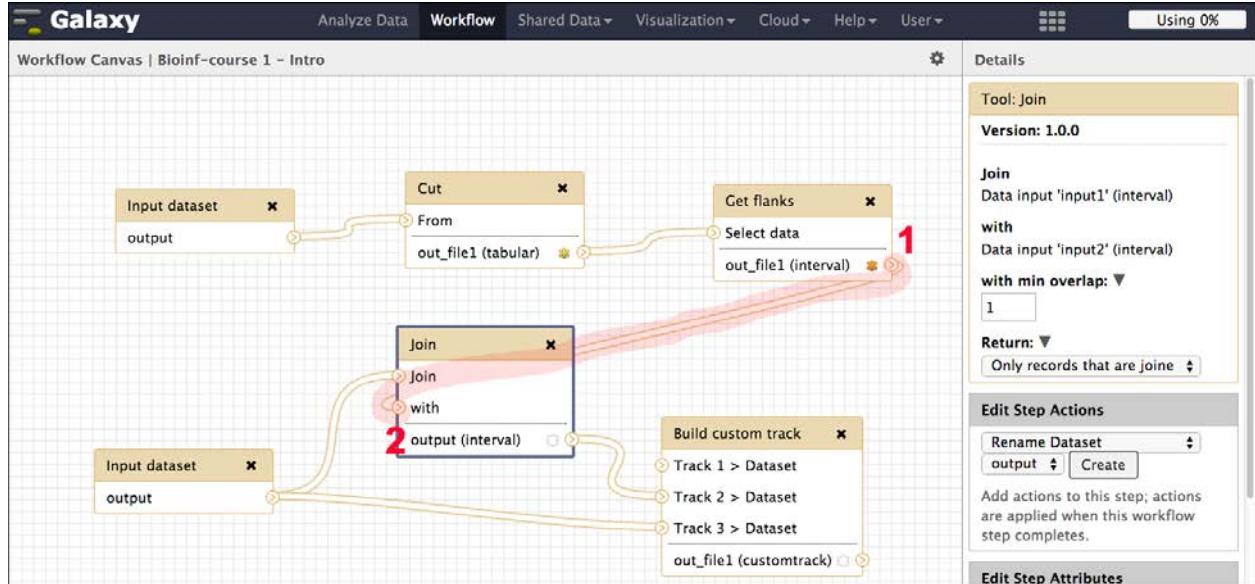


Figure 40: Connecting workflow steps 1.

We also want to show our promoters in the output UCSC track that we create as a result, but it is not connected to it either. We fix that by dragging a connector file from the output dataset of the **Get flanks** step (1) to the **Build custom track** input (2) (see Figure 41).

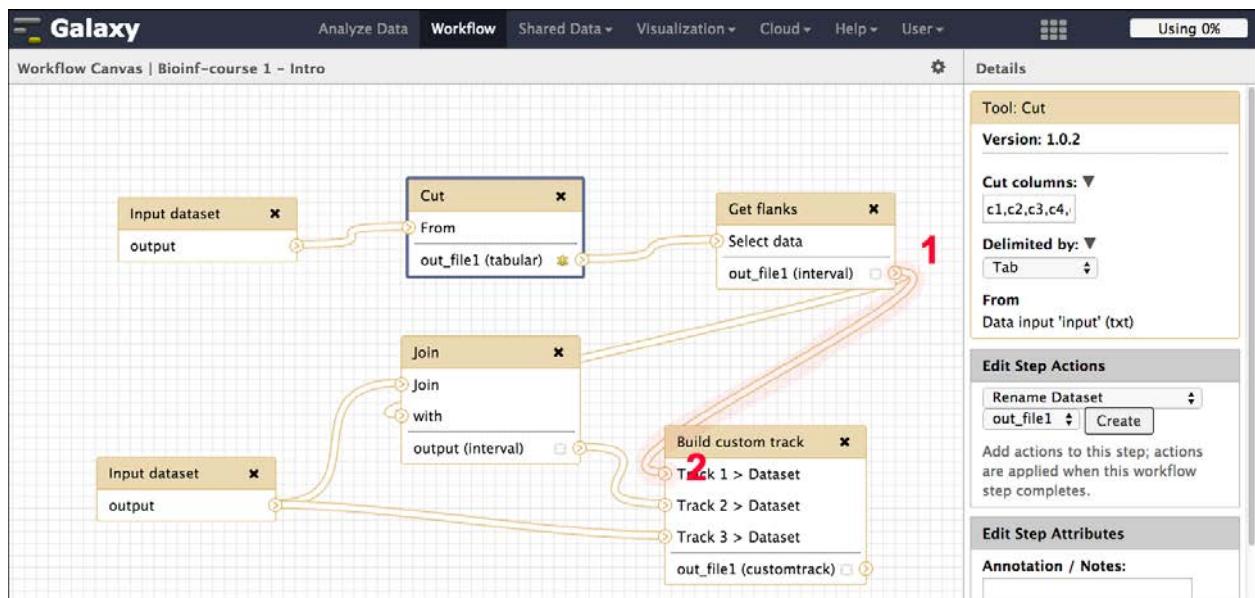


Figure 41: Connecting workflow steps 2.

Next we rename the second input dataset into the workflow in (1) to SNPs in the Details pane (2) (see Figure 42).

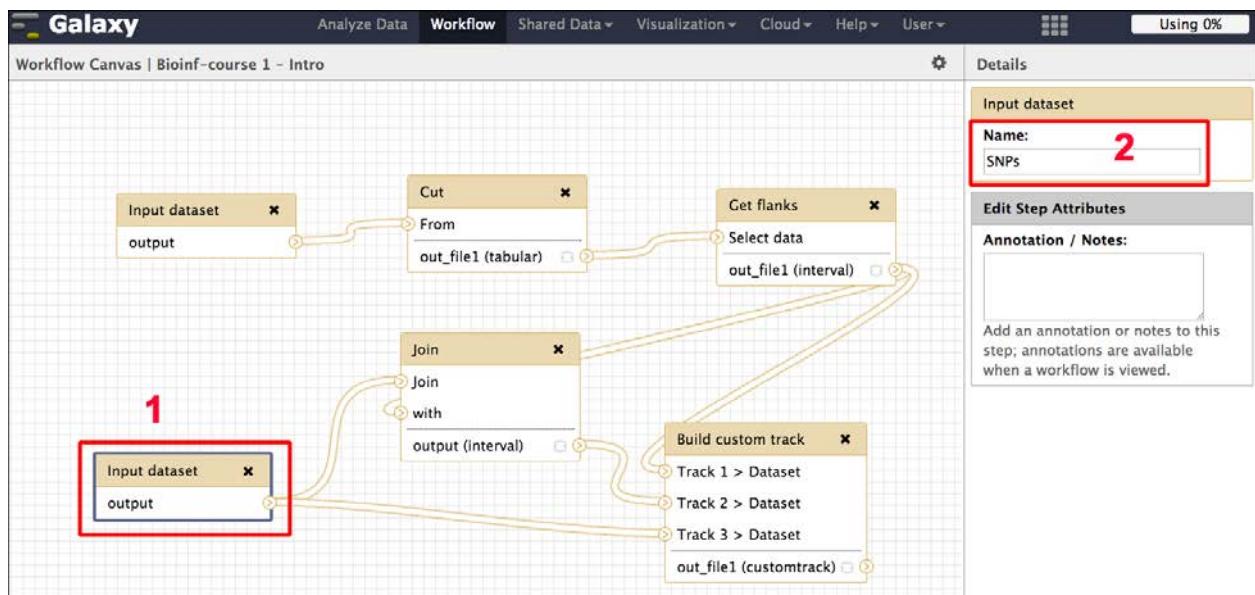


Figure 42: Renaming workflow steps.

Finally, we save the workflow (1) (see Figure 43).

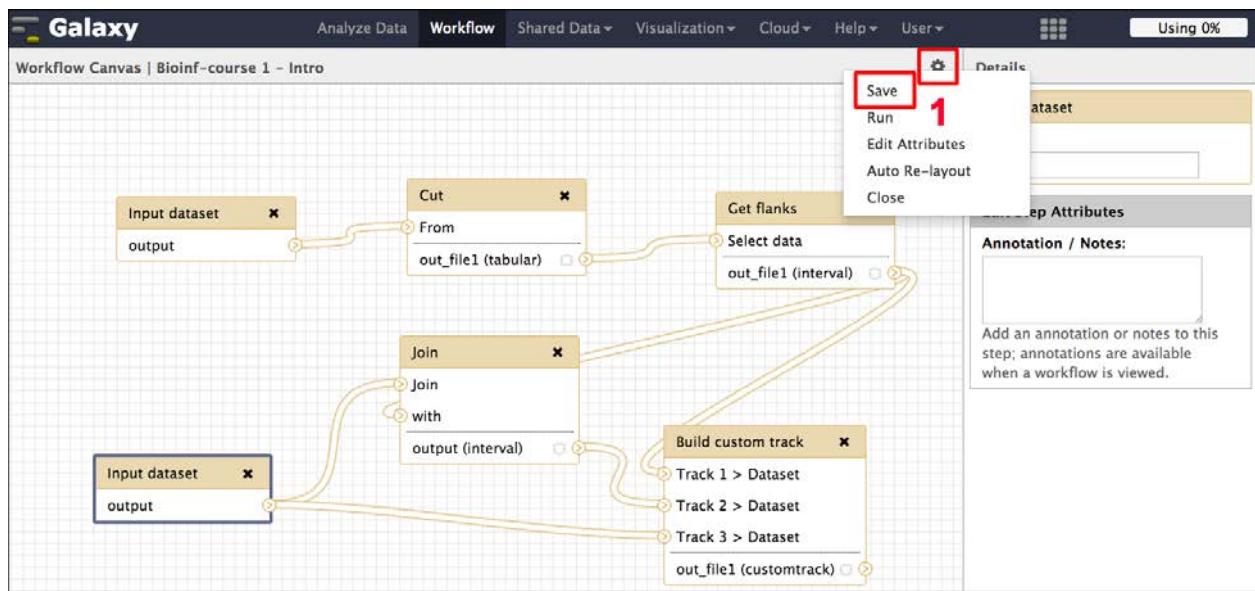


Figure 43: Saving the workflow.

2.2.3 Applying workflows to your data

Now that we have the workflow let's run it. First go to the **Workflow** panel and select the workflow and hit **Run** (1) (see Figure 44).

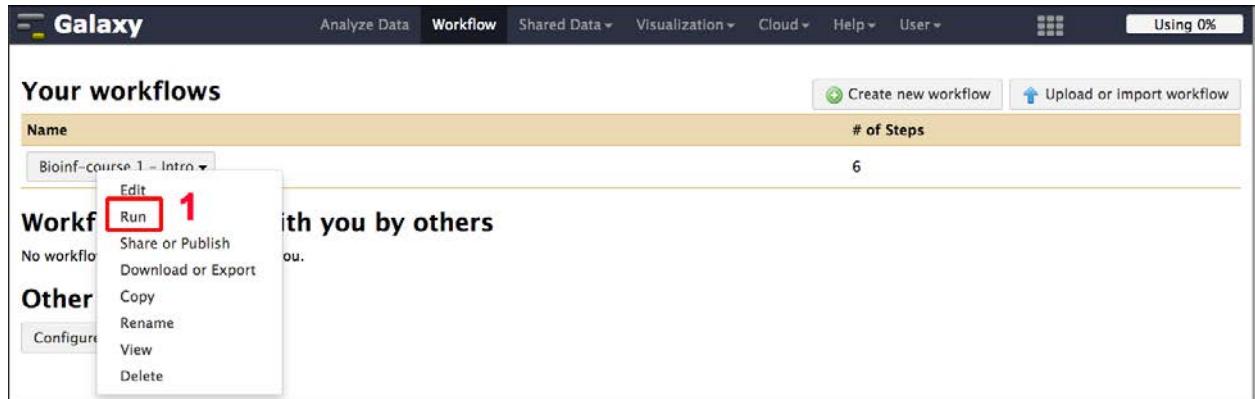


Figure 44: Executing a workflow.

Now we see the workflow and we can expand each section by clicking on the headers (see Figure 45). We choose an appropriate dataset for the **knownGenes UCSC** (1) and the **SNPs** (2). We can see in that the dataset of **Step 1** (knownGenes) is used in **Step 3** and that the output from **Step 3** is used in **Step 4**, exactly what we want (3). We also see that we join the results from **Step 4** with our **SNPs** input dataset from **Step 2** (4). Just specify your geneset and SNPs and click the **Run workflow** button (see Figure 45).

Running workflow "Bioinf-course 1 - Intro"

Step 1: Input dataset

knownGenes UCSC
18: mm9_chrX_promoter
type to filter

Step 2: Input dataset

SNPs
10: mm9_chrX_SNP128
type to filter

Step 3: Cut (version 1.0.2)

Cut columns
c1,c2,c3,c4,c5,c6
Delimited by
Tab
From
Output dataset 'output' from step 1

Step 4: Get flanks (version 1.0.0)

Select data
Output dataset 'out_file1' from step 3
Region
Around Start
Location of the flanking region/s
Upstream
Offset
0
Length of the flanking region(s)
5000

Step 5: Join (version 1.0.0)

Join
Output dataset 'output' from step 2
with
Output dataset 'out_file1' from step 4
with min overlap
1
Return
Only records that are joined (INNER JOIN)

Step 6: Build custom track (version 1.0.0)

Send results to a new history named: Bioinf-course 2

Run workflow

History

search datasets

Bioinf-course 1
8 shown, 14 deleted
204.5 MB

22: Build custom track on data 10, data 21, and data 19
21: SNPs at_promoter
19: mm9_chrX_promoter
18: mm9_chrX_promoter
14: mm9_knownGene_chrX_short
13: mm9_ChIP_chr19_control
11: mm9_knownGene_chrX
10: mm9_chrX_SNP128

Figure 45: Workflow running options.

2.3 References

Blankenberg D, Von Kuster G, Coraor N, Ananda G, Lazarus R, Mangan M, Nekrutenko A, Taylor J. **Galaxy: a web-based genome analysis tool for experimentalists.** *Current Protocols in Molecular Biology. 2010 Jan; Chapter 19:Unit 19.10.1-21.*

Giardine B, Riemer C, Hardison RC, Burhans R, Elnitski L, Shah P, Zhang Y, Blankenberg D, Albert I, Taylor J, Miller W, Kent WJ, Nekrutenko A. **Galaxy: a platform for interactive large-scale genome analysis.** *Genome Research. 2005 Oct; 15(10):1451-5.*

Goecks J, Nekrutenko A, Taylor J and The Galaxy Team. **Galaxy: a comprehensive approach for supporting accessible, reproducible, and transparent computational research in the life sciences.** *Genome Biol. 2010 Aug 25;11(8):R86.*

2.4 Web links

This tutorial: <http://sschmeier.github.io/bioinf-workshop/galaxy-intro/>

Galaxy: <http://galaxyproject.org/>

Galaxy Wiki: <http://wiki.galaxyproject.org/>

Galaxy mailing lists: <http://wiki.galaxyproject.org/MailingLists>

Galaxy learning material: <https://wiki.galaxyproject.org/Learn>