

# **PBGL QTL-BSA Analysis v1.0**

**A Training Tutorial**  
**[DRAFT]**

**Anibal E. Morales-Zambrana**  
**Norman Warthmann**

**Plant Breeding and Genetics Laboratory**  
**FAO/IAEA Joint Division**  
**Seibersdorf, Austria**

Created: June, 2021  
Last updated: 21 June 2021

**Please note:** *This is not an official IAEA publication but is made available as working material. The material has not undergone an official review by the IAEA. The views expressed do not necessarily reflect those of the International Atomic Energy Agency or its Member States and remain the responsibility of the contributors. The use of particular designations of countries or territories does not imply any judgement by the publisher, the IAEA, as to the legal status of such countries or territories, of their authorities and institutions or of the delimitation of their boundaries. The mention of names of specific companies or products (whether or not indicated as registered) does not imply any intention to infringe proprietary rights, nor should it be construed as an endorsement or recommendation on the part of the IAEA.*

# Contents

<b>1</b>	<b>Background</b>	<b>2</b>
<b>2</b>	<b>Installations - Virtual Environments and Software Packages</b>	<b>3</b>
2.1	Miniconda3 (conda) and Mamba . . . . .	3
2.2	Git Installation and Repo Cloning . . . . .	4
2.3	Required Libraries with Mamba . . . . .	4
2.3.1	Automatically (faster) . . . . .	5
2.3.2	Manually (slower) . . . . .	5
<b>3</b>	<b>Running Jupyter</b>	<b>6</b>
3.1	Editing the Configuration File . . . . .	6
3.2	Running a RCNV_seq-template Jupyter Notebook . . . . .	7
3.2.1	Import Necessary Libraries and Functions . . . . .	8
3.2.2	Configuration File Path Definition . . . . .	8
3.2.3	Plot Allele Frequencies . . . . .	8
<b>4</b>	<b>Example Tutorial - QTL-BSA Analysis of Sorghum</b>	<b>9</b>
4.1	Data Download . . . . .	9
4.1.1	Linux Terminal . . . . .	9
4.1.2	Web-Browser . . . . .	10
4.2	General Installations . . . . .	10
4.3	Github Repository Cloning . . . . .	10
4.4	Virtual Environment Creation . . . . .	11
4.5	Configuration File Editing . . . . .	11
4.6	Jupyter Notebook Analysis . . . . .	12
<b>5</b>	<b>References</b>	<b>18</b>

---

# 1 Background

Quantitative trait locus - bulk segregant analysis (QTL-BSA) using allele frequencies, bcftools, Python, Jupyter Notebooks, Miniconda3, Mamba, and Git.

All the commands run in a Linux terminal are preceded by the \$ prompt sign. To run a command, copy/paste the command without the \$ sign. Those commands run in a Jupyter Notebook are preceded by the *In [ ]:*

---

**Note:** This is not an official IAEA publication but is made available as working material. The material has not undergone an official review by the IAEA. The views expressed do not necessarily reflect those of the International Atomic Energy Agency or its Member States and remain the responsibility of the contributors. The use of particular designations of countries or territories does not imply any judgement by the publisher, the IAEA, as to the legal status of such countries or territories, of their authorities and institutions or of the delimitation of their boundaries. The mention of names of specific companies or products (whether or not indicated as registered) does not imply any intention to infringe proprietary rights, nor should it be construed as an endorsement or recommendation on the part of the IAEA.

---

## 2 Installations - Virtual Environments and Software Packages

Before installing any necessary software, it is recommended to check if the computer is running 32-bit or 64-bit for downloading Miniconda3. Run the following to verify the system:

```
$ uname -m
```

### 2.1 Miniconda3 (conda) and Mamba

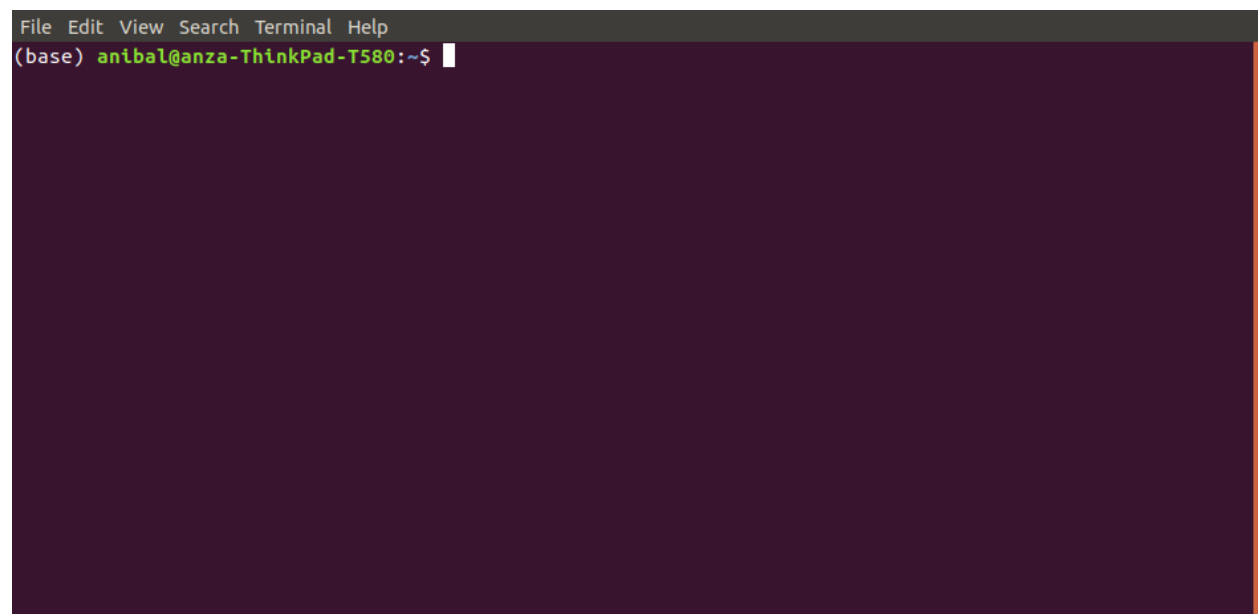
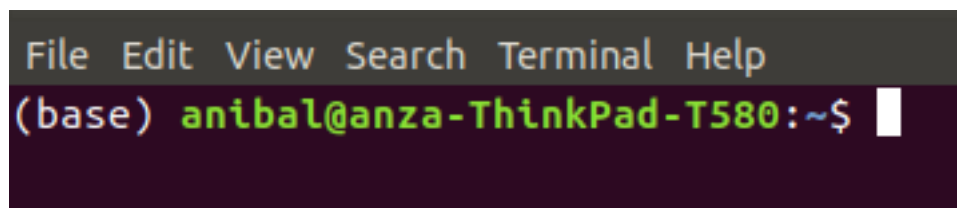
Download the Miniconda3, or simply “conda”, installer:

- [Miniconda3 installer for Linux](#)

Run the downloaded installer (for a 64-bit system):

```
$ bash Miniconda3-latest-Linux-x86_64.sh
```

Open a new terminal window for conda to take effect. The word (*base*) should appear in front of the computer name in the terminal window, like so:

A screenshot of a terminal window with a dark purple background. The title bar at the top reads "File Edit View Search Terminal Help". The first line of the terminal shows the prompt "(base) antibal@anza-ThinkPad-T580:~\$" with a white cursor. The rest of the terminal area is empty.A close-up screenshot of the terminal window showing the prompt "(base) antibal@anza-ThinkPad-T580:~\$" with a white cursor. The title bar "File Edit View Search Terminal Help" is visible above the prompt.

Verify the installation and update conda in new terminal window with:

```
$ conda env list
$ conda update --all
$ conda upgrade --all
```

Install mamba library/package manager that will be used for installing software dependencies of the tool:

```
$ conda install mamba --yes
```

## 2.2 Git Installation and Repo Cloning

Git is required for cloning locally (downloading a copy to your local computer) the PBGL QTL-BSA Github repository. Git and Github are used for version control of software. It keeps track of development, releases, and issues of a software project.

Install **git** for cloning the **pbgl-qtl-bsa** software repository from Github, where the latest version of the tool resides:

```
$ mamba install git --yes
```

After the installation, clone PBGL's QTL-BSA repository, **pbgl-qtl-bsa**, to the local computer in any desired directory.

```
$ git clone https://github.com/pbgl/pbgl-qtl-bsa.git
```

The cloning process will depict the following:

```
Cloning into 'pbgl-qtl-bsa'...
remote: Enumerating objects: 139, done.
remote: Counting objects: 100% (139/139), done.
remote: Compressing objects: 100% (116/116), done.
remote: Total 139 (delta 21), reused 129 (delta 16), pack-reused 0
Receiving objects: 100% (139/139), 6.90 MiB | 298.00 KiB/s, done.
Resolving deltas: 100% (21/21), done.
```

The **pbgl-qtl-bsa** repository should have been cloned successfully. Verify that the download is complete by listing the folders/files in the directory.

```
$ ls -l
```

The folder called **pbgl-qtl-bsa** should be listed in the directory.

## 2.3 Required Libraries with Mamba

QTL-BSA has multiple dependencies, listed below:

- Bcftools
- Python
  - pandas
  - matplotlib
  - scikit-allel
  - seaborn
  - natsort
- Jupyter Notebook

There are two ways to create the **qtl-bsa** virtual environment and install the necessary libraries to run QTL-BSA: automatically or manually.

### 2.3.1 Automatically (faster)

One YAML file, **environment.yml**, is provided to automatically create a virtual environment and install the dependent libraries through mamba. The file creates the **qtl-bsa** virtual environment, along Jupyter Notebook, and the Python libraries. Run the **environment.yml**:

```
$ mamba env create --file envs/environment.yml
```

Once done, a list of the virtual environments available can be seen by running:

```
$ conda env list
```

Activate (enter) the recently-created virtual environment **qtl-bsa**:

```
$ conda activate qtl-bsa
```

Once done, the virtual environment should be activated and all the necessary packages should be installed. This can be verified with:

```
$ conda list
```

### 2.3.2 Manually (slower)

To manually create and activate an environment, run:

```
$ conda create --name qtl-bsa  
$ conda activate qtl-bsa
```

Start running the installations of the necessary libraries:

```
$ mamba install python=3.6.7 notebook natsort bcftools pandas matplotlib seaborn  
↪scikit-allel --yes
```

Once done, all the necessary packages should be installed. This can be verified with:

```
$ conda list
```

## 3 Running Jupyter

To activate Jupyter, run the following in the terminal:

```
$ jupyter notebook
```

This command will start a Jupyter session inside the directory the command is run. The user can navigate between directories, visualize files, and edit files in a web browser by clicking on directories or files, respectively.

Look for the directory **pbgl-qt1-bsa** and click on it. Click on **tool** directory, which contains three directories and two Jupyter Notebooks. Here is a breakdown of each:

- *Allele\_Frequency\_Plots\_Computomics*:
  - directory that will contain both tab-files and images output after running a QTL-BSA analysis
- *config*:
  - directory containing configuration files specifying file paths, parameter definitions, list of samples, and list of chromosomes
- *scripts*:
  - directory containing Python scripts with functions for plotting allele frequencies
- two Jupyter Notebooks:
  - QTL-BSA-sorghum-example.ipynb
    - \* example analysis of a comparison between a control and mutant of sorghum
  - QTL-BSA.ipynb
    - \* template for the user

---

**Note:** Jupyter lets the user duplicate, rename, move, download, view, or edit files in a web browser. This can be done by clicking the box next to a file and choosing accordingly.

---

### 3.1 Editing the Configuration File

In order to run the QTL-BSA Jupyter Notebook, the user needs to feed it with a configuration file (**config-allele-freq.yml**) that specifies the path to the vcf file, list of samples, chromosomes to analyze, and window/step size definitions for calculating and plotting allele frequencies.

The configuration file **config-allele-freq.yml** can be found in the **pbgl-qt1-bsa/tool/config** directory. The configuration file contains the following fields:

```
# VCF File information (unzipped .vcf)
vcf_file:
  path:
  name:
  extension:

# parameters to plot allele frequencies
window_size:
step_size:

# list of samples
```

(continues on next page)

```

samples:
  control:
  mutant:
  F2_wild_type:
  F2_mutant:

# chromosomes to analyze
chromosomes:
  -

```

---

**Note:** The user needs to edit **config-allele-freq.yml** to point towards a vcf file; specify chromosomes to analyze; and define the parameters to plot allele frequencies. The paths specified in this tutorial manual may not match the paths of the user's computer.

---

One example configuration file is provided (**config-allele-freq-sorghum-example.yml**). The configuration file **config-allele-freq.yml** contains multiple fields to be defined by the user.

- *vcf\_file*:
  - *path* - path to directory containing the vcf file
  - *name* - name of the vcf file
  - *extension* - type of file; only handles **.vcf** files at the moment
- *window\_size* - integer specifying the window size to allele frequencies
- *step\_size* - integer specifying the step size of the sliding window
- *samples*:
  - *control* - control parent
  - *mutant* - mutant parent
  - *F2\_wild\_type* - F2 organism showing wild type phenotype
  - *F2\_mutant* - F2 organism showing mutant phenotype
- *chromosomes* - list of chromosomes to analyze

## 3.2 Running a RCNV\_seq-template Jupyter Notebook

---

**Note:** It is recommended to duplicate the **QTL-BSA.ipynb** notebook and then renaming the copy before doing any edits to the notebook.

---

In the **pbgl-qtl-bsa/tool** directory, click on **QTL-BSA.ipynb** and a new tab in your web-browser will open the notebook.

The notebook contains cells that are populated by text or code. Instructions are provided in the notebook to guide the user. To run a cell, click on the corresponding cell and click on the *Run* button on the top of the notebook. Another way to run a cell can be done by clicking on the corresponding cell and pressing **Ctrl + Enter** or **Shift + Enter**.

The notebook consists of 3 sections:

1. Import Necessary Libraries and Functions
2. Configuration File Path Definition



### 3. Plot Allele Frequencies

The third section **Plot Allele Frequencies** has 3 subsections:

1. Unweighted Window Sizes
2. Weighted Window Sizes from Configuration File
3. Weighted Window Sizes Defined in Jupyter Notebook

#### 3.2.1 Import Necessary Libraries and Functions

Imports the functions needed to plot the allele frequencies. All the libraries are imported from **pbgl-qtl-*bsa*/tool/scripts** directory.

```
In [ ]: from scripts.plot_allele_freqs import *
```

#### 3.2.2 Configuration File Path Definition

Defines the path and name of the configuration path. Configuration files can be found in the **pbgl-qtl-*bsa*/tool/config** directory.

```
In [ ]: config = "config/config-allele-freq.yml"
```

#### 3.2.3 Plot Allele Frequencies

Contains three functions to plot the allele frequencies by extracting information from the configuration file defined previously. These are:

1. Plotting Unweighted Window Sizes

```
In [ ]: plot_allele_frequencies_raw(config)
```

This function will plot the allele frequencies of an F2 mutant against the F2 wild-type. It does not take into consideration neither the window size nor the step size. The function has no additional parameters.

2. Plotting Weighted Window Sizes from Configuration File

```
In [ ]: plot_allele_frequencies_weighted(config)
```

This function will plot the allele frequencies of an F2 mutant against the F2 wild type. It takes into consideration both the window size and step size.

Running the function as shown will use the values defined in *window\_size* and *step\_size* inside the configuration file used.

3. Plotting Weighted Window Sizes from Jupyter Notebook

```
In [ ]: plot_allele_frequencies_weighted(config, window_size= , step_size= )
```

This function provides the user to edit the *window\_size* and *step\_size* parameters in-place in the Jupyter Notebook without having to go back and edit the configuration file. If the parameters *window\_size* and *step\_size* are empty, the function will extract these parameter values from the configuration file.

## 4 Example Tutorial - QTL-BSA Analysis of Sorghum

In this section of the manual, an example analysis of sorghum will be shown in a step-by-step process. The data has been filtered to show variant locations where the control has genotype '0/0' and the mutant has genotype '1/1'. The QTL-BSA analysis will depict an artifact on chromosome Chr04.

The tutorial is divided between the following sections:

1. Data Download
2. General Installations
3. Github Repository Cloning
4. Virtual Environment Creation
5. Configuration File Editing
6. Jupyter Notebook Analysis

### 4.1 Data Download

Running this tutorial requires one variant call format (VCF) file of sorghum crop: **freebayes\_D2.filtered.vcf**. It is used in genetics for storing variations of gene sequences.

The following links can be used to download the necessary VCF file and its correspondig indexed **.tbi** file:

- freebayes\_D2.filtered.vcf
  - [https://bsslinnovlnafalpocl.blob.core.windows.net/sample-container/2021\\_Training/freebayes\\_D2.filtered.vcf](https://bsslinnovlnafalpocl.blob.core.windows.net/sample-container/2021_Training/freebayes_D2.filtered.vcf)
- freebayes\_D2.filtered.vcf.gz.tbi
  - [https://bsslinnovlnafalpocl.blob.core.windows.net/sample-container/2021\\_Training/freebayes\\_D2.filtered.vcf.gz.tbi](https://bsslinnovlnafalpocl.blob.core.windows.net/sample-container/2021_Training/freebayes_D2.filtered.vcf.gz.tbi)

There are two additional ways to download the **.vcf** file and its respective index, besides clicking the links above:

1. Linux terminal
2. Web-Browser

#### 4.1.1 Linux Terminal

Open a new terminal and navigate to a directory of choice. We recommend creating a directory to store the data and running *wget* in the respective location:

```
$ mkdir vcf_files
$ cd vcf_files
$ wget https://bsslinnovlnafalpocl.blob.core.windows.net/sample-container/2021_
↪Training/freebayes_D2.filtered.vcf
$ wget https://bsslinnovlnafalpocl.blob.core.windows.net/sample-container/2021_
↪Training/freebayes_D2.filtered.vcf.gz.tbi
```

### 4.1.2 Web-Browser

Open a web-browser of preference. Copy/paste the links provided above in the address bar. This should automatically begin the download. Move the downloaded files to a location of personal preference.

## 4.2 General Installations

Open a web browser and copy/paste the following link to download Miniconda3:

- <https://docs.conda.io/en/latest/miniconda.html#linux-installers>

After download, open a new terminal window, navigate to the directory with the downloaded Miniconda3 installer, and run the installation.

---

**Note:** The downloaded Miniconda3 installer file might not match the one run in this example. Please, type the corresponding name of the **.sh** file downloaded.

---

```
$ bash Miniconda3-latest-Linux-x86_64.sh
```

Once the Miniconda3 installation is done, close the terminal and open a new one. The word (*base*) should be present to the left of the computer name in the prompt. Update conda and install mamba.

```
$ conda update --all --yes
$ conda upgrade --all --yes
$ conda install mamba --yes
```

## 4.3 Github Repository Cloning

Git or Github is used for storage and version control of software projects. Git is used to manage in a local Linux terminal. First, git will need to be installed. In a new terminal window, run the installation command:

```
$ mamba install git --yes
```

After installing git, create a location to store Github repositories, navigate into it, and clone (download a copy locally) of PBGL's **pbgl-ctl-bsa** repository.

```
$ mkdir Github
$ cd Github
$ git clone https://github.com/pbgl/pbgl-ctl-bsa.git
```

A successful cloning will output the following:

```
Cloning into 'pbgl-ctl-bsa'...
remote: Enumerating objects: 139, done.
remote: Counting objects: 100% (139/139), done.
remote: Compressing objects: 100% (116/116), done.
remote: Total 139 (delta 21), reused 129 (delta 16), pack-reused 0
Receiving objects: 100% (139/139), 6.90 MiB | 298.00 KiB/s, done.
Resolving deltas: 100% (21/21), done.
```

Navigate into the cloned repository,

```
$ cd pbgl-ctl-bsa
```

## 4.4 Virtual Environment Creation

Once inside the **pbgl-qtl-bsa** directory, create the **qtl-bsa** virtual environment, install the necessary libraries, and activate the newly created **qtl-bsa** virtual environment.

```
$ mamba env create --file envs/environment.yml
$ conda activate qtl-bsa
```

The name (*qtl-bsa*) environment should be reflected to the left of the computer name in the terminal command prompt.

## 4.5 Configuration File Editing

Open a Jupyter session by running,

```
$ jupyter notebook
```

Navigate to **pbgl-qtl-bsa/tool/config** and click on **config-allele-freq-sorghum-example.yml**. This will open the configuration file in a new web-browser tab. Copy/paste the following in the configuration file.

---

**Note:** The *path* to the vcf file will not match. It needs to be edited accordingly to point towards the location of the vcf file stored locally in the user's computer.

---

```
vcf_file:
  path: /home/anibal/vcf_files
  name: freebayes_D2.filtered
  extension: vcf

window_size: 500000
step_size: 100000

samples:
  control: con--all
  mutant: D2
  F2_wild_type: D2_F2_TT
  F2_mutant: D2_F2_tt

chromosomes:
  - Chr01
  - Chr02
  - Chr03
  - Chr04
  - Chr05
  - Chr06
  - Chr07
  - Chr08
  - Chr09
  - Chr10
```

Save the file and close the tab.

## 4.6 Jupyter Notebook Analysis

In the open Jupyter session, navigate to the **pbgl-qtl-bsa/tool** directory and click on the **QTL-BSA.ipynb** Jupyter Notebook. Begin by running the **Import Necessary Libraries and Functions** section. This can be done by clicking on the cell to be run followed by clicking the *Run* button on the top of the Jupyter Notebook. This cell will import the functions defined in the **pbgl-qtl-bsa/tool/scripts**.

```
In [ ]: from scripts.plot_allele_freqs import *
```

Type the correct name of the configuration file being loaded under the **Configuration File Path Definitions** section. The configuration file **config-allele-freq-sorghum-example.yml** is loaded from the **pbgl-qtl-bsa/tool/config** directory.

```
In [ ]: config = "config/config-allele-freq-sorghum-example.yml"
```

Run the **Unweighted Window Sizes** subsection under the **Plot Allele Frequencies** section. This function will plot the allele frequencies without taking into consideration *window\_size* and *step\_size*; the output plot is named **freebayes\_D2.filtered.pdf** stored in **pbgl-qtl-bsa/tool/Allele\_Frequency\_Plots\_Computomics**.

```
In [ ]: plot_allele_frequencies_raw(config)
```

This function will also create a tabulated **.tsv** file named **freebayes\_D2.filtered.tsv** stored in **pbgl-qtl-bsa/tool/Allele\_Frequency\_Plots\_Computomics**. This file can be exported to any statistical software tool for further analysis. The file contains the following fields:

- *CHROM* - chromosome ID
- *POS* - position in chromosome in base-pair
- *REF* - reference sequence
- *ALT* - alternate sequence
- *RO* - reference allele observation count
- *AO* - alternate allele observation count
- *GT* - genotype
- *GQ* - Genotype Quality, the Phred-scaled marginal (or unconditional) probability of the called genotype
- *SampleRO* - wild-type allele observation count
- *SampleAO* - mutant allele observation count

All output files are stored in the directory **pbgl-qtl-bsa/tool/Allele\_Frequency\_Plots\_Computomics**.

In order to plot the allele frequencies using the window and step sizes, run the subsection **Weighted Window Sizes from Configuration File**. The parameters *window\_size* and *step\_size* are extracted from the configuration file.

```
In [ ]: plot_allele_frequencies_weighted(config)
```

The plot using *window\_size=500000* and *step\_size=100000* from the configuration file outputs the following figure:

In order to modify the *window\_size* and *step\_size* parameters in the Jupyter Notebook without the need of remodifying the configuration file, the *plot\_allele\_frequencies\_weighted* function has two parameters that can be defined to create allele frequencies plots with varying window and step sizes.

Run the **Weighted Window Sizes Defined in Jupyter Notebook** subsection with bigger window and step sizes.

```
In [ ]: plot_allele_frequencies_weighted(config, window_size=5000000 , step_
↪size=1000000 )
```

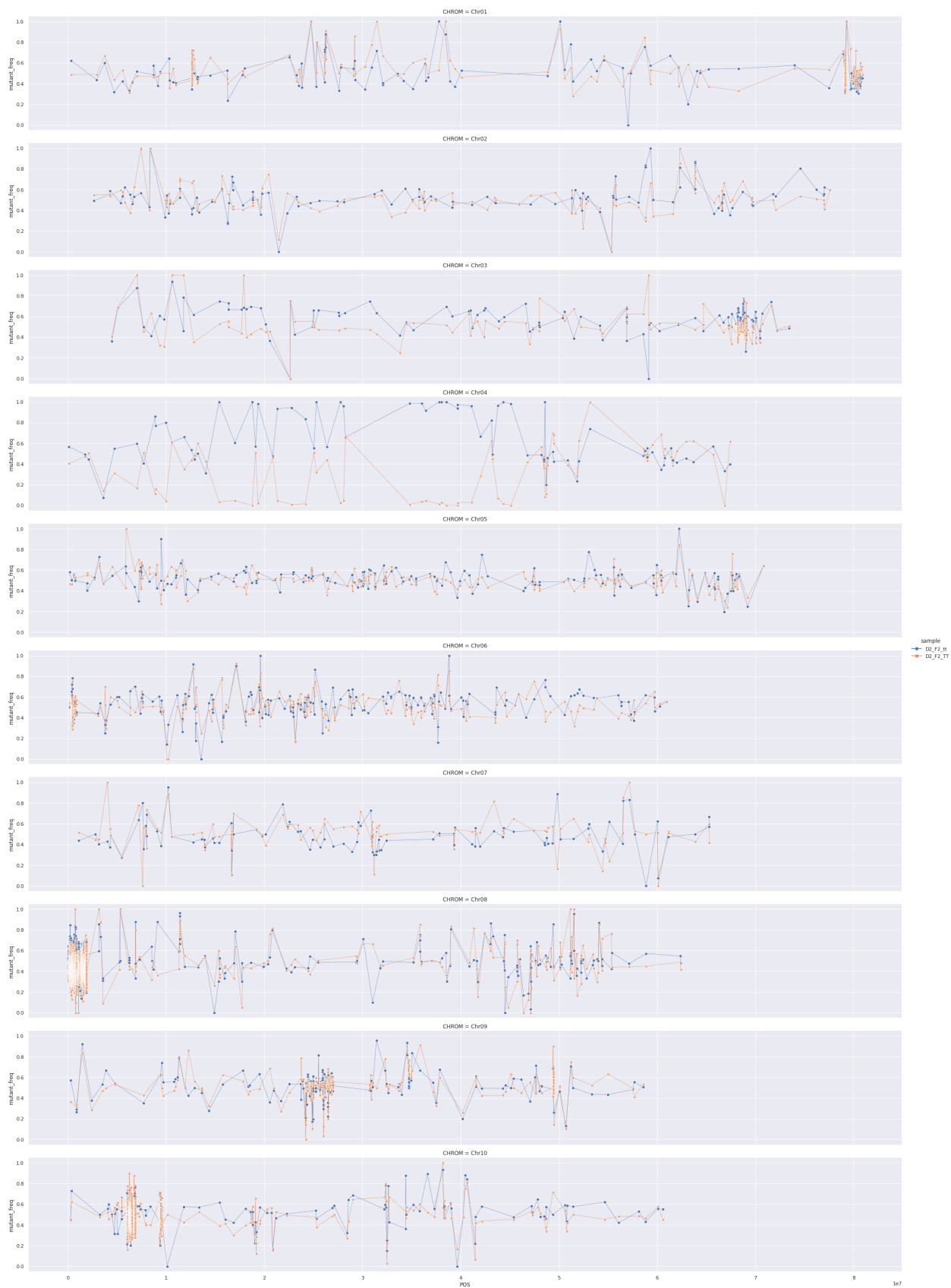


Fig. 1: Allele frequency plot with unweighted window and step sizes

```
freebayes_D2.filtered.tsv x +
localhost:8889/edit/tool/Allele_Frequency_Plots_Computomics/freebayes_D2.filtered.tsv
jupyter freebayes_D2.filtered.tsv 7 minutes ago Logout
File Edit View Language current mode

1 #Samples:con-all,D2,D2_F2_tt,D2_F2_TT
2 CHROM POS REF ALT RO AO GT GQ SampleR0 SampleA0
3 Chr01 344698 C T 39 43 0/0,1/1,0/1,0/1 22,19,137,155 6,0,14,19 0,2,23,18
4 Chr01 2943267 T A 140 109 0/0,1/1,0/1,0/1 134,45,140,140 30,0,66,44 0,16,51,42
5 Chr01 3751995 T C 27 20 1/1,0/0,0/0,0/0 21,16,77,55 2,2,15,8 6,0,10,4
6 Chr01 4720049 G A 174 106 0/0,1/1,0/1,0/1 142,18,142,142 30,0,80,64 0,19,37,50
7 Chr01 5567202 G A 112 118 1/1,0/0,0/1,0/1 98,80,160,160 2,20,39,51 20,0,53,45
8 Chr01 6237654 A G 26 39 1/1,0/0,0/1,0/1 24,31,42,3 2,9,10,5 8,0,20,11
9 Chr01 6582529 CCACTG CG 205 147 0/0,1/1,0/1,0/1 160,45,160,160 31,0,95,79 0,21,67,59
10 Chr01 7047748 A G 60 60 1/1,0/0,0/1,0/1 17,20,138,137 0,2,31,27 1,0,29,30
11 Chr01 8720466 T C 247 222 0/0,1/1,0/1,0/1 99,38,139,139 3,0,136,108 0,1,127,94
12 Chr01 8720551 GTTTTTTTTG GTTTTTTTTG 165 153 0/0,1/1,0/1,0/1 55,31,135,135 0,0,92,73 0,1,73,79
13 Chr01 8728296 A T 139 153 0/0,1/1,0/1,0/1 26,53,160,160 2,2,66,69 0,6,88,59
14 Chr01 9151187 GTTTTTTTTTC GTTTTTTTTC 162 115 0/0,1/1,0/1,0/1 147,41,147,147 28,0,75,59 0,17,46,52
15 Chr01 9396322 G A 41 42 0/0,1/1,0/1,0/1 16,19,146,146 3,0,21,17 0,3,22,17
16 Chr01 10330228 GC GCATCCCGTAGCTGGCGCGCAAGATCCGAC 30 23 1/1,0/0,0/1,0/1 24,24,143,132 0,4,18,8 5,0,10,8
17 Chr01 10361552 A G 196 134 0/0,1/1,0/1,0/1 160,31,160,160 22,0,96,78 0,19,72,43
18 Chr01 10710975 G A 53 56 1/1,0/0,0/1,0/1 21,24,160,160 0,5,24,24 2,0,34,20
19 Chr01 10972432 G A 75 110 1/1,0/0,0/1,0/1 104,20,160,160 0,12,32,31 14,0,47,49
20 Chr01 12616225 G A 133 103 0/0,1/1,0/1,0/1 143,75,143,143 37,0,59,37 0,26,49,28
21 Chr01 12616497 T G 151 115 0/0,1/1,0/1,0/1 135,33,144,144 28,0,59,64 0,13,56,46
22 Chr01 12616756 T C 142 133 0/0,1/1,0/1,0/1 66,64,147,147 7,0,80,55 0,13,57,63
23 Chr01 12616805 A T 166 136 0/0,1/1,0/1,0/1 108,42,160,160 12,0,80,74 0,12,66,58
24 Chr01 12616990 GAAAAAAAAC GAAAAAAAAC 162 169 0/0,1/1,0/1,0/1 105,111,138,138 23,0,77,62 0,20,74,75
25 Chr01 12617272 CT CGT 256 200 0/0,1/1,0/1,0/1 160,83,160,160 32,0,115,109 0,28,85,87
26 Chr01 12617582 A G 154 170 0/0,1/1,0/1,0/1 76,137,160,160 17,0,59,78 0,30,84,56
27 Chr01 12617785 C T 207 171 0/0,1/1,0/1,0/1 160,64,160,160 31,0,94,82 0,18,93,60
28 Chr01 12617905 T C 155 161 0/0,1/1,0/1,0/1 98,108,160,160 21,0,68,66 0,20,74,67
29 Chr01 12617949 CAAAAAAAAG CAAAAAAAAG 143 148 0/0,1/1,0/1,0/1 97,96,160,160 22,1,65,55 0,27,70,51
30 Chr01 12617972 GTTTTTTTC GTTTTTTTC 142 153 0/0,1/1,0/1,0/1 98,132,160,160 24,0,66,52 0,26,70,57
31 Chr01 12618051 G A 155 129 0/0,1/1,0/1,0/1 138,62,139,139 26,0,70,59 0,18,68,43
32 Chr01 12618230 T C 110 99 0/0,1/1,0/1,0/1 71,47,144,144 11,0,53,46 0,11,52,36
33 Chr01 12618741 C T 124 137 0/0,1/1,0/1,0/1 61,84,160,160 14,0,61,49 0,12,70,55
34 Chr01 12618821 G A 151 147 0/0,1/1,0/1,0/1 82,73,152,152 13,0,67,71 0,13,73,61
35 Chr01 12618895 C G 182 172 0/0,1/1,0/1,0/1 132,110,160,160 26,0,79,77 0,25,78,69
36 Chr01 12619687 G T 175 188 0/0,1/1,0/1,0/1 81,106,160,160 20,0,89,66 1,15,87,85
37 Chr01 12619918 C G 85 73 0/0,1/1,0/1,0/1 74,42,160,160 14,0,42,29 0,12,22,39
38 Chr01 12620033 C T 62 76 0/0,1/1,0/1,0/1 22,68,141,141 7,0,37,18 0,12,26,38
39 Chr01 12620102 C T 70 79 0/0,1/1,0/1,0/1 26,64,145,145 6,0,41,23 0,12,25,42
40 Chr01 12620521 AT CC 108 79 0/0,1/1,0/1,0/1 134,18,139,139 35,0,35,38 0,9,47,23
41 Chr01 12620578 A C 67 63 0/0,1/1,0/1,0/1 92,41,160,160 24,0,20,23 0,10,32,21
42 Chr01 12620620 A G 40 47 0/0,1/1,0/1,0/1 34,44,151,151 11,0,14,15 0,9,21,17
43 Chr01 12620681 A G 76 82 0/0,1/1,0/1,0/1 66,53,160,160 18,0,31,27 0,9,42,31
```

Fig. 2: Output tabulated file

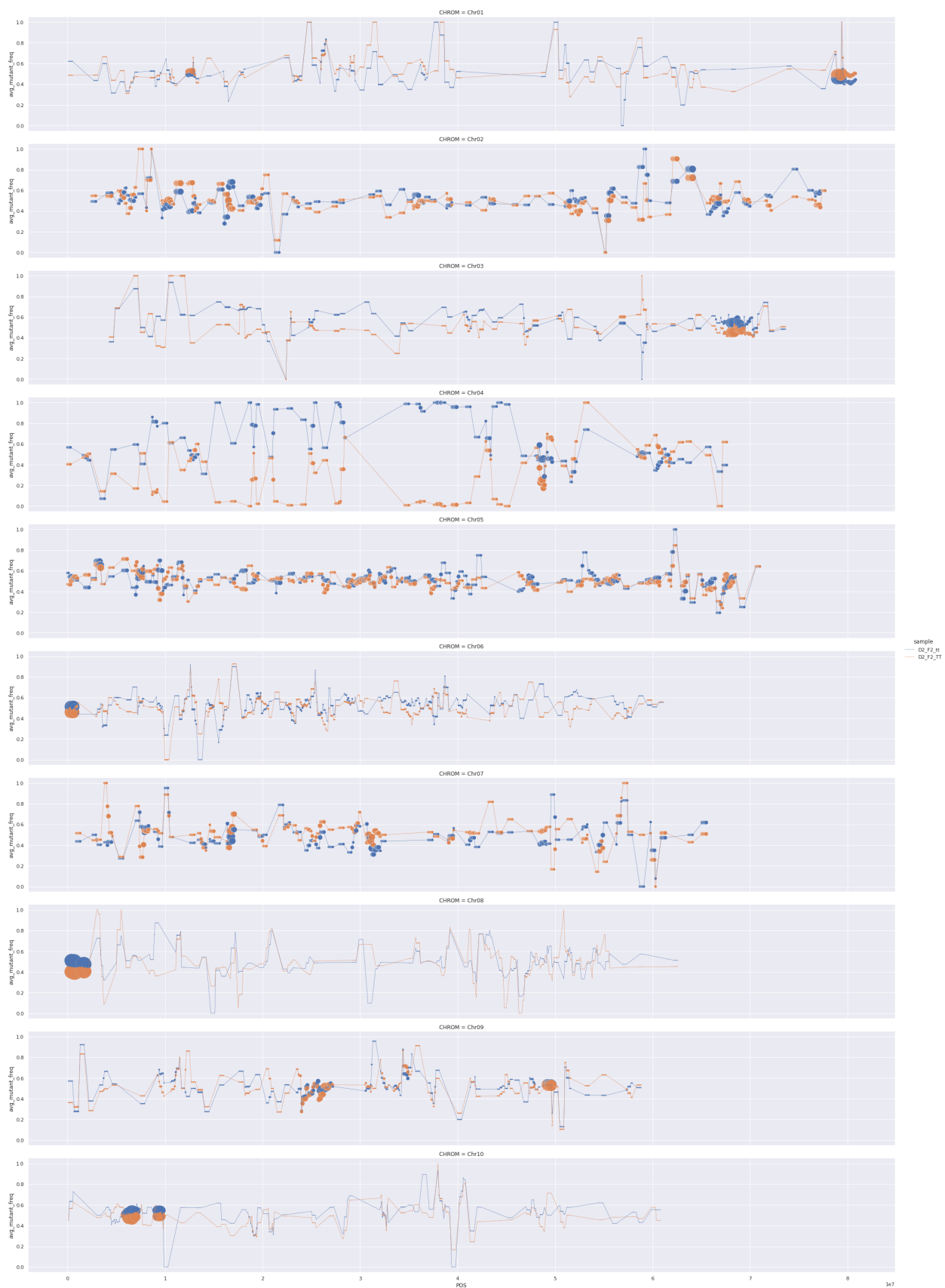


Fig. 3: Allele frequency plot with window\_size=500000 and step\_size=100000



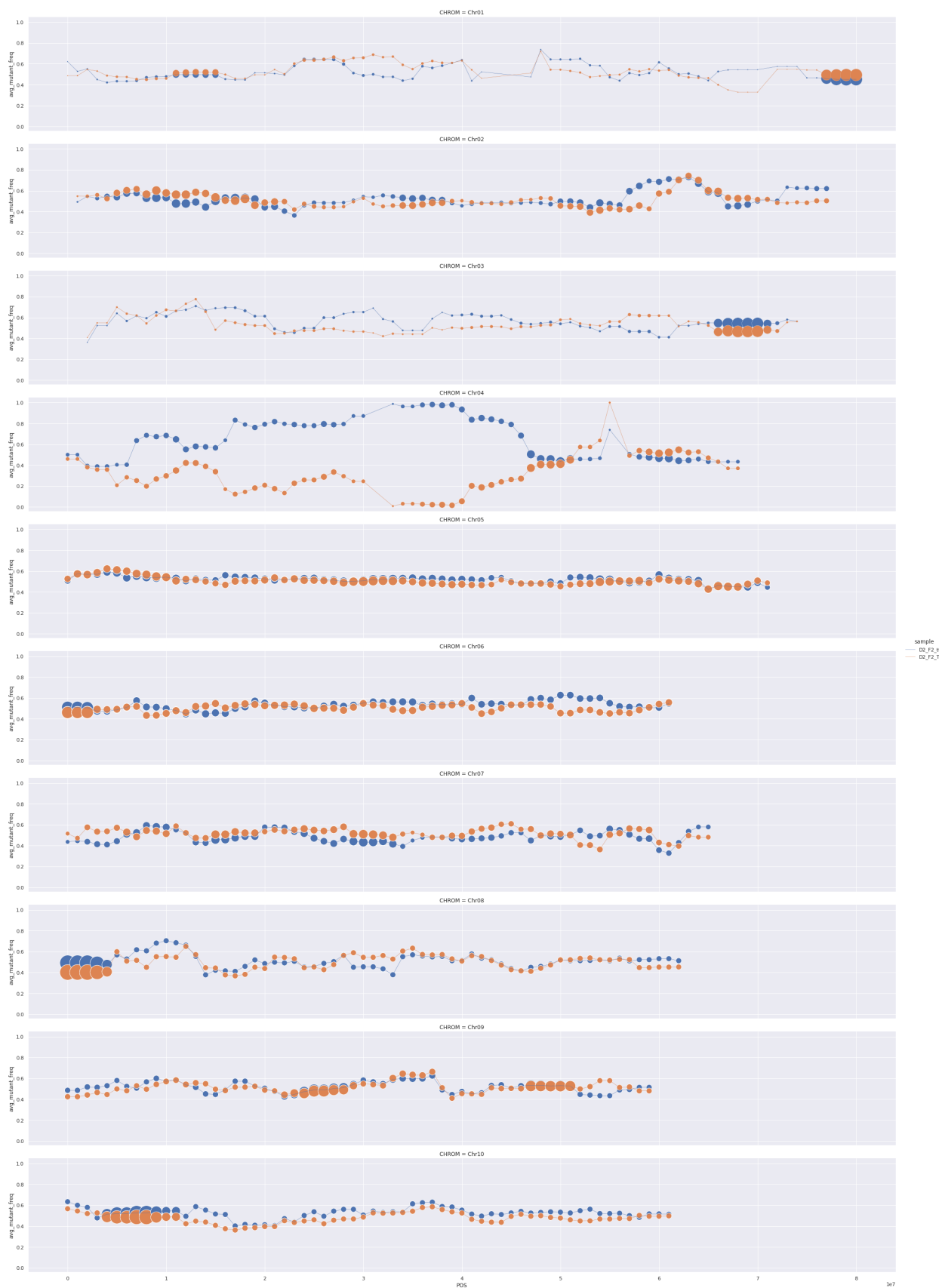


Fig. 4: Allele frequency plot with window\_size=5000000 and step\_size=1000000

This last plot using *window\_size=5000000* and *step\_size=1000000* clearly shows a variant present in Chr04. Knowing where artifacts like this one is located plays a major role in identifying and visualizing variants in an organism. The user can, in turn, use different tools, like the Integrative Genomics Viewer (IGV), for further analysis.

This culminates the tutorial.

## 5 References

### GitHub repositories:

- [pbgl/pbgl-qtl-bsa](#)
- [amora197/pbgl-qtl-bsa](#)
- [AnzaGhaffar/QTL-Snakemake-Workflow](#)