Exploratory Data Analysis: Sassafras

Capstone

Samantha Harper

11 November 2024

Contents

Background

Research Question

The research question is: Can we use machine learning algorithms to mine SNPs to find gene or gene regions of interest between natural cultivars (strains) of Sassafras?

Hypothesis

Hypothesis: Underlying genes, as identified by SNPs, in Sassafras are influenced by environmental factors because environmental pressure can cause mutations to persist in a population that is unique to each area.

Prediction

Prediction: Populations of Sassafras that are under high environmental pressure are more likely to have many predictive SNPs due to evolutionary influences.

Methods

Data set and format

The data set contains genetic data from *Sassafras tzumu*. Data was collected from 106 individuals across 13 populations in China. The genetic material was sequenced to create the data set, which identified 1,862 single nucleotide polymorphisms (SNPs) and is presented in a Variant Call Format (VCF) file. A VCF file contains three regions, the meta region, the fix region, and the gt region. Each contains different information about the genetic data.

Cleaning and data preparation

In order to access and manipulate the VCF file, several additional R packages were used. The vcfR package allows for loading and manipulation of the data, while the adegenet package was helpful in creating several of the visualizations. Going forward it may be useful to use ChromR objects as well.

Exploratory Data Analysis

My approach to exploratory data analysis was to engage with the various R objects designed to hold genetic data and to explore the standard graphs and visualizations for those objects. Since the data is going to require some changes to be compatable with a machine learning model, I focused on understanding the big picture of the SNPs. The graphs below show the distribution of the SNPs as well as the distribution of the alleles. I also wanted to highlight any missing data that could have an impact on the ML model going forward.

Results

```
if (!requireNamespace("vcfR", quietly = TRUE)) {
  install.packages("vcfR", verbose = FALSE)
}
library(vcfR)
```

```
##
##    *****       ***   vcfR   ***       *****
##    This is vcfR 1.15.0
##      browseVignettes('vcfR') # Documentation
##      citation('vcfR') # Citation
##    *****       *****      *****       *****
```

```
vcf <- read.vcfR( "Data/SNP.vcf", verbose = FALSE )
#examine VCF Object
vcf
```

```
## ***** Object of Class vcfR *****
## 106 samples
## 9177 CHROMs
## 11,862 variants
## Object size: 14.5 Mb
## 0 percent missing data
## *****        *****         *****
```

```
if (!requireNamespace("adegenet", quietly = TRUE)) {
  install.packages("adegenet")
}
library(adegenet)
```

```
## Loading required package: ade4
```

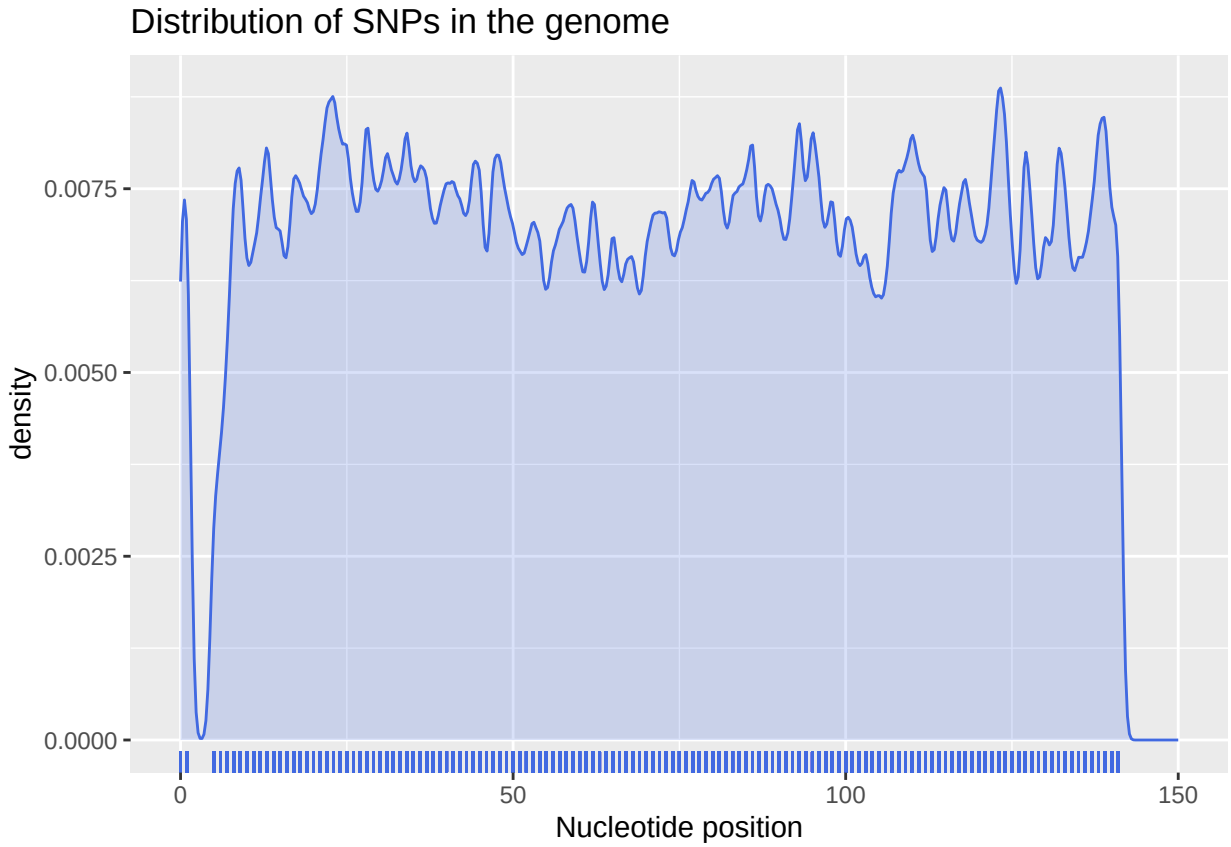
```
##
##    /// adegenet 2.1.10 is loaded ////////////
##
##    > overview: '?adegenet'
##    > tutorials/doc/questions: 'adegenetWeb()'
##    > bug reports/feature requests: adegenetIssues()
```

A genlight object is created to store and manipulate genetic data.

```
#Create a genlight object
x <- vcfR2genlight(vcf)
x
```

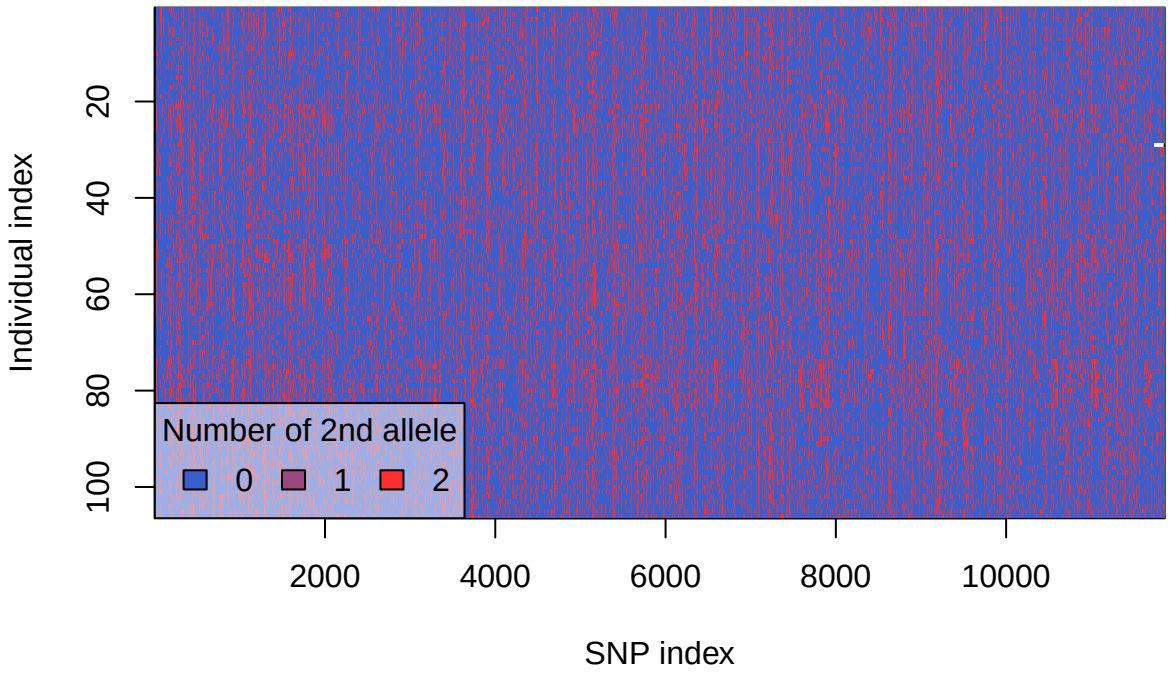
```
##  /// GENLIGHT OBJECT /////////
##
##  // 106 genotypes,  11,862 binary SNPs, size: 2.1 Mb
##  48588 (3.86 %) missing data
##
##  // Basic content
##    @gen: list of 106 SNPbin
##
##  // Optional content
##    @ind.names:  106 individual labels
##    @loc.names:  11862 locus labels
##    @chromosome: factor storing chromosomes of the SNPs
##    @position: integer storing positions of the SNPs
##    @other: a list containing: elements without names
```

```
snpposi.plot(position(x), genome.size=150, codon=FALSE)
```



Looking at the disribution of the SNPs throughout the genome, we can see that they are fairly evenly spaced.

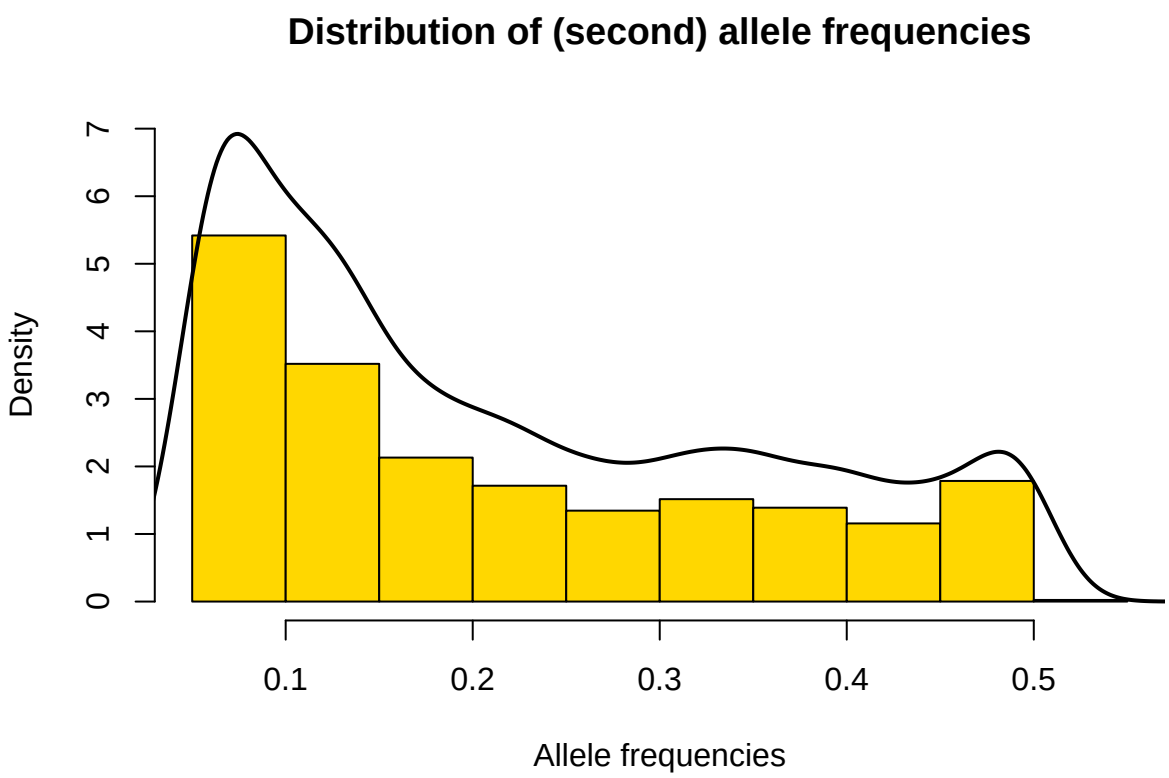
```
x.dist <- dist(x)
glPlot(x)
```



Here we can see that there are up to two alternative bases and how those are distributed throughout the

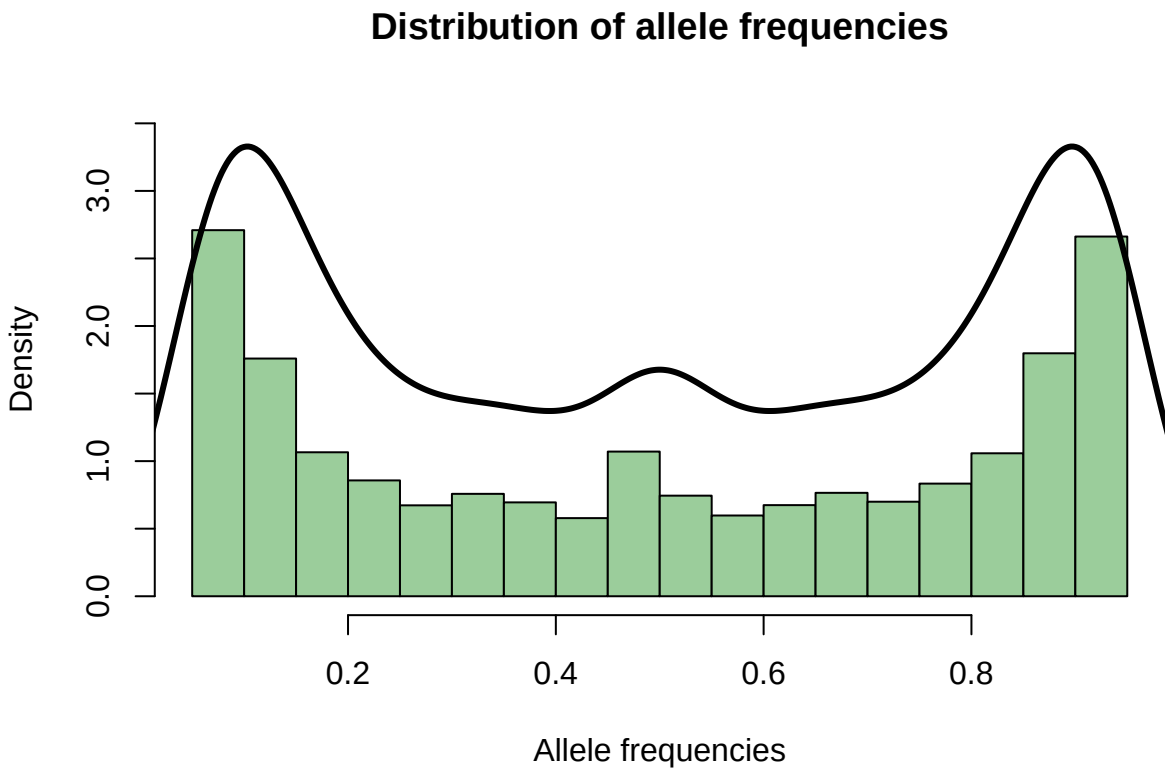
samples. We can see some light banding in the data suggesting that different populations may have differing proportions of alternative alleles.

```
myFreq <- glMean(x)
hist(myFreq, proba=TRUE, col="gold", xlab="Allele frequencies",
main="Distribution of (second) allele frequencies", ylim=c(0,7))
temp <- density(myFreq)
lines(temp$x, temp$y*1.5,lwd=2)
```



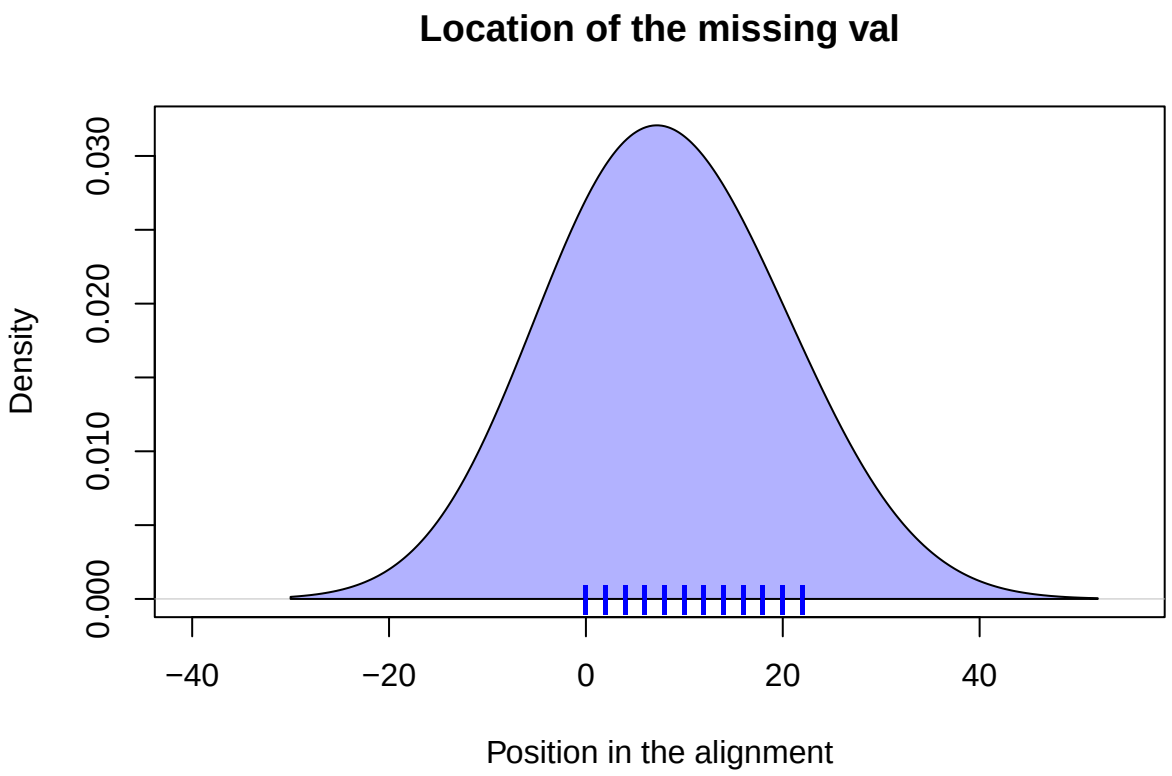
This graph shows that most SNPs have a low frequency, but there are some that have higher frequencies as well. This means that most of the SNPs in this set appear fewer times.

```
myFreq <- glMean(x)
myFreq <- c(myFreq, 1-myFreq)
hist(myFreq, proba=TRUE, col="darkseagreen3", xlab="Allele frequencies",
main="Distribution of allele frequencies", nclass=20, ylim = c(0,3.5))
temp <- density(myFreq, bw=.05)
lines(temp$x, temp$y*2,lwd=3)
```



This can also be visualized as a symmetric graph due to the nature of alleles having two options.

```
temp <- density(glNA(x), bw=10)
plot(temp, type="n", xlab="Position in the alignment", main="Location of the missing val", xlim=c(-40,5
polygon(c(temp$x,rev(temp$x)), c(temp$y, rep(0,length(temp$x))), col=transp("blue",.3))
points(glNA(x), rep(0, nLoc(x)), pch="|", col="blue")
```



Here we can see the locations of the missing data within the genome. They seem to be focused early in the sequence.

```
snp_counts <- table(vcf@fix[, "CHROM"])
#snp_counts
#This was supposed to be a table of SNPs per chromosome, but it is much too large to print
```

Discussion

The key results show that this is a very large and interesting dataset. While most of the allele frequencies are very high or low, there are a number of alleles that show promise for being influenced by environment. Additionally, the banding present in the index plot below, suggests that different populations do show different distributions of SNPs. This is promising for our future goals to identify specific SNPs, and therefore specific genes, that are linked with environmental factors. While there is some missing data, it seems fairly localized and not impactful for our model as we move forward.

As I move forward, the most important data processing step is to produce data in a format compatable with machine learning models. There are txt files that may lend themselves to easier manipulation. Some R packages such as 'fuc' have built-in functions to split VCF data that may be useful.

Appendix

Data Dictionary

```
data <- vcfR2tidy(vcf)
```

```
## Extracting gt element GT

## Extracting gt element DP

## Extracting gt element AD

## Extracting gt element GL
```

```
data_dict <- data$meta
data_dict
```

```
## # A tibble: 6 x 5
##   Tag    ID    Number Type    Description
##   <chr>  <chr> <chr>  <chr>   <chr>
## 1 INFO   NS    1      Integer Number of Samples With Data
## 2 INFO   AF    .      Float   Allele Frequency
## 3 FORMAT gt_GT 1      String  Genotype
## 4 FORMAT gt_DP 1      Integer Read Depth
## 5 FORMAT gt_AD 1      Integer Allele Depth
## 6 FORMAT gt_GL .      Float   Genotype Likelihood
```

Code

```
#META Data provides information about the file as well as the abbreviations used elsewhere in the file
#MAKE TABLE???
queryMETA(vcf)
```

```
## [1] "INFO=ID=NS"   "INFO=ID=AF"   "FORMAT=ID=GT" "FORMAT=ID=DP" "FORMAT=ID=AD"
## [6] "FORMAT=ID=GL"
```

```
queryMETA(vcf, element = 'GT')
```

```
## [[1]]
## [1] "FORMAT=ID=GT"         "Number=1"             "Type=String"
## [4] "Description=Genotype"
```

```
queryMETA(vcf, element = 'DP')
```

```
## [[1]]
## [1] "FORMAT=ID=DP"           "Number=1"               "Type=Integer"
## [4] "Description=Read Depth"
```

```
queryMETA(vcf, element = 'AD')
```

```
## [[1]]
## [1] "FORMAT=ID=AD"             "Number=1"
## [3] "Type=Integer"             "Description=Allele Depth"
```

```
queryMETA(vcf, element = 'GL')
```

```
## [[1]]
## [1] "FORMAT=ID=GL"                    "Number=."
## [3] "Type=Float"                      "Description=Genotype Likelihood"
```

```
#
vcf_field_names(vcf)
```

```
## # A tibble: 2 x 5
##   Tag   ID    Number Type    Description
##   <chr> <chr> <chr>  <chr>   <chr>
## 1 INFO  NS    1      Integer Number of Samples With Data
## 2 INFO  AF    .      Float   Allele Frequency
```

```
#sample names
colnames(vcf@gt)
```

```
##   [1] "FORMAT" "FD02"   "FD04"   "FD05"   "FD06"   "FD08"   "FD09"   "FD11"
##   [9] "FD12"   "FD15"   "FD18"   "GZS01"  "GZS02"  "GZS03"  "GZS04"  "GZS05"
##  [17] "GZS07"  "GZS08"  "GZS13"  "GZS17"  "GZS18"  "HS01"   "HS02"   "HS03"
##  [25] "HS04"   "HS05"   "HS06"   "JGS01"  "JGS02"  "JHS04"  "JHS06"  "JHS08"
##  [33] "JHS09"  "JHS10"  "JHS11"  "JHS13"  "JHS15"  "JHS16"  "JHS18"  "LCS02"
##  [41] "LCS04"  "LCS05"  "LCS07"  "LCS08"  "LCS09"  "LCS15"  "LCS16"  "LCS18"
##  [49] "LCS20"  "LS01"   "LS02"   "LS03"   "LS04"   "LS05"   "ML01"   "ML02"
##  [57] "ML03"   "ML05"   "ML06"   "ML07"   "ML08"   "ML12"   "ML13"   "ML14"
##  [65] "MS02"   "MS03"   "MS04"   "MS06"   "MS07"   "MS08"   "MS09"   "MS11"
##  [73] "MS12"   "MS13"   "SS01"   "SS03"   "SS04"   "SS08"   "SS09"   "SS10"
##  [81] "SS15"   "SS16"   "SS18"   "SS19"   "TMS01"  "TMS02"  "TMS05"  "TMS10"
##  [89] "TMS11"  "TMS13"  "TMS14"  "TMS15"  "TMS16"  "TTS01"  "TTS04"  "TTS07"
##  [97] "TTS09"  "TTS10"  "TTS11"  "TTS12"  "TTS13"  "TTS17"  "TTS20"  "WYS01"
## [105] "WYS02"  "WYS03"  "WYS04"
```

```
#Metadata
data$meta
```

```
## # A tibble: 6 x 5
##   Tag    ID    Number Type    Description
##   <chr>  <chr> <chr>  <chr>   <chr>
## 1 INFO   NS    1      Integer Number of Samples With Data
## 2 INFO   AF    .      Float   Allele Frequency
## 3 FORMAT gt_GT 1      String  Genotype
## 4 FORMAT gt_DP 1      Integer Read Depth
## 5 FORMAT gt_AD 1      Integer Allele Depth
## 6 FORMAT gt_GL .      Float   Genotype Likelihood
```

```
#Summary of the all the SNPs
data$fix
```

```
## # A tibble: 11,862 x 10
```

```
##    ChromKey CHROM   POS ID     REF   ALT    QUAL FILTER    NS AF
##       <int> <chr> <int> <chr>  <chr> <chr> <dbl> <chr>  <int> <chr>
##  1     3155 15      111 1_15   T     G        NA PASS      NA <NA>
##  2     3816 23       65 2_23   T     C        NA PASS      NA <NA>
##  3     3816 23       98 3_23   T     C        NA PASS      NA <NA>
##  4     3945 25       79 4_25   T     C        NA PASS      NA <NA>
##  5     4617 35       23 5_35   T     C        NA PASS      NA <NA>
##  6     5501 49        0 6_49   A     T        NA PASS      NA <NA>
##  7     6088 57       87 7_57   G     A        NA PASS      NA <NA>
##  8     7452 75       44 8_75   G     A        NA PASS      NA <NA>
##  9     2396 135      25 9_135  C     T        NA PASS      NA <NA>
## 10     3129 147      33 10_147 T     C        NA PASS      NA <NA>
## # i 11,852 more rows
```

```
#Summary of the all the samples
data$gt
```

```
## # A tibble: 1,257,372 x 8
##    ChromKey   POS Indiv gt_GT gt_DP gt_AD gt_GL     gt_GT_alleles
##       <int> <int> <chr> <chr> <int> <int> <chr>     <chr>
##  1     3155   111 FD02  0/0       7     7 .,9.7,.   T/T
##  2     3816    65 FD02  0/0      36    22 .,49.91,. T/T
##  3     3816    98 FD02  1/0      36    22 .,49.91,. C/T
##  4     3945    79 FD02  0/0      11    11 .,15.25,. T/T
##  5     4617    23 FD02  0/0      17    17 .,23.57,. T/T
##  6     5501     0 FD02  0/0      26    26 .,36.04,. A/A
##  7     6088    87 FD02  0/0      26    26 .,36.04,. G/G
##  8     7452    44 FD02  1/0      46    31 .,63.77,. A/G
##  9     2396    25 FD02  0/1      14     6 .,19.41,. C/T
## 10     3129    33 FD02  1/0      26    12 .,36.04,. C/T
## # i 1,257,362 more rows
```