SNP Variation Analysis

Qahhar

2023-10-31

INTRODUCTION

Welcome to Week 2 of the contest:

- Here we are to analysze 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 emphasise the need for diversity in human sequencing projects.

Description of the columns in the dataset:

Abbreviations in the dataset

- **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)
- **RSID**: (RS NUMBER, WHEN AVAILABLE)
- **CHR**: CHROMOSOME
- **POS**: GENOMIC POSIION (BASE PAIR) - hg19/hg37 BUILD
- **EFFECT_ALLELE**: Mutant allele sequence
- **OTHER_ALLELE**: Reference Allele Sequence
- **EFFECT_ALLELE_FREQ**: (Minor allele frequency)
- **BETA**: Odds probability (6 significant figures)
- **SE**: Standard Error(3 significant figures)
- **P**: P-value
- **N**: Sample size

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 Certainly! Here are all the questions you can address with the given dataset:

- How many SNPs are significant (p-value < 0.01) for variability in height (MAF > 0.01) in all the super populations?
- 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?
- What is the average effect size (BETA) for SNPs with a minor allele frequency (MAF) greater than 0.01?
- Are there any SNPs associated with height that show a different effect size (BETA) between different super populations?
- Is there a correlation between the effect allele frequency (EFFECT_ALLELE_FREQ) and the effect size (BETA) for SNPs associated with height?
- Can you identify the SNP (RSID) with the highest and lowest effect size (BETA) for each super population?

Let's Get it!!

Firstly,

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

```
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
```

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

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

```
library(skimr)
library(ggplot2)
library(devtools)
```

```
## Loading required package: usethis
```

```
library(patchwork)
library(knitr)
library(kableExtra)
```

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

```
library(reshape2)
```

```
##
## Attaching package: 'reshape2'
##
## The following object is masked from 'package:tidyr':
##
##     smiths
```

Secondly;

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

```
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

```
#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"
```

```r
#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~
```

```r
# 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

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

```
## [1] 582
```

```
# 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 occurence of through duplication across a row

Renaming column The provided dataset seems to have column names like “SNPID,” “RSID,” “CHR,” etc., which are meaningful and descriptive. But it’s best to have your columns as lowercase

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

colnames(renamed_df)
```

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

Convert it to a simpler name

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

```
## [1] 24806    12
```

Given our gene_var dataset, we can begin further exploration

EDA

Firstly, I'd like to summarize and understand how the categorical columns reflect on the