**This file contains my comments/explanations on the script here**

**(**[**https://github.com/heejungshim/multiscale\_analysis/blob/master/analysis/simulation/sample\_size/simulation\_manydsQTL\_v1/prepareData/script/prepare.data.for.simulation.R**](https://github.com/heejungshim/multiscale_analysis/blob/master/analysis/simulation/sample_size/simulation_manydsQTL_v1/prepareData/script/prepare.data.for.simulation.R)**). My comments start with [HJ] and are in blue.**

#!/usr/bin/env Rscript

## Aim : This file contains Rscript to prepare phenotype and genotype data for 578 dsQTLs for simulations. The 578 dsQTLs are idetinfied by Shim and Stephens 2014 (see its Supplementary Materials for details of those dsQTLs). This information will be used in simulation.

## I modified two scripts: "/mnt/lustre/home/shim/wavelets/revision/code/simulation.explore.578.sites.R", "/mnt/lustre/home/shim/multiscale\_analysis/analysis/simulation/sample\_size/simulation\_footprint/code/better.effect.size.R", and `~/multiscale\_analysis/analysis/simulation/sample\_size/simulation\_578/code/prepare.data.for.simulation.R'

##

## Usage R CMD BATCH --no-save --no-restore "--args ss=$SGE\_TASK\_ID" prepare.data.for.simulation.R

## See /mnt/lustre/home/shim/multiscale\_analysis/analysis/simulation/sample\_size/simulation\_manydsQTL\_v1/com/com.Data.sh

##

##

## Copyright (C) 2014 Heejung Shim

##

## License: GPL3+

## ss = 305

args = (commandArgs(TRUE))

eval(parse(text=args[[1]]))

setwd("/mnt/lustre/home/shim/multiscale\_analysis")

multiscale.analysis.repodir <- scan(".multiscale\_analysis.repodir.txt", what=character())

**[HJ] It’s this repository here:** [**https://github.com/heejungshim/multiscale\_analysis/**](https://github.com/heejungshim/multiscale_analysis/)

WaveQTL.repodir <- scan(".WaveQTL.repodir.txt", what=character())

**[HJ] It’s this repository here:** [**https://github.com/heejungshim/WaveQTL**](https://github.com/heejungshim/WaveQTL)

## set working directory

wd.path = paste0(multiscale.analysis.repodir, "/analysis/simulation/sample\_size/simulation\_manydsQTL\_v1/prepareData/")

setwd(wd.path)

## set path to files

## Path to directory which contain DNase-seq data as hdf5 format,

hdf5.data.path = "/mnt/lustre/data/internal/genome\_db/hg18/dnase/"

**[HJ] new path in spartan: /data/cephfs/punim0614/shared/shared\_data/internal/multi.scale/WaveQTL/hg18/dnase/**

## Path to mappability information as hdf5 format

hdf5.mapp.path = "/mnt/lustre/data/internal/genome\_db/hg18/mappability/roger\_20bp\_mapping\_uniqueness.h5"

**[HJ]new path in spartan:**

**/data/cephfs/punim0614/shared/shared\_data/internal/multi.scale/WaveQTL/hg18/mappability/roger\_20bp\_mapping\_uniqueness.h5**

## path to directory which contains information on SNPs located region of interest

geno.info.dir.path = "/mnt/lustre/home/shim/wavelets/data/DNase/geno\_01\_step1/geno/"

**[HJ] new path in spartan:**

**/data/cephfs/punim0614/shared/shared\_data/internal/multi.scale/WaveQTL/DNase/geno\_01\_step1/geno/**

## path to directory which contains genotype data

geno.dir.path = "/mnt/lustre/home/shim/wavelets/data/DNase/geno\_01\_step1/geno\_maf/"

**[HJ] new path in spartan:**

**/data/cephfs/punim0614/shared/shared\_data/internal/multi.scale/WaveQTL/DNase/geno\_01\_step1/geno\_maf/**

This is a subset of the above’s /geno/, filtered down based on 4th column > 0.05 (minor allele frequency)

The files here are maf\_chr1.1.geno (maf chromosome 1.1)

## path to directory which contains location information on 578 sites

locus.path = "/mnt/lustre/home/shim/wavelets/data/DNase/region\_01\_sel\_step1/"

**[HJ] new path in spartan:**

**/data/cephfs/punim0614/shared/shared\_data/internal/multi.scale/WaveQTL/DNase/region\_01\_sel\_step1/**

**This contains information on 50K randomly selected regions. 578 regions are a subset of 50K regions.**

Don’t really need the info on the 578 regions as that was WaveQTL. We want to go further here (to the 50k).

## path to output directory

output.path = paste0(multiscale.analysis.repodir, "/analysis/simulation/sample\_size/simulation\_manydsQTL\_v1/data/")

## read a list of individual IDs

inds.IDs = scan(paste0(WaveQTL.repodir, "/data/Shim\_2014\_etc/DNaseI.individuals.oneline.txt"), what="")

**[HJ] new path in spartan:**

**/data/cephfs/punim0614/shared/shared\_data/internal/multi.scale/WaveQTL/YRI70\_genotype/DNaseI.individuals.oneline.txt**

**It contains YRI 70 individuals’ IDs.**

**Or, you can find it here:**

[**https://github.com/heejungshim/WaveQTL/tree/master/data/Shim\_2014\_etc**](https://github.com/heejungshim/WaveQTL/tree/master/data/Shim_2014_etc)

Struggling to read in on server. Click on the link in github.

## read functions to read DNase data and preprocess

source(paste0(multiscale.analysis.repodir, "/src/R/prepare.DNase.funcs.R"))

**[HJ] path in github:**

[**https://github.com/heejungshim/multiscale\_analysis/blob/master/src/R/prepare.DNase.funcs.R**](https://github.com/heejungshim/multiscale_analysis/blob/master/src/R/prepare.DNase.funcs.R)

**This contains the function “read.DNase.data” which we will use later.**

This function is documented in comments\_on\_mycode\_2.docx

source(paste0(multiscale.analysis.repodir, "/src/R/utils.R"))

**[HJ] path in github:**

[**https://github.com/heejungshim/multiscale\_analysis/blob/master/src/R/utils.R**](https://github.com/heejungshim/multiscale_analysis/blob/master/src/R/utils.R)

**This contains the function “get.counts.h5” which is called in the function “read.DNase.data”.**

The above function contains get.counts.h5, which is found in the above github link.

## read information on 578 sites

data = read.table("/mnt/lustre/home/shim/wavelets/revision/etc/simu.578.sites.txt", header=T)

##names(data)

##[1] "index" "chr" "site" "genoIX" "FDR.10.wave"

**[HJ] path in spartan:**

**/data/cephfs/punim0614/shared/shared\_data/internal/multi.scale/WaveQTL/DNase/etc/simu.578.sites.txt**

**We won’t focus on 578 sites. “chr” indicates chromosome of the region. “site” indicates row number of the region in “chrX.loc” file in /data/cephfs/punim0614/shared/shared\_data/internal/multi.scale/WaveQTL/DNase/region\_01\_sel\_step1/**

**“genoIX” indicates the index (in chrX.X.geno file in /data/cephfs/punim0614/shared/shared\_data/internal/multi.scale/WaveQTL/DNase/geno\_01\_step1/geno\_maf/) of the SNP with strongest association – we probably won’t use this information because we are supposed to run for all SNPs (in cis area with MAF > 0.05).**

chr.list = data$chr

site.list = data$site

genoIX.list = data$genoIX

## for each dsQTL, let's prepare data

#for(ss in 1:578){

chr = chr.list[ss]

site = site.list[ss]

genoIX = genoIX.list[ss]

## read location information

**[HJ] the following code extract chromosome name (“chrIX”), start position of region (“locus.start”), and end position of region (“locus.end”) from “chrX.loc” files in /data/cephfs/punim0614/shared/shared\_data/internal/multi.scale/WaveQTL/DNase/region\_01\_sel\_step1/**

path = paste0(locus.path, "chr", chr, ".loc")

loc\_dat = read.table(path, as.is = TRUE)

chrIX = loc\_dat[site,1]

locus.start = loc\_dat[site,2]

locus.end = loc\_dat[site, 3] - 1

## path to genotype information (to correct for cutting preference)

geno.info.path = paste0(geno.info.dir.path, "maf\_chr", chr, ".", site, ".geno")

**[HJ] these files (“maf\_chrX.X.geno” in /data/cephfs/punim0614/shared/shared\_data/internal/multi.scale/WaveQTL/DNase/geno\_01\_step1/geno/) contain information (e.g, position) on all SNPs in cis-area for a given region. We will use this position information to handle cutting preference inside “read.DNase.data” function.**

## run function to read DNase data

res = read.DNase.data(hdf5.data.path = hdf5.data.path, hdf5.mapp.path = hdf5.mapp.path, geno.info.path = geno.info.path, inds.IDs = inds.IDs, chrIX = chrIX, locus.start = locus.start , locus.end = locus.end)

**[HJ] this function is implemented in** [**https://github.com/heejungshim/multiscale\_analysis/blob/master/src/R/prepare.DNase.funcs.R**](https://github.com/heejungshim/multiscale_analysis/blob/master/src/R/prepare.DNase.funcs.R)

phenoD = ceiling(res$DNase.dat)

**[HJ] DNase-seq counts from the function “read.DNase.data” (res$DNase.dat) are average from two strands. So they are not necessary counts, so using ceiling function, we made them counts data.**

**[HJ] the following code is to obtain genotypes for SNP with strongest association. But we won’t need this information.**

## get genotype informaiton

geno.path = paste0(geno.dir.path, "chr", chr, ".", site, ".geno")

genoF = read.table(geno.path, as.is = TRUE)

genoD = as.numeric(genoF[genoIX, 4:73])

## output information to files

path.output = paste0(output.path, "pheno.dat.", ss)

write.table(phenoD, path.output, row.names = FALSE, col.names = FALSE, quote = FALSE)

path.output = paste0(output.path, "orig.geno.dat.", ss)

cat(genoD, file = path.output)

We already have:

* Perfect genotype data (maf genotype)

Need to get:

* Phenotype data (the 1024 count data) for the 50k region

Chr1.1.

ChrNum.region.

SNP location:

Chr1.702091 A G 0.018 0 702091

Ie. Name (with location), minor alleles, minor allele freq, something, then the loc (again)

Mappability; not all locations are mapped as easily/well as others, so we need to account for this. That’s the mappability data – helps us take this into account

Submitting array job in bash?

**Number of regions:**

In the dir: "/data/gpfs/projects/punim0614/shared/shared\_data/internal/multi.scale/WaveQTL/DNase/region\_01\_sel\_step1/"

[bklaw@spartan-login2 region\_01\_sel\_step1]$ wc -l \*.loc

2271 chr10.loc

2924 chr11.loc

2525 chr12.loc

839 chr13.loc

1786 chr14.loc

1713 chr15.loc

2683 chr16.loc

3309 chr17.loc

854 chr18.loc

3449 chr19.loc

4745 chr1.loc

1584 chr20.loc

702 chr21.loc

1828 chr22.loc

3330 chr2.loc

2608 chr3.loc

1753 chr4.loc

2145 chr5.loc

2120 chr6.loc

2445 chr7.loc

1894 chr8.loc

2492 chr9.loc

49999 total

49,999 regions, dispersed throughout the 22 chromosomes!

**Individuals:**

NAxxxx appear to be the IDs of the 70 individuals