HW1

Huiyu Hu

1/18/2018

# Part 1

* Finished the GitHub setting.

# Part 2

1. How many persons are in the data set (statisticians call this n)? How many SNPs are in the data set (statisticians call this p)?

awk 'END { print NR }' /home/m280-data/hw1/merge-geno.fam

## 959

awk 'END { print NR }' /home/m280-data/hw1/merge-geno.bim

## 8348674

* Answer: n = 959; p = 8348674

1. Which chromosomes does this data set contain? How many SNPs are in each chromosome?

awk < /home/m280-data/hw1/merge-geno.bim '{print $1}' | uniq -c

## 1309299 1  
## 1215399 3  
## 1090185 5  
## 980944 7  
## 732013 9  
## 815860 11  
## 602809 13  
## 491208 15  
## 477990 17  
## 393615 19  
## 239352 21

* Answer:

1. MAP4 (microtubule-associated protein 4) is a gene on chromosome 3 spanning positions 47,892,180 bp – 48,130,769 bp. How many SNPs are located within MAP4 gene?

awk '{if ($1 == 3 && $4 >= 47892180 && $4 <= 48130769) print}' /home/m280-data/hw1/merge-geno.bim | wc -l

## 894

* Answer: There are 894 SNPs lovated within MAP4 gene.

1. Reformat

* For .bim file:

echo " 2.40 = FILE FORMAT VERSION NUMBER." > /home/huiyuhu/biostat-m280-2018-winter/hw1/merge-geno.lalala  
 echo '8348674 = NUMBER OF SNPS LISTED HERE.' >> /home/huiyuhu/biostat-m280-2018-winter/hw1/merge-geno.lalala  
 awk '{OFS = ","} {print $2, $1, $4}' /home/m280-data/hw1/merge-geno.bim >> /home/huiyuhu/biostat-m280-2018-winter/hw1/merge-geno.lalala  
 head /home/huiyuhu/biostat-m280-2018-winter/hw1/merge-geno.lalala

## 2.40 = FILE FORMAT VERSION NUMBER.  
## 8348674 = NUMBER OF SNPS LISTED HERE.  
## 1-54490,1,54490  
## 1-55550,1,55550  
## 1-57033,1,57033  
## 1-57064,1,57064  
## 1-57818,1,57818  
## 1-58432,1,58432  
## 1-58448,1,58448  
## 1-58814,1,58814

* For .fam file
* Comments: 1, change seperator to “,”; 2, strip the string T2DG from the IDs; 3, convert sex from number to char; 4, remove missing value.

awk '{OFS = ","} {print $1, $2, $3, $4, $5, $6}' /home/m280-data/hw1/merge-geno.fam | sed 's/T2DG//g' | awk -F, -v sex="" '{OFS = ","} {if ($5 == 1) sex = "M"; else sex = "F"} {print $1,$2,$3,$4, sex,$6}' | awk -F, '{OFS=","} { for (i = 1; i <= NF; i++) {$i = ($i == 0 ? "" : $i)}; print }' > /home/huiyuhu/biostat-m280-2018-winter/hw1/merge-geno.hehehe   
head -20 /home/huiyuhu/biostat-m280-2018-winter/hw1/merge-geno.hehehe

## 2,0200001,,,M,  
## 2,0200002,,,F,  
## 2,0200003,,,F,  
## 2,0200004,,,F,  
## 2,0200005,,,M,  
## 2,0200006,,,M,  
## 2,0200007,,,F,  
## 2,0200008,,,F,  
## 2,0200009,,,F,  
## 2,0200012,,,M,  
## 2,0200013,,,M,  
## 2,0200018,,,M,  
## 2,0200023,,,F,  
## 2,0200024,,,M,  
## 2,0200027,,,F,  
## 2,0200031,0200001,0200015,M,  
## 2,0200032,0200001,0200015,F,  
## 2,0200033,0200001,0200015,F,  
## 2,0200034,0200001,0200015,F,  
## 2,0200035,0200001,0200015,F,

# Part 3

# These assignments will be removed later  
 rep <- 50  
 seed <- 280  
 n <- 100  
 dist <- 't1'  
  
 ## parsing command arguments  
 for (arg in commandArgs(TRUE)) {  
 eval(parse(text=arg))  
 }  
   
   
 ## check if a given integer is prime  
 isPrime = function(n) {  
 if (n <= 3) {  
 return (TRUE)  
 }  
 if (any((n %% 2:floor(sqrt(n))) == 0)) {  
 return (FALSE)  
 }  
 return (TRUE)  
 }  
   
 ## estimate mean only using observation with prime indices  
 estMeanPrimes = function (x) {  
 n = length(x)  
 ind = sapply(1:n, isPrime)  
 return (mean(x[ind]))  
 }  
   
 # step 1: set random seed  
 set.seed(seed)  
   
 sum1 <- 0  
 sum2 <- 0  
 for ( i in 1:rep) {  
 # step 2: generate data according to argument dist  
 if (dist == "gaussian"){  
 x = rnorm(n,0,1)  
 }  
 else if (dist == "t1"){  
 x = rt(n, df=1)  
 }  
 else if (dist == "t5"){  
 x = rt(n, df=5)  
 }  
   
 # estimate mean  
 emp <- estMeanPrimes(x)  
 emc <- mean(x)  
 sum1 <- sum1 + (emp-0)^2  
 sum2 <- sum2 + (emc-0)^2  
 }  
   
 result <- paste(sum1/rep, sum2/rep, sep = " ")  
 #return(result)  
 result

## [1] "2808.42566728365 230.237205563387"

# autoSim.R  
rep <- 50  
seed <- 280  
distTypes = c("gaussian", "t1", "t5")  
nVals = seq(100, 500, by=100)  
  
for (n in nVals) {  
 for (dist in distTypes) {  
 oFile = paste("n", n, "\_", dist, ".txt", sep="")  
 arg = paste("seed=", seed," n=", n," dist=\\\"", dist,"\\\" rep=",rep, sep="")  
 sysCall = paste("nohup Rscript runSim.R ", arg, " > ", oFile)  
 system(sysCall)  
 print(paste("sysCall=", sysCall, sep=""))  
 }  
}

## [1] "sysCall=nohup Rscript runSim.R seed=280 n=100 dist=\\\"gaussian\\\" rep=50 > n100\_gaussian.txt"  
## [1] "sysCall=nohup Rscript runSim.R seed=280 n=100 dist=\\\"t1\\\" rep=50 > n100\_t1.txt"  
## [1] "sysCall=nohup Rscript runSim.R seed=280 n=100 dist=\\\"t5\\\" rep=50 > n100\_t5.txt"  
## [1] "sysCall=nohup Rscript runSim.R seed=280 n=200 dist=\\\"gaussian\\\" rep=50 > n200\_gaussian.txt"  
## [1] "sysCall=nohup Rscript runSim.R seed=280 n=200 dist=\\\"t1\\\" rep=50 > n200\_t1.txt"  
## [1] "sysCall=nohup Rscript runSim.R seed=280 n=200 dist=\\\"t5\\\" rep=50 > n200\_t5.txt"  
## [1] "sysCall=nohup Rscript runSim.R seed=280 n=300 dist=\\\"gaussian\\\" rep=50 > n300\_gaussian.txt"  
## [1] "sysCall=nohup Rscript runSim.R seed=280 n=300 dist=\\\"t1\\\" rep=50 > n300\_t1.txt"  
## [1] "sysCall=nohup Rscript runSim.R seed=280 n=300 dist=\\\"t5\\\" rep=50 > n300\_t5.txt"  
## [1] "sysCall=nohup Rscript runSim.R seed=280 n=400 dist=\\\"gaussian\\\" rep=50 > n400\_gaussian.txt"  
## [1] "sysCall=nohup Rscript runSim.R seed=280 n=400 dist=\\\"t1\\\" rep=50 > n400\_t1.txt"  
## [1] "sysCall=nohup Rscript runSim.R seed=280 n=400 dist=\\\"t5\\\" rep=50 > n400\_t5.txt"  
## [1] "sysCall=nohup Rscript runSim.R seed=280 n=500 dist=\\\"gaussian\\\" rep=50 > n500\_gaussian.txt"  
## [1] "sysCall=nohup Rscript runSim.R seed=280 n=500 dist=\\\"t1\\\" rep=50 > n500\_t1.txt"  
## [1] "sysCall=nohup Rscript runSim.R seed=280 n=500 dist=\\\"t5\\\" rep=50 > n500\_t5.txt"

1. Summary

ns <- c()  
methods <- c()  
for (n in seq(100, 500, by=100)){  
 ns <- append(ns, c(n, ""))  
 methods <- append(methods, c("PrimeAvg", "SampAvg"))  
}  
  
table <- data.frame(  
 n = ns,  
 method = methods,  
 t\_1 = NA, t\_5 = NA, Gaussian = NA  
 )  
  
#loop to write in data  
distTypes = c("t1", "t5","gaussian")  
nVals = seq(100, 500, by=100)  
i <- 0  
j <- 0  
for (n in nVals) {  
 i <- i + 1  
 j <- 0  
 for (dist in distTypes) {  
 j <- j + 1  
 iFile = paste("n", n, "\_", dist, ".txt", sep="")  
 a <- read.table(iFile)  
 a <- as.data.frame(a)  
 table[2\*i-1, j + 2] <- a[1,2]  
 table[2\*i, j + 2] <- a[1,3]  
 }  
}  
  
#table  
library(knitr)  
kable(table, "markdown")

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| n | method | t\_1 | t\_5 | Gaussian |
| 100 | PrimeAvg | 2808.42570 | 0.0664284 | 0.0414674 |
|  | SampAvg | 230.23720 | 0.0167508 | 0.0094834 |
| 200 | PrimeAvg | 799.31393 | 0.0418479 | 0.0187544 |
|  | SampAvg | 95.87976 | 0.0076665 | 0.0071637 |
| 300 | PrimeAvg | 446.40235 | 0.0241224 | 0.0200254 |
|  | SampAvg | 53.57742 | 0.0054711 | 0.0035344 |
| 400 | PrimeAvg | 19.45133 | 0.0243314 | 0.0112689 |
|  | SampAvg | 20.53370 | 0.0035170 | 0.0029707 |
| 500 | PrimeAvg | 217.26890 | 0.0116763 | 0.0111769 |
|  | SampAvg | 20.78780 | 0.0033550 | 0.0025920 |