SNP MATRIX RETRIEVAL

Compiled by - Ayush

About

BACKGROUND

Arabidopsis Thaliana

Arabidopsis thaliana is a model plant species widely utilized in scientific research.
Family – ***Brassicaceae family*** (thale cress)

<u>CHARACTERISTICS:</u>

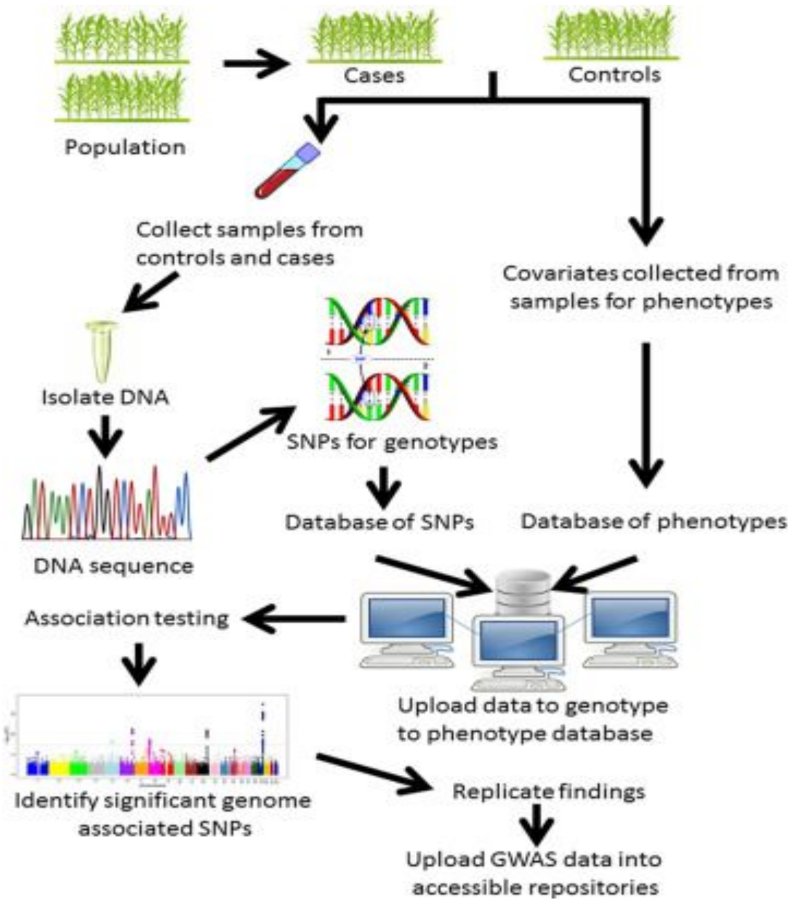
- Small Genome Size: has a relatively small genome compared to other plant species, with approximately 135 million base pairs. This compact genome makes it easier to study and analyze genetic variations.

- Short Life Cycle: The life cycle of Arabidopsis thaliana is relatively short, completing its growth and reproduction within a few weeks. This fast life cycle allows researchers to conduct experiments and observe multiple generations in a relatively short period.

- Genetic Tool: It possesses a wide array of genetic tools and resources, including a fully sequenced genome, mutant collections, and genetic transformation techniques.

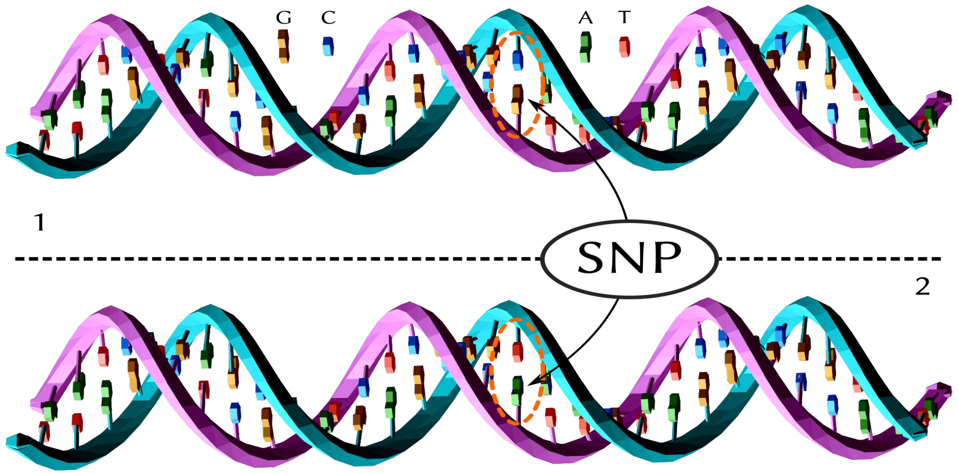
GWAS

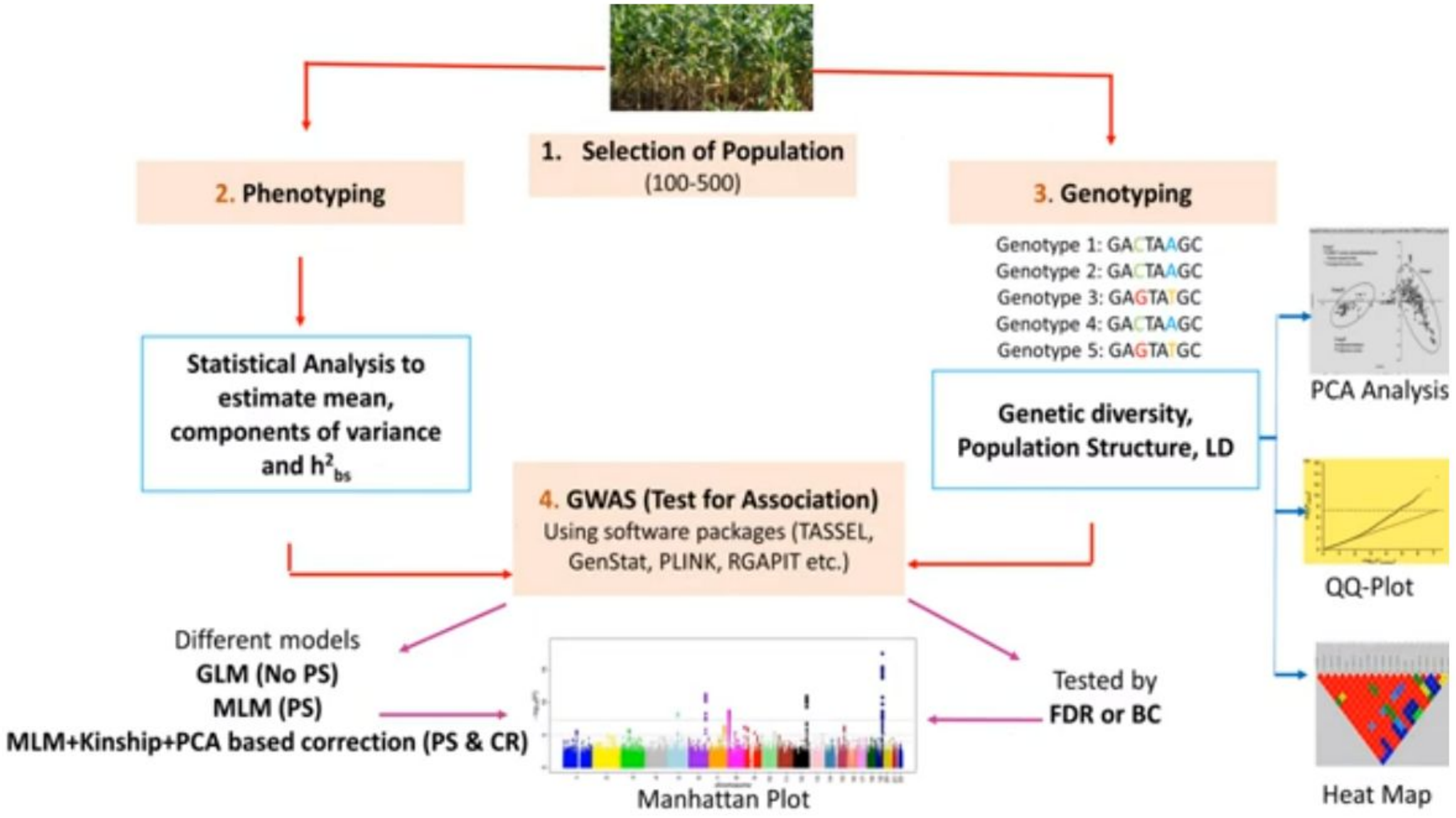
- GWAS analyzes millions of genetic variants across the entire genome.
- It focuses on common genetic variants with modest effects on the trait or disease of interest.
- GWAS reveals the polygenic nature of traits and identifies multiple loci associated with the phenotype.
- It provides insights into the genetic architecture and heritability of traits.
- Integration with functional genomics helps understand the biological mechanisms underlying genetic associations.
- GWAS has advanced our understanding of genetic contributions to complex traits and diseases.


Population
Cases
Controls
Collect samples from controls and cases
Covariates collected from samples for phenotypes
Isolate DNA
SNPs for genotypes
DNA sequence
Database of SNPs
Database of phenotypes
Association testing
Upload data to genotype to phenotype database
Identify significant genome associated SNPs
Replicate findings
Upload GWAS data into accessible repositories

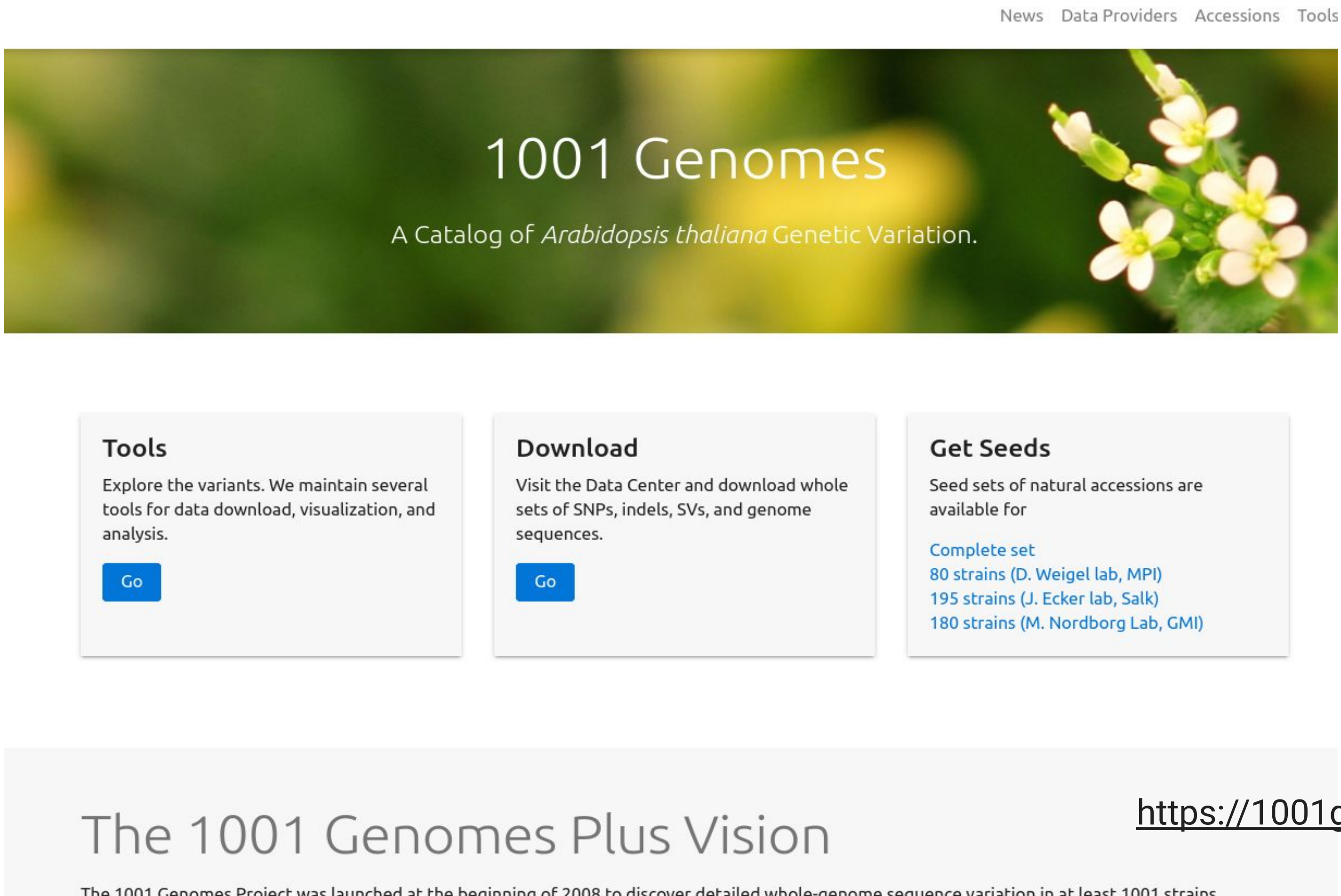
SNPs

- Genetic marker used in paper.
- Single nucleotide differences in a DNA sequence.
- SNPs present in the coding regions can affect protein function or gene expression ,leading to different phenotypes or traits.




1. Selection of Population
(100-500)
2. Phenotyping
3. Genotyping
Genotype 1: GACTAAGC
Genotype 2: GACTAAGC
Genotype 3: GAGTATGC
Genotype 4: GACTAAGC
Genotype 5: GAGTATGC
Statistical Analysis to estimate mean, components of variance and h^2_{bs}
Genetic diversity, Population Structure, LD
PCA Analysis
QQ-Plot
Heat Map
4. GWAS (Test for Association)
Using software packages (TASSEL, GenStat, PLINK, RGAPIT etc.)
Different models
GLM (No PS)
MLM (PS)
MLM+Kinship+PCA based correction (PS & CR)
Manhattan Plot
Tested by
FDR or BC

1001 Genome project



1001 Genomes

A Catalog of *Arabidopsis thaliana* Genetic Variation.

Tools

Explore the variants. We maintain several tools for data download, visualization, and analysis.

Go

Download

Visit the Data Center and download whole sets of SNPs, indels, SVs, and genome sequences.

Go

Get Seeds

Seed sets of natural accessions are available for

Complete set
80 strains (D. Weigel lab, MPI)
195 strains (J. Ecker lab, Salk)
180 strains (M. Nordborg Lab, GMI)

The 1001 Genomes Plus Vision

The 1001 Genomes Project was launched at the beginning of 2008 to discover detailed whole-genome sequence variation in at least 1001 strains

https://1001genomes.org/

DATASET

The dataset used for the study that investigated the global pattern of polymorphism in Arabidopsis thaliana through the analysis of 1,135 genomes is a comprehensive collection of genomic data from diverse populations of Arabidopsis thaliana.

Here are some key points about the dataset:

Sample Size: The dataset comprises genomic information from 1,135 individual Arabidopsis thaliana plants. This large sample size allows for a robust analysis and enhances the representation of genetic variation within the species.

Geographic Diversity: The dataset includes genomes from different geographic regions across the globe. By capturing genetic diversity from various populations, the dataset enables the examination of regional patterns of polymorphism and their potential correlation with environmental factors or historical events.

Genomic Variation Data: The dataset provides information about genetic variations, such as single nucleotide polymorphisms (SNPs), insertions, deletions, and structural variations, across the genomes of the 1,135 Arabidopsis thaliana plants. These variations serve as the basis for analyzing the global pattern of polymorphism in the species.

DATASET OVERVIEW

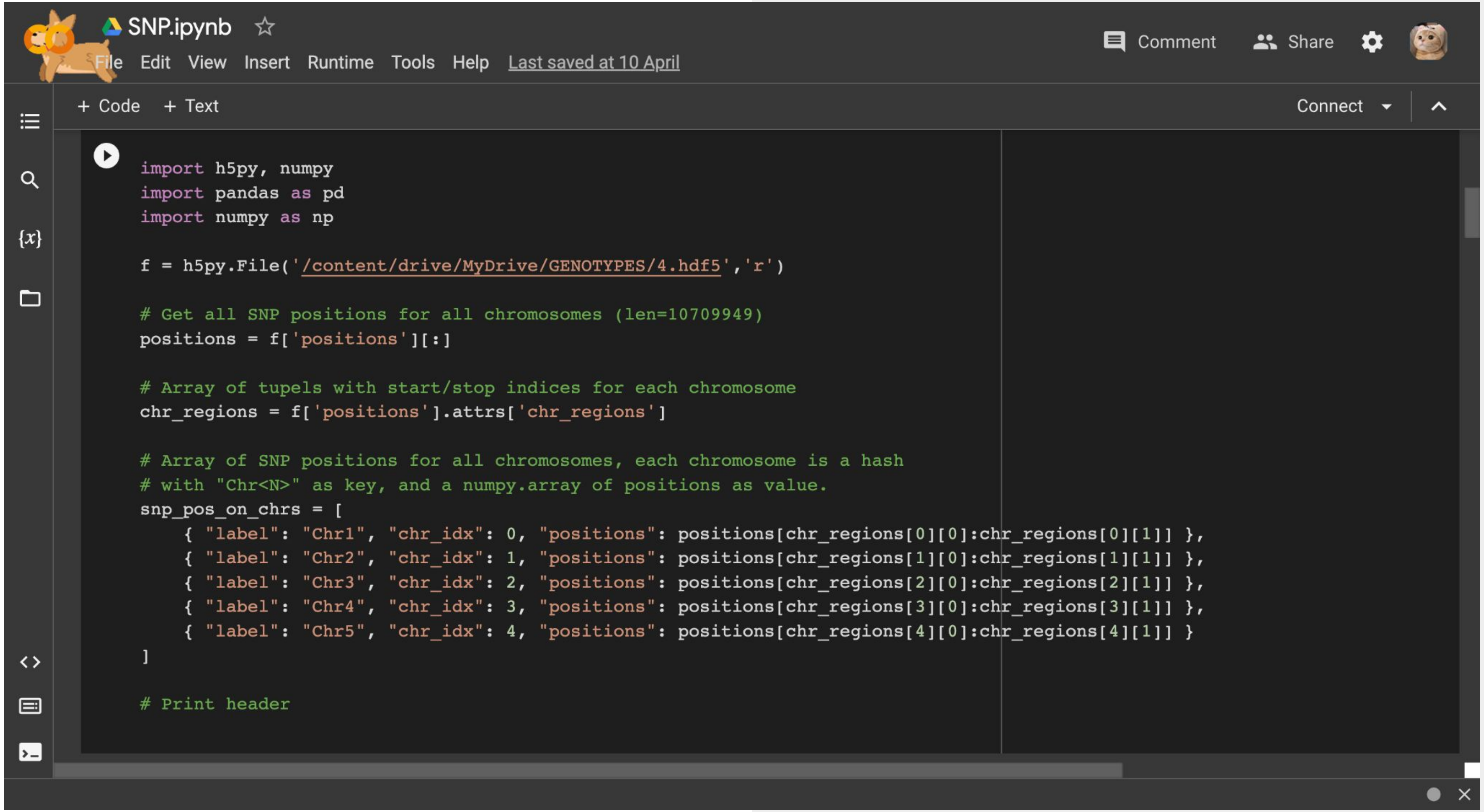
Dataset was in the HDF5 format (Hierarchical Data Format 5 ,is a data file format commonly used for storing and organizing large and complex datasets.) → **132 GB**

The keys in the dataset:

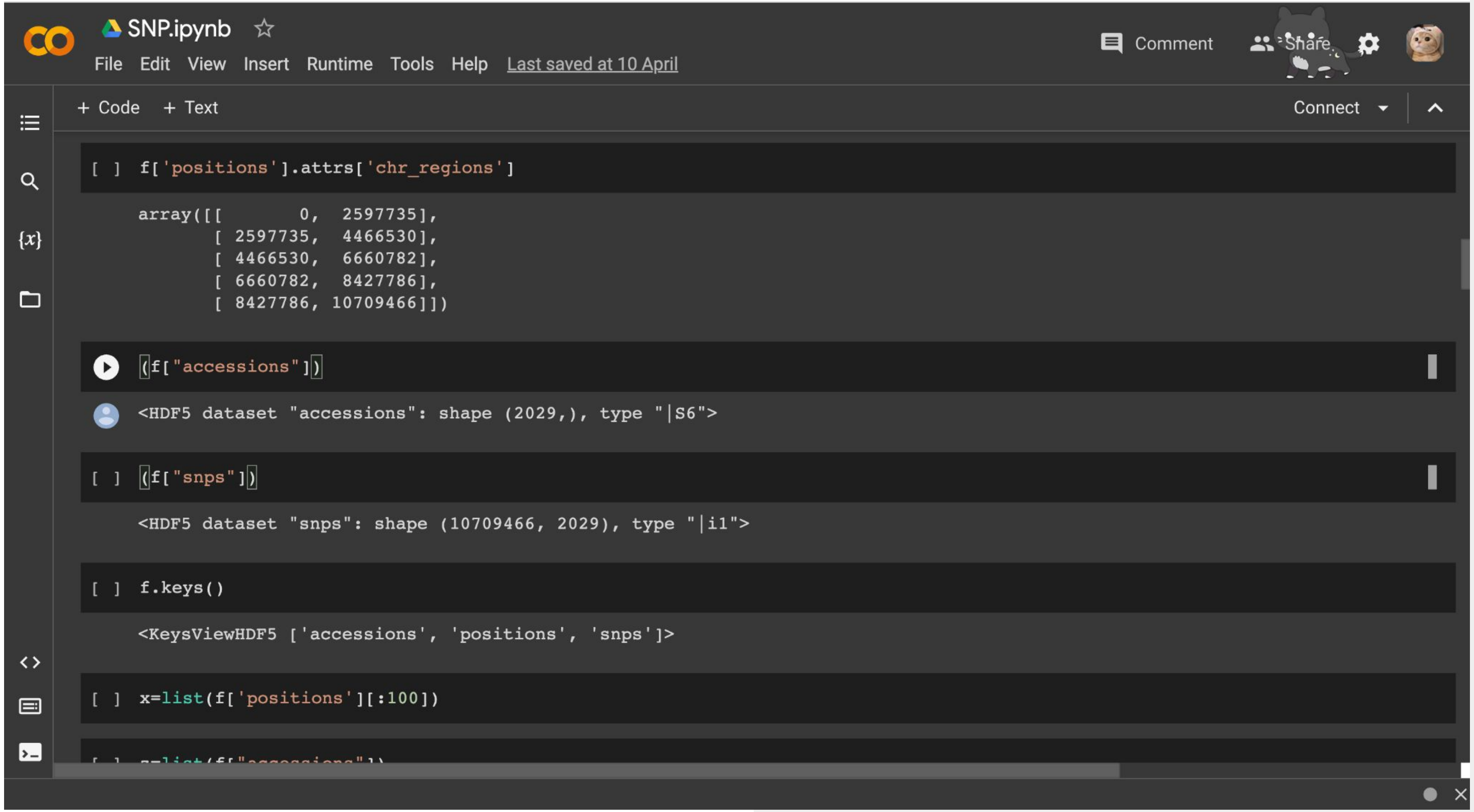
- “Accessions” → shape (2029 ,)
- “Positions” → shape (10709466 ,)
  Contains two attributes → [chromosome , chromosomal regions]

  Chromosomes = [1 , 2 , 3 , 4 , 5]

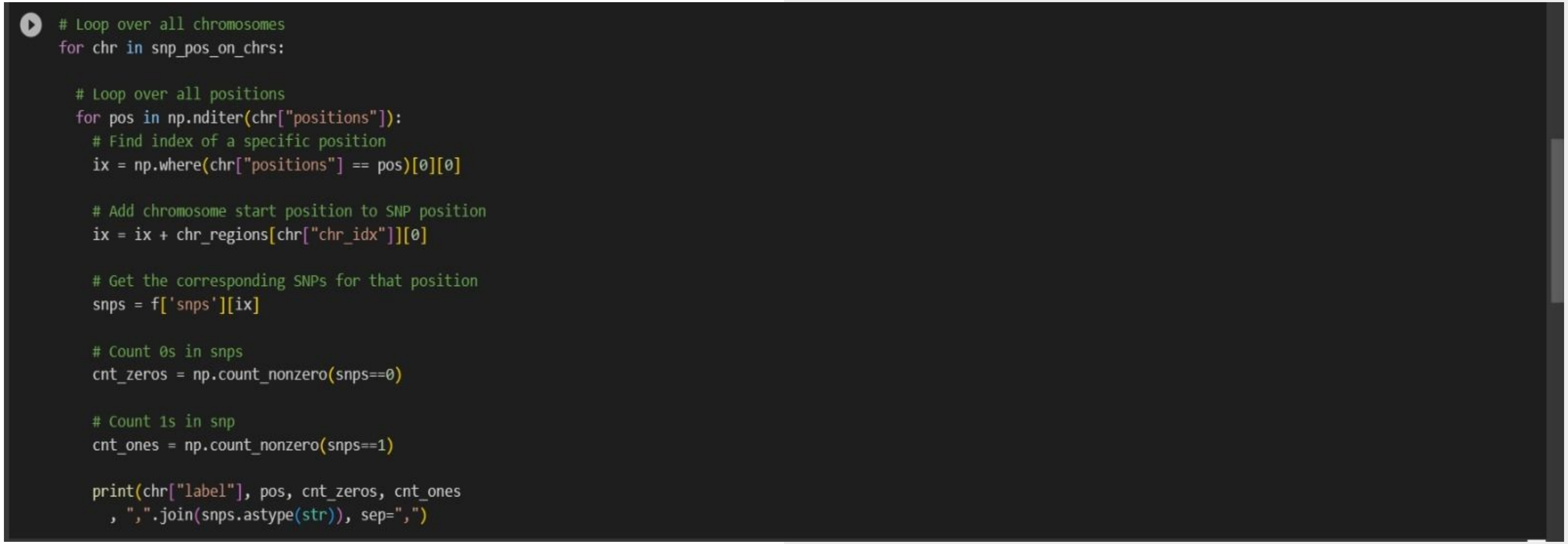
- “SNPS”
  Snp → shape (10709466 , 2029)


SNP.ipynb
File Edit View Insert Runtime Tools Help Last saved at 10 April
Comment Share
+ Code + Text
Connect

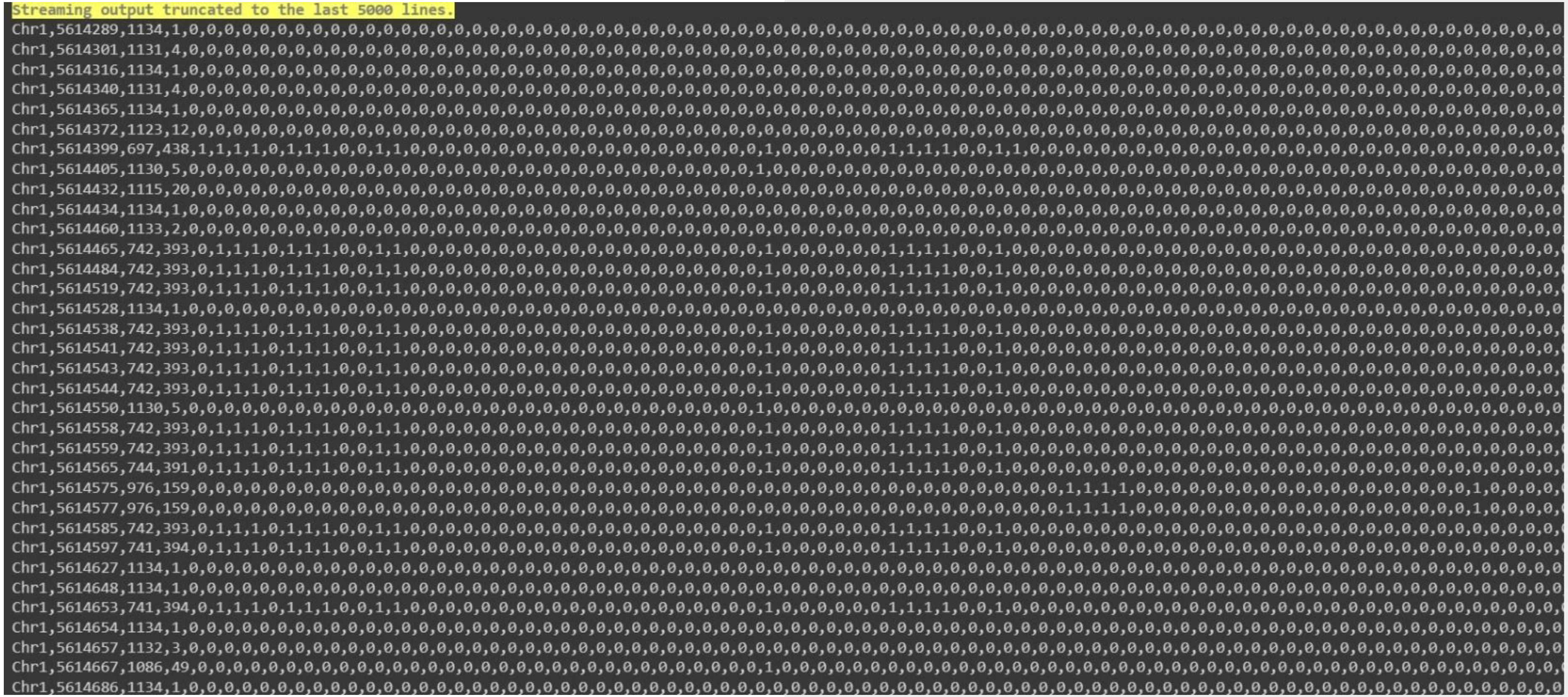
```python
import h5py, numpy
import pandas as pd
import numpy as np

f = h5py.File('/content/drive/MyDrive/GENOTYPES/4.hdf5','r')

# Get all SNP positions for all chromosomes (len=10709949)
positions = f['positions'][:]

# Array of tupels with start/stop indices for each chromosome
chr_regions = f['positions'].attrs['chr_regions']

# Array of SNP positions for all chromosomes, each chromosome is a hash
# with "Chr<N>" as key, and a numpy.array of positions as value.
snp_pos_on_chrs = [
    { "label": "Chr1", "chr_idx": 0, "positions": positions[chr_regions[0][0]:chr_regions[0][1]] },
    { "label": "Chr2", "chr_idx": 1, "positions": positions[chr_regions[1][0]:chr_regions[1][1]] },
    { "label": "Chr3", "chr_idx": 2, "positions": positions[chr_regions[2][0]:chr_regions[2][1]] },
    { "label": "Chr4", "chr_idx": 3, "positions": positions[chr_regions[3][0]:chr_regions[3][1]] },
    { "label": "Chr5", "chr_idx": 4, "positions": positions[chr_regions[4][0]:chr_regions[4][1]] }
]

# Print header
```


SNP.ipynb
File Edit View Insert Runtime Tools Help Last saved at 10 April
Comment
Share
+ Code + Text
Connect

```
[ ] f['positions'].attrs['chr_regions']

    array([[       0,  2597735],
           [ 2597735,  4466530],
           [ 4466530,  6660782],
           [ 6660782,  8427786],
           [ 8427786, 10709466]])

    (f["accessions"])

    <HDF5 dataset "accessions": shape (2029,), type "|S6">

[ ] (f["snps"])

    <HDF5 dataset "snps": shape (10709466, 2029), type "|i1">

[ ] f.keys()

    <KeysViewHDF5 ['accessions', 'positions', 'snps']>

[ ] x=list(f['positions'][:100])
```



```
# Loop over all chromosomes
for chr in snp_pos_on_chrs:

  # Loop over all positions
  for pos in np.nditer(chr["positions"]):
    # Find index of a specific position
    ix = np.where(chr["positions"] == pos)[0][0]

    # Add chromosome start position to SNP position
    ix = ix + chr_regions[chr["chr_idx"]][0]

    # Get the corresponding SNPs for that position
    snps = f['snps'][ix]

    # Count 0s in snps
    cnt_zeros = np.count_nonzero(snps==0)

    # Count 1s in snp
    cnt_ones = np.count_nonzero(snps==1)

    print(chr["label"], pos, cnt_zeros, cnt_ones
      , ",".join(snps.astype(str)), sep=",")
```

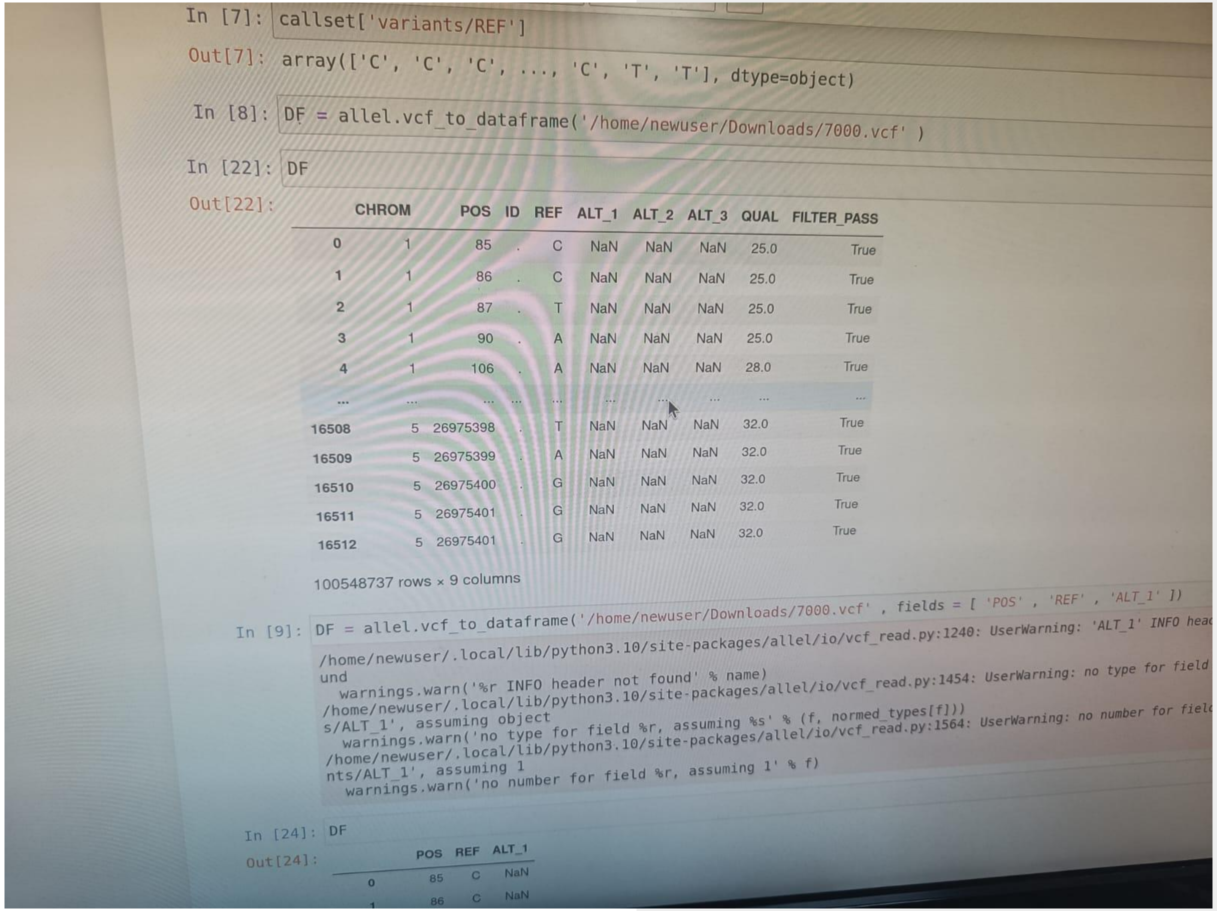


```
Streaming output truncated to the last 5000 lines.
Chr1,5614289,1134,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
Chr1,5614301,1131,4,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
Chr1,5614316,1134,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
Chr1,5614340,1131,4,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
Chr1,5614365,1134,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
Chr1,5614372,1123,12,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
Chr1,5614399,697,438,1,1,1,1,0,1,1,1,0,0,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,1,1,1,1,0,0,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
Chr1,5614405,1130,5,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
Chr1,5614432,1115,20,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
Chr1,5614434,1134,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
Chr1,5614460,1133,2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
Chr1,5614465,742,393,0,1,1,1,0,1,1,1,0,0,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,1,1,1,1,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
Chr1,5614484,742,393,0,1,1,1,0,1,1,1,0,0,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,1,1,1,1,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
Chr1,5614519,742,393,0,1,1,1,0,1,1,1,0,0,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,1,1,1,1,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
Chr1,5614528,1134,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
Chr1,5614538,742,393,0,1,1,1,0,1,1,1,0,0,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,1,1,1,1,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
Chr1,5614541,742,393,0,1,1,1,0,1,1,1,0,0,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,1,1,1,1,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
Chr1,5614543,742,393,0,1,1,1,0,1,1,1,0,0,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,1,1,1,1,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
Chr1,5614544,742,393,0,1,1,1,0,1,1,1,0,0,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,1,1,1,1,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
Chr1,5614550,1130,5,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
Chr1,5614558,742,393,0,1,1,1,0,1,1,1,0,0,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,1,1,1,1,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
Chr1,5614559,742,393,0,1,1,1,0,1,1,1,0,0,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,1,1,1,1,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
Chr1,5614565,744,391,0,1,1,1,0,1,1,1,0,0,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,1,1,1,1,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
Chr1,5614575,976,159,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,
Chr1,5614577,976,159,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,
Chr1,5614585,742,393,0,1,1,1,0,1,1,1,0,0,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,1,1,1,1,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
Chr1,5614597,741,394,0,1,1,1,0,1,1,1,0,0,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,1,1,1,1,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
Chr1,5614627,1134,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
Chr1,5614648,1134,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
Chr1,5614653,741,394,0,1,1,1,0,1,1,1,0,0,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,1,1,1,1,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
Chr1,5614654,1134,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
Chr1,5614657,1132,3,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
Chr1,5614667,1086,49,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
Chr1,5614686,1134,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
```

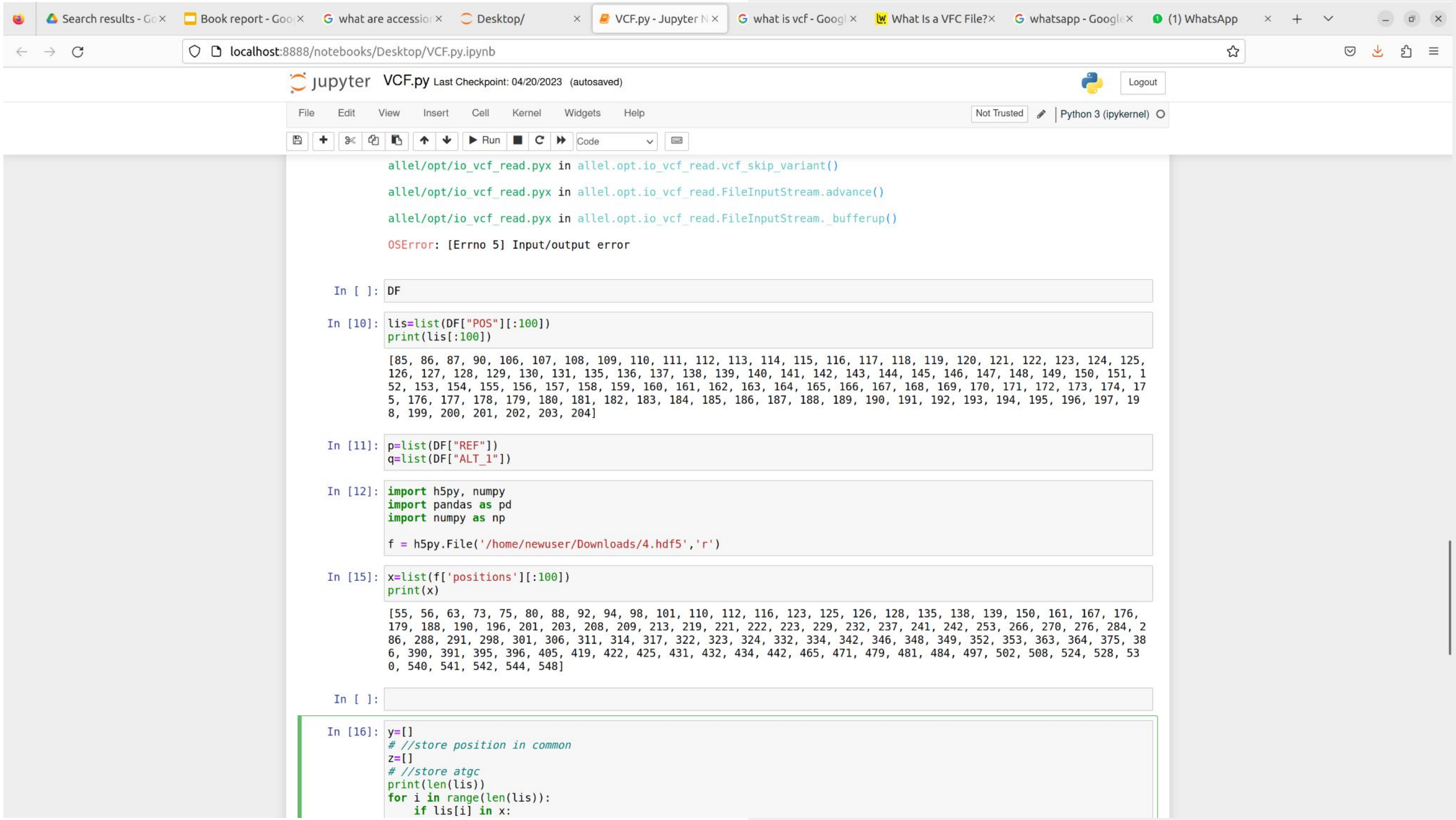
Libraries/Packages

- H5py
- Numpy
- Pandas

VCF (Variant Call Format) Approach



```
In [7]:  callset['variants/REF']

Out[7]:  array(['C', 'C', 'C', ..., 'C', 'T', 'T'], dtype=object)

In [8]:  DF = allel.vcf_to_dataframe('/home/newuser/Downloads/7000.vcf' )

In [22]: DF

Out[22]:
         CHROM       POS  ID  REF  ALT_1  ALT_2  ALT_3  QUAL  FILTER_PASS
    0        1        85   .    C    NaN    NaN    NaN  25.0         True
    1        1        86   .    C    NaN    NaN    NaN  25.0         True
    2        1        87   .    T    NaN    NaN    NaN  25.0         True
    3        1        90   .    A    NaN    NaN    NaN  25.0         True
    4        1       106   .    A    NaN    NaN    NaN  28.0         True
  ...      ...       ...  ...  ...    ...    ...    ...   ...          ...
16508        5  26975398   .    T    NaN    NaN    NaN  32.0         True
16509        5  26975399   .    A    NaN    NaN    NaN  32.0         True
16510        5  26975400   .    G    NaN    NaN    NaN  32.0         True
16511        5  26975401   .    G    NaN    NaN    NaN  32.0         True
16512        5  26975401   .    G    NaN    NaN    NaN  32.0         True

100548737 rows × 9 columns

In [9]:  DF = allel.vcf_to_dataframe('/home/newuser/Downloads/7000.vcf' , fields = [ 'POS' , 'REF' , 'ALT_1' ])

/home/newuser/.local/lib/python3.10/site-packages/allel/io/vcf_read.py:1240: UserWarning: 'ALT_1' INFO head
und
  warnings.warn('%r INFO header not found' % name)
/home/newuser/.local/lib/python3.10/site-packages/allel/io/vcf_read.py:1454: UserWarning: no type for field
s/ALT_1', assuming object
  warnings.warn('no type for field %r, assuming %s' % (f, normed_types[f]))
/home/newuser/.local/lib/python3.10/site-packages/allel/io/vcf_read.py:1564: UserWarning: no number for fiel
nts/ALT_1', assuming 1
  warnings.warn('no number for field %r, assuming 1' % f)

In [24]: DF

Out[24]:
       POS  REF  ALT_1
    0   85    C    NaN
    1   86    C    NaN
```



```
allel/opt/io_vcf_read.pyx in allel.opt.io_vcf_read.vcf_skip_variant()

allel/opt/io_vcf_read.pyx in allel.opt.io_vcf_read.FileInputStream.advance()

allel/opt/io_vcf_read.pyx in allel.opt.io_vcf_read.FileInputStream._bufferup()

OSError: [Errno 5] Input/output error
```

```
In [ ]: DF
```

```
In [10]: lis=list(DF["POS"][:100])
         print(lis[:100])
```

```
[85, 86, 87, 90, 106, 107, 108, 109, 110, 111, 112, 113, 114, 115, 116, 117, 118, 119, 120, 121, 122, 123, 124, 125,
126, 127, 128, 129, 130, 131, 135, 136, 137, 138, 139, 140, 141, 142, 143, 144, 145, 146, 147, 148, 149, 150, 151, 1
52, 153, 154, 155, 156, 157, 158, 159, 160, 161, 162, 163, 164, 165, 166, 167, 168, 169, 170, 171, 172, 173, 174, 17
5, 176, 177, 178, 179, 180, 181, 182, 183, 184, 185, 186, 187, 188, 189, 190, 191, 192, 193, 194, 195, 196, 197, 19
8, 199, 200, 201, 202, 203, 204]
```

```
In [11]: p=list(DF["REF"])
         q=list(DF["ALT_1"])
```

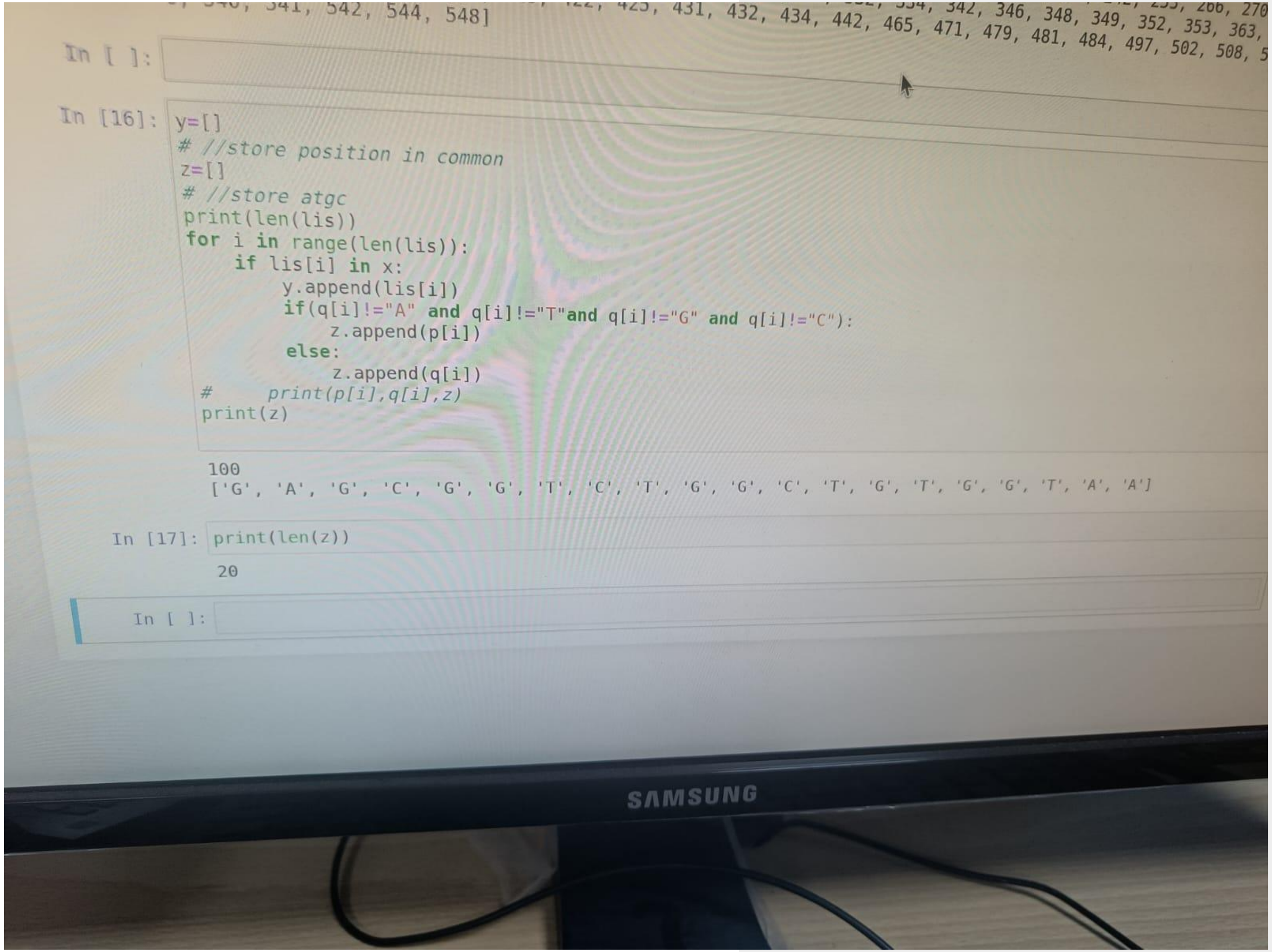
```
In [12]: import h5py, numpy
         import pandas as pd
         import numpy as np

         f = h5py.File('/home/newuser/Downloads/4.hdf5','r')
```

```
In [15]: x=list(f['positions'][:100])
         print(x)
```

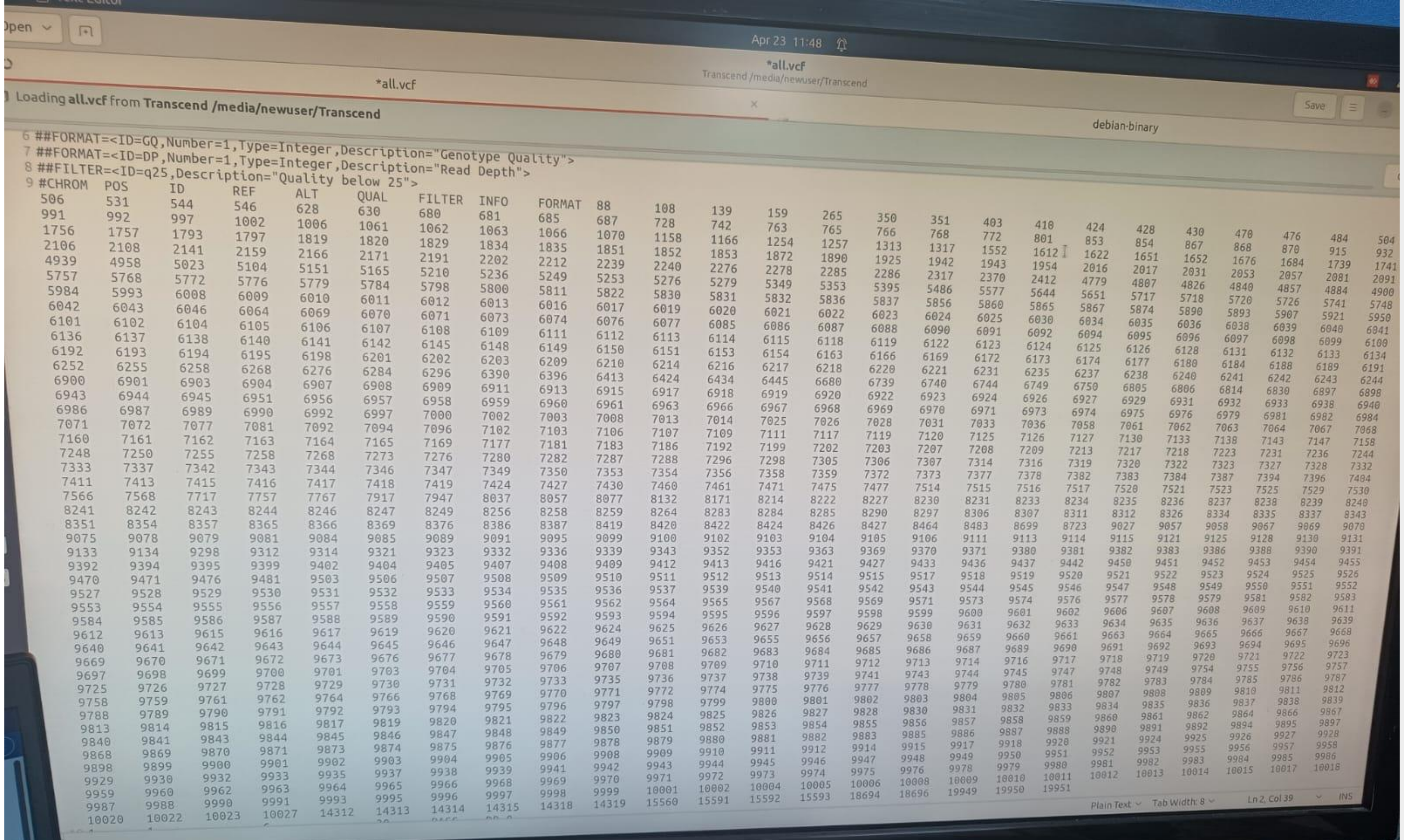
```
[55, 56, 63, 73, 75, 80, 88, 92, 94, 98, 101, 110, 112, 116, 123, 125, 126, 128, 135, 138, 139, 150, 161, 167, 176,
179, 188, 190, 196, 201, 203, 208, 209, 213, 219, 221, 222, 223, 229, 232, 237, 241, 242, 253, 266, 270, 276, 284, 2
86, 288, 291, 298, 301, 306, 311, 314, 317, 322, 323, 324, 332, 334, 342, 346, 348, 349, 352, 353, 363, 364, 375, 38
6, 390, 391, 395, 396, 405, 419, 422, 425, 431, 432, 434, 442, 465, 471, 479, 481, 484, 497, 502, 508, 524, 528, 53
0, 540, 541, 542, 544, 548]
```

```
In [ ]:
```

```
In [16]: y=[]
         # //store position in common
         z=[]
         # //store atgc
         print(len(lis))
         for i in range(len(lis)):
             if lis[i] in x:
```



```
541, 542, 544, 548]                    425, 431, 432, 434, 442, 465, 471, 479, 481, 484, 497, 502, 508, 5
                                                     342, 346, 348, 349, 352, 353, 363,
                                                                                  266, 270

In [ ]:

In [16]: y=[]
         # //store position in common
         z=[]
         # //store atgc
         print(len(lis))
         for i in range(len(lis)):
             if lis[i] in x:
                 y.append(lis[i])
                 if(q[i]!="A" and q[i]!="T"and q[i]!="G" and q[i]!="C"):
                     z.append(p[i])
                 else:
                     z.append(q[i])
         #     print(p[i],q[i],z)
         print(z)

         100
         ['G', 'A', 'G', 'C', 'G', 'G', 'T', 'C', 'T', 'G', 'G', 'C', 'T', 'G', 'T', 'G', 'G', 'T', 'A', 'A']

In [17]: print(len(z))

         20

In [ ]:
```



Loading **all.vcf** from **Transcend /media/newuser/Transcend**

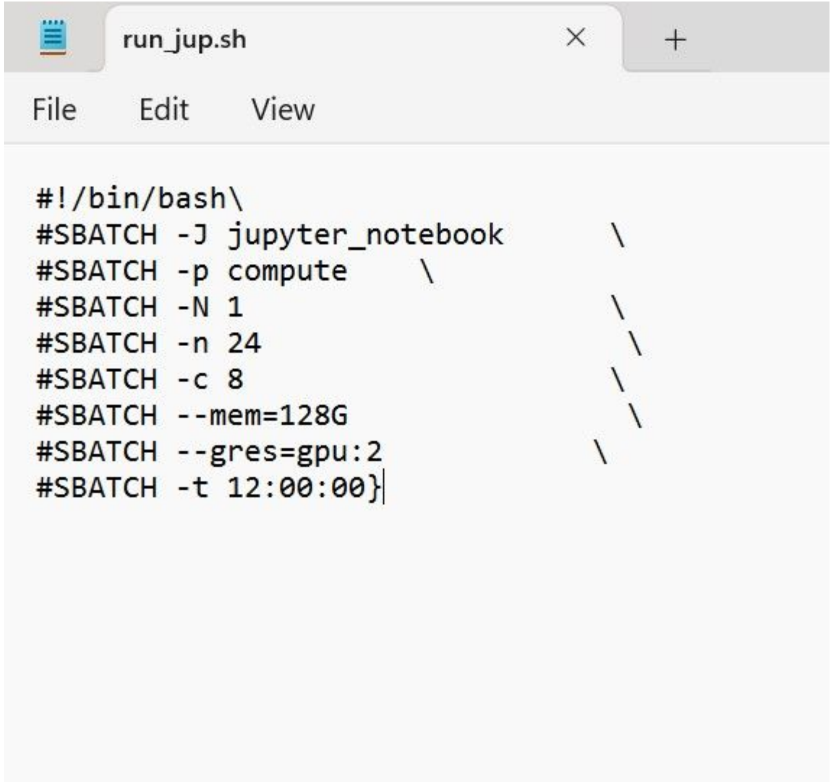
```
##FORMAT=<ID=GQ,Number=1,Type=Integer,Description="Genotype Quality">
##FORMAT=<ID=DP,Number=1,Type=Integer,Description="Read Depth">
##FILTER=<ID=q25,Description="Quality below 25">
#CHROM  POS     ID      REF     ALT     QUAL    FILTER  INFO    FORMAT  88      108     139     159     265     350     351     403     410     424     428     430     470     476     484     504
506     531     544     546     628     630     680     681     685     687     728     742     763     765     766     768     772     801     853     854     867     868     870     915     932
991     992     997     1002    1006    1061    1062    1063    1066    1070    1158    1166    1254    1257    1313    1317    1552    1612    1622    1651    1652    1676    1684    1739    1741
1756    1757    1793    1797    1819    1820    1829    1834    1835    1851    1852    1853    1872    1890    1925    1942    1943    1954    2016    2017    2031    2053    2057    2081    2091
2106    2108    2141    2159    2166    2171    2191    2202    2212    2239    2240    2276    2278    2285    2286    2317    2370    2412    4779    4807    4826    4840    4857    4884    4900
4939    4958    5023    5104    5151    5165    5210    5236    5249    5253    5276    5279    5349    5353    5395    5486    5577    5644    5651    5717    5718    5720    5726    5741    5748
5757    5768    5772    5776    5779    5784    5798    5800    5811    5822    5830    5831    5832    5836    5837    5856    5860    5865    5867    5874    5890    5893    5907    5921    5950
5984    5993    6008    6009    6010    6011    6012    6013    6016    6017    6019    6020    6021    6022    6023    6024    6025    6030    6034    6035    6036    6038    6039    6040    6041
6042    6043    6046    6064    6069    6070    6071    6073    6074    6076    6077    6085    6086    6087    6088    6090    6091    6092    6094    6095    6096    6097    6098    6099    6100
6101    6102    6104    6105    6106    6107    6108    6109    6111    6112    6113    6114    6115    6118    6119    6122    6123    6124    6125    6126    6128    6131    6132    6133    6134
6136    6137    6138    6140    6141    6142    6145    6148    6149    6150    6151    6153    6154    6163    6166    6169    6172    6173    6174    6177    6180    6184    6188    6189    6191
6192    6193    6194    6195    6198    6201    6202    6203    6209    6210    6214    6216    6217    6218    6220    6221    6231    6235    6237    6238    6240    6241    6242    6243    6244
6252    6255    6258    6268    6276    6284    6296    6390    6396    6413    6424    6434    6445    6680    6739    6740    6744    6749    6750    6805    6806    6814    6830    6897    6898
6900    6901    6903    6904    6907    6908    6909    6911    6913    6915    6917    6918    6919    6920    6922    6923    6924    6926    6927    6929    6931    6932    6933    6938    6940
6943    6944    6945    6951    6956    6957    6958    6959    6960    6961    6963    6966    6967    6968    6969    6970    6971    6973    6974    6975    6976    6979    6981    6982    6984
6986    6987    6989    6990    6992    6997    7000    7002    7003    7008    7013    7014    7025    7026    7028    7031    7033    7036    7058    7061    7062    7063    7064    7067    7068
7071    7072    7077    7081    7092    7094    7096    7102    7103    7106    7107    7109    7111    7117    7119    7120    7125    7126    7127    7130    7133    7138    7143    7147    7158
7160    7161    7162    7163    7164    7165    7169    7177    7181    7183    7186    7192    7199    7202    7203    7207    7208    7209    7213    7217    7218    7223    7231    7236    7244
7248    7250    7255    7258    7268    7273    7276    7280    7282    7287    7288    7296    7298    7305    7306    7307    7314    7316    7319    7320    7322    7323    7327    7328    7332
7333    7337    7342    7343    7344    7346    7347    7349    7350    7353    7354    7356    7358    7359    7372    7373    7377    7378    7382    7383    7384    7387    7394    7396    7404
7411    7413    7415    7416    7417    7418    7419    7424    7427    7430    7460    7461    7471    7475    7477    7514    7515    7516    7517    7520    7521    7523    7525    7529    7530
7566    7568    7717    7757    7767    7917    7947    8037    8057    8077    8132    8171    8214    8222    8227    8230    8231    8233    8234    8235    8236    8237    8238    8239    8240
8241    8242    8243    8244    8246    8247    8249    8256    8258    8259    8264    8283    8284    8285    8290    8297    8306    8307    8311    8312    8326    8334    8335    8337    8343
8351    8354    8357    8365    8366    8369    8376    8386    8387    8419    8420    8422    8424    8426    8427    8464    8483    8699    8723    9027    9057    9058    9067    9069    9070
9075    9078    9079    9081    9084    9085    9089    9091    9095    9099    9100    9102    9103    9104    9105    9106    9111    9113    9114    9115    9121    9125    9128    9130    9131
9133    9134    9298    9312    9314    9321    9323    9332    9336    9339    9343    9352    9353    9363    9369    9370    9371    9380    9381    9382    9383    9386    9388    9390    9391
9392    9394    9395    9399    9402    9404    9405    9407    9408    9409    9412    9413    9416    9421    9427    9433    9436    9437    9442    9450    9451    9452    9453    9454    9455
9470    9471    9476    9481    9503    9506    9507    9508    9509    9510    9511    9512    9513    9514    9515    9517    9518    9519    9520    9521    9522    9523    9524    9525    9526
9527    9528    9529    9530    9531    9532    9533    9534    9535    9536    9537    9539    9540    9541    9542    9543    9544    9545    9546    9547    9548    9549    9550    9551    9552
9553    9554    9555    9556    9557    9558    9559    9560    9561    9562    9564    9565    9567    9568    9569    9571    9573    9574    9576    9577    9578    9579    9581    9582    9583
9584    9585    9586    9587    9588    9589    9590    9591    9592    9593    9594    9595    9596    9597    9598    9599    9600    9601    9602    9606    9607    9608    9609    9610    9611
9612    9613    9615    9616    9617    9619    9620    9621    9622    9624    9625    9626    9627    9628    9629    9630    9631    9632    9633    9634    9635    9636    9637    9638    9639
9640    9641    9642    9643    9644    9645    9646    9647    9648    9649    9651    9653    9655    9656    9657    9658    9659    9660    9661    9663    9664    9665    9666    9667    9668
9669    9670    9671    9672    9673    9676    9677    9678    9679    9680    9681    9682    9683    9684    9685    9686    9687    9689    9690    9691    9692    9693    9694    9695    9696
9697    9698    9699    9700    9701    9703    9704    9705    9706    9707    9708    9709    9710    9711    9712    9713    9714    9716    9717    9718    9719    9720    9721    9722    9723
9725    9726    9727    9728    9729    9730    9731    9732    9733    9735    9736    9737    9738    9739    9741    9743    9744    9745    9747    9748    9749    9754    9755    9756    9757
9758    9759    9761    9762    9764    9766    9768    9769    9770    9771    9772    9774    9775    9776    9777    9778    9779    9780    9781    9782    9783    9784    9785    9786    9787
9788    9789    9790    9791    9792    9793    9794    9795    9796    9797    9798    9799    9800    9801    9802    9803    9804    9805    9806    9807    9808    9809    9810    9811    9812
9813    9814    9815    9816    9817    9819    9820    9821    9822    9823    9824    9825    9826    9827    9828    9830    9831    9832    9833    9834    9835    9836    9837    9838    9839
9840    9841    9843    9844    9845    9846    9847    9848    9849    9850    9851    9852    9853    9854    9855    9856    9857    9858    9859    9860    9861    9862    9864    9866    9867
9868    9869    9870    9871    9873    9874    9875    9876    9877    9878    9879    9880    9881    9882    9883    9885    9886    9887    9888    9890    9891    9892    9894    9895    9897
9898    9899    9900    9901    9902    9903    9904    9905    9906    9908    9909    9910    9911    9912    9914    9915    9917    9918    9920    9921    9924    9925    9926    9927    9928
9929    9930    9932    9933    9935    9937    9938    9939    9941    9942    9943    9944    9945    9946    9947    9948    9949    9950    9951    9952    9953    9955    9956    9957    9958
9959    9960    9962    9963    9964    9965    9966    9968    9969    9970    9971    9972    9973    9974    9975    9976    9978    9979    9980    9981    9982    9983    9984    9985    9986
9987    9988    9990    9991    9993    9995    9996    9997    9998    9999    10001   10002   10004   10005   10006   10008   10009   10010   10011   10012   10013   10014   10015   10017   10018
10020   10022   10023   10027   14312   14313   14314   14315   14318   14319   15560   15591   15592   15593   18694   18696   19949   19950   19951
```

HPC SERVER

Work Done on HPC –

- ❑ Learning to work on HPC using slurm
- ❑ Setting up suitable environment to run file on slurm
- ❑ Setting up Jupyter notebook as per requirement of the project to run on slurm
- ❑ Testing the environment

Creating S-batch File According to our needs.



```
#!/bin/bash\
#SBATCH -J jupyter_notebook      \
#SBATCH -p compute    \
#SBATCH -N 1                     \
#SBATCH -n 24                     \
#SBATCH -c 8                     \
#SBATCH --mem=128G                \
#SBATCH --gres=gpu:2            \
#SBATCH -t 12:00:00}
```

Contributions:-

Ayush

- ➔ Written code for Extracting Data from VCF file as per desired format .
- ➔ Worked on HPC server and learned to use slurm
- ➔ Setting up Jupyter Environment and solving errors while doing so.
- ➔ Created Demonstration using small Dataset of how the code on actual Data will work.
- ➔ Analyzing and Detecting the right Approach when discussing with my Professors.Giving updates to Professor.
- ➔ Knowing about the Dataset in hands to work with it and code accordingly