5-8-2020

CliqueSNV: GUI and Deployment on Galaxy

Sakshitha Channadi

CLIQUESNV: GUI AND DEPLOYMENT ON GALAXY

by

SAKSHITHA CHANNADI

Under the Direction of Alex Zelikovsky, PhD

ABSTRACT

Next Generation Sequencing (NGS) is a powerful technology that has enabled sequencing of thousands to millions of DNA molecules simultaneously. NGS can be applied to any species and source of DNA including viral genomes. CliqueSNV is a novel reference-based tool for reconstruction of viral variants from NGS data. In this thesis, we present a graphical user interface for CliqueSNV installed on Galaxy and as a standalone application such that microbiologists can use the tool without having to type any command line instructions and can also run multiple experiments and include it into pipelines.

INDEX WORDS: CliqueSNV, Galaxy, Next generation sequencing, Graphical user interface, Haplotype calling, SNP calling.

CLIQUESNV: GUI AND DEPLOYMENT ON GALAXY

by

SAKSHITHA CHANNADI

A Thesis Submitted in Partial Fulfillment of the Requirements for the Degree of

Master of Science

in the College of Arts and Sciences

Georgia State University

2020

CLIQUESNV: GUI AND DEPLOYMENT ON GALAXY

by

SAKSHITHA CHANNADI

Committee Chair: Alex Zelikovsky

Committee: Pavel Skums

Robert Harrison

Electronic Version Approved:

Office of Graduate Services

College of Arts and Sciences

Georgia State University

May 2020

DEDICATION

I dedicate my dissertation work to my family and many friends. A special feeling of gratitude to my loving parents whose words of encouragement and push for tenacity ring in my ears.

ACKNOWLEDGEMENTS

I wish to record my sincere thanks to my advisor, Dr. Alex Zelikovsky for his patient guidance, insightful suggestions and support which made my graduate studies a rewarding experience.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

1. NGS: Next Generation Sequencing
2. SNP: Single Nucleotide Polymorphism
3. SNV: Single Nucleotide Variant
4. GUI: Graphical User Interface
5. VCF: Variant Call Format

1 INTRODUCTION

Next Generation Sequencing (NGS) is a powerful platform that has enabled the sequencing of thousands to millions of DNA molecules simultaneously. This powerful tool is revolutionizing fields such as personalized medicine, genetic diseases, and clinical diagnostics by offering a high throughput option with the capability to sequence multiple individuals at the same time. NGS can be applied to any species including viruses and source of DNA.

Certain RNA viruses such as human immunodeficiency virus (HIV) and hepatitis C virus (HCV) display a huge amount of genetic diversity, in this thesis we will discuss about the diversity that exits within a single individual because we are interested in intra-host genomic diversity of viruses. This diversity is relevant also from a medical point of view, for example low frequency viral variants can be involved in the development of drug resistance. CliqueSNV is a novel reference-based tool for reconstruction of viral variants from NGS data. It primarily identifies linked SNVs and constructs allele graphs using highly efficient data structures. This approach finds maximal cliques in a graph with vertices corresponding to alleles (in previously published methods, they considered vertices to represent reads) which highly improves the performance because for viruses, allele graph is much smaller than the read graph. CliqueSNV was initially developed as a command line tool. In this thesis, we present a graphical user interface for CliqueSNV on Galaxy so that research scientists can easily use the tool without having to type the commands needed to run it. Galaxy can be viewed as a workflow management system used in the field of Bioinformatics. This system typically provides a graphical user interface for specifying what type of data to operate on, what are the steps needed to be taken, and in what order to do them.

1.1 Next Generation Sequencing

Analyzing DNA sequences have aided in several scientific fields since the 1900's from Sanger Sequencing to Applied Biosystems to the Genome Analyzer, scientists clearly have an insatiable hunger for finding the next best technology to accurately sequence large portions of DNA. Next Generation Sequencing also known as NGS has fundamentally changed the scientific world. It is important to know that the costs regarding next generation sequencing are significantly lower than in any other point in DNA sequencing history. Illumina is a largely successful biotechnology company that plays a large role in next generation sequencing. There are four steps that can briefly explain the sophisticated process regarding this version of DNA sequencing.

- Library Preparation
- Cluster Generation
- Sequencing
- Data Analysis

Using next generation sequencing, millions of DNA sequences can quickly be processed at astonishingly low rates. Illumina is a largely successful biotechnology company that plays a large role in next generation sequencing. Below are three different types of illumina sequencers,

- Illumina MiSeq
- Illumina HiSeq
- Illumina NovaSeq

	MiSeq	HiSeq	NovaSeq	Sanger
Reads (millions)	30	3,000	13,000	0.0004
Gigabases/day	7	500	4000	0.001

Table 1: Comparison of the output of Illumina sequencers with Sanger sequencing platforms

1.2 Intra-Host Viral Population

Some RNA viruses such as human immunodeficiency virus (HIV) and hepatitis C virus (HCV) display a huge amount of genetic diversity, within each individual patient there is a whole set of viruses that are genetically related but different from each other and if we compare virus populations between patients they display even more heterogeneity on the genomic level. So, this diversity is a result of evolutionary dynamics of viruses. RNA viruses have very high mutation rates, they tend to have short generation times and they typically exist in very large population sizes. These parameters generate a huge amount of genomic diversity in relatively short time scale so there is a lot diversity both between different infected hosts called inter-host diversity and also within each infected individual called intra-host genomic diversity. In this section, we will discuss about the diversity that exits within a single individual because we are interested in intra-host genomic diversity of viruses. This diversity is relevant also from a medical point of view, for example low frequency viral variants can be involved in the development of drug resistance. The composition of different variants changes over time and at the time of treatment failure, the traditional approach of analyzing the virus population was based on sanger sequencing which only provides the average of the virus population so the consensus sequence of the viruses, it is not able to resolve variants of lower frequency. But, next generation sequencing (NGS) can be performed at very deep coverage and then it allows us to quantify viral variants that pre-existed in the virus population also at lower frequencies. So, this is a way to increase the sensitivity of viral diagnostics. Using NGS for viral sequencing and viral diagnostics comes with a number of opportunities but also challenges because NGS technologies on one hand allow for assessing the genetic diversity of intra-host virus populations but on the other hand, this task is complicated by the fact that this sequencing reads tend to be much shorter than the genomic interval we are

interested in and also these reads are typically error prone so both the amplification process as well as the sequencing step itself introduces errors. These technical errors need to be separated from the true biological variation in the sample that we are interested in.

1.3 Calling Haplotypes and SNPs

Before discussing various existing tools that can be used for calling haplotypes and SNPs, let us get a brief overview of what is haplotyping and SNP calling.

Haplotyping of heterogeneous viral populations refers to the reconstruction of the full-length genomic variants and their frequencies. NGS technologies now provide versatile opportunities to study viral quasispecies and allows multiple coverage of highly variable viral genomic regions. This high coverage is essential for capturing rare variants, but haplotyping of heterogeneous viral populations is a complicated method because of various factors. These factors include the need to identify and preserve low frequency variants, large number of sequencing reads and the need to assemble an unknown number of closely related viral sequences. In recent years, a number of tools have been proposed for haplotyping of heterogeneous viral populations such as PredictHaplo, Savage, SHORAH, HaploClique, etc., these tools are highly reliable for many applications but accurate viral haplotyping still remains a challenge.

SNP calling also known as variant calling refers to the determination of positions where there are polymorphisms or positions where at least one of the bases varies from a reference sequence. The reason we need algorithms to do variant calling is because sequencers produce a lot of data that's really noisy and when we observe the data closely, we can see that there seem to be a lot of different mutations but we don't know whether it's really missing artifact or if it's a real variant. Various tools have been proposed to do variant calling such as VirVarSeq, V-phaser,

MinVar, etc. Even though these tool exploited linkage of nucleotide variants, they did not take into account sequencing errors when deciding whether two variants are linked.

CliqueSNV however, uses linkage between single nucleotide variations (SNVs) to distinguish them from sequencing errors efficiently. It constructs an allele graph with edges connecting linked SNVs and identifies true viral variants by merging cliques of that graph using combinatorial optimization techniques. It is a novel reference based method which reconstructs closely related low-frequency intra-host viral variants from noisy next-generation and third-generation sequencing data thereby eliminating the need for preliminary error correction and assembly and infers haplotypes from patterns in distributions of SNVs in sequencing reads which makes it suitable for both long single-molecule reads (PacBio) as well as short paired reads (Illumina).

1.4 Problem Formulation

Given: CliqueSNV software tool

Develop:

1. GUI for Standalone Application
2. User Interface for CliqueSNV on Galaxy
3. Run tests on different samples with various parameters

1.5 Contributions

Our contributions include the following,

- Standalone application for CliqueSNV software tool using Eclipse IDE.

- Deployment of CliqueSNV on Galaxy, an open-source workflow management system used for analysis of sequencing data.
- Validation of the developed user interfaces on the real NGS data samples.

1.6 Roadmap

This thesis is organized as follows: A literature about existing haplotyping and variant calling tools is discussed in Chapter 2. Chapter 3 gives the detailed description of development of the GUI as a standalone application. Chapter 4 discusses the deployment of CliqueSNV on Galaxy. Finally, Chapter 5 draws the conclusion.

2 EXISTING TOOLS

In this chapter, we will discuss about few state-of-the-art haplotyping methods and how they are implemented.

2.1 PredictHaplo

In recent years, a number of computational tools for inference of viral quasispecies populations from “noisy” NGS data have been proposed. PredictHaplo is one such software tool which aims at the reconstruction of haplotypes from next generation sequencing data, a typical application example is to reconstruct HIV-1 haplotypes present in a blood sample from a patient based on a set of reads. It was specifically designed for identifying haplotypes in an HIV-1 population and it reformulates the original problem in terms of a non-standard clustering problem, where reads are points in some metric space and haplotypes are clusters. It uses probabilistic clustering to solve the global haplotype reconstruction problem. It takes a fastq file as input and generates set of haplotypes as output. Even though PredictHaplo has been proved as a useful algorithm in many applications, it still faces a challenge of accurate and scalable viral haplotyping. PredictHaplo artificially mutated ten haplotypes from a single HIV-1 reference genome at varying proportions and often the pairwise divergence between the strains used to represent “real” HIV-1 haplotype diversity was unreported in its studies.

When CliqueSNV was validated on simulated and experimental data and was compared with PredictHaplo, it significantly outperformed PredictHaplo in both accuracy and running time. Moreover, installation and usage of PredictHaplo is challenging because of the lack of description and instructions regarding the config file while CliqueSNV is easier to install and use.

2.2 MinVar

MinVar is a virus genotyping tool which uses deep sequencing data to find mutations discussing drug resistance in HIV-1 and HCV populations. It was developed as a command line python tool. It takes a fastq file as input and generates a vcf file as output. It relies on other state-of-the-art bioinformatics algorithms for different stages of its pipeline, such as

- Bwa - for read alignment.
- GATK - for base quality recalibration.
- LoFreq - a fast and sensitive variant caller capable of calling both SNPs and indels (insertions and deletions in the genome).

In the current study of MinVar, it was compared to an existing analysis tool, VirVarSeq, specifically developed for Illumina data and evaluated the performance of MinVar using both Sanger sequencing and two different deep sequencing platforms. The discussion in the paper highlights that although deep sequencing is in principle able to detect mutations at frequencies around 1% and lower, considering the error rates of the RT-PCR, 5% should be considered as conservative but reliable threshold for the detection of drug resistance mutations in clinical practice. They base this estimation on the variant frequencies in unmixed samples and the drop in precision/recall below 5%. Even with a 5% threshold, deep sequencing is considerably more sensitive than the currently applied methods in routine diagnostics laboratories. Although lower frequency drug resistance mutations are certainly present, their clinical relevance is currently not known and they are therefore not considered for guiding treatment adaptations by major algorithms. MinVar was compared to VirVarSeq, another pipeline to call low frequency variants at the codon level developed specifically for Illumina data. MinVar employs a more conservative approach in calling low frequency variants that results in better precision at the cost of a slight

decrease in recall. It should be emphasized that VirVarSeq had to be run on our Illumina data without the last step of its workflow. The impossibility to run this step on Miseq data has been reported to VirVarSeq authors by other groups and is probably due to the presence of gaps in the distribution of quality scores. In order to show how MinVar can be successfully applied to different sequencing technologies, they have also applied it to data obtained with Roche 454 GS Junior. Although Roche announced the imminent phase out of the system, evaluating a pipeline also on this sequencing technology is still of interest because it challenges versatility and robustness of a tool and also because technologies still in use like Ion Torrent share some similarities with it. In fact, since Illumina Miseq and Roche 454 GS Junior use different chemistries, they offer a diverse spectrum of coverage, read length, and nature of sequencing errors. GS Junior has less but typically longer reads, and its most common errors are indels (insertions and deletions in the genome). They have demonstrated in their discussion that MinVar can be run with good results on both platforms, indicating that it can be applied successfully to other sequencing platforms too, provided that the input is in fastq format.

It is important to note that in the current study of MinVar, they have only analyzed the *pol* gene of HIV but they are yet to extend their study to genotyping other regions as well as entirely different viruses (*e.g.*, HBV, HCV). Unlike MinVar, CliqueSNV is also suitable for both long single-molecule reads (PacBio) as well as short paired reads (Illumina). Also, MinVar only accepts input in fastq format and does not provide an option to set any parameters such as minimum read support, etc., whereas CliqueSNV does provide an option to set the parameters.

2.3 CliqueSNV

CliqueSNV is a state of the art haplotyping method that aims to reconstruct viral variants from noisy next generation sequencing data and third-generation sequencing data.

It consists of the following steps:

- Finding linked SNV pairs - CliqueSNV uses pairs of linked SNVs which have been previously introduced for the 2SNV method.
- Constructing the allele graph - The allele graph $G = (V, E)$ consists of vertices corresponding to minor alleles and edges corresponding to linked pairs of minor alleles from different positions. There are no isolated vertices in G since the minor alleles are only considered if they are linked to other minor alleles. If the intra-host population consists of very similar haplotypes, then the graph G is very sparse.
- Finding maximal cliques in the allele graph - Ideally, the individual minor alleles distinguishing a viral haplotype from the consensus should all be pairwise adjacent to the allele graph $G = (V, E)$. Therefore, CliqueSNV looks for maximal cliques in G. Although the MAX CLIQUE is a well-known NP-complete problem and there may be an exponential number of maximal cliques in G, a standard Bron - Kerbosch algorithm requires little computational time since G is very sparse.
- Merging cliques in the clique graph - We first find all pairs of cliques p and q which are unlikely to come from the same haplotype and decide whether p and q are forbidden pairs or adjacent pairs.
- Finding consensus viral variants for merged cliques - Let S be the set of all positions that belong to at least one clique. Let qS be an *empty clique* corresponding to a haplotype with all major alleles in S. For each read r restricted to the positions in S, we assign r to the

closest clique q (which can be qS), i.e. clique q which differs from r in the minimum number of positions in S. In case of a tie, we assign r to all closest cliques.

For each clique q, CliqueSNV finds the consensus $v(q)$ of all restricted reads assigned to q. Then $v(q)$ is extended from S to a full-length haplotype by setting all non-S positions to major alleles.

- Expectation-maximization (EM) Algorithm for Estimating Variant Frequencies - We estimate the frequencies of the reconstructed viral variants via an EM algorithm.

The above steps explaining the algorithm are taken from CliqueSNV paper. Validation of different haplotype reconstruction methods should simultaneously answer two general questions: (i) how close are the reconstructed and true variants and (ii) how narrow is the reconstructed and true variant frequency distribution. Previous studies report high variation in results addressing these questions likely due to the challenge of simultaneously addressing them. Here we propose to use the Earth Mover's Distance (EMD) as a distance measure for populations, which generalizes edit distances between genomes of individual variants.

The ability to accurately infer the structure of intra-host viral populations makes CliqueSNV applicable for studying evolution and examining genomic compositions in RNA viruses. However, we envision that the application of our method can be extended to other highly heterogeneous genomic populations, such as metagenomes, immune repertoires, and cancer cells.

3 GUI FOR CLIQUESNV

In this chapter, we will discuss the development of a graphical user interface for CliqueSNV as a standalone application.

3.1 What is GUI?

Before discussing the development of graphical user interface for CliqueSNV, we need to first understand what exactly refers to a graphical user interface. A graphical user interface can be defined as something that makes interacting with the computer easier. Before graphical user interface came into existence, we had to write commands to perform even simple tasks on a computer like opening a folder, copying files from one folder to another, etc. Graphical user interface provides buttons, icons and many other interactive visual components which we can click on to perform the same tasks like opening a folder, copying files from one folder to another, etc., There are many different ways in which we can develop a graphical user interface based on what language we are using. In Java, there is one popular framework for developing a GUI called Swing which is pretty outdated but teaches us the basic concepts. There is one other java GUI framework called JavaFX which has also been around for a long time but only gained traction after Oracle purchased Java from Sun. A GUI includes a range of user interactive elements — which just means all the elements that are visible when you are working in an application. Some of these elements are listed as follows,

- Input controls such as buttons, dropdown lists, checkboxes, and text fields.
- Informational elements such as labels, banners, icons, or notification dialogs.
- Navigational elements, including sidebars, breadcrumbs, and menus.

3.2 Development of GUI

In this section, we will discuss about the development of a graphical user interface for CliqueSNV tool. As discussed earlier in the previous sections, CliqueSNV was initially developed as a command line tool, this thesis presents a graphical user interface for CliqueSNV. The source code for CliqueSNV was initially developed in java and it was already an executable version so the programming language used to develop the graphical user interface presented in this paper is also java. The programming platform for developing the GUI in this thesis is Eclipse IDE. Although there are different frameworks for developing a GUI in java, the one used in this thesis is java swing framework. Swing is an API in java and is a part of java foundation classes (JFC), it is used to build graphical user interfaces in java. Swing API is much more light weight compared to abstract window toolkit (AWT) which is one other way of developing graphical user interfaces in java. Various user interface elements of java swing are used to design the graphical user interface for CliqueSNV. Some of them include,

- JFrame - It acts as a starting point of any swing application and we must create it so that we can add other components like title, buttons, etc., in it. It is basically a window of our GUI application that holds other visually interactive elements and frame is an instance of JFrame. In our case, a JFrame is created with the name “Clique SNV” which is the title of our standalone GUI application.
- JButton - This is a button that any user using the application can click on and some action is performed based on what it is designed for. In our GUI, we have an input button (SAM File), an output button (Output) and an execute button (Run). The input button is set to specifically perform an action of choosing a file from users local system for which a swing UI element called JFileChooser has been used in the code, when users click on the input

button, they are prompted to choose a file from their local system as input and the file must be in sam format because CliqueSNV only takes sam files as input. The output button is set to specifically perform an action of selecting a location/path from users local system for which a swing UI element called JFileChooser has been used in the code, when users click on the output button, they are prompted to select a location/path from their local system where they would want their output file to be stored. The run button is simply used to execute the application after all the required fields are populated, it throws an alert if any of the required fields are empty.

- JRadioButton – This user interface element is used to create a radio button such that the users can only select one option from multiple options. There are two radio buttons in the GUI of CliqueSNV, one for each NGS technology. The two NGS technologies from which the users have a choice of selecting only one option are pacbio and illumina sequencing technologies.
- JCheckBox - This user interface element is used to create a checkbox which can turn its state from “on” to “off” or from “off” to “on” when a user clicks on it. There are two checkboxes in the GUI of CliqueSNV, one for each method. The purpose of using checkboxes is to give a choice to the users to execute both the methods, namely haplotyping and SNP calling either simultaneously or separately.
- JTextBoxes - This user interface element is used to create a text field that allows the editing of a single line text. There are two text boxes to define minimum read support of SNP pairs in the GUI of CliqueSNV, one is for absolute frequency and the other is for relative frequency. Users can enter their desired numeric values in these text boxes but their default values are set to 10 and 0.00033 for absolute and relative frequency respectively so even if

the users don't enter any value in these text fields, they are automatically populated with their respective default values.

Let us now discuss about the installation of CliqueSNV standalone application and the steps needed to run the application,

- Installation of CliqueSNV standalone application - The GUI for CliqueSNV standalone application presented in this thesis is developed as an executable jar file. The users can click on the jar file (CliqueSNV.jar) to download it and click on Install. To finish the installation, the users must follow the steps prompted until they finish the process of installation.
- Steps to run CliqueSNV standalone application – After installing the executable jar file of CliqueSNV, users can click on the file and follow the below steps to run it
    i. Click on "SAM File" button in the GUI window after which the users are prompted to choose a file in sam format from their local system. This is an input for CliqueSNV.
    ii. Click on "Output" button in the GUI window after which the users are prompted to choose a location/path to store their output file in their local system.
    iii. Select the type of NGS technology from the list of options provided, users can only select one option here.
    iv. Then select the method from the list of options provided, users can select multiple options here.
    v. The next step is to provide a numeric value in the two text fields provided for absolute and relative frequency respectively, note that their default values are taken if users don't enter any values.

vi. Click on “Run” to execute the application. This might take some time depending on the size of input file.

vii. Check the output file location to view the generated output files.

viii. Once the output files are generated, the same steps above can be repeated to run CliqueSNV on a different input file or the application can be closed by clicking “x” symbol on the top right corner.

3.3 Running CliqueSNV GUI for real NGS data samples

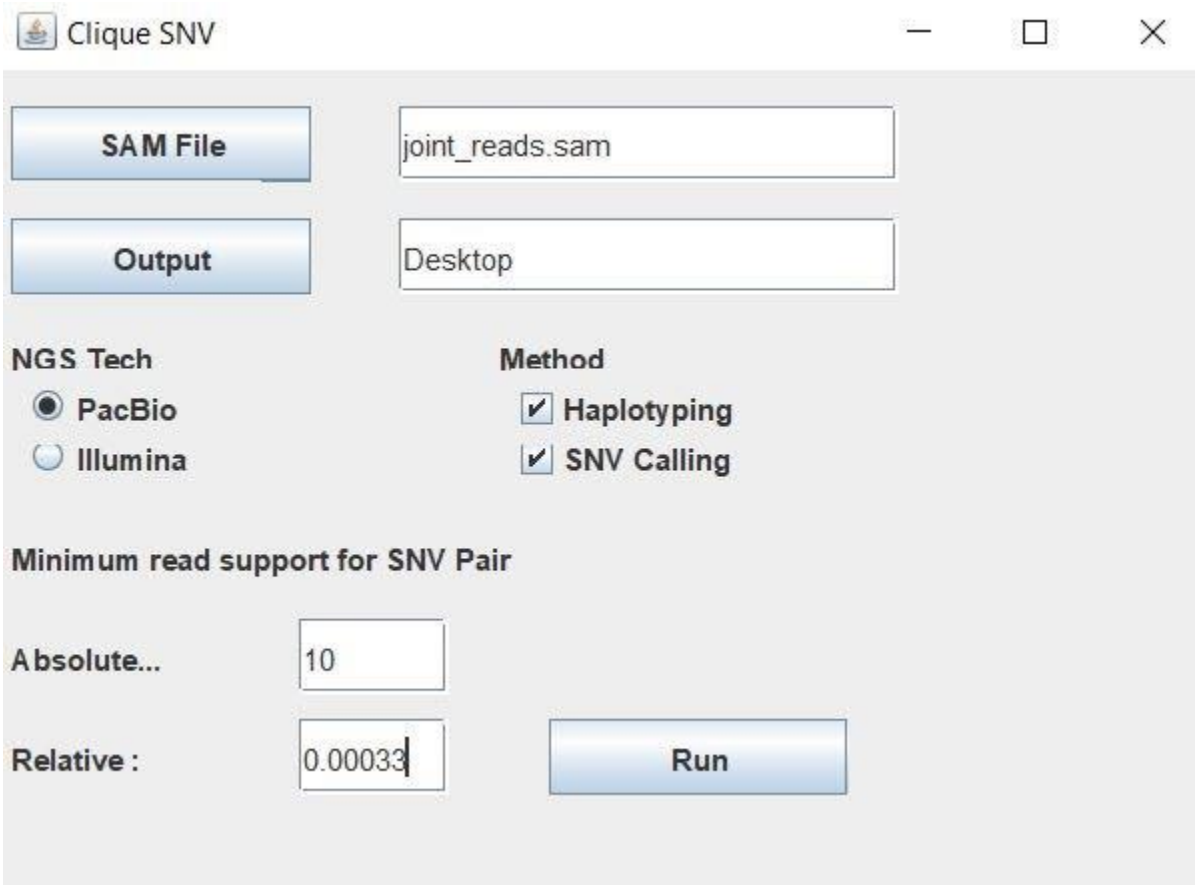

Figure 1: CliqueSNV GUI (Standalone application)

4 DEPLOYMENT OF CLIQUESNV ON GALAXY

4.1 Introduction to Galaxy

Galaxy is an open-source workflow management system used in the field of Bioinformatics. It can be defined as a one-stop platform for sequence analysis tools. This system typically provides a graphical user interface for specifying what type of data to operate on, what are the steps needed to be taken, and in what order to do them. It can also be defined as a website where we can upload our sequence and can perform any number of modifications, manipulations and analysis.

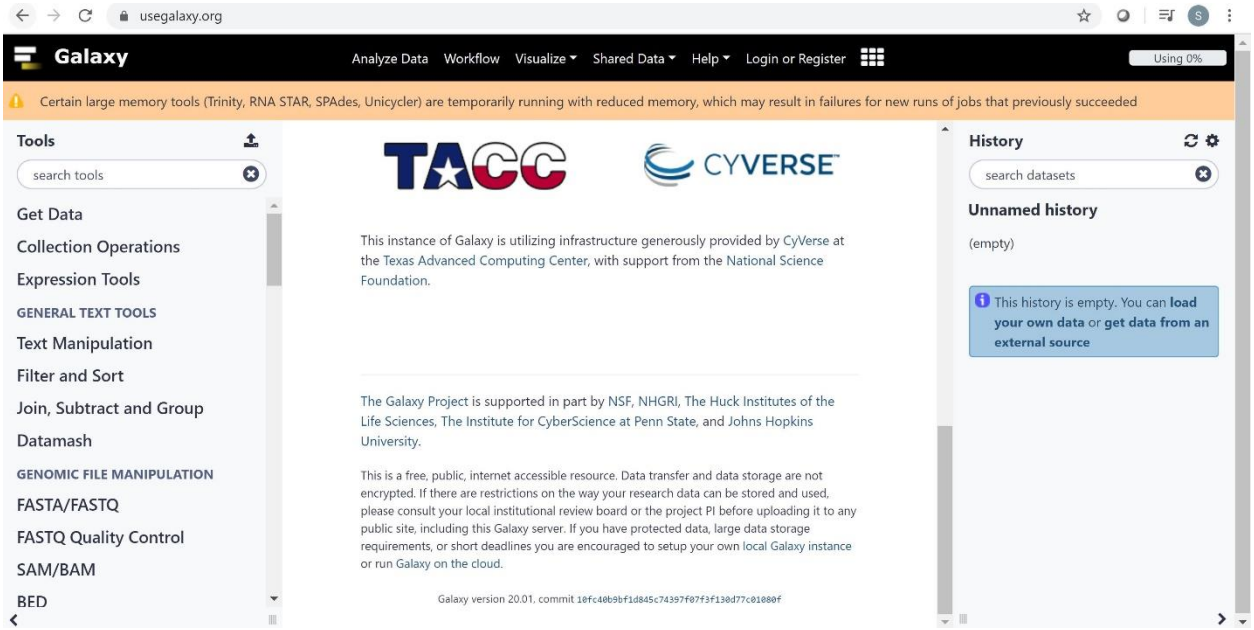

Figure 2: Galaxy website

Figure 2 is a brief overview of how a basic Galaxy website looks like, let us now understand how it works. The right pane in Galaxy is history where we can see the sequences that we have uploaded and any manipulations that we have performed in the order that they were performed. The left pane contains a list of tools that Galaxy has available starting with get data which we are going to discuss further with the help of a figure.

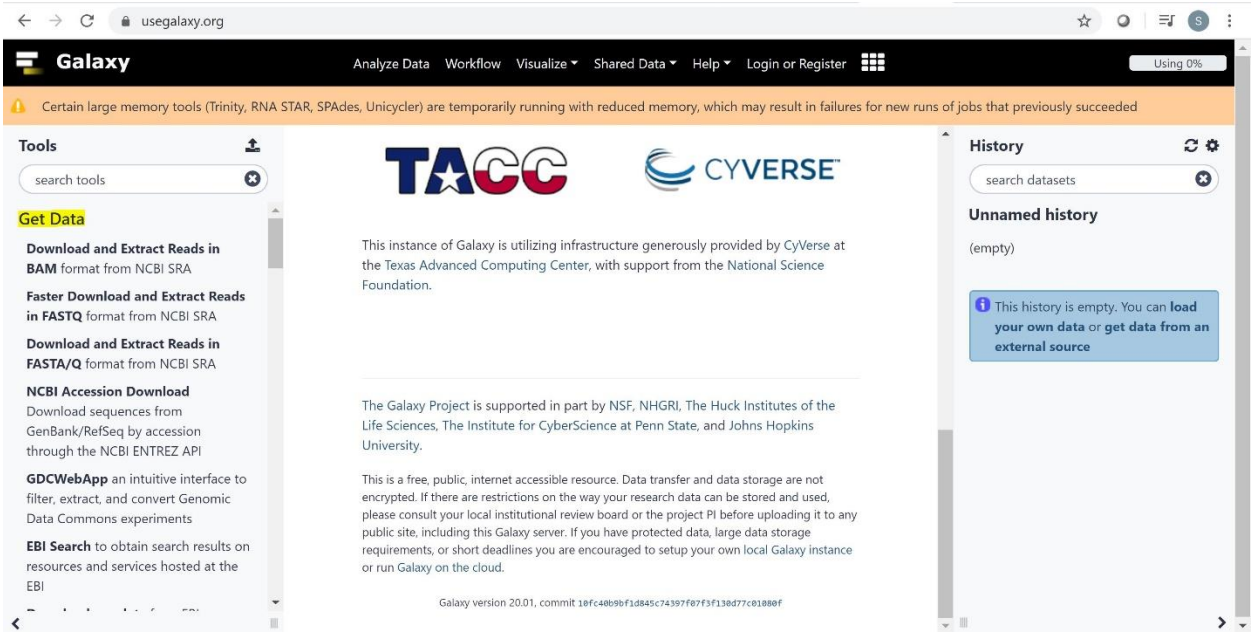

Figure 3: Get Data tool in Galaxy

If we click on get data tool (highlighted in figure 3) on the left pane, we can see a list of servers from which we can transfer data directly to Galaxy. Under get data, there is *upload file from your computer* and when we click on that, we can see that a new panel appears in the middle of the Galaxy screen (shown in figure 4). Middle of the Galaxy screen is where the tool that we are actually working with is shown. There are two options to upload files to Galaxy, the first option is *choose local file* and the second option is *paste/fetch data* ((both these options are highlighted in figure 4). When we click on the first option, it lets us to directly upload a file from our computer and when we click on the second option, it lets us to put in http or ftp URL and transfer the file directly to galaxy. We can see our uploaded files in Galaxy history on the right pane.

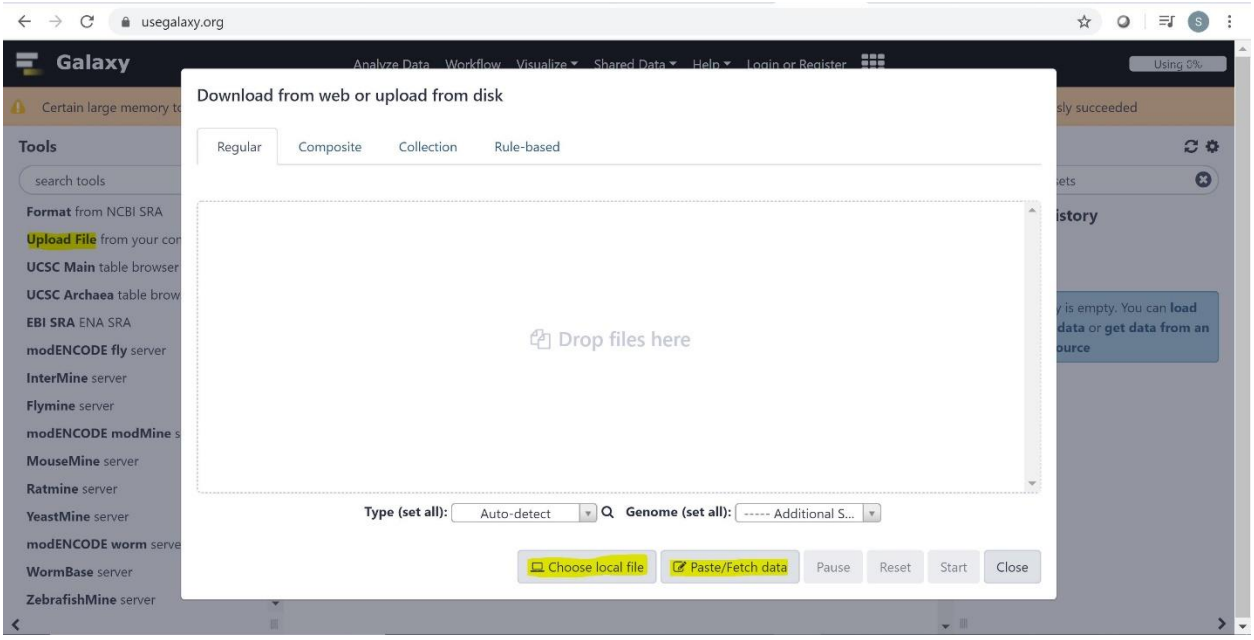

Figure 4: How to upload files on Galaxy

4.2 Steps of Deployment

In this section, we will discuss about the steps of deployment of CliqueSNV on Galaxy. Below figures explain the step by step procedure of deployment.

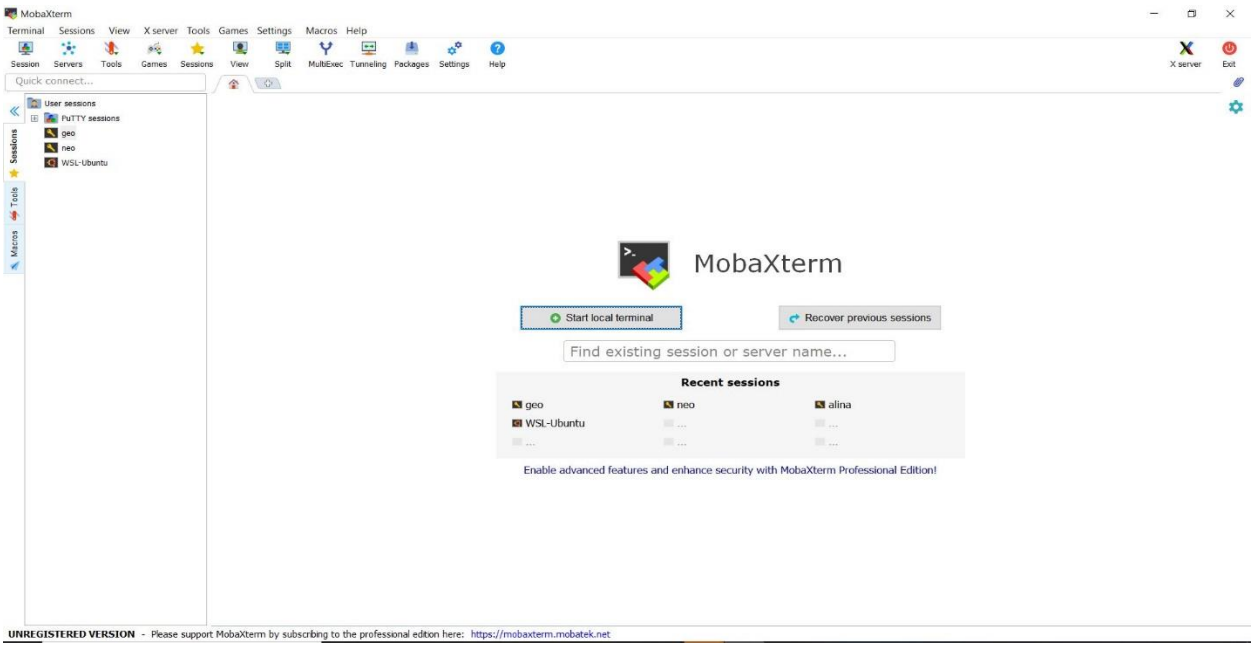

Figure 5: SSH client used

We can use any SSH client to remotely connect to the server on which Galaxy is installed, we have used MobaXterm for our purpose. Using MobaXterm, we connected remotely to https://gene.engr.uconn.edu/ server where we first installed and tested the user interface of CliqueSNF. Then we ran a command to become a galaxy user on it since we had permissions to do so. Note that all the deployment steps must be implemented only as a galaxy user.

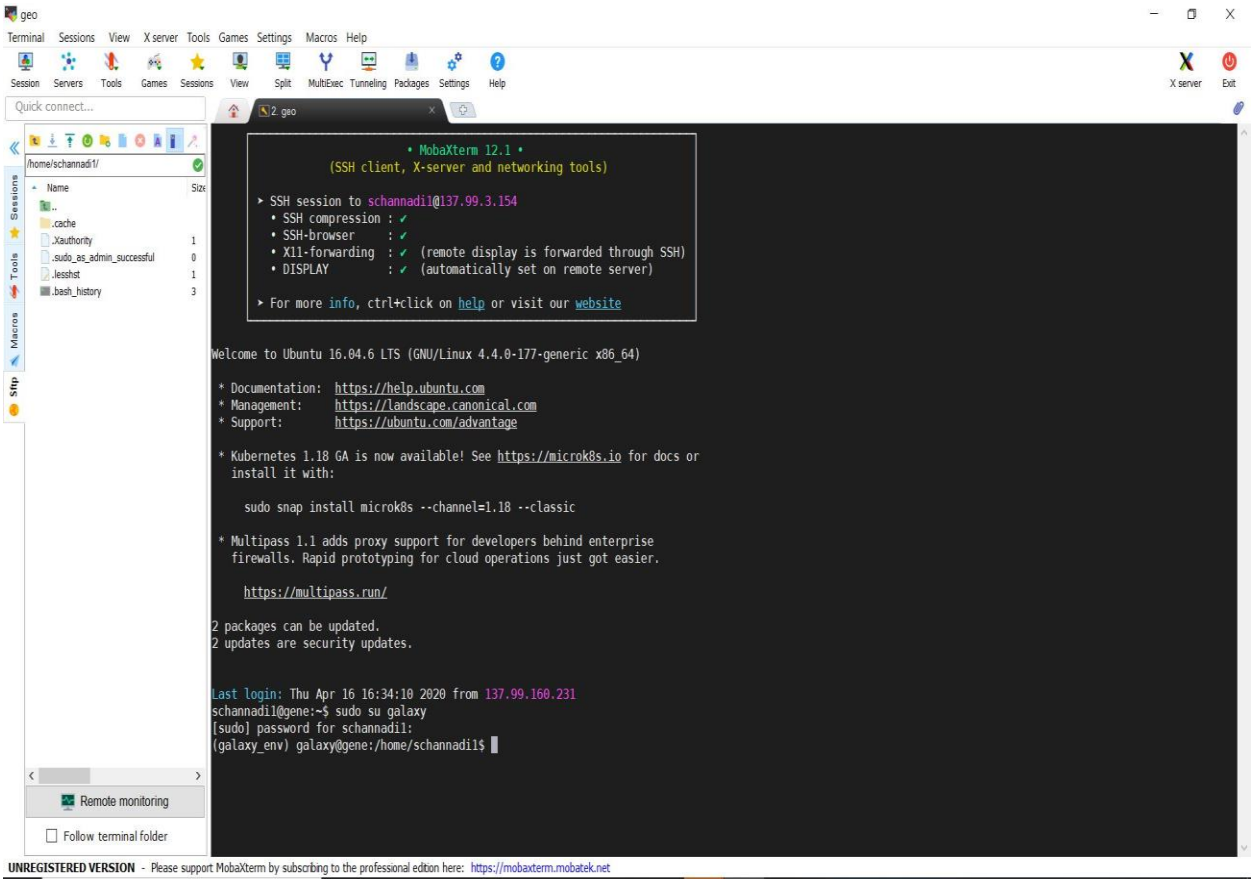

Figure 6: Commands to become galaxy user

The above figure shows the commands needed to become a galaxy user on the server that has Galaxy installed on it. User that is trying to impersonate as a galaxy user must be added to the sudo group otherwise the user will receive an error.

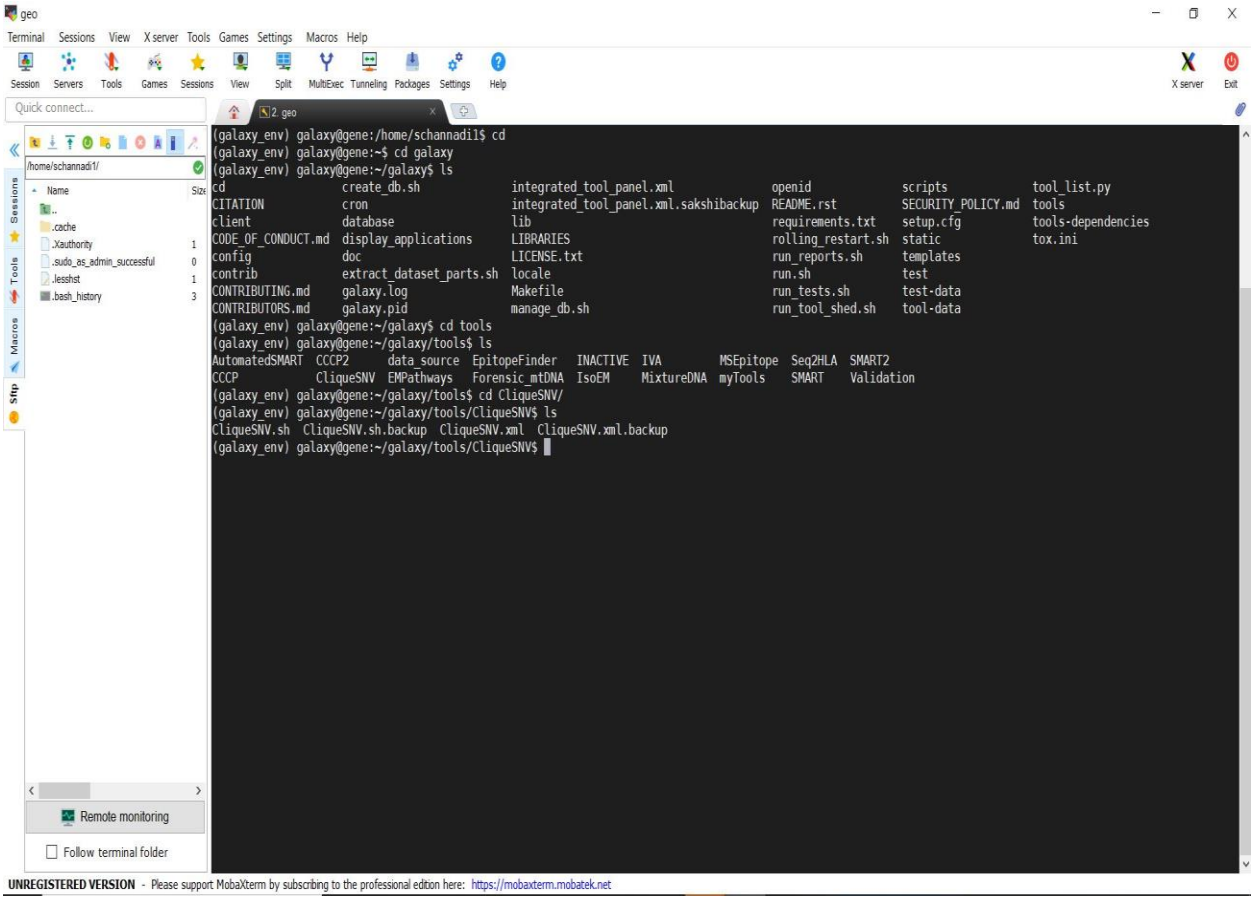

Figure 7: Directory structure that must be maintained on Galaxy to deploy new tools

There is a directory called *tools* which is present under galaxy by default. This is our working directory where we need to create a directory for our tool. From the above figure, we can see that there is a directory that is created for CliqueSNV by the same name. Under this directory we have our bash script and the xml file which are basically our source codes. All of this can be seen in the above figure and this exact directory structure must be maintained to deploy any new tools on Galaxy.

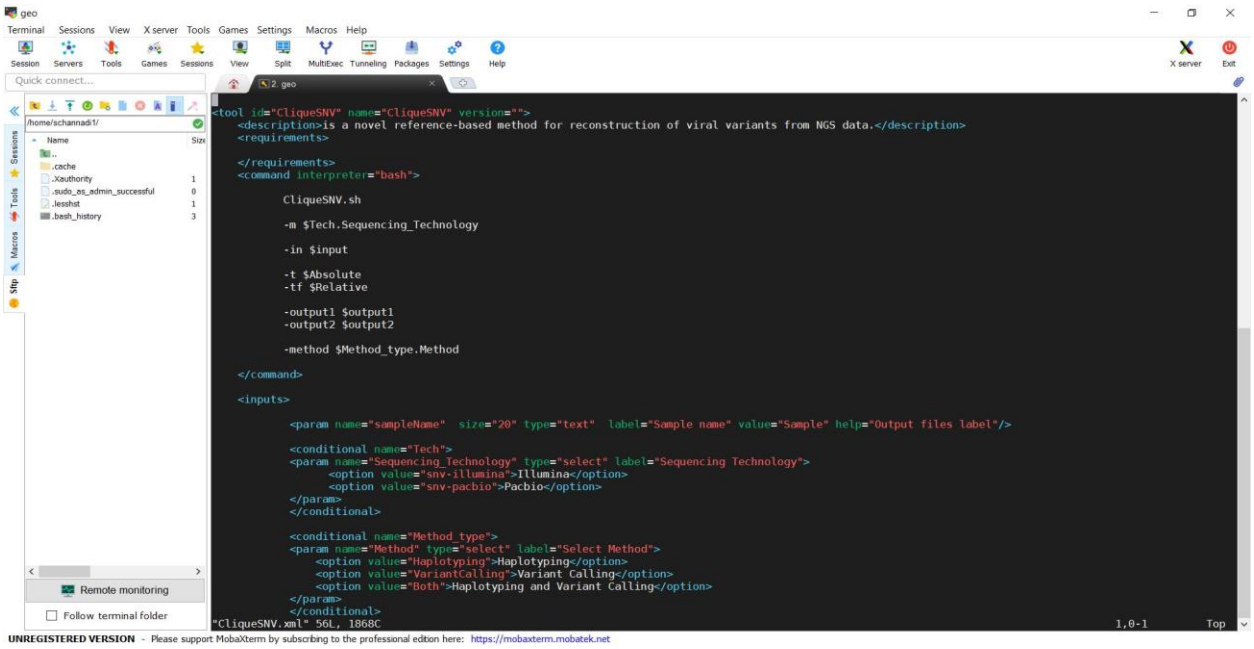

Figure 8: Sample xml file 1

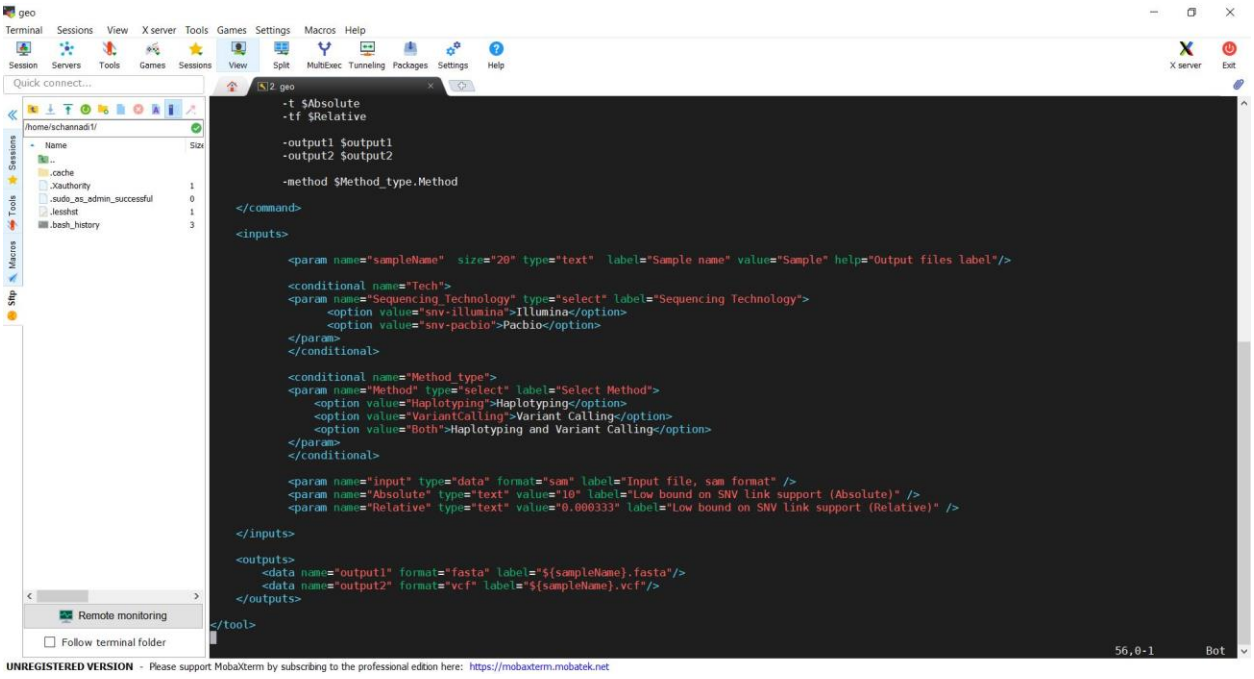

Figure 9: Sample xml file 2

Figure 8 and Figure 9 show the sample xml code that is used to develop the user interface for CliqueSNV on Galaxy.

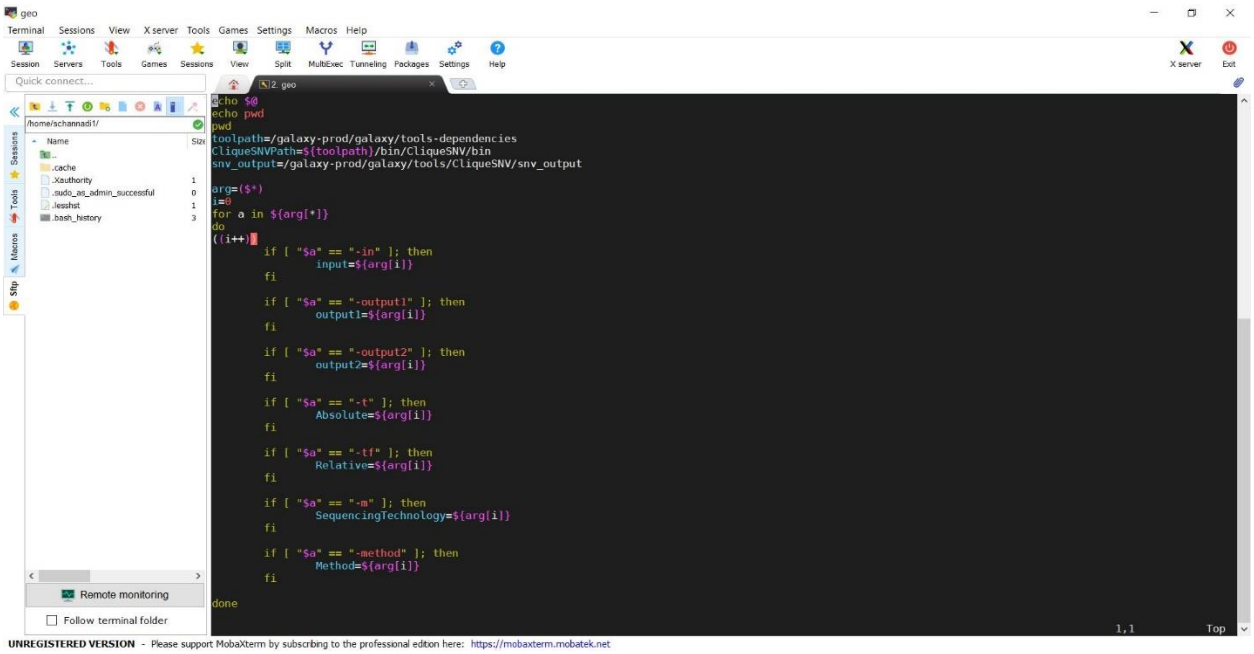

Figure 10: Sample bash script 1

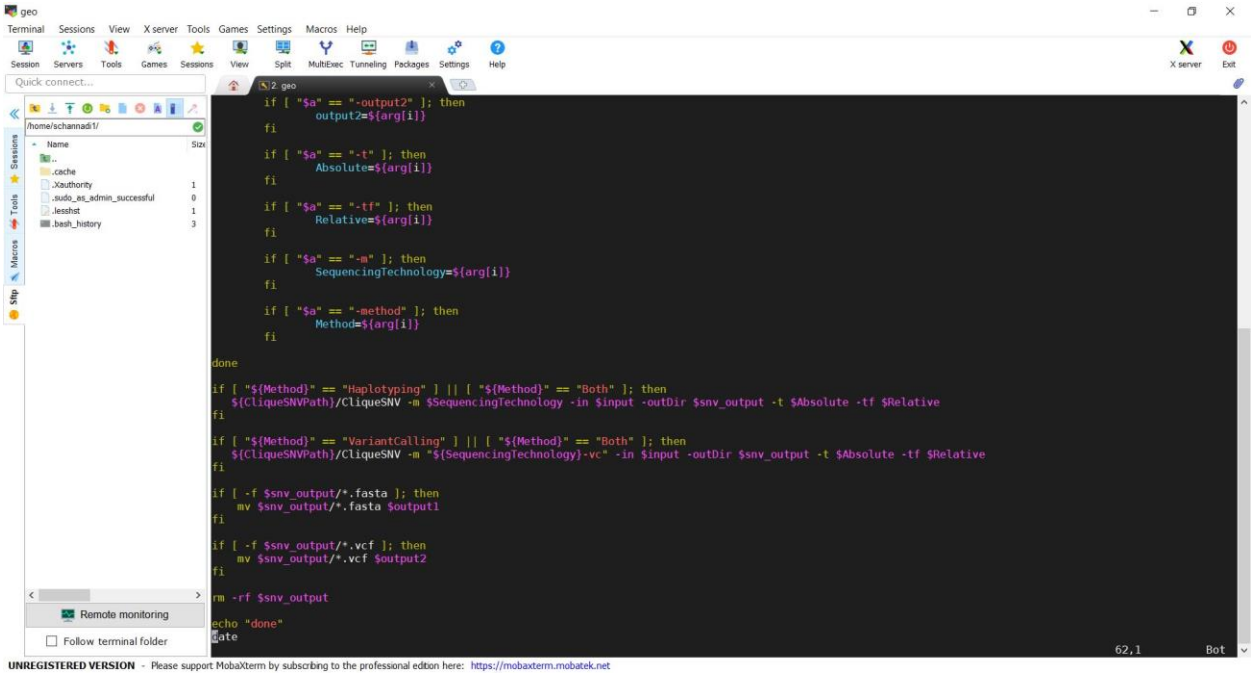

Figure 11: Sample bash script 2

Figure 10 and Figure 11 show the sample xml code that is used to develop the user interface for CliqueSNV on Galaxy.

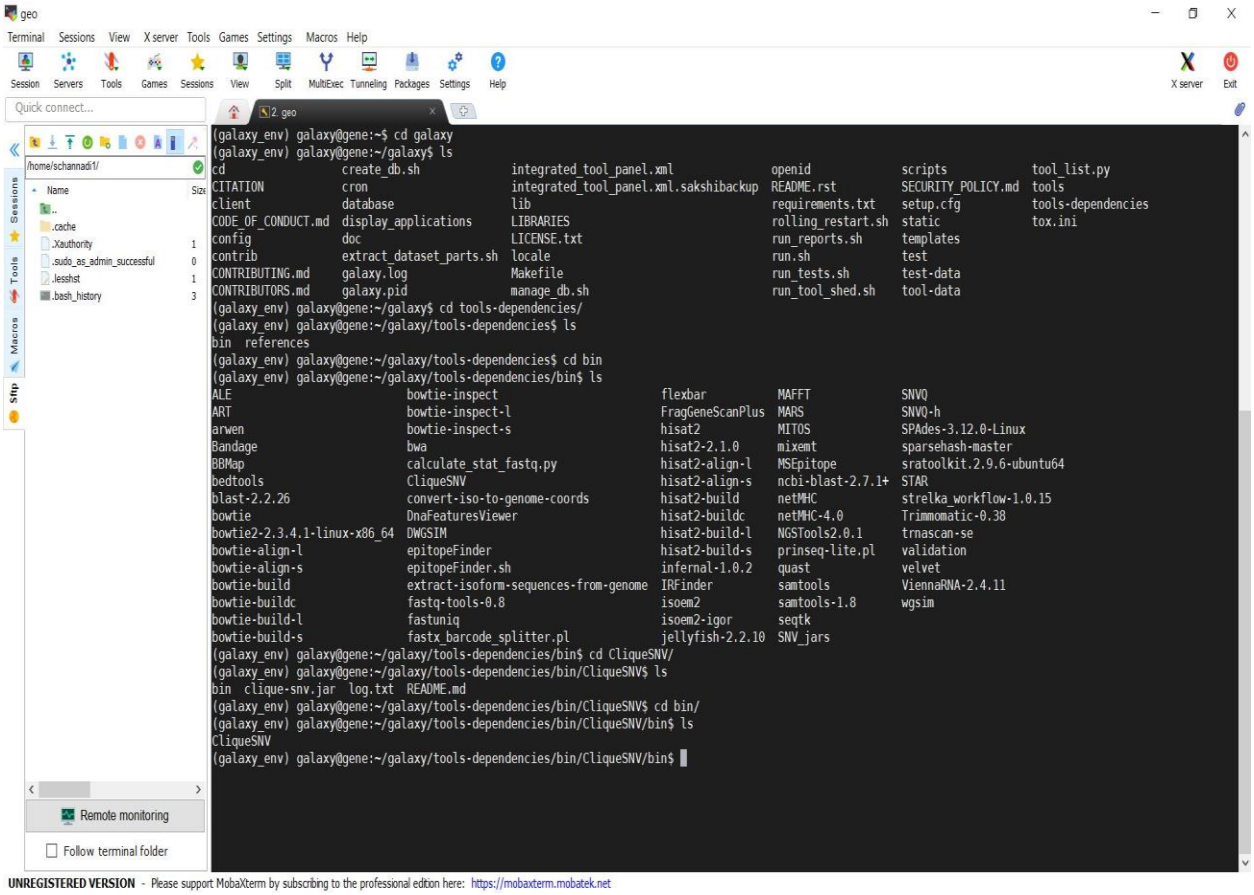

Figure 12: Directory structure to add any tool dependencies

There is a directory called *tools-dependencies/bin* which is present under galaxy by default. This is a directory under which we need to place all our dependencies like source code, etc., for our tool that we are deploying on Galaxy. We need to create a directory for our tool under this directory just like we did under *tools.* In our case we created a new directory under *tools-dependencies/bin* called *CliqueSNV* and placed our executable jar file that contains the entire source code under this new directory. All of this shown in the above figure.

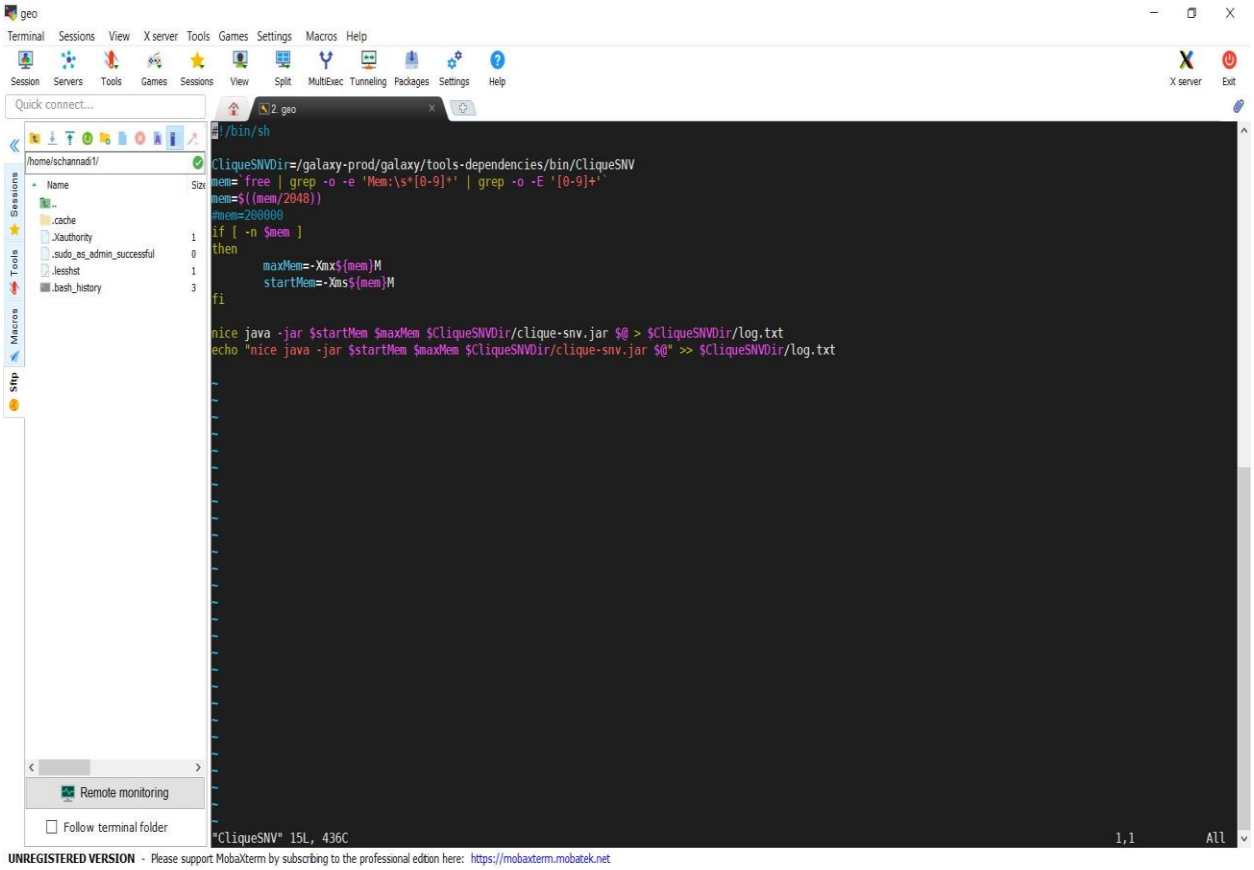

Figure 13: Sample shell script for allocating memory to CliqueSNV

We need to create a new directory called *bin* under our new directory that we just created in the above step and write a shell script under this directory for memory allocation of our tool that we are going to deploy. From figure 12, we can see that a shell script has been written under this new bin directory. The above figure shows a sample shell script for allocating memory to the tool that is being deployed. This script varies depending on the language used to develop the algorithm for the tool that is being deployed.

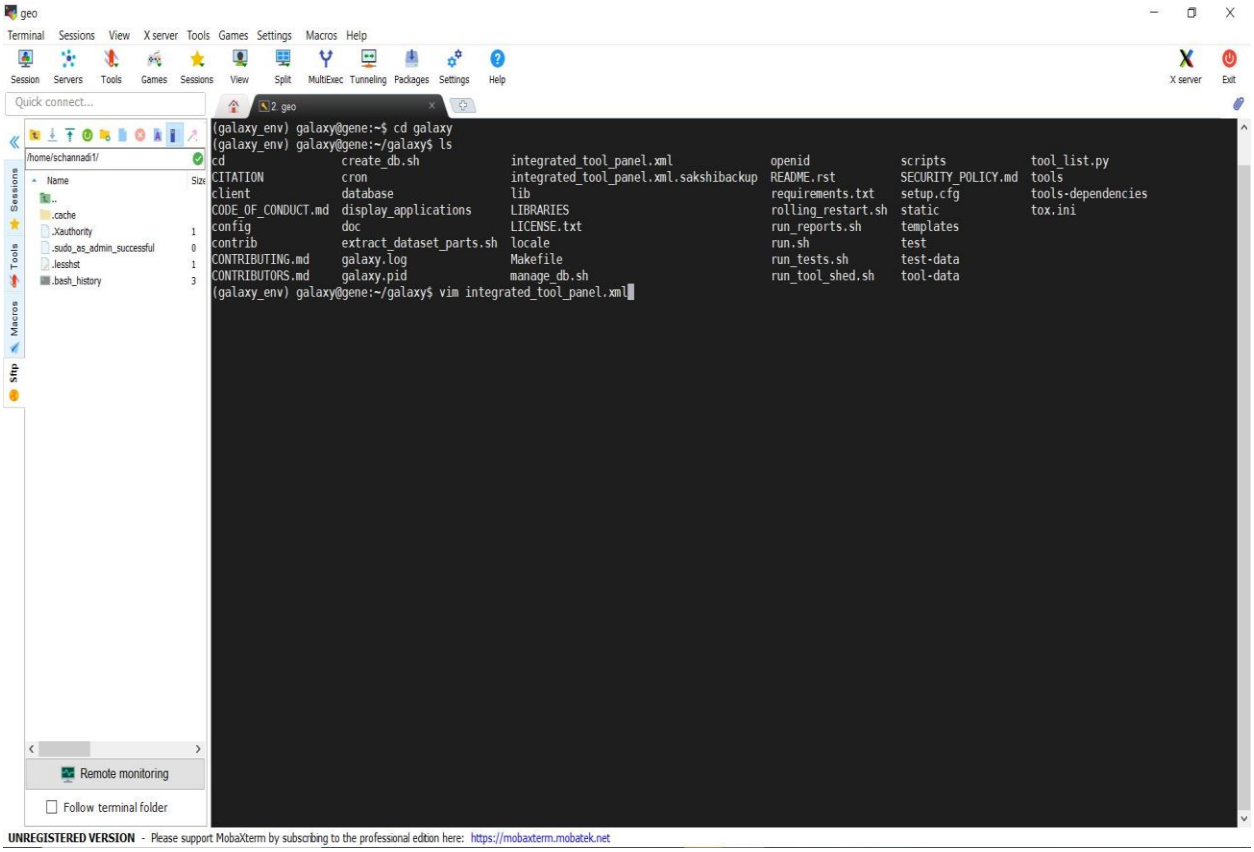

Figure 14: XML file responsible for making user interface of CliqueSNV visible on Galaxy

Once the source code has been written and the memory has been allocated for CliqueSNV, it is time to tweak the xml file that is under galaxy which contains information about all the tools that can be made visible on Galaxy web interface. The xml file with the name *integrated_tool_panel.xml* is the file that is needed to be tweaked to make our newly developed user interface for CliqueSNV visible on the website of Galaxy.

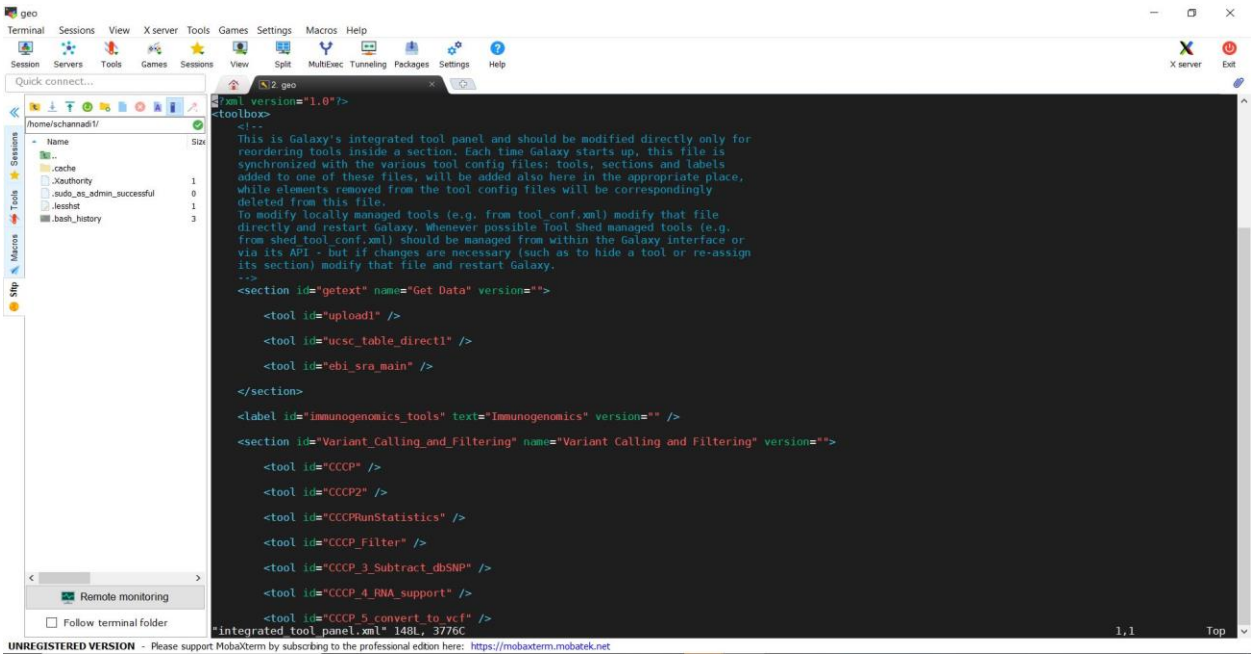

Figure 15: Integrated tool panel xml file 1

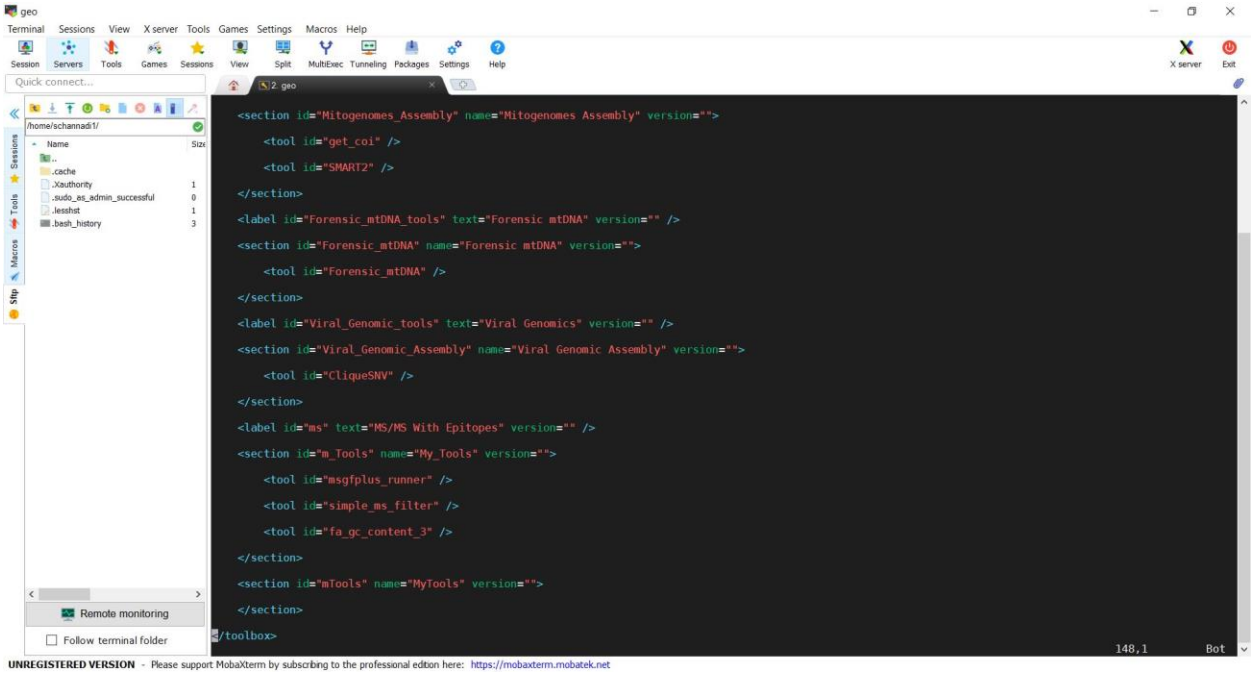

Figure 16: Integrated tool panel xml file 2

Figure 15 and Figure 16 show the sample xml code for integrated tool panel that is used to make CliqueSNV visible on Galaxy website.

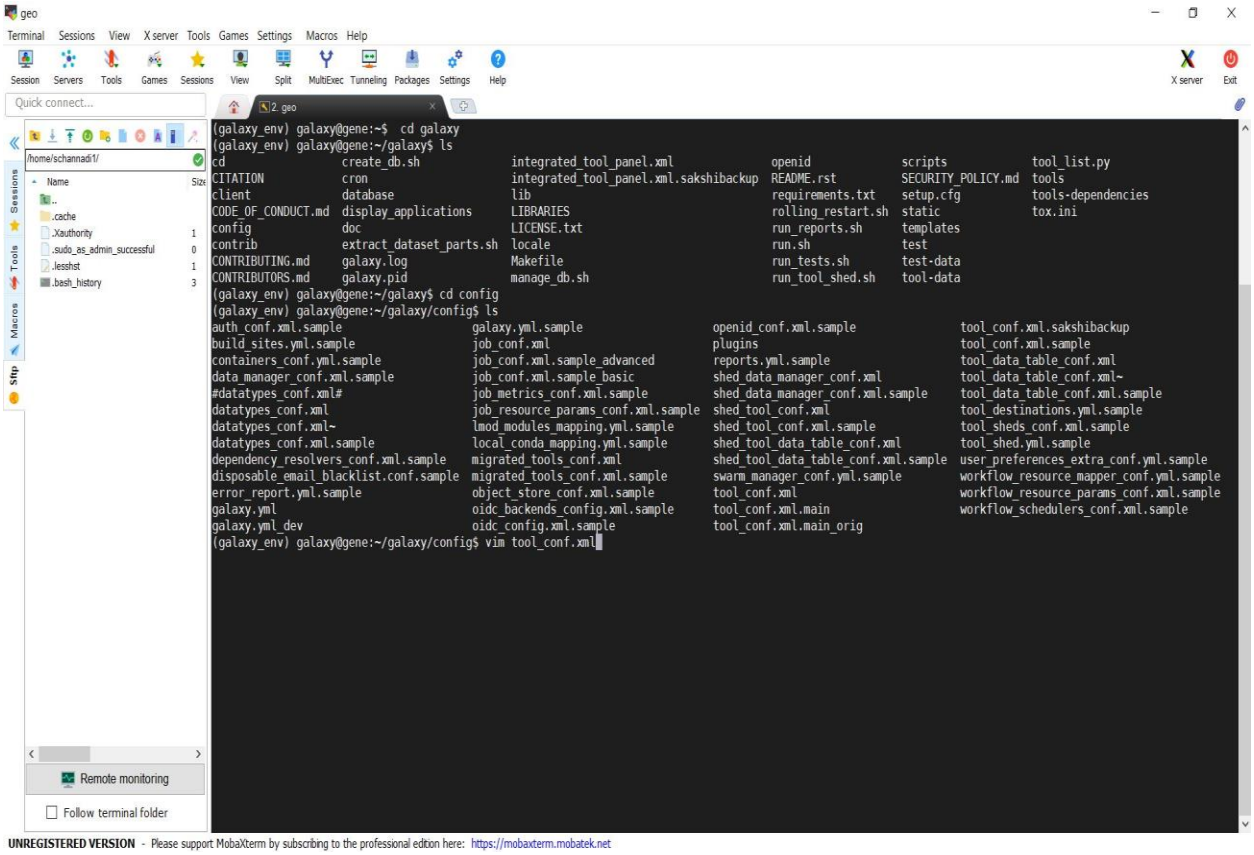

Figure 17: Config xml file of galaxy

We also need to tweak the config files under galaxy to make CliqueSNV visible on the website of Galaxy. Under galaxy, there is a directory called *config* that is present by default and it contains all the config files of galaxy. Here, we need to tweak the xml file called *tool_conf.xml.* It is important to always maintain a backup of all these files that we are tweaking to avoid loss of original files in case there is some unexpected error. Also, galaxy server must be restarted to apply all the changes to the user interface of our tools on Galaxy website.

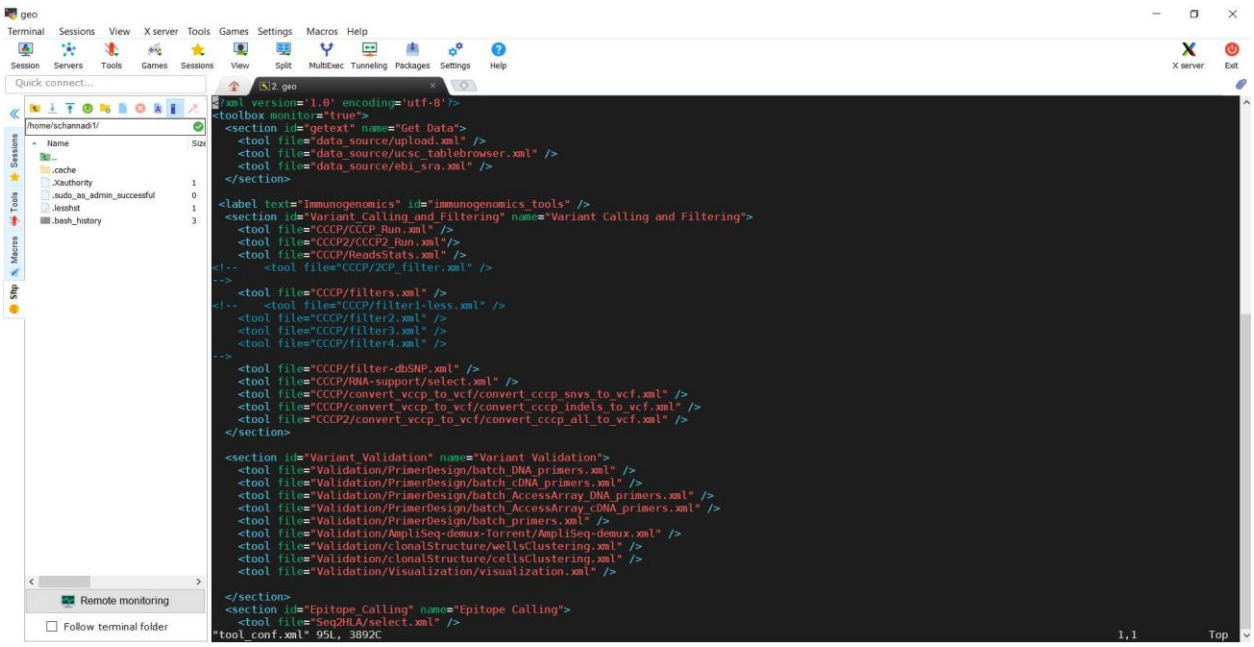

Figure 18: Tool config xml file 1

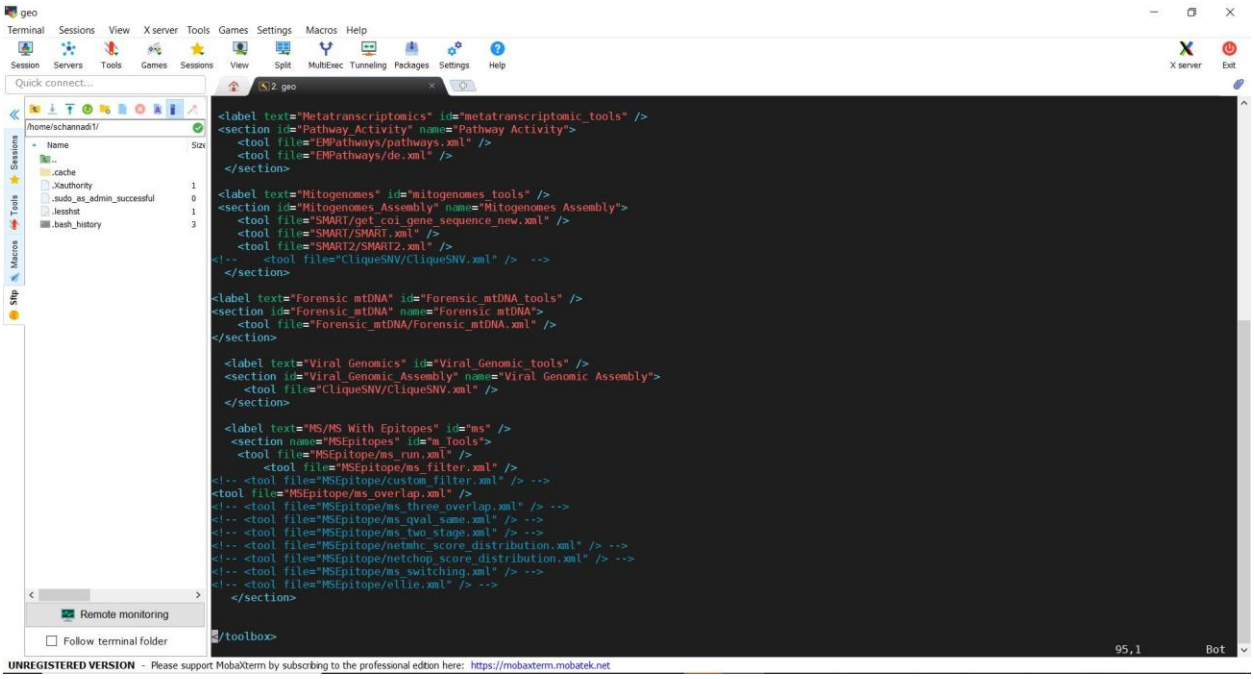

Figure 19: Tool config xml file 2

Figure 18 and Figure 19 show the sample xml code for tool config xml that is used to make CliqueSNV visible on Galaxy website.

4.3 User Interface of CliqueSNV on Galaxy

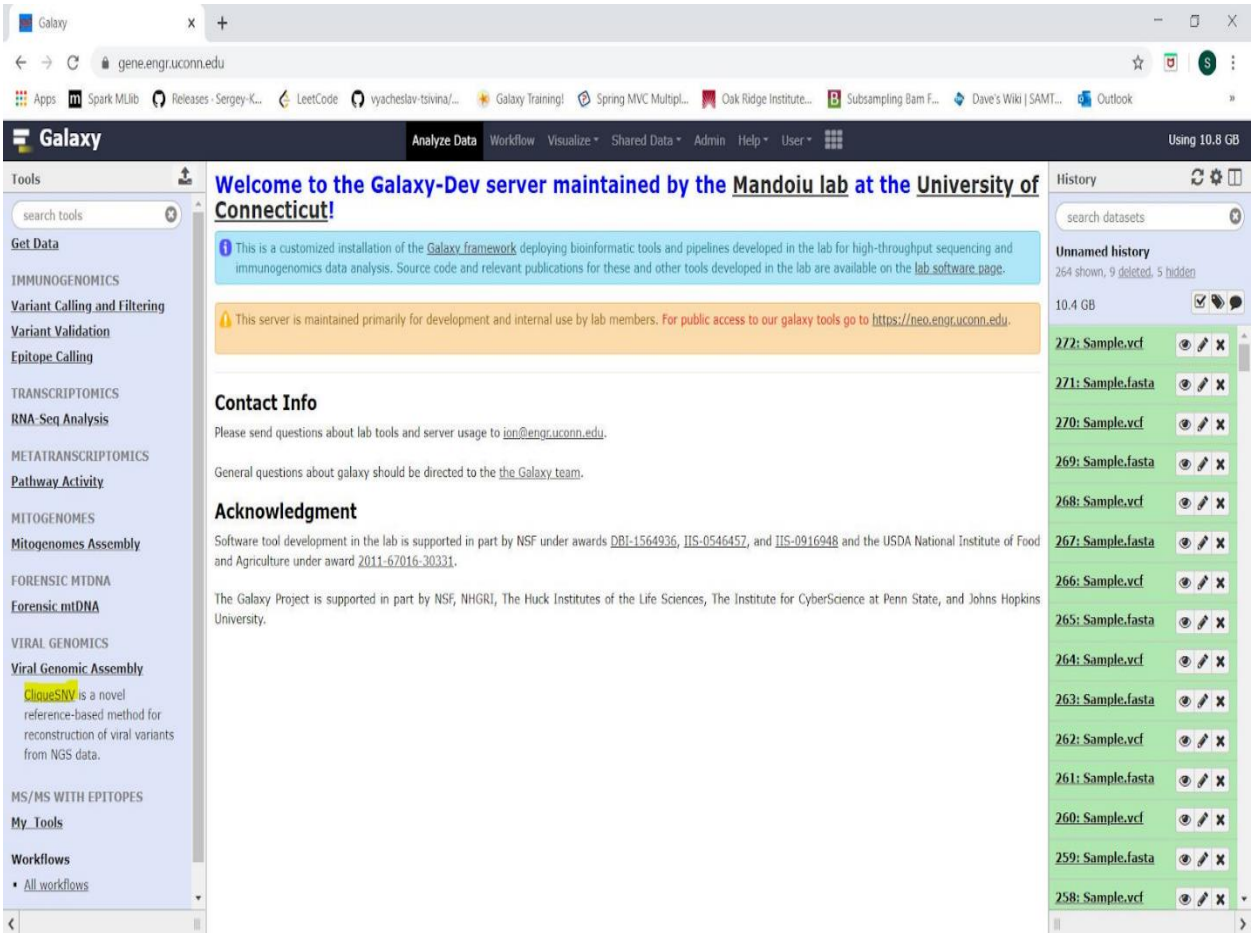

Figure 20: Where to find CliqueSNV on Galaxy

The figure above shows where to find CliqueSNV on Galaxy, users can simply create an account or use it for free on the public server of Galaxy. It is on the left pane of Galaxy which contains a list of tools under the *viral genomics* section. The user interface of CliqueSNV appears in the middle of the screen when users click on CliqueSNV.

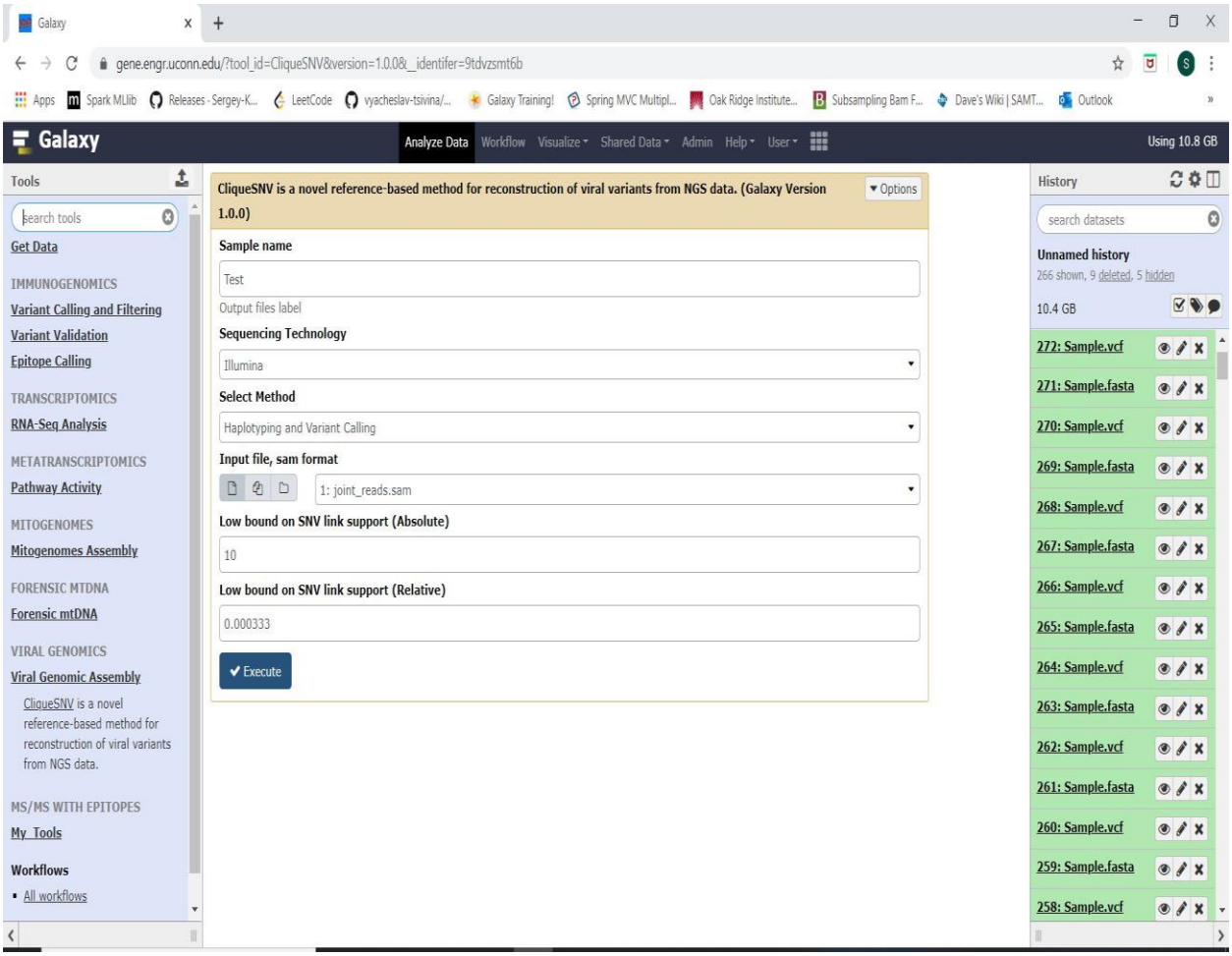

Figure 21: User interface of CliqueSNV on Galaxy

The figure above shows the user interface of CliqueSNV in middle of the screen of Galaxy. *Sample name* is a text field where the users can give a name to their execution at that instance. The name given as *sample name* is also the name of the output that users can see in the history on the right pane. This is one of the advantages of Galaxy because users can execute multiple tools at once and can check their results later because they are stored under history with the sample name that the users have given.

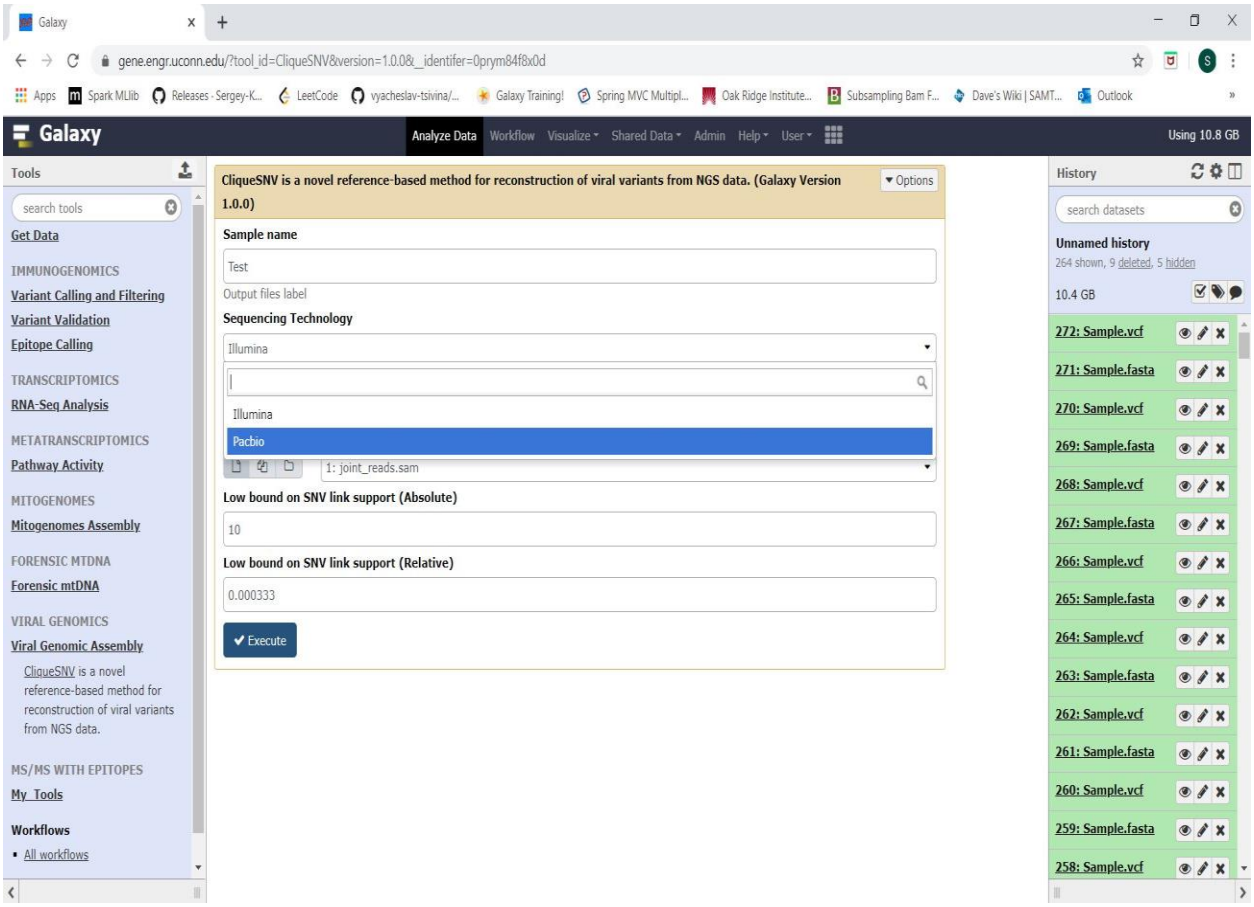

Figure 22: Sequencing technologies supported by CliqueSNV

The above figure shows a dropdown of the sequencing technologies that CliqueSNV supports. The name of the sequencing technology selected by the users is displayed under *sequencing technology* and users can click on it to change their selection at any point before execution. CliqueSNV eliminates the need for preliminary error correction and assembly and infers haplotypes from patterns in distributions of SNVs in sequencing reads which makes it suitable for both long single-molecule reads (PacBio) as well as short paired reads (Illumina). The user interface of CliqueSNV is designed in such a way that even a novice computer user can run the tool without any external help.

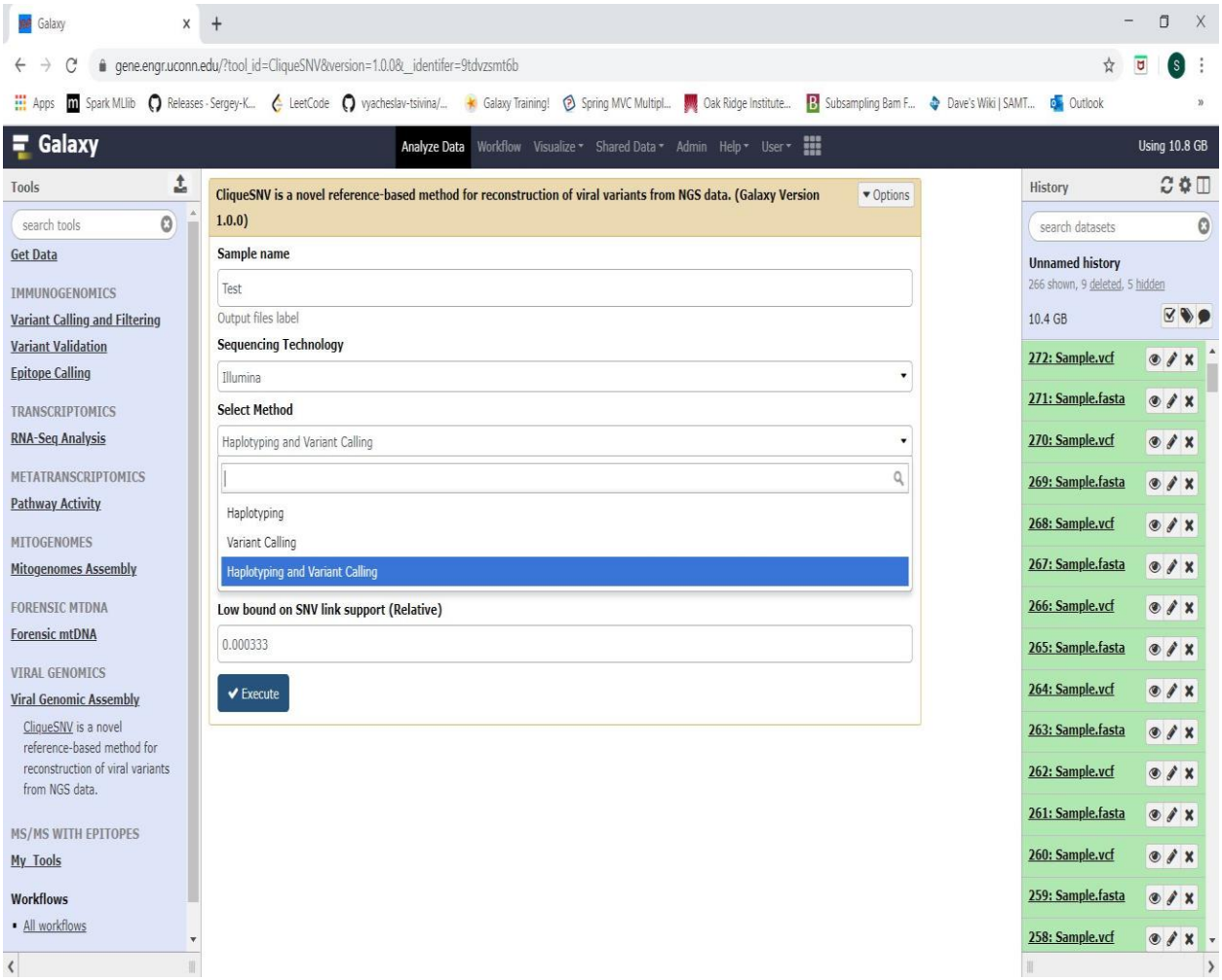

Figure 23: Different methods which we can run for CliqueSNV on input samples

The above figure shows a dropdown of various methods that CliqueSNV supports. The name of the method selected by the users is displayed under *select method* and users can click on it to change their selection at any point before execution. The user interface of CliqueSNV on Galaxy also provides an option to execute haplotyping and variant calling on a sample input either simultaneously or separately.

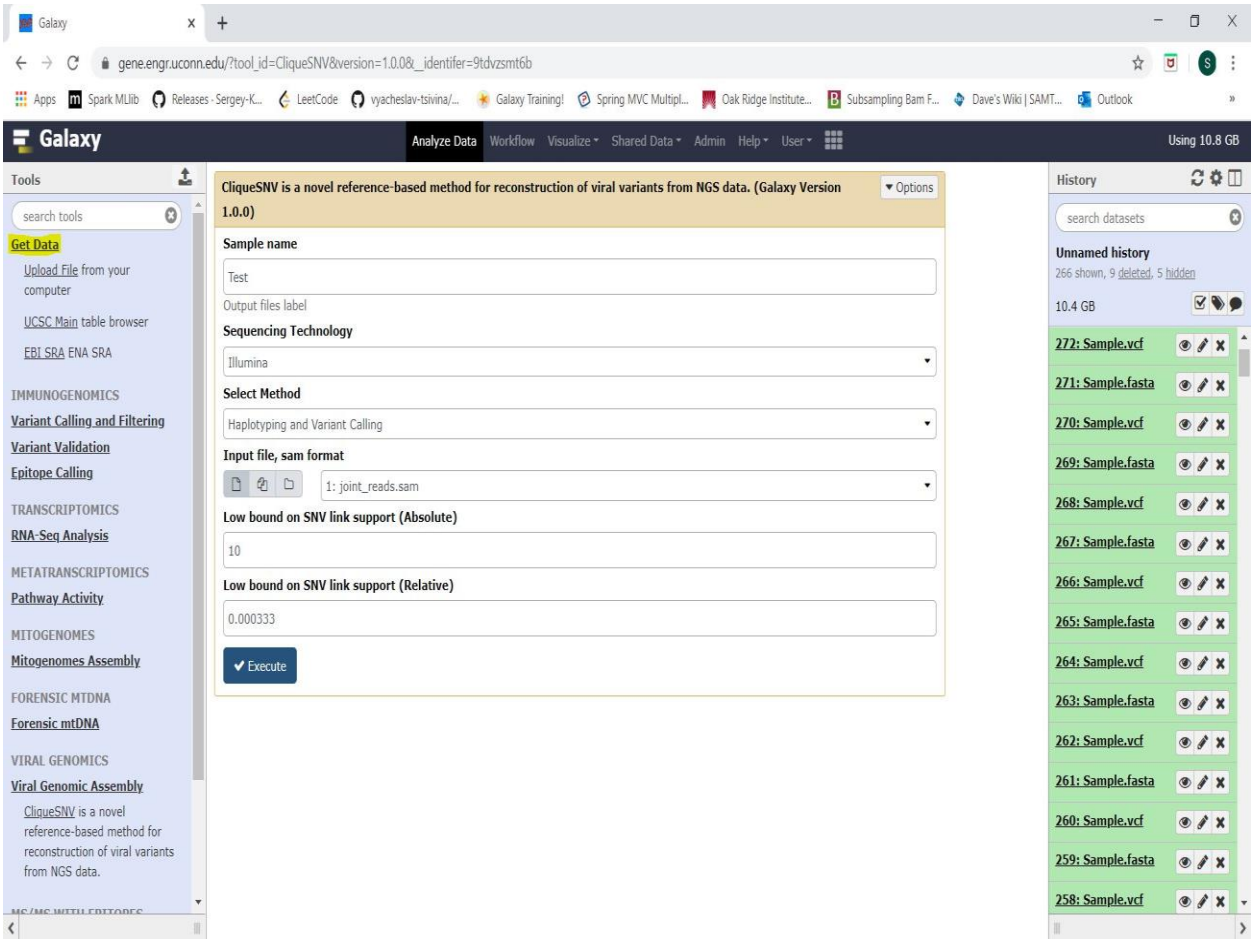

Figure 24: How to upload input samples on Galaxy

The above figure shows how to upload input samples on Galaxy. If the users click on the *get data* highlighted on the left pane, a list of servers appears from which users can transfer data directly to Galaxy. When users click on one of those servers, they can see that a new panel appears in the middle of the Galaxy screen. Typically there is a list of several servers under *get data* in a traditional Galaxy server but this can be modified by the Galaxy administrator and hence the list contains only three servers in the above figure.

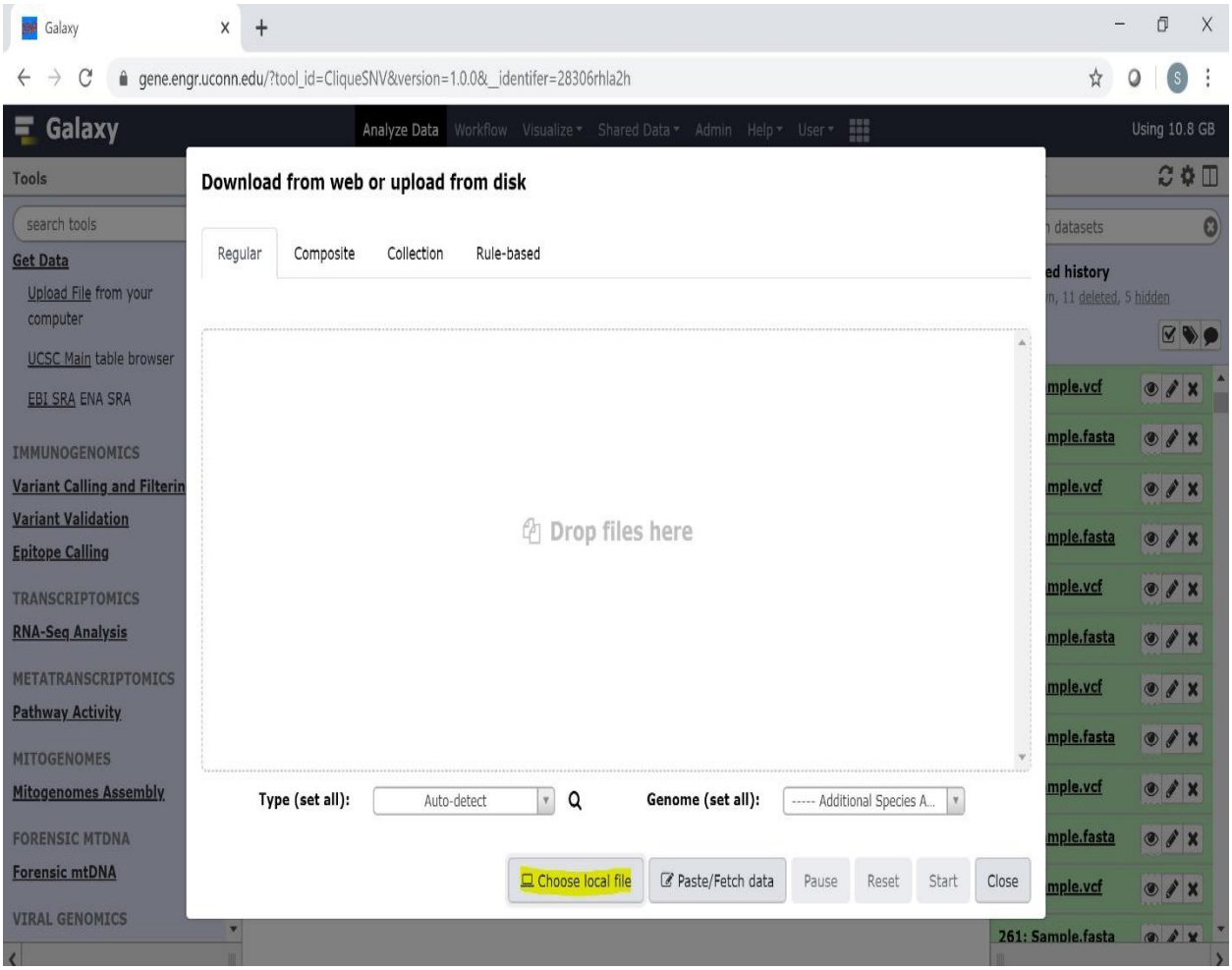

Figure 25: How to upload files from local computer to Galaxy

The above figure shows a panel that appears when users click on *upload file* under *get data.* This panel contains two options to upload files to Galaxy, the first option is *choose local file* which lets the users to directly upload a file from their computer and the second option is *paste/fetch data* which lets the users to put in http or ftp URL and transfer the file directly to galaxy. Users can see their uploaded files in Galaxy history on the right pane.

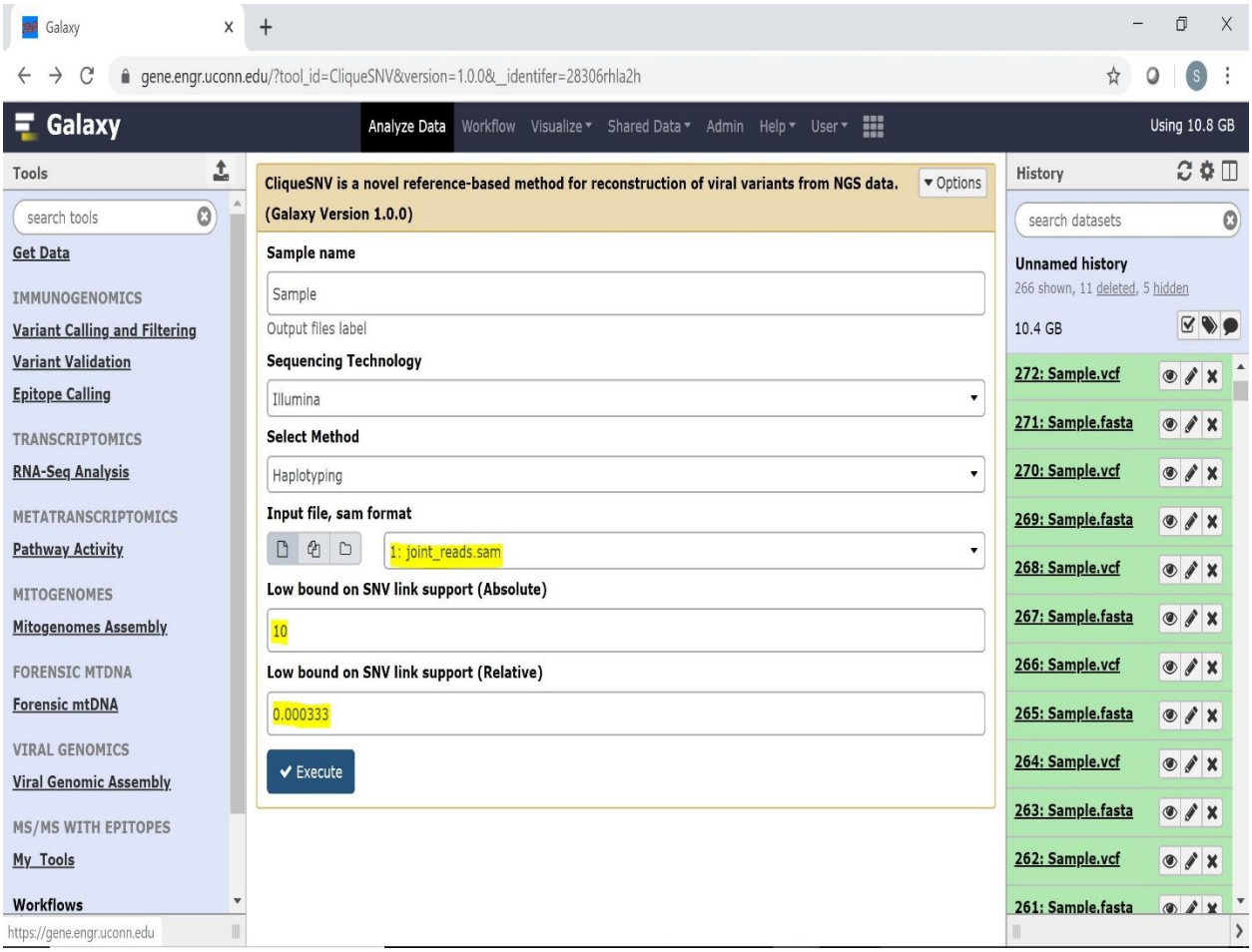

Figure 26: How to set parameters for CliqueSNV on Galaxy

The above figure shows the steps to be followed once the input file has been uploaded on Galaxy. User interface of CliqueSNV provides two text fields to set parameters called *low bound on SNV link support (Absolute)* and *low bound on SNV link support (Relative)*. These text fields are already populated with their default values, users can edit these values or choose to run the tool with the default values. There is a button named *execute* at the end which is clickable and is used to run the tool.

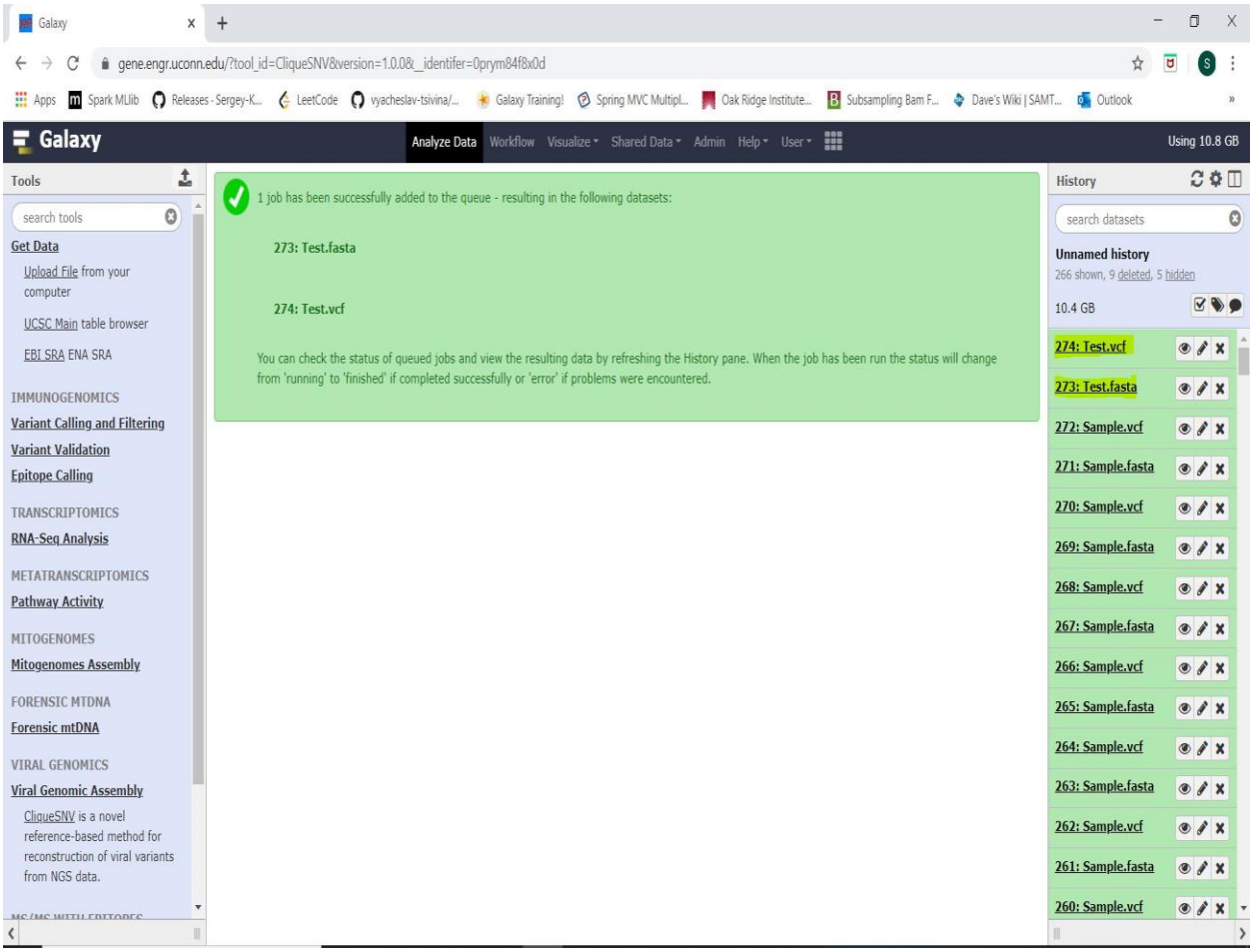

Figure 27: Where to find results of CliqueSNV on Galaxy

The above figure shows the results under history on the right pane of Galaxy. Once the users click on *execute* button, the expected results are displayed under history with the sample name that the user has given and the status of the job changes from ‘running’ to ‘finished’ if completed successfully or ‘error’ if problems were encountered. The users can check either the results or the error logs in the history pane depending on the status of their execution.

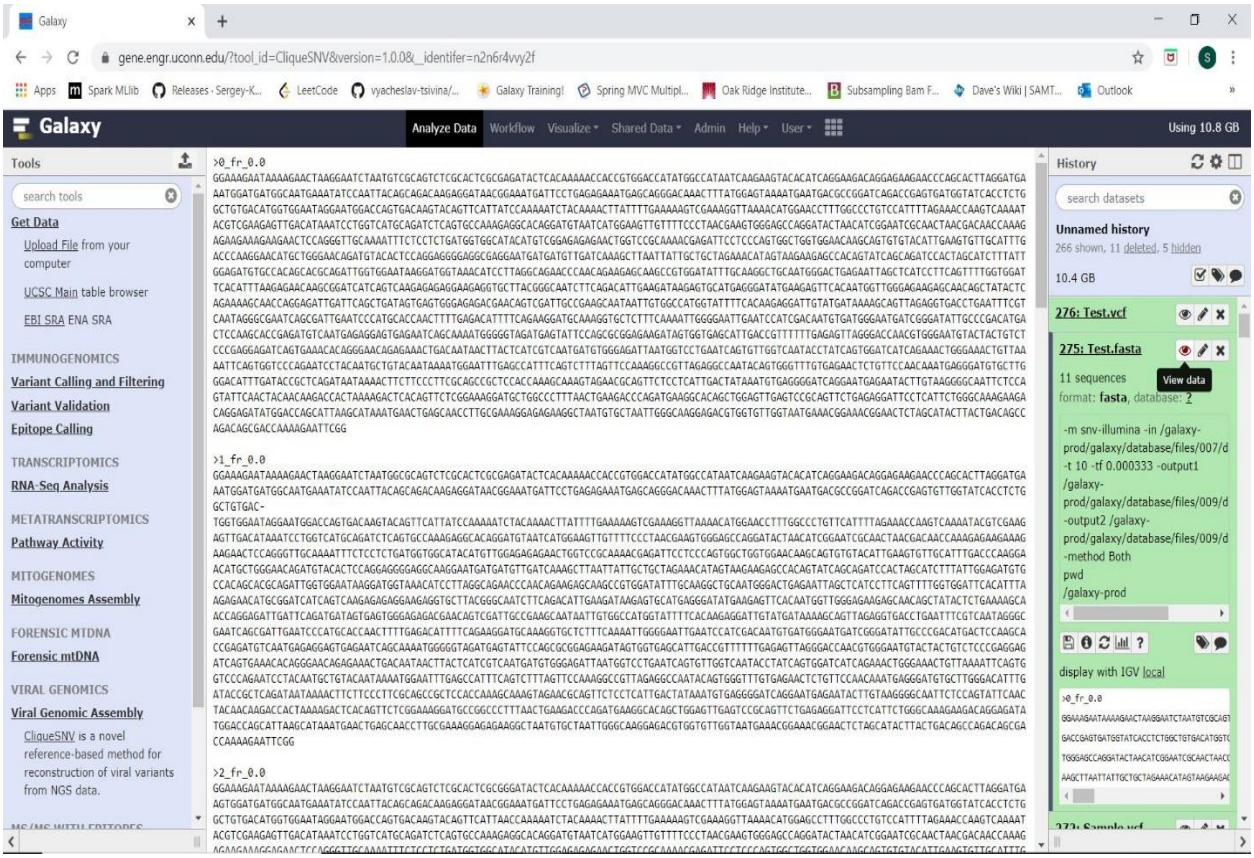

Figure 28: Results when we run haplotyping method of CliqueSNV on Galaxy

The above figure shows the results of executing CliqueSNV with haplotyping method. The output file of haplotyping has an extension of ".fasta" and this file is visible in the history pane with the name "SampleName.fasta". When users click on the output file name, it is expanded and the users can see the number of sequences obtained in the output file just under the filename. Users can also click on the eye shaped icon just beside the output file name to view the results on Galaxy itself or click on the save icon under the output file name to save it on their local system for further analysis.

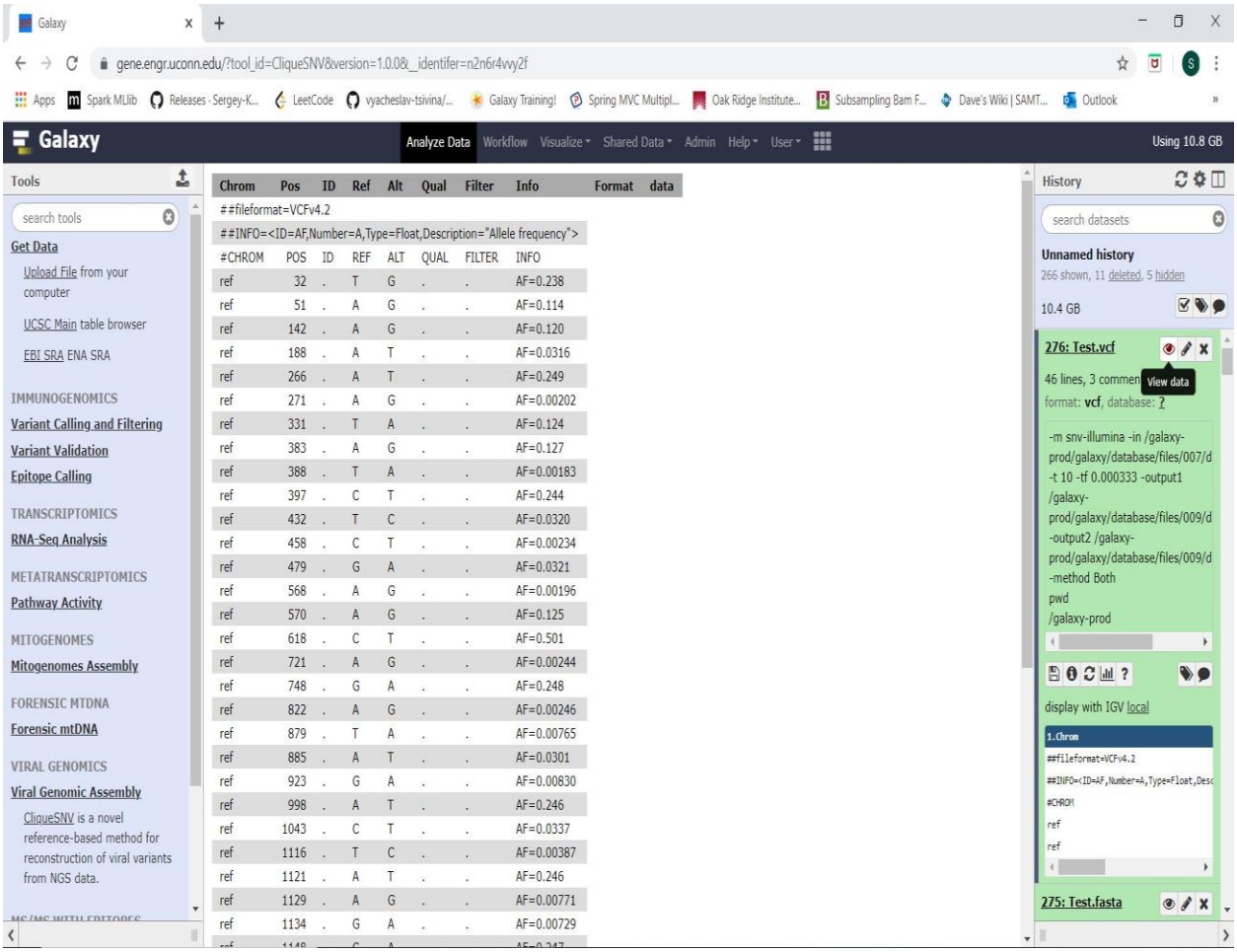

Figure 29: Results when we run variant calling method of CliqueSNV on Galaxy

The above figure shows the results of executing CliqueSNV with variant calling method. The output file of variant calling has an extension of ".vcf" and this file is visible in the history pane with the name "SampleName.vcf". When users click on the output file name, it is expanded and the users can see the number of lines obtained in the output file just under the filename. Users can also click on the eye shaped icon just beside the output file name to view the results on Galaxy itself or click on the save icon under the output file name to save it on their local system for further analysis.

4.4 Sample runs for NGS datasets

In the CliqueSNV article, two experimental and two simulated datasets were used to measure its performance compared to other state of the art methods. Some of these datasets were used to test the standalone application of CliqueSNV and the user interface of CliqueSNV on Galaxy.

5 CONCLUSIONS

In this thesis, we have successfully developed a graphical user interface for CliqueSNV as a standalone application and also deployed it on Galaxy. For the future work, we plan to extend CliqueSNV pipeline on Galaxy with an option to obtain 95% confidence intervals for each allele and haplotype frequency using bootstrapping.

REFERENCES

[1]. S. Prabhakaran, M. Rey, O. Zagordi, N. Beerenwinkel, and V. Roth. "HIV haplotype inference using a propagating Dirichlet process mixture model." IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB), **11**(1): 182–191, 2014

[2]. Huber M., Metzner K.J., Geissberger F.D., Shah C., Leemann C., Klimkait T., Böni J., Trkola A., Zagordi O. "MinVar: A rapid and versatile tool for HIV-1 drug resistance genotyping by deep sequencing." J. Virol. Methods. 2017; 240:7–13. doi: 10.1016/j.jviromet.2016.11.008.

[3]. Knyazev S., Tsyvina V., Melnyk A., Malygina T., Porozov Y.B., Campbell E., Switzer W.M., Skums P., Zelikovsky A., 2018. "CliqueSNV: Scalable reconstruction of intra-host viral populations from NGS reads." bioRxiv1–8.https://doi.org/10.1101/264242.

[4]. A. Artyomenko, N.C. Wu, S. Mangul, E. Eskin, R. Sun and A. Zelikovsky. "Long single-molecule reads can resolve the complexity of the influenza virus composed of rare, closely related mutant variants." Journal of Computational Biology, 24(6): 558–570, 2017.

[5]. C. Bron and J. Kerbosch. Algorithm 457: "Finding all cliques of an undirected graph. Commun." ACM, **16**(9): 575–577, Sept. 1973.

[6]. M. Nicolae, S. Mangul, I. Mandoiu, and A. Zelikovsky. "Estimation of alternative splicing isoform frequencies from rna-seq data." Algorithms for Molecular Biology, **6**: 9, 2011.

[7]. S. Peleg, M. Werman, and H. Rom. "A unified approach to the change of resolution: Space and gray-level." IEEE Transactions on Pattern Analysis and Machine Intelligence, **11**(7): 739–742, 1989