"How to simulate data with three clusters from real data"

Heejung Shim

February 7, 2016

Contents

1 Introduction

Ideas for simulating data from real data has been described in Chapter 5 in Supplementary Material of Shim and Stephens 2015. We will use DNase-seq data presented in Shim and Stephens 2015. You can obtain data from WaveQTL repo. Simulation scheme uses functions provided in utils_shim repo.

2 Simulation scheme

```
## set up working directory (you need to change this).
setwd('/home/hjshim/d/projects/HTS-Clustering');

## read path to repositories
WaveQTL.repodir =scan(".WaveQTL.repodir.txt",what=character())
utils_shim.repodir =scan(".utils_shim.repodir.txt",what=character())

## read functions for simulation. We will use a function 'sample.from.Binomial.
    with.Overdispersion'. See the file for detaied description of arguments.
source(paste0(utils_shim.repodir, "/R/utils_multiscale.R"))
```

```
## read DNase-seq data. This DNase-seq data has been obtained as average of two
    strands. For simulation, we convert them to count data.
pheno.dat = as.matrix(read.table(paste0(WaveQTL.repodir, "/data/dsQTL/chr17
    .10160989.10162012.pheno.dat")))
pheno.dat = ceiling(pheno.dat)

## read genotype data and convert them to three genotypes
sel_geno_IX = 11
geno.dat = as.numeric(read.table(paste0(WaveQTL.repodir, "/data/dsQTL/chr17
    .10160989.10162012.2kb.cis.geno"), as.is = TRUE)[sel_geno_IX, -(1:3)])
geno.dat = round(geno.dat)
table(geno.dat)
## geno.dat
## 0 1 2
## 28 30 12

## create three DNase-seq data sets for three genotypes.
DNase0 = pheno.dat[geno.dat==0,]
DNase1 = pheno.dat[geno.dat==1,]
DNase2 = pheno.dat[geno.dat==2,]

## sample 10 curves from each cluster. I put over.dispersion = 0.001, but you
    could chnage this value.
num.sam = 10
DNase = DNase0
total.count = as.numeric(apply(DNase, 2, sum))
mu.sig = rep(1/dim(DNase)[1], dim(DNase)[2])
over.dispersion=0.001
data0 = sample.from.Binomial.with.Overdispersion(num.sam, total.count, mu.sig,
    over.dispersion)

num.sam = 10
DNase = DNase1
total.count = as.numeric(apply(DNase, 2, sum))
mu.sig = rep(1/dim(DNase)[1], dim(DNase)[2])
over.dispersion=0.001
data1 = sample.from.Binomial.with.Overdispersion(num.sam, total.count, mu.sig,
    over.dispersion)

num.sam = 10
```

```
DNase = DNase2
total.count = as.numeric(apply(DNase, 2, sum))
mu.sig = rep(1/dim(DNase)[1], dim(DNase)[2])
over.dispersion=0.001
data2 = sample.from.Binomial.with.Overdispersion(num.sam, total.count, mu.sig,
    over.dispersion)
```