```
setwd("c:/AvramopoulosProject")

######## QUICK SCAN TO DETERMINE WHERE THE DATA BEGINS

for(i in 1:24){
x<-scan(dir()[i],skip=11,nlines=1,what="character",sep=",")
print(x[1])
}

#########  QUICK SCAN, VERIFYING A UNIQUE INDIVIDUAL IDENTIFIER PER FILE --- MULTIPLE THREADS (SESSIONS)
OF R

system(paste(Sys.getenv("COMSPEC"),"/c","rbat.bat"),show.output.on.console = F,invisible=F)  ### AVOID
MEMORY ISSUES, READ 9 FILES AT A TIME
Sys.sleep(120)
system(paste(Sys.getenv("COMSPEC"),"/c","rbat1.bat"),show.output.on.console = F,invisible=F)  ### " "
Sys.sleep(90)
system(paste(Sys.getenv("COMSPEC"),"/c","rbat2.bat"),show.output.on.console = F,invisible=F)  ### " ", 6
FILES

x<-read.table("output.txt")

#############################################################################################
####################  WRITE THE DB FILE   ##############################

setwd("c:/AvramopoulosProject")
library(RSQLite)  ### LOAD THE RSQLITE LIBRARY (NOTE: THE DBI LIBRARY LOADS AUTOMATICALLY)
dbfile<-"AVRDB.dbsql"  ### THE DB FILENAME TO WORK WITH
drv<-dbDriver("SQLite")  ### LOAD THE SQLite DRIVER
con<-dbConnect(drv,dbname = dbfile)  ### CONNECT TO THE DATABASE

names<-vector(mode="character")  #### STORE THE TABLE NAMES IN THE names VECTOR

for(i in 1:24){                                                  ### R RUNS OUT OF MEMORY, READING IN
ALL OF THE RECORDS... SPLIT INTO MULTIPLE COMPONENTS
fil<-paste("Avramopoulos_1MD_051909_FinalReport",i,".csv",sep="")
x<-scan(fil,skip=11,what="character",sep=",")  ### RETRIEVE THE DATA FOR THE ith ID OF THE GIVEN TEXT
FILE, SKIPPING OVER HEADER INFORMATION
names[i]<-paste("TAB",as.character(i),sep="")
mat<-as.data.frame(matrix(x,nrow=length(x)/21,ncol=21,byrow=T))   ##### nrow = 1,199,187... I.E., NEARLY
1.2 MILLION SNPs PER INDIVIDUAL
dbWriteTable(con,names[i],mat,append=F,row.names=F)
}
dbListTables(con)
mat<-matrix(0,24,1)
for(i in 1:24){
mat[i,1]<-names[i]}
mat<-as.data.frame(mat)
dbWriteTable(con,"TABNAMES",mat,append=F,row.names=F)
rs<-dbSendQuery(con,"Select * from TABNAMES")
d1<-fetch(rs,n=-1)
dbClearResult(rs)
dbDisconnect(con)



#############################################################################################################
############################
######################### DETERMINE THE TABLES WHICH "LINE-UP" BY PARTICIPANT ID, EACH OF bl (BLOOD) AND
wga (BUCCAL) CODING ######
#############################################################################################################
############################

setwd("c:/AvramopoulosProject")
library(RSQLite)  ### LOAD THE RSQLITE LIBRARY
dbfile<-"AVRDB.dbsql"
drv<-dbDriver("SQLite")
con<-dbConnect(drv,dbname = dbfile)
mat<-matrix(0,24,2)
mat1<-matrix(0,23,3)
dbListTables(con)
rs<-dbSendQuery(con,"select * from TABNAMES")
d1<-fetch(rs,n=-1)
```

```
dbClearResult(rs)
mat[,1]<-d1[,1]

for(i in 1:24){
rs<-dbSendQuery(con,paste("select * from ",mat[i,1],sep=""))
d2<-fetch(rs,n=1)
dbClearResult(rs)
mat[i,2]<-d2[1,3]
}

mat1[,1]<-mat[-22,1]  #### "CIDR...." ESSENTIALLY, PARTICIPANT WITH ID=3025 HAS BUCCAL SNP DATA, BUT NO
BLOOD DATA -- OMIT THIS PARTICIPANT
mat1[,2]<-mat[-22,2]

for(i in 1:23){
if(substr(mat1[i,2],1,1)=='w') mat1[i,3]<-"bucc"  ### THE ID FIELD WHICH BEGINS WITH A 'w', INDICATES
BUCCAL SNP DATA
if(substr(mat1[i,2],1,1)=='b') mat1[i,3]<-"blood" ### THE ID FIELD WHICH BEGINS WITH A 'b', INDICATES
BLOOD SNP DATA

#### GET PARTICIPANT ID'S ############### EXAMPLE ID CODE:  wga_191@0074372742
pos1<-1
pos2<-1
while(substr(mat1[i,2],pos1,pos1)!='_') pos1<-pos1+1  ### WHERE DOES THE ID CODING BEGIN?
pos1<-pos1+1
pos2<-pos1
while(substr(mat1[i,2],pos2,pos2)!='@') pos2<-pos2+1  ### WHERE DOES THE ID CODING END?
pos2<-pos2-1
mat1[i,2]<-substr(mat1[i,2],pos1,pos2)
}


mat1<-mat1[order(mat1[,2]),]                          ### ORDER THE DATA BY ID
for(i in 1:23) if(sum(mat1[i,2]==mat1[,2])<2) break();  ### DETERMINE THE ROW POSITION OF THE ID FOR
WHICH ONLY A SINGLE SNP RECORD EXISTS... I.E., ID=3025 (BUCCAL DATA ONLY)

#########  THE ID'S RETRIEVED
mat2<-mat1[-i,]                    ### OMIT PARTICIPANT WITH ID=3025

dbDisconnect(con)

mat2

#### COLUMN 1 = TABLE NAME WITHIN THE DB FILE
#### COLUMN 2 = PARTICIPANT IDENTIFIER
#### COLUMN 3 = WHICH SAMPLE, BLOOD ("blood") OR BUCCAL ("bucc")

      [,1]    [,2]   [,3]
 [1,] "TAB5"  "1280" "blood"
 [2,] "TAB11" "1280" "bucc"
 [3,] "TAB10" "1467" "blood"
 [4,] "TAB16" "1467" "bucc"
 [5,] "TAB1"  "191"  "bucc"
 [6,] "TAB17" "191"  "blood"
 [7,] "TAB13" "2110" "blood"
 [8,] "TAB19" "2110" "bucc"
 [9,] "TAB9"  "2680" "bucc"
[10,] "TAB18" "2680" "blood"
[11,] "TAB4"  "3033" "bucc"
[12,] "TAB15" "3033" "blood"
[13,] "TAB12" "4034" "bucc"
[14,] "TAB21" "4034" "blood"
[15,] "TAB3"  "4149" "blood"
[16,] "TAB24" "4149" "bucc"
[17,] "TAB7"  "697"  "blood"
[18,] "TAB14" "697"  "bucc"
[19,] "TAB20" "877"  "blood"
[20,] "TAB23" "877"  "bucc"
[21,] "TAB2"  "974"  "blood"
[22,] "TAB8"  "974"  "bucc"
>

##############################################################################################
```

```
#########################################################################
#################  RETRIEVE DATA FROM THE DB FILE ###################### -- "TOP"
### CREATE A DB FILE, SNPDAT(TOP).DB, CONTAINING (FOR THE MOMENT) TWO TABLES.
### TABLE1: SNPDATBLOOD - CONTAINS THE SNP DATA FOR THE BLOOD SAMPLES
### TABLE2: SNPDATBUCC - CONTAINS THE SNP DATA FOR THE BUCCAL SAMPLES
#########################################################################
setwd("c:/AvramopoulosProject")


for(k in 1:29){   ## k = NUMBER OF SETS OF 40,000 SNPs RETRIEVED (BY READING IN A SUBSET OF SNPs AT A
TIME, WE AVOID MEMORY ISSUES IN R) --- Max(k)=29 --- RETRIEVE THE INITIAL 1,160,000 (=29*40000) SNPs

dbfile<-"AVRDB.dbsql"
drv<-dbDriver("SQLite")
con<-dbConnect(drv,dbname = dbfile)

blind<-1   ### HOW MANY BLOOD SAMPLES HAVE WE RETRIEVED?
bucind<-1  ### HOW MANY BUCCAL SAMPLES HAVE WE RETRIEVED?

for(j in 1:dim(mat2)[1]){  ### RETRIEVE THE DATA BY TABLE... A TOTAL OF DIM(mat2)[1] (=22) TABLES EXIST
WITHIN THE DB FILE (ACTUALLY, 24 EXIST... WE ONLY NEED TO RETRIEVE THE DATA FOR 22 OF THEM)
phra<-paste("select V6,V7 from ",mat2[j,1],sep="")  ## GET THE TWO ALLELE CODES FOR 40,000 SNPs
rs<-dbSendQuery(con,phra)
d2<-fetch(rs,n=40000*k)     ## RETRIEVE THE RECORDS.  ONLY 40,000 ARE NEEDED... I.E., RETRIEVE ONLY THE
"MINIMUM" NEEDED, SUCH THAT k*40000 SNPs HAVE BEEN RETRIEVED TO THIS END


if(blind==1 & mat2[j,3]=='blood')
{snp1<-cbind(c((1+40000*(k-1)):(40000*k)),d2[(1+40000*(k-1)):(40000*k),]); blind<-0
}else if(mat2[j,3]=='blood') {snp1<-cbind(snp1,d2[(1+40000*(k-1)):(40000*k),])
}else if(bucind==1 & mat2[j,3]=='bucc')
{snp2<-cbind(c((1+40000*(k-1)):(40000*k)),d2[(1+40000*(k-1)):(40000*k),]); bucind<-0
}else if(mat2[j,3]=='bucc') {snp2<-cbind(snp2,d2[(1+40000*(k-1)):(40000*k),])}

dbClearResult(rs)
} # for j
dbDisconnect(con)

#### THE snp1/snp2 DATA FRAMES (EACH) CONTAIN 40,000 SNP BI-ALLELEIC CODES FOR 11 INDIVIDUALS... I.E.,
440,000 DATA VALUES
con1<-dbConnect(drv,dbname = "SNPDAT(TOP).DB")

if(k==1){ dbWriteTable(con1,"SNPDATBLOOD",snp1,append=F,row.names=F)  ### STORE THE INITIAL 40,000 SNPs
          dbWriteTable(con1,"SNPDATBUCC",snp2,append=F,row.names=F)
}else {dbWriteTable(con1,"SNPDATBLOOD",snp1,append=T,row.names=F)      ### OTHERWISE APPEND 40,000
ADDITIONAL SNPs
       dbWriteTable(con1,"SNPDATBUCC",snp2,append=T,row.names=F)
}
dbDisconnect(con1)
write(paste("Total Sets of 40,000 SNPs: ",k,sep=""),"out.txt")  ### THIS IS A VERY TIME CONSUMING (HOURS)
PROCESS...  THIS OUTPUT FILE TELLS US WHERE TO START UP IN CASE OF A COMPUTER SHUTDOWN
}

#########################################################################################
######################### REMAINING 39,187 SNPs ###########################################
#########################################################################################

setwd("c:/AvramopoulosProject")
library(RSQLite)  ### LOAD THE RSQLITE LIBRARY
dbfile<-"AVRDB.dbsql"
drv<-dbDriver("SQLite")

con<-dbConnect(drv,dbname = dbfile)
#rs<-dbSendQuery(con,"select * from TABNAMES")  ### RETRIEVE THE TABLE NAMES (INDIVIDUAL ID'S) FROM THE
DB
#d1<-fetch(rs,n=-1)                              ### d1 CONTAINS THE UNIQUE IDENTIFIERS
#dbClearResult(rs)



blind<-1   ### HOW MANY BLOOD SAMPLES HAVE WE RETRIEVE?
bucind<-1  ### HOW MANY BUCCAL SAMPLES HAVE WE RETRIEVED?
```

```
for(j in 1:dim(mat2)[1]){  ### RETRIEVE THE DATA BY TABLE... A TOTAL OF DIM(mat2)[1] (=22) TABLES EXIST
WITHIN THE DB FILE (ACTUALLY, 24 EXIST... WE ONLY NEED TO RETRIEVE THE DATA FOR 22 OF THEM)
phra<-paste("select V6,V7 from ",mat2[j,1],sep="")  ## GET THE TWO ALLELE CODES FOR 40,000 SNPs
rs<-dbSendQuery(con,phra)
d2<-fetch(rs,n=-1)     ## RETRIEVE THE REMAINING RECORDS

if(blind==1 & mat2[j,3]=='blood') {snp1<-cbind(c((1+10000*116):(1199187)),d2[1160001:1199187,]); blind<-0
}else if(mat2[j,3]=='blood') {snp1<-cbind(snp1,d2[1160001:1199187,])
}else if(bucind==1 & mat2[j,3]=='bucc') {snp2<-cbind(c((1+10000*116):(1199187)),d2[1160001:1199187,]);
bucind<-0
}else if(mat2[j,3]=='bucc') {snp2<-cbind(snp2,d2[1160001:1199187,])}

dbClearResult(rs)
} # for j
dbDisconnect(con)


con<-dbConnect(drv,dbname = "SNPDAT(TOP).DB")
dbWriteTable(con,"SNPDATBLOOD",snp1,append=T,row.names=F)
dbWriteTable(con,"SNPDATBUCC",snp2,append=T,row.names=F)

dbDisconnect(con)


#########################################################################################################################
###


#########################################################################
#################  RETRIEVE DATA FROM THE DB FILE ###################### -- "AB"
### CREATE A DB FILE, SNPDAT(AB).DB, CONTAINING (FOR THE MOMENT) TWO TABLES.
### TABLE1: SNPDATBLOOD - CONTAINS THE SNP DATA FOR THE BLOOD SAMPLES
### TABLE2: SNPDATBUCC - CONTAINS THE SNP DATA FOR THE BUCCAL SAMPLES
#########################################################################
setwd("c:/AvramopoulosProject")


for(k in 1:29){   ## k = NUMBER OF SETS OF 40,000 SNPs RETRIEVED (BY READING IN A SUBSET OF SNPs AT A
TIME, WE AVOID MEMORY ISSUES IN R) --- Max(k)=29 --- RETRIEVE THE INITIAL 1,160,000 (=29*40000) SNPs

dbfile<-"AVRDB.dbsql"
drv<-dbDriver("SQLite")
con<-dbConnect(drv,dbname = dbfile)

blind<-1   ### HOW MANY BLOOD SAMPLES HAVE WE RETRIEVED?
bucind<-1  ### HOW MANY BUCCAL SAMPLES HAVE WE RETRIEVED?

for(j in 1:dim(mat2)[1]){  ### RETRIEVE THE DATA BY TABLE... A TOTAL OF DIM(mat2)[1] (=22) TABLES EXIST
WITHIN THE DB FILE (ACTUALLY, 24 EXIST... WE ONLY NEED TO RETRIEVE THE DATA FOR 22 OF THEM)
phra<-paste("select V6,V7 from ",mat2[j,1],sep="")  ## GET THE TWO ALLELE CODES FOR 40,000 SNPs
rs<-dbSendQuery(con,phra)
d2<-fetch(rs,n=40000*k)     ## RETRIEVE THE RECORDS.  ONLY 40,000 ARE NEEDED... I.E., RETRIEVE ONLY THE
"MINIMUM" NEEDED, SUCH THAT k*40000 SNPs HAVE BEEN RETRIEVED TO THIS END


if(blind==1 & mat2[j,3]=='blood')
{snp1<-cbind(c((1+40000*(k-1)):(40000*k)),d2[(1+40000*(k-1)):(40000*k),]); blind<-0
}else if(mat2[j,3]=='blood') {snp1<-cbind(snp1,d2[(1+40000*(k-1)):(40000*k),])
}else if(bucind==1 & mat2[j,3]=='bucc')
{snp2<-cbind(c((1+40000*(k-1)):(40000*k)),d2[(1+40000*(k-1)):(40000*k),]); bucind<-0
}else if(mat2[j,3]=='bucc') {snp2<-cbind(snp2,d2[(1+40000*(k-1)):(40000*k),])}

dbClearResult(rs)
} # for j
dbDisconnect(con)

#### THE snp1/snp2 DATA FRAMES (EACH) CONTAIN 40,000 SNP BI-ALLELEIC CODES FOR 11 INDIVIDUALS... I.E.,
440,000 DATA VALUES
con1<-dbConnect(drv,dbname = "SNPDAT(TOP).DB")

if(k==1){ dbWriteTable(con1,"SNPDATBLOOD",snp1,append=F,row.names=F)  ### STORE THE INITIAL 40,000 SNPs
          dbWriteTable(con1,"SNPDATBUCC",snp2,append=F,row.names=F)
```

```
}else {dbWriteTable(con1,"SNPDATBLOOD",snp1,append=T,row.names=F)      ### OTHERWISE APPEND 40,000
ADDITIONAL SNPs
       dbWriteTable(con1,"SNPDATBUCC",snp2,append=T,row.names=F)
}
dbDisconnect(con1)
write(paste("Total Sets of 40,000 SNPs: ",k,sep=""),"out.txt")  ### THIS IS A VERY TIME CONSUMING (HOURS)
PROCESS...  THIS OUTPUT FILE TELLS US WHERE TO START UP IN CASE OF A COMPUTER SHUTDOWN
}

#############################################################################
######################### REMAINING 39,187 SNPs ###############################
#############################################################################

setwd("c:/AvramopoulosProject")
library(RSQLite)  ### LOAD THE RSQLITE LIBRARY
dbfile<-"AVRDB.dbsql"
drv<-dbDriver("SQLite")

con<-dbConnect(drv,dbname = dbfile)
#rs<-dbSendQuery(con,"select * from TABNAMES")  ### RETRIEVE THE TABLE NAMES (INDIVIDUAL ID'S) FROM THE
DB
#d1<-fetch(rs,n=-1)                              ### d1 CONTAINS THE UNIQUE IDENTIFIERS
#dbClearResult(rs)



blind<-1   ### HOW MANY BLOOD SAMPLES HAVE WE RETRIEVE?
bucind<-1  ### HOW MANY BUCCAL SAMPLES HAVE WE RETRIEVED?

for(j in 1:dim(mat2)[1]){  ### RETRIEVE THE DATA BY TABLE... A TOTAL OF DIM(mat2)[1] (=22) TABLES EXIST
WITHIN THE DB FILE (ACTUALLY, 24 EXIST... WE ONLY NEED TO RETRIEVE THE DATA FOR 22 OF THEM)
phra<-paste("select V6,V7 from ",mat2[j,1],sep="")  ## GET THE TWO ALLELE CODES FOR 40,000 SNPs
rs<-dbSendQuery(con,phra)
d2<-fetch(rs,n=-1)     ## RETRIEVE THE REMAINING RECORDS

if(blind==1 & mat2[j,3]=='blood') {snp1<-cbind(c((1+10000*116):(1199187)),d2[1160001:1199187,]); blind<-0
}else if(mat2[j,3]=='blood') {snp1<-cbind(snp1,d2[1160001:1199187,])
}else if(bucind==1 & mat2[j,3]=='bucc') {snp2<-cbind(c((1+10000*116):(1199187)),d2[1160001:1199187,]);
bucind<-0
}else if(mat2[j,3]=='bucc') {snp2<-cbind(snp2,d2[1160001:1199187,])}

dbClearResult(rs)
} # for j
dbDisconnect(con)


con<-dbConnect(drv,dbname = "SNPDAT(AB).DB")
dbWriteTable(con,"SNPDATBLOOD",snp1,append=T,row.names=F)
dbWriteTable(con,"SNPDATBUCC",snp2,append=T,row.names=F)

dbDisconnect(con)



#############################################################################################################################
#######
################################# CREATE UPDATED BLOOD/BUCCAL TABLES
################################# 1. WE WANT ONLY AUTOSOMAL SNPs; 2. OBTAIN GENOTYPE PROPORTIONS
#############################################################################################################################
#######

setwd("c:/AvramopoulosProject")
library(RSQLite)  ### LOAD THE RSQLITE LIBRARY
dbfile<-"SNPDAT(AB).DB"
drv<-dbDriver("SQLite")
con<-dbConnect(drv,dbname = dbfile)
con1<-dbConnect(drv,dbname = dbfile)
con2<-dbConnect(drv,dbname = "AVRDB.dbsql")   #### EXTRACT THE SNP NAMES FROM THIS DB
con4<-dbConnect(drv,dbname = "SNPDAT(AB)1.DB")  ### THE DB FILE TO CONTAIN UPDATED INFO
dbListTables(con)

rs<-dbSendQuery(con,"select * from SNPDATBLOOD")
rs1<-dbSendQuery(con1,"select * from SNPDATBUCC")
```

```
rs2<-dbSendQuery(con2,"select V1 from TAB1")

for(i in 1:12){  ### PROCEED TO EXTRACT DATA BY INDIVIDUAL/ALLELE, 100,000 RECORDS AT A TIME

if(i<12) { d1<-fetch(rs,n=100000) ### RETRIEVE 100,000 RECORDS FROM BLOOD
           d2<-fetch(rs1,n=100000) ### RETRIEVE 100,000 RECORDS FROM BUCC
           d3<-fetch(rs2,n=100000) ### RETRIEVE 100,000 SNP NAMES
}else{
           d1<-fetch(rs,n=-1)
           d2<-fetch(rs1,n=-1)
           d3<-fetch(rs2,n=-1)
}

d1new<-matrix('0',dim(d1)[1],11)  ### CREATE A MATRIX TO KEEP TABS ON THE GENOTYPES
d2new<-matrix('0',dim(d2)[1],11)  ### "                                          "

for(j in 1:dim(d1new)[2]) d1new[,j]<-paste(d1[,2*j],d1[,2*j+1],sep="")  ### AFFIX GENOTYPES
for(j in 1:dim(d2new)[2]) d2new[,j]<-paste(d2[,2*j],d2[,2*j+1],sep="")

d1<-cbind(d1,apply(d1new=='AA',1,sum))  ### HOW MANY INDIVIDUALS WITH 'AA' GENOTYPE?  (BLOOD)
d1<-cbind(d1,apply(d1new=='AB',1,sum))
d1<-cbind(d1,apply(d1new=='BA',1,sum))
d1<-cbind(d1,apply(d1new=='BB',1,sum))
d1<-cbind(d1,as.matrix(apply(d1[,24:27],1,max))[,1])
d1[,28]<-d1[,28]/as.matrix(apply(d1[,24:27],1,sum))[,1]

d2<-cbind(d2,apply(d2new=='AA',1,sum))  ### HOW MANY INDIVIDUALS WITH 'AA' GENOTYPE?  (BUCC)
d2<-cbind(d2,apply(d2new=='AB',1,sum))
d2<-cbind(d2,apply(d2new=='BA',1,sum))
d2<-cbind(d2,apply(d2new=='BB',1,sum))
d2<-cbind(d2,as.matrix(apply(d2[,24:27],1,max))[,1])
d2[,28]<-d2[,28]/as.matrix(apply(d2[,24:27],1,sum))[,1]

names(d1)[24:27]<-c('AA','AB','BA','BB')
names(d1)[1]<-"SNPID"
names(d1)[28]<-"PROPVAR"
d1[,1]<-d3[,1]
d1<-cbind(d1,(substr(d1[,1],1,2)=='rs')*1)  ### AUTOSOMAL SNP ID
names(d1)[29]<-'AUTOSNPIND'                 ### AUTOSOMAL SNP INDICATOR COLUMN
d1<-d1[d1[,29]>0,]                          ### KEEP ONLY AUTOSOMAL SNPs

names(d2)[24:27]<-c('AA','AB','BA','BB')
names(d2)[1]<-"SNPID"
d2[,1]<-d3[,1]
d2<-cbind(d2,(substr(d2[,1],1,2)=='rs')*1)
names(d1)[28]<-"PROPVAR"
names(d2)[29]<-'AUTOSNPIND'
d2<-d2[d2[,29]>0,]

if(i==1) {dbWriteTable(con4,"BLOODUP",d1,append=F,row.names=F);
dbWriteTable(con4,"BUCCUP",d2,append=F,row.names=F)
} else {dbWriteTable(con4,"BLOODUP",d1,append=T,row.names=F);
dbWriteTable(con4,"BUCCUP",d2,append=T,row.names=F)}

} ## for i
dbClearResult(rs)
dbClearResult(rs1)
dbClearResult(rs2)
dbDisconnect(con)
dbDisconnect(con1)
dbDisconnect(con2)
dbDisconnect(con4)

#########################################################################################################
##################################  PREPARE THE DATA FOR PLINK ##########################################
#########################################################################################################

setwd("c:/AvramopoulosProject")
library(RSQLite)  ### LOAD THE RSQLITE LIBRARY
drv<-dbDriver("SQLite")
con<-dbConnect(drv,dbname = "SNPDAT(AB)1.DB")
dbListTables(con)
rs<-dbSendQuery(con,"select * from BLOODUP")       ### WE ARE GOING TO RUN THE HWE TEST IN PLINK, USING
```

```
THE BLOOD SNP DATA

i<-0
d.im<-0
while(dbHasCompleted(rs)==FALSE){
d1<-fetch(rs,n=200000)
d.im<-d.im+dim(d1)[1]
for(j in 2:23) d1[d1[,j]=='-',j]<-'0'  ##### CHANGE MISSING ALLELE VALUES TO ZEROS
d1[,1]<-c((d.im-dim(d1)[1]+1):d.im)
if(i==0)
{write.table(cbind(rep(1,dim(d1)[1]),d1[,1],rep(0,dim(d1)[1]),rep(0,dim(d1)[1]),d1[,2:23]),"AVR.tped",row
.names=F,col.names=F,quote=F)
i<-1
} else
write.table(cbind(rep(1,dim(d1)[1]),d1[,1],rep(0,dim(d1)[1]),rep(0,dim(d1)[1]),d1[,2:23]),"AVR.tped",row.
names=F,col.names=F,quote=F,append=T)
} ### while

dbClearResult(rs)
dbDisconnect(con)
d.im               ########    1145510

################## EXECUTE PLINK ON BLOOD SAMPLES ###################################

setwd("c:/AvramopoulosProject")

system(paste(Sys.getenv("COMSPEC"),"/c","AVR.bat"),show.output.on.console = F,invisible=F)
Sys.sleep(120)
#### WAIT UNTIL PLINK COMPLETES, PRIOR TO MOVING ON HERE ########### ~ 2 MINUTES

options(warn=-1)  ### SUPRESS WARNING MESSAGES
setwd("c:/AvramopoulosProject")
d<-read.table("plink.hwe",nrows=-1,header=T,fill=T,colClasses="character")
dim(d)  ### 1145510       9
names(d)
### [1] "CHR"    "SNP"    "TEST"   "A1"     "A2"     "GENO"   "O.HET." "E.HET."
### [9] "P"

unique(d$E.HET.[d$P==""])  ### "" "1"
unique(d$O.HET.[d$P==""])  ### "0" "1"
sum(d$E.HET.[d$P==""]=="")  ### 8870

d2<-d[d$E.HET.=="" & d$P=="",]
d2[1:10,]
unique(d2$O.HET)   #### "1"

########################################## MERGE HWE P-VALUES INTO THE BLOOD GENOTYPE SAMPLES ######

library(RSQLite)  ### LOAD THE RSQLITE LIBRARY
drv<-dbDriver("SQLite")
con<-dbConnect(drv,dbname = "SNPDAT(AB)1.DB")
con1<-dbConnect(drv,dbname = "SNPDAT(AB)2.DB")  ### UPDATED DB FILE
dbListTables(con)
rs<-dbSendQuery(con,"select * from BLOODUP")

i<-0
di.m<-1
while(dbHasCompleted(rs)==FALSE){
d1<-fetch(rs,n=100000)
d1[,1]<-c(di.m:(di.m+dim(d1)[1]-1))
d1<-cbind(d1,d[di.m:(di.m+dim(d1)[1]-1),9])
di.m<-di.m+dim(d1)[1]
names(d1)[30]<-'HWEPVAL'
d1[apply(d1[,24:27],1,sum)==0,28]<-0


if(i==0) dbWriteTable(con1,"BLOODUP",d1,append=F,row.names=F)
else dbWriteTable(con1,"BLOODUP",d1,append=T,row.names=F)
i<-i+1
} ### while
dbClearResult(rs)
dbDisconnect(con)
dbDisconnect(con1)
```

```
########################################## CREATE THE CONCORDANCE MAP DATABASE ########################

setwd("c:/AvramopoulosProject")

library(RSQLite)  ### LOAD THE RSQLITE LIBRARY
drv<-dbDriver("SQLite")
con<-dbConnect(drv,dbname = "SNPDAT(AB)1.DB")       #### EXTRACT BUCC SAMPLE/SNP IDs SNP DATA FROM THIS
DATABASE FILE
con1<-dbConnect(drv,dbname = "SNPDAT(AB)2.DB")      #### EXTRACT BLOOD SAMPLE/HWE/PROPORTION SNP DATA
FROM THIS DATABASE FILE
con2<-dbConnect(drv,dbname = "SNPDATCONCMAP.DB")    #### STORE SNP IDs, PROPORTION VARIATION, HWE
PVALUES, AND CONCORDANCE MAP

rs<-dbSendQuery(con,"select * from BUCCUP")
rs1<-dbSendQuery(con1,"select * from BLOODUP")

i<-0                                  #### INDICATOR i=1 IF THE LOOP HAS CYCLED ONCE THROUGH, i=0
OTHERWISE
while(dbHasCompleted(rs)==FALSE){     #### CONTINUE EXTRACTING RECORDS UNTIL THERE ARE NO FURTHER
RECORDS TO EXTRACT
d<-fetch(rs,n=100000)                 #### GET 100,000 SNPs AT A TIME FROM THE BUCCAL RECORDS
d1<-fetch(rs1,n=100000)               #### GET 100,000 SNPs AT A TIME FROM THE BLOOD RECORDS

for(j in 2:23){                        #### SNP CODING FOR BI-ALLELIC ALLELES BEGINS WITH COLUMN 2 (11
PARTICIPANTS)
if(j==2) conc<-(d[,j]==d1[,j])*1       #### DO THE JTH COLUMN OF ALLELE CODES "LINE-UP" BETWEEN THE
BUCCAL AND BLOOD SAMPLES?
else conc<-cbind(conc,(d[,j]==d1[,j])*1) #### APPEND TO THE DATA FRAME
}

conc<-as.data.frame(conc)
names(conc)<-mat2[,2]
row.names(conc)<-d[,1]
conc<-as.data.frame(cbind(conc,d1[,24:30]))   #### APPEND HWE P-VALUE, PROPORTION OF SNP VARIATION TO THE
CONCORDANCE MAP



if(i==0) {dbWriteTable(con2,"CONCORDANCEMAP",conc,append=F,row.names=T);i<-1
} else dbWriteTable(con2,"CONCORDANCEMAP",conc,append=T,row.names=T)
}


dbClearResult(rs); dbClearResult(rs1)
dbDisconnect(con); dbDisconnect(con1); dbDisconnect(con2)




###################### CONCORDANCE MAPS ####################################################



setwd("c:/AvramopoulosProject")

library(RSQLite)  ### LOAD THE RSQLITE LIBRARY
drv<-dbDriver("SQLite")
con1<-dbConnect(drv,dbname = "SNPDATCONCMAP.DB")
dbListTables(con1)
rs<-dbSendQuery(con1,"select * from CONCORDANCEMAP")
d1<-fetch(rs,n=-1)
dbClearResult(rs)
dbDisconnect(con1)
dim(d1)               #### 1145510      30


(sum(d1[,30]=='')-8870)/dim(d1)[1]  #### 0.2278225 ~ 22.78% of SNPs HAVE HWE P-VALUES EQUAL TO ZERO

sum(apply(d1[,24:27],1,sum)==0)   #### 8870 --- MISSING ALL GENOTYPES
d<-d1[apply(d1[,24:27],1,sum)!=0,]  ###### GET RID OF THE OBSERVATIONS W/MISSING GENOTYPES
```

```
dim(d)                              ####  1136640      30

d[d[,30]=='',30]<-0                 #### REPLACE '' HWE P-VALUES WITH ZEROS

apply(d[,2:23],2,sum)/dim(d)[1]     #### OVERALL CONCORDANCE

    X1280  X1280__1     X1467  X1467__1      X191   X191__1     X2110  X2110__1
0.9386516 0.9402968 0.7370883 0.7489240 0.9735105 0.9742223 0.9194925 0.9229466
    X2680  X2680__1     X3033  X3033__1     X4034  X4034__1     X4149  X4149__1
0.9801793 0.9806667 0.9559298 0.9585709 0.9716656 0.9724425 0.9810767 0.9814708
     X697   X697__1      X877   X877__1      X974   X974__1
0.4741017 0.5129205 0.7387291 0.7540884 0.9772795 0.9780757

dim(d[d[,30]!=0,])    ####  875667 30     SO, THERE ARE 875,667 SNPs WITH HWE P-VALUES NOT EQUAL TO ZERO


apply(d[d[,30]!=0,2:23],2,sum)/875667      #### CONCORDANCE FOR THE NON-ZERO HWE P-VALUES


   X1280  X1280__1     X1467  X1467__1      X191   X191__1     X2110  X2110__1
0.9392200 0.9402924 0.7368554 0.7462289 0.9754244 0.9757294 0.9237713 0.9263636
    X2680  X2680__1     X3033  X3033__1     X4034  X4034__1     X4149  X4149__1
0.9816346 0.9818378 0.9601549 0.9625326 0.9723262 0.9727294 0.9829890 0.9831466
     X697   X697__1      X877   X877__1      X974   X974__1
0.4666728 0.4966557 0.6820218 0.7000675 0.9802585 0.9805417
```