SNP Variation Analysis

Qahhar

2023-11-07

INTRODUCTION

Welcome to Week 2 of the contest:

- Here we are to analys=ze a dataset that's an extract/subset of the data from GEUVADIS, a human population genome wide association studies for height across 5 super populations of the world.

What is Our Problem Statement?

Common single-nucleotide polymorphisms (SNPs) are predicted to explain over 50% of phenotypic variation in human height, but identifying the specific variants and associated regions requires huge sample sizes.

Given a set of over 25,000 unique SNPs, Your task is to identify and describe these interesting SNPs and their behaviour across the 5 super populations. Overall, we want you to use data to emphasize the need for diversity in human sequencing projects.

Description of the columns in the dataset:

Abbreviations in the data

- **AFR**: African (mostly African American)
- **EAS**: East-Asian
- **SAS**: South-Asian
- **HIS**: Hispanic
- **EUR**: European

Column Description (12)

- **SNPID**: represented as CHR:POS:REF:ALT; is a unique identifier for each SNP
- **RSID**: RS NUMBER; is a reference identifier for known SNPs
- **CHR**: CHROMOSOME; specifies the chromosome where the SNP is located
- **POS**: GENOMIC POSIION (BASE PAIR); gives the specific position of the SNP on the chromosome
- **EFFECT__ALLELE**: Mutant allele sequence; represents one of the alleles at the SNP that is associated with a particular effect, in our case height variation
- **OTHER__ALLELE**: Reference Allele Sequence
- **EFFECT__ALLELE__FREQ**: Minor allele frequency; indicates how frequently the "effect allele" occurs in the population.
- **BETA**: Odds probability; is a measure of how much the effect allele affects the outcome. A negative beta suggests a decrease in height, and a positive beta suggests an increase
- **SE**: Standard Error; associated with the beta, representing the uncertainty in the effect estimate.

- **P**: P-value which tells us how statistically significant the association between the SNP and height variation is.
- **N**: Sample size, indicating how many individuals' data were used in the analysis Imagine: n value here is 100692, which means that data for this particular SNP (snpid) was available for 100,692 individuals of African ancestry. Here it indicates the number of individuals from a specific ancestry or super population for whom data on that SNP is available.

Questions to be answered:

This phase focuses on EDA, Visualization and Reporting, as expected questions would be asked. We have to understand the trends/insights that could be uncovered from this data

Here are the list of questions that sparked my interest :

- How many SNPs are significant (p-value < 0.01) for variability in height (MAF > 0.01) in all the super populations?
- What is the average effect size (BETA) for SNPs with a minor allele frequency (MAF) greater than 0.01, using the significant SNPs?
- What is the level of correlation between the effect allele frequency (EFFECT_ALLELE_FREQ) and the effect size (BETA) for SNPs associated with height, if there is any?
- How much of Europeans' genetic variability can/cannot be found in other super populations? Does this provide enough argument for increasing the diversity of sequencing projects?
- For European SNPs found in other super populations as well, do their statistical measures differ?

Let's Get it!!

Firstly,

We load all necessary packages for this analysis. This is good practice.

```r
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.3     v readr     2.1.4
## v forcats   1.0.0     v stringr   1.5.0
## v ggplot2   3.4.4     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.0
## v purrr     1.0.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```r
library(tibble)
library(janitor)
```

```
##
## Attaching package: 'janitor'
##
## The following objects are masked from 'package:stats':
##
##     chisq.test, fisher.test
```

```r
library(ggplot2)
library(patchwork)
library(knitr)
library(kableExtra)
```

```
##
## Attaching package: 'kableExtra'
##
## The following object is masked from 'package:dplyr':
##
##     group_rows
```

```r
library(corrplot)
```

```
## corrplot 0.92 loaded
```

Secondly;

We retrieve the dataset and read it into a more suitable for analysis, that is, a dataset

```r
url <- "https://raw.githubusercontent.com/HackBio-Internship/public_datasets/main/R/datasets/Contests/hu

#Downloading the dataset, using destfile to assign it a file name
download.file(url, destfile = "SNP.tsv")

#Read the file in, it has a space delimeter

df_SNP <- read.delim("SNP.tsv", sep = " ")

# Reset the row names to create a new index to make for smoother appearance
rownames(df_SNP) <- NULL

head(df_SNP, n = 3)
```

```
##             SNPID      RSID CHR       POS EFFECT_ALLELE OTHER_ALLELE
## 1  1:32296525:C:T rs4949473   1  32296525             C            T
## 2  1:49121231:A:G  rs319993   1  49121231             A            G
## 3 1:171155103:C:T rs6657314   1 171155103             T            C
##   EFFECT_ALLELE_FREQ        BETA      SE        P      N ANCESTRY
## 1              0.299 -0.00311101 0.00496 0.530811 100692  AFRICAN
## 2              0.686 -0.00340862 0.00492 0.488601 104293  AFRICAN
## 3              0.852 -0.00436463 0.00644 0.497941 104294  AFRICAN
```

DESCRIPTIVE SUMMARY OF THE DATA

In a bid to explore the data, understanding the dimensions of said data and grasping statistical summary of interesting aspects of our data

```r
#Returns the full dimensions (no of rows and columns) of the data
dim(df_SNP)
```

```
## [1] 25000    12
```

```r
# Extracts the column names
colnames(df_SNP)
```

```
##  [1] "SNPID"              "RSID"               "CHR"
##  [4] "POS"                "EFFECT_ALLELE"      "OTHER_ALLELE"
##  [7] "EFFECT_ALLELE_FREQ" "BETA"               "SE"
## [10] "P"                  "N"                  "ANCESTRY"
```

```
#Provides a brief glimpse into our data
glimpse(df_SNP)
```

```
## Rows: 25,000
## Columns: 12
## $ SNPID              <chr> "1:32296525:C:T", "1:49121231:A:G", "1:171155103:C:~
## $ RSID               <chr> "rs4949473", "rs319993", "rs6657314", "rs2816213", ~
## $ CHR                <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, ~
## $ POS                <int> 32296525, 49121231, 171155103, 179183766, 31139078,~
## $ EFFECT_ALLELE      <chr> "C", "A", "T", "C", "C", "G", "T", "T", "G", "T", "~
## $ OTHER_ALLELE       <chr> "T", "G", "C", "T", "A", "A", "C", "G", "T", "C", "~
## $ EFFECT_ALLELE_FREQ <dbl> 0.2990, 0.6860, 0.8520, 0.2630, 0.5780, 0.5470, 0.0~
## $ BETA               <dbl> -0.003111010, -0.003408620, -0.004364630, -0.007882~
## $ SE                 <dbl> 0.00496, 0.00492, 0.00644, 0.00534, 0.00480, 0.0046~
## $ P                  <dbl> 0.5308110, 0.4886010, 0.4979410, 0.1396730, 0.11148~
## $ N                  <int> 100692, 104293, 104294, 104294, 104294, 100692, 104~
## $ ANCESTRY           <chr> "AFRICAN", "AFRICAN", "AFRICAN", "AFRICAN", "AFRICA~
```

```
# Returns high level statistical summary of our dataset
summary(df_SNP)
```

```
##     SNPID               RSID                CHR              POS
##  Length:25000       Length:25000       Min.   : 1.000   Min.   :    67365
##  Class :character   Class :character   1st Qu.: 4.000   1st Qu.: 31819538
##  Mode  :character   Mode  :character   Median : 8.000   Median : 71068010
##                                        Mean   : 8.571   Mean   : 79495862
##                                        3rd Qu.:13.000   3rd Qu.:115694815
##                                        Max.   :22.000   Max.   :249222450
##
##  EFFECT_ALLELE      OTHER_ALLELE       EFFECT_ALLELE_FREQ      BETA
##  Length:25000       Length:25000       Min.   :0.0000     Min.   :-1.53806
##  Class :character   Class :character   1st Qu.:0.0919     1st Qu.:-0.00539
##  Mode  :character   Mode  :character   Median :0.2670     Median :-0.00005
##                                        Mean   :0.3389     Mean   : 0.00030
##                                        3rd Qu.:0.5470     3rd Qu.: 0.00515
##                                        Max.   :1.0000     Max.   : 1.93485
##                                                           NA's   :194
##       SE                 P                N              ANCESTRY
##  Min.   :0.00104   Min.   :0.0000   Min.   :    482   Length:25000
##  1st Qu.:0.00358   1st Qu.:0.1163   1st Qu.:  46408   Class :character
##  Median :0.00654   Median :0.3729   Median : 100692   Mode  :character
##  Mean   :0.01802   Mean   :0.4087   Mean   : 374820
##  3rd Qu.:0.00944   3rd Qu.:0.6742   3rd Qu.: 264725
##  Max.   :1.07000   Max.   :0.9999   Max.   :1597374
##  NA's   :194       NA's   :194
```

This has enables us to retrive the information that the dataset contains 25000 observations/rows whilst having 12 columns as stated previously in the introduction Our preview also tells us that we have FIVE character/string columns, namely

- SNPID, RSID , EFFECT_ALLELE , OTHER_ALLELE, ANCESTRY

DATA CLEANING

Before going any further, it is best practice to perform simple data cleaning techniques on a given dataset.

A data cleaning cadence would be

- Observe and deal with missing values
- Search for through put duplicates in the data
- Make sure all data are in the right data type(they are)

Missing Values

```
# Count rows with at least one NA
count <- sum(rowSums(is.na(df_SNP)) > 0)

cat("Number of rows with at least one NA:", count, "\n")
```

```
## Number of rows with at least one NA: 194
```

```
# What is the column with a large sumn of missing values
sum(is.na(df_SNP$P))
```

```
## [1] 194
```

```
sum(is.na(df_SNP$SE))
```

```
## [1] 194
```

```
sum(is.na(df_SNP$BETA))
```

```
## [1] 194
```

```
# Create a subset data with no missing values to analyze
# First, create a logical vector of complete cases
complete_rows <- complete.cases(df_SNP)

# Subset the data frame to retain only rows with complete data
filtered_df <- df_SNP[complete_rows, ]
nrow(filtered_df)
```

```
## [1] 24806
```

complete.cases(df) returns a logical vector that is **TRUE** for rows with complete data (no missing values) and FALSE for rows with missing values.

We use this logical vector to subset the original data frame df, keeping only the rows where complete_rows is TRUE. This results in a new data frame called filtered_df that contains only rows with complete data & contain only the rows that have no missing values.

Duplicates

```
# Check for duplicate rows
duplicate_rows <- duplicated(filtered_df)

# Subset the data frame to retain only the first occurrence of each unique row
no_duplicates <- filtered_df[!duplicate_rows, ]
nrow(no_duplicates)
```

```
## [1] 24806
```

We find that there is no single occurrence where there is through duplication across an entire row

Dropping Columns

For our analysis, certain columns seems to not be important. Let me explain

rsid:

The `rsid` column contains reference SNP cluster IDs, which are unique identifiers used to reference specific SNPs in genomic databases. it is beneficial to remove the `rsid` column because, similar to `snpid`, it's primarily an identifier for SNPs and doesn't contribute directly to the analysis. Removing it helps streamline the dataset for the analysis of SNP significance and population differences.

```
col_dropped <- no_duplicates[, !(names(no_duplicates) %in% c("RSID"))]
```

Renaming column

The provided dataset seems to have column names like "SNPID," "POS" "CHR," etc., which are meaningful and descriptive. But it's best to have column names in lowercase

```
renamed_df <- rename_with(col_dropped, tolower)

colnames(renamed_df)
```

```
##  [1] "snpid"               "chr"                 "pos"
##  [4] "effect_allele"       "other_allele"        "effect_allele_freq"
##  [7] "beta"                "se"                  "p"
## [10] "n"                   "ancestry"
```

Convert it to a simpler name,

```
gene_var <- data.frame(renamed_df)

dim(gene_var)
```

```
## [1] 24806    11
```

Given our gene_var dataset, we can begin further exploration

EDA

This section would revolve around the `Ancestry` column, which contains 5 super populations of our data set, to understand the significant difference of height per category based on SNPs found

```
unique(gene_var$ancestry)
```

```
## [1] "AFRICAN"    "EUROPEAN"   "SOUTH_ASIA" "EAST_ASIA"  "HISPANIC"
```

```
tabyl(gene_var$ancestry)
```

```
##  gene_var$ancestry    n   percent
##            AFRICAN 4999 0.2015238
##          EAST_ASIA 4811 0.1939450
##           EUROPEAN 4998 0.2014835
##           HISPANIC 4999 0.2015238
##         SOUTH_ASIA 4999 0.2015238
```

From this data, we can see that the category that probably contributed to the missing values in the df_SNP data set was "EAST_ASIA" But it's evident that each group would have contributed 5000 values in the original data set

How many SNPs are significant (p-value $<$ 0.01) for variability in height (MAF $>$ 0.01) in all the super populations?

To identify SNPs significantly associated with height variability, a filtering approach is employed based on p-values (p-value $<$ 0.01) and Minor Allele Frequency (MAF $>$ 0.01). MAF quantifies the frequency of the less common allele in the population.

The rationale is to focus on SNPs where the effect allele (associated with height) is relatively common. This is reflected in the use of `effect_allele_freq` as a proxy for MAF, as the more common the effect allele, the more likely it is to have a substantial impact on height.

```r
significant_snps <- gene_var %>%
  filter(p < 0.01, effect_allele_freq > 0.01)

# Count the number of significant SNPs
num_significant_snps <- nrow(significant_snps)

# Print the result
cat("Number of significant SNPs:", num_significant_snps, "\n")
```

```
## Number of significant SNPs: 2253
```

What is the average effect size (BETA) for SNPs with a minor allele frequency (MAF) greater than 0.01 per super population?

```r
significant_snps %>%
  group_by(ancestry) %>%
  filter(effect_allele_freq > 0.01) %>%
  summarize(beta_mean = round(mean(beta), 5))
```

```
## # A tibble: 5 x 2
##   ancestry   beta_mean
##   <chr>          <dbl>
## 1 AFRICAN     -0.00154
## 2 EAST_ASIA   -0.00085
## 3 EUROPEAN    -0.0008
## 4 HISPANIC    -0.00339
## 5 SOUTH_ASIA   0.00057
```

What is the level of correlation between numerical columns and the effect size (BETA) for SNPs associated with height, if there is any?

It is safe to assume that SNPs with a p - value lower than 0.01 have a higher significance and higher chance of causing variability within height, so we would continue to work with the `significant_snps` data set

If correlation coefficient, calculated, is close to 1, it suggests a strong positive correlation, while a value close to -1 indicates a strong negative correlation. If the correlation coefficient is close to 0, it suggests little to no correlation between the variables.

```r
# Calculate the correlations between "BETA" and (n, se, p, frequency)

columns_to_analyze <- c("effect_allele_freq", "se", "n", "p")

correlation_values <- sapply(columns_to_analyze, function(col_name) {
  cor(significant_snps$beta, significant_snps[, col_name])
})

# Create a data frame to store the results
correlation_df <- data.frame( Column = columns_to_analyze, Correlation = correlation_values)

correlation_df <- correlation_df %>%
  arrange(desc(Correlation))

# Visuals
library(reshape2)
```

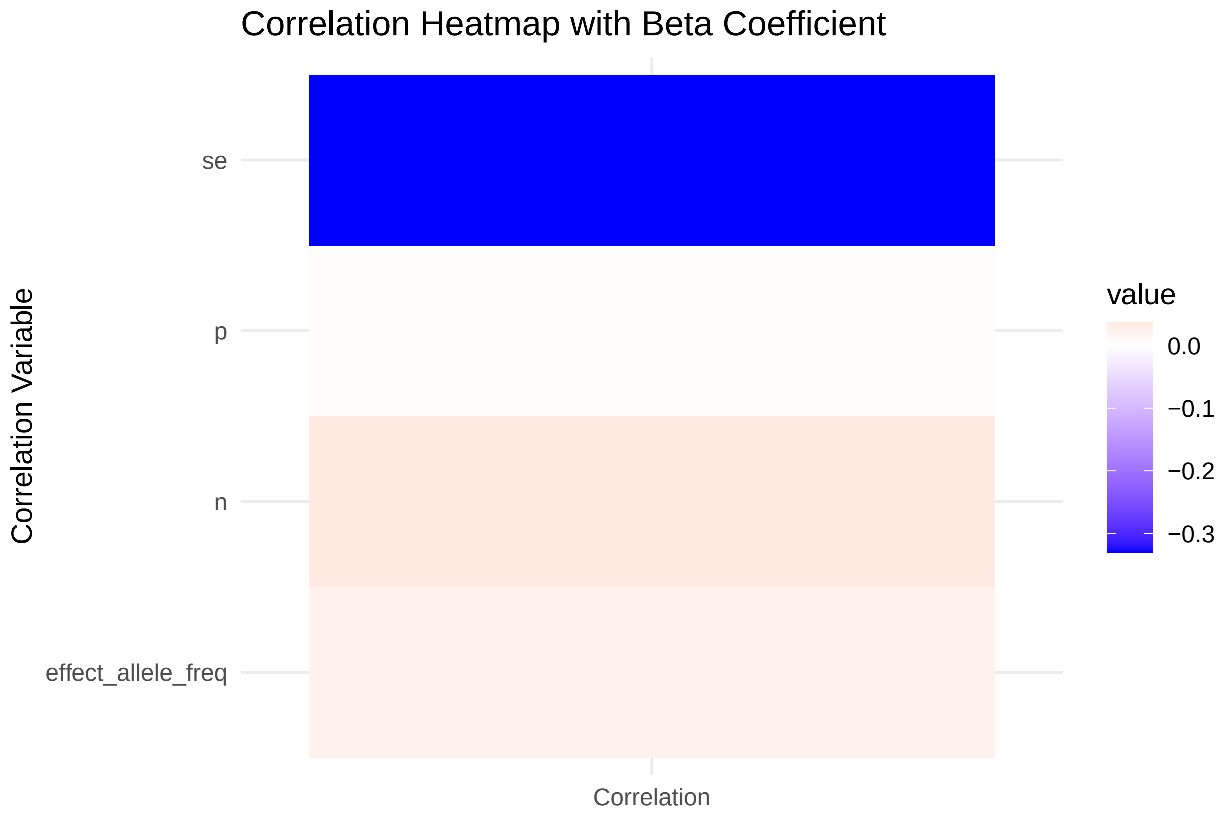
```
##
## Attaching package: 'reshape2'

## The following object is masked from 'package:tidyr':
##
##     smiths
```

```
# Melt the data frame for heatmap plotting
melted_data <- melt(correlation_df, id.vars = "Column")

# Create a heatmap
heatmap_plot <- ggplot(melted_data, aes(x = variable, y = Column, fill = value)) +
  geom_tile() +
  scale_fill_gradient2(low = "blue", mid = "white", high = "red", midpoint = 0) +
  theme_minimal() +
  labs(title = "Correlation Heatmap with Beta Coefficient") +
  xlab(' ') +
  ylab("Correlation Variable")

print(heatmap_plot)
```



```
# corr_effect_allele <- cor(significant_snps$effect_allele_freq, significant_snps$beta)
```

- `SE`(standard error) has a negative correlation with `BETA.` This means that as `SE` increases, `BETA.` tends to decrease.
- `N`(sample size), `P` (p-value), `EFFECT_ALLELE_FREQ` all have positive correlation with `BETA.` This means

Table 1: Count of Significant SNP's Per Ancestry

		Metric	
ancestry	count_sig_snps	beta_mean	allele_freq_mean
AFRICAN	294	-0.0015	0.3984
EAST_ASIA	300	-0.0009	0.4037
EUROPEAN	1371	-0.0008	0.3721
HISPANIC	199	-0.0034	0.3741
SOUTH_ASIA	89	0.0006	0.3701

that as they increase, `BETA.` tends to increase.

`SE` has the strongest negative correlation with `BETA`and `N`, `P`and `EFFECT_ALLELE_FREQ` have positive correlations, with varying strengths (`N` > `EFFECT_ALLELE_FREQ` > `P`). So, in terms of absolute correlation strength:

1. `SE` (strong negative correlation)
2. `N` (moderate positive correlation)
3. `P` (weak positive correlation)
4. `EFFECT_ALLELE_FREQ` (weak positive correlation)

Return a table of statistical significance from the SNPs (p_value < 0.1) data frame retrieve

```
summary_table <- significant_snps %>%
  group_by(ancestry) %>%
  summarize(count_sig_snps = n(), beta_mean = mean(beta), allele_freq_mean = mean(effect_allele_freq))

# Making it an explanatory table
table <- kable(summary_table, digits=4,
             caption="Count of Significant SNP's Per Ancestry") %>%
  add_header_above(c(" " = 2,  "Metric" = 2))

table
```

These results provide insights into the genetic variability of height across different super populations.

For instance, the `EUROPEAN` super population has the highest number of significant SNPs, indicating a strong genetic influence on height in this population.

The `beta coefficients` give an idea of the strength and direction of the genetic effect. A negative beta suggests a decrease in height, and a positive beta suggests an increase

- African: The average beta is -0.0015 (a decrease in height).
- East Asia: The average beta is -0.0009 (a decrease in height).
- European: The average beta is -0.0008 (a decrease in height).
- Hispanic: The average beta is -0.0034 (a significant decrease in height).
- South Asia: The average beta is 0.0006 (a slight increase in height).

Comparing the beta values, we can see that the HISPANIC super group has the most negative and significant beta value, indicating the largest decrease in height associated with the significant SNPs.

The magnitude of the beta value indicates the strength and direction of the effect. A larger absolute value of beta suggests a more substantial impact on the trait (height). In this context, the super group with the most negative (or lowest) average beta has the most significant effect on height because it experiences the greatest decrease in height associated with these SNPs.

`Allele frequencies` show the allele's prevalence in each population. The categories have a close range of frequencies, suggesting that the occurrence of these SNPs on the trait isn't too different per group

How much of Europeans genetic variability can/cannot be found in other super populations? Does this provide enough argument for increasing the diversity of sequencing projects?

This implies comparing the overlap of SNPs found in the European super population with those in each of the other super populations. We could calculate the percentage of SNPs that match between European and each other super population

```
shared_snps <- gene_var %>%
  filter(ancestry == "EUROPEAN") %>%
  select(snpid) %>%
  semi_join(gene_var %>% filter(ancestry != "EUROPEAN"), by = "snpid")

head(shared_snps, n = 3)
```

```
##             snpid
## 1  1:28378384:C:T
## 2 1:147381444:A:C
## 3 1:208450001:C:T
```

To test if the code effectively extracts snpid's that are found in multiple ancestries, we use this SNP, `3:129980909:A:G`

```
gene_var %>%
  filter(snpid == "3:129980909:A:G") %>%
  distinct(ancestry)
```

```
##    ancestry
## 1  EUROPEAN
## 2 EAST_ASIA
```

We find that this particular SNP is found both the `European` & `East Asia` super population.

And so, we:

Calculate the total count of SNPs for each super population, filtering out the Europeans

```
total_counts <- gene_var %>%
  filter(ancestry != "EUROPEAN") %>%
  group_by(ancestry) %>%
  summarise(total_snps = n())
```

Calculate the count of shared SNPs for each super population

```
shared_counts <- gene_var %>%
  filter(ancestry != "EUROPEAN") %>%
  group_by(ancestry) %>%
  summarise(count_shared_snps = sum(snpid %in% shared_snps$snpid)) %>%
  mutate(european_snp_count = c(4998))
```

Combine the datasets

```
comparison <- total_counts %>%
  left_join(shared_counts, by = "ancestry") %>%
  mutate(
    shared_percentage = (count_shared_snps / total_snps) * 100)
```

```
comparison
```

```
## # A tibble: 4 x 5
##   ancestry   total_snps count_shared_snps european_snp_count shared_percentage
##   <chr>           <int>             <int>              <dbl>             <dbl>
## 1 AFRICAN          4999                14               4998             0.280
## 2 EAST_ASIA        4811                28               4998             0.582
## 3 HISPANIC         4999                22               4998             0.440
## 4 SOUTH_ASIA       4999                14               4998             0.280
```

I believe with the number of shared genetic variability seen that this provides enough argument for increasing the diversity of sequencing projects, in this case amongst super populations

For European SNPs found in other super populations as well, do their statistical measures differ

It is clear that there are shared SNPs between super populations in our dataset, how do we intend to understand, if there is a clear difference between the super populations sharing the SNP.

Let's pick the first

```
head(shared_snps$snpid, n = 1)
```

```
## [1] "1:28378384:C:T"
```

"1:28378384:C:T" is the SNP of interest. So we start by extracting a data frame, containing only this value

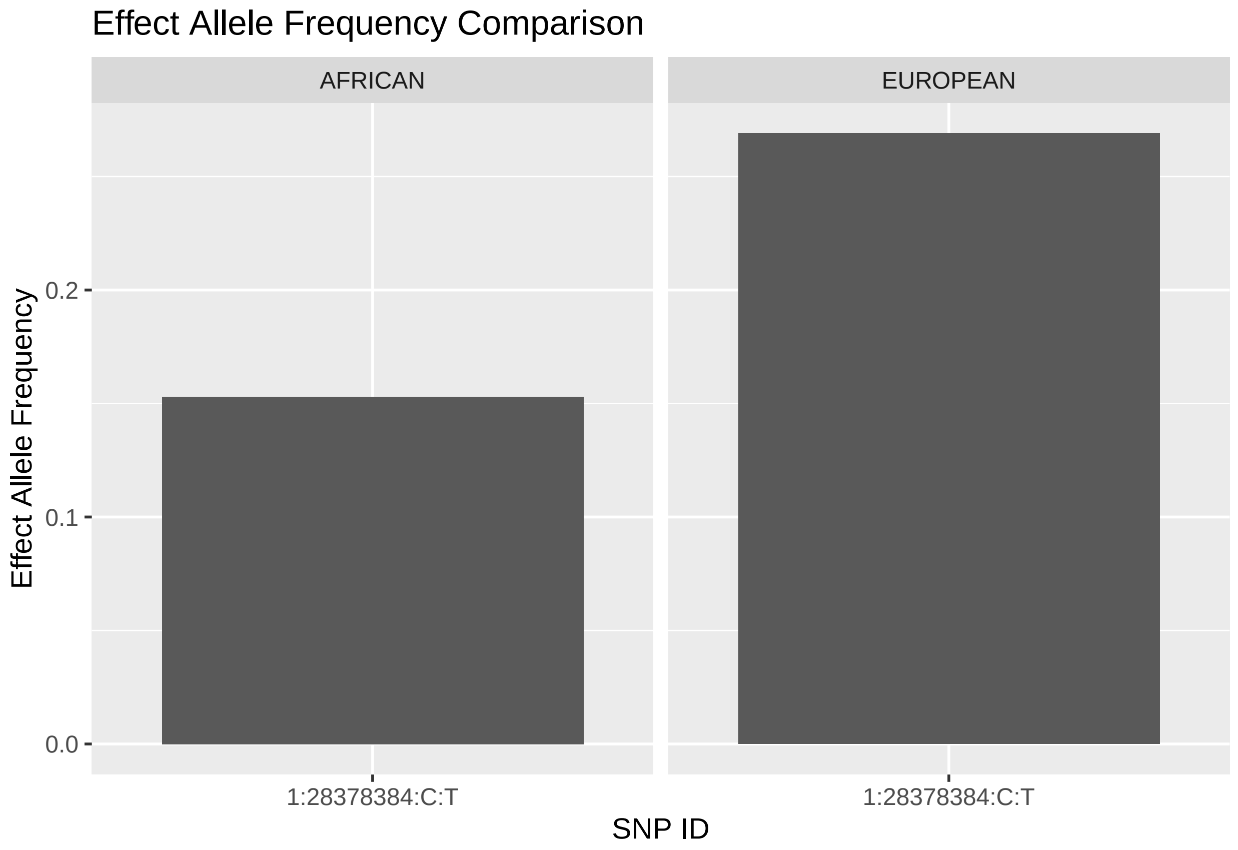
```
snp <- gene_var %>%
  filter(snpid == "1:28378384:C:T")

snp %>%
  distinct(ancestry)
```

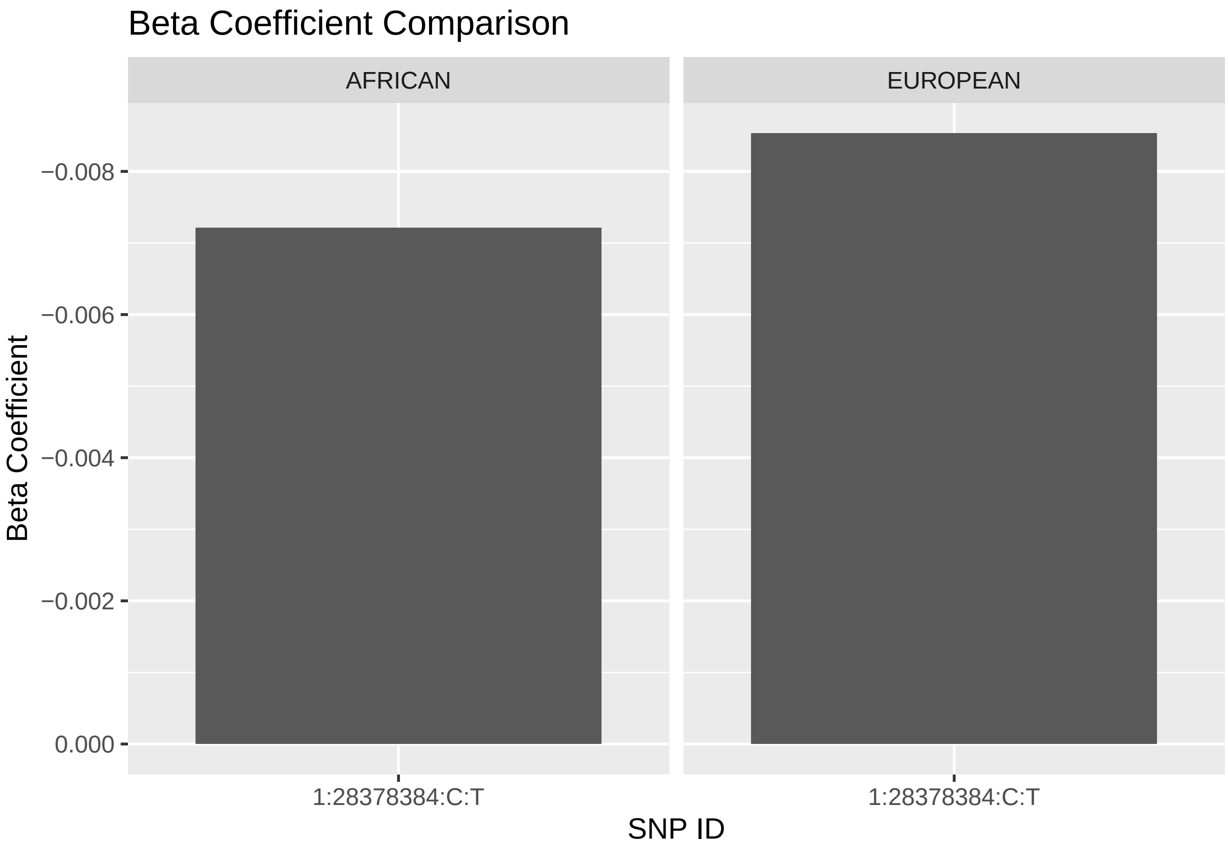
```
##   ancestry
## 1  AFRICAN
## 2 EUROPEAN
```

Visualize statistical measures to get a clear understanding of any differences between these groups

```
ggplot(snp, aes(x = snpid, y = effect_allele_freq)) +
  geom_bar(stat = "identity", position = "dodge") +
  labs(title = "Effect Allele Frequency Comparison", x = "SNP ID", y = "Effect Allele Frequency")+
  facet_wrap(~ancestry)
```



```
ggplot(snp, aes(x = snpid, y = beta)) +
  geom_bar(stat = "identity", position = "dodge") +
  labs(title = "Beta Coefficient Comparison", x = "SNP ID", y = "Beta Coefficient")+
  facet_wrap(~ancestry) +
  scale_y_reverse()
```



Basically for the SNP retrieved from European descent has a higher odds probability (Beta) score, as well as a higher frequency mark of occurring in the `European` descent as opposed to it's counterpart, `Africa`

Thank you!