**PennCNV**

**http://penncnv.openbioinformatics.org/en/latest/**

Crop files

cut -f 1-6 FullDataTable2.txt >sample01\_tob220\_P10.txt

cut -f 1-3,7-9 FullDataTable2.txt >sample02\_tob198\_P9.txt

cut -f 1-3,10-12 FullDataTable2.txt >sample03\_tob205\_P10.txt

cut -f 1-3,13-15 FullDataTable2.txt >sample04\_tob206\_P11.txt

cut -f 1-3,16-18 FullDataTable2.txt >sample05\_tob220\_P21.txt

cut -f 1-3,19-21 FullDataTable2.txt >sample06\_tob198\_P32.txt

cut -f 1-3,22-24 FullDataTable2.txt >sample07\_tob074\_P10.txt

cut -f 1-3,25-27 FullDataTable2.txt >sample08\_tob206\_P19.txt

cut -f 1-3,28-30 FullDataTable2.txt >sample09\_tob224\_P18.txt

cut -f 1-3,31-33 FullDataTable2.txt >sample10\_hSARM1.txt

cut -f 1-3,34-36 FullDataTable2.txt >sample11\_hSARM1\_K193R.txt

cut -f 1-3,37-39 FullDataTable2.txt >sample12\_hSARM1\_DN.txt

cut -f 1-3,25-27 fulldatatable.txt >tob220\_mt.txt

Convert pfb to txt

awk '{print "chr"$2"\t"$3"\t"$3+1"\t"$1"\t"$4}' hc12v1.hg18.pfb > pfb\_to\_usc.txt

Convert gcmodel to txt

awk '{print "chr"$2"\t"$3"\t"$3+1"\t"$1"\t"$4}' hc12v1.hg18.gcmodel > gcmodel\_to\_usc.txt

Use gcsc genome browser liftover tool to convert pfb and gcmodel from hg18 to hg19

Rename all samples as sample\_\*, for example if sample named TOB220.txt, rename as sample\_TOB220.txt

#CNV calling for autosomes with gc model adjustment

perl detect\_cnv.pl -test -hmm lib/hhall.hmm -pfb hg19finalpfb.pfb sample\*.txt -log sampleall.adjusted.log -out sampleall.adjusted.rawcnv -gcmodel hg19finalgcmodel.gcmodel

perl detect\_cnv.pl -test -hmm lib/hhall.hmm -pfb H9WT\_xy\_as\_x\_pfb.pfb --chrx H9\_CLN3\_xy\_as\_x.txt -log H9\_CLN3\_autosome.log -out H9\_CLN3\_autosome.adjusted.rawcnv -gcmodel hg19finalgcmodel.gcmodel

#CNV calling for chr X

perl detect\_cnv.pl -test -hmm lib/hhall.hmm -pfb hg19finalpfb.pfb --chrx sample\*.txt -log sampleall\_gender.adjusted.log -out sampleall\_gender.adjusted.rawcnv -gcmodel hg19finalgcmodel.gcmodel

#CNV calling for chr XY- rename chr xy to chr x, then compare against parental pfb

perl detect\_cnv.pl -test -hmm lib/hhall.hmm -pfb x\_fib\_CLN3\_pfb.pfb --chrx x\_IPS\_CLN3.txt -log x\_iPS\_CLN32.log -out x\_IPS\_CLN32.rawcnv -gcmodel hg19finalgcmodel.gcmodel

If using docker:

# ”desktop/test” directory has hmm, pfb, sample.txt, gcmodel file

sudo docker run -v ${PWD}://home/ubuntu/penncnv -w /home/ubuntu/penncnv/ romanhaa/penncnv detect\_cnv.pl -test -hmm hhall.hmm -pfb hg19finalpfb.pfb sample\*.txt -log sampleall.log -out sampleall.rawcnv -gcmodel hg19finalgcmodel.gcmodel

sudo docker run -v ${PWD}://home/ubuntu/Desktop/test -w /home/ubuntu/Desktop/test romanhaa/penncnv detect\_cnv.pl -test -hmm hhall.hmm -pfb hg19finalpfb.pfb tob220\_lys.txt -log tob220\_lys2.log -out tob220\_lys2.rawcnv -gcmodel hg19finalgcmodel.gcmodel

sudo docker run -v ${PWD}://home/ubuntu/Desktop/test -w /home/ubuntu/Desktop/test romanhaa/penncnv detect\_cnv.pl -test -hmm hhall.hmm -pfb TOB220.pfb tob220\_lys.txt -log tob220\_lys.log -out tob220\_lys.rawcnv

**QuantiSNP**

[**https://sites.google.com/site/quantisnp/**](https://sites.google.com/site/quantisnp/)

**#to change drive**

**C:**

Set the path by doing one of the following:

NOTE: <mcr\_root> is the directory where MCR is installed

on the target machine.

On Windows systems:

\* Add the MCR directory to the environment variable by opening

a command prompt and issuing the DOS command:

**set PATH=C:\Program Files (x86)\MATLAB\MATLAB Compiler Runtime\v710\runtime\win32;%PATH%**

CdProgram Files (x86)

Cd QuantiSNP

#this detects snp in sex chromosome as well

C:\Program Files (x86)\QuantiSNP>**quantisnp2.exe --outdir "C:\Users\schear\Desktop\output" --config "C:\Program Files (x86)\QuantiSNP\params.dat" --levels "C:\Program Files (x86)\QuantiSNP\levels.dat" --gcid C:\Program Files (x86)\QuantiSNP\b37 --plot --genotype --sampleid H9\_WT --input-files "C:\Users\schear\Desktop\signal\H9WT.txt”**

C:\Program Files (x86)\QuantiSNP>**quantisnp2.exe --outdir "C:\Users\schear\Desktop\output" --config "C:\Program Files (x86)\QuantiSNP\params.dat" --levels "C:\Program Files (x86)\QuantiSNP\levels.dat" --gcid C:\Program Files (x86)\QuantiSNP\b37 --plot --genotype --sampleid H9\_CLN3 --input-files "C:\Users\schear\Desktop\signal\H9CLN3.txt”**

C:\Program Files (x86)\QuantiSNP>quantisnp2.exe --outdir "C:\Users\schear\Desktop\output" --config "C:\Program Files (x86)\QuantiSNP\params.dat" --levels "C:\Program Files (x86)\QuantiSNP\levels.dat" --gcid C:\Program Files (x86)\QuantiSNP\b37 --plot --genotype --sampleid FIB\_CLN3\_2 --input-files "C:\Users\schear\Desktop\signal\FIB\_CLN3\_2.txt”

**R-manhattan plot : method 1 to view without Chr XY**

library(qqman)

library(ggplot2)

library(magrittr)

library(stringr)

library(tidyverse)

library(dplyr)

#export table with Name,Chr,Position,samples from genomestudio,and save as fulltable.csv

setwd("P:/project/43. virtual karyotyping/R manhattan plot")

data<-read.csv("fulltable.csv", header=T, sep=",")

head(data)

#select samples

data2<-data%>%select(Name,Chr,Position,**IPS\_G1\_2**.B.Allele.Freq)

head(data2)

#remove chr XY

data2<-subset(data2, Chr!="XY")

#replace chr X with integer 23

data3 <- data2 %>%

mutate\_all(funs(str\_replace(., "X", "23")))

#replace chr Y with integer 24

data4 <- data3 %>%

mutate\_all(funs(str\_replace(., "Y", "24")))

#replace all NaN with NA

data5<-data4 %>% mutate\_if(is.character, str\_replace\_all, pattern ='NaN', replacement = 'NA')

#rename fourth column to Baf

names(data5)[4] <- "Baf"

head(data5)

#omit all NAs

data6<-subset(data5, Baf!="NA")

head(data6)

#change column class to numeric

data6$Chr<-as.numeric(as.character(data6$Chr))

data6$Position<-as.numeric(as.character(data6$Position))

data6$Baf<-as.numeric(as.character(data6$Baf))

#check class, ensure Chr, Position,Baf columns are numerical

str(data6)

#manhattan plot

manhattan(data6,bp="Position", p = "Baf",chr="Chr", snp="Name",logp = FALSE, ylab = "B Allele Frequency", genomewideline = FALSE, cex=0.8,chrlabs = c(1:24),suggestiveline = FALSE, main = "IPS\_G1\_2")

**#Method 2: to include chr XY as well**

library(qqman)

library(ggplot2)

library(magrittr)

library(stringr)

library(tidyverse)

library(dplyr)

setwd("P:/project/43. virtual karyotyping/R manhattan plot")

data<-read.csv("fulltable.csv", header=T, sep=",")

head(data)

#select samples

data2<-data%>%select(Name,Chr,Position,**TOB220**.B.Allele.Freq)

head(data2)

#select Chr XY and rename as 25

data3<-dplyr::filter(data2, Chr=="XY")

data4<-data3%>%mutate\_all(funs(str\_replace(.,"XY","25")))

#bind data4 and data2

data5<-rbind(data2,data4)

#remove overlapping rows Chr XY

data6<-dplyr::filter(data5,Chr!="XY")

#replace chr X as integer 23

data7 <- data6 %>%

mutate\_all(funs(str\_replace(., "X", "23")))

#replace chr Y as integer 24

data8 <- data7 %>%

mutate\_all(funs(str\_replace(., "Y", "24")))

#replace all NaN with NA

data9<-data8 %>% mutate\_if(is.character, str\_replace\_all, pattern ='NaN', replacement = 'NA')

#rename fourth column to Baf

names(data9)[4] <- "Baf"

head(data9)

#omit all rows with NAs

data10<-subset(data9, Baf!="NA")

#check str

str(data10)

#change column class to numeric

data10$Chr<-as.numeric(as.character(data10$Chr))

data10$Position<-as.numeric(as.character(data10$Position))

data10$Baf<-as.numeric(as.character(data10$Baf))

#check class, ensure Chr, Position,Baf columns are numerical

str(data10)

#manhattan plot

manhattan(data10,bp="Position", p = "Baf",chr="Chr", snp="Name",logp = FALSE, ylab = "B Allele Frequency", genomewideline = FALSE, cex=0.8,suggestiveline = FALSE, main = "IPS\_TOB220")

**#method 3 : to view chr XY as part of chr X**

#method 2 to view Chromosome XY as part of Chr X

setwd("P:/PhD project/43. virtual karyotyping/R manhattan plot")

data<-read.csv("fulltable.csv", header=T, sep=",")

head(data)

#select samples

data2<-data%>%select(Name,Chr,Position,TOB220.B.Allele.Freq)

head(data2)

#select Chr XY and rename as 25

data3<-dplyr::filter(data2, Chr=="XY")

#rename Chr XY as Chr X

data4<-data3%>%mutate\_all(funs(str\_replace(.,"XY","X")))

#bind data4 and data2

data5<-rbind(data2,data4)

#remove overlapping rows Chr XY

data6<-dplyr::filter(data5,Chr!="XY")

data7 <- data6 %>%

mutate\_all(funs(str\_replace(., "X", "23")))

data8 <- data7 %>%

mutate\_all(funs(str\_replace(., "Y", "24")))

data9<-data8 %>% mutate\_if(is.character, str\_replace\_all, pattern ='NaN', replacement = 'NA')

#rename fourth column to Baf

names(data9)[4] <- "Baf"

head(data9)

#omit all NAs

data10<-subset(data9, Baf!="NA")

#check str

str(data10)

#change column class to numeric

data10$Chr<-as.numeric(as.character(data10$Chr))

data10$Position<-as.numeric(as.character(data10$Position))

data10$Baf<-as.numeric(as.character(data10$Baf))

#check class

str(data10)

#manhattan plot

manhattan(data10,bp="Position", p = "Baf",chr="Chr", snp="Name",logp = FALSE, ylab = "B Allele Frequency", genomewideline = FALSE,

cex=0.8,suggestiveline = FALSE, main = "IPS\_TOB220")