

```

##=====
#       File name: recontrData-all.R
#       Author: Jianying Li
#       Comment: used to convert normal to data with 16 variables
##=====
library(Rlab)

## OS specific directories:

mac.os  <- "/Users/lil1/"
linux   <- "~//"
windows <- "X:/"
#root <- windows
root <- mac.os

##=====
source (paste (root, "myGit/mixturemodel/Scripts/cleaningFuncs.R", sep = ""))
source (paste (root, "myGit/mixturemodel/Scripts/simDt_functions.R", sep = ""))
source (paste (root, "myGit/mixturemodel/Scripts/reconstrDtFunctions.R", sep = ""))
parameters <- para4()

## normal sample

#dt.dir <- paste (root, "/myGit/mixturemodel/cleanedData/Normal/", sep="")

## This is for phase II
dt.dir <- paste (root, "/myGit/workingWithYicheng/phase-I-data/cleanedData/Normal/", sep="")
lab <- "n"
files <- list.files (path = dt.dir, pattern=".rda")
reconed <- reconstruct(files, parameters)
normal.temp <- reconed[,-1]
dim(t(normal.temp))[1]
label <- rep("n", dim(t(normal.temp))[1])
normal.out <- cbind(t(as.data.frame(normal.temp)), as.data.frame(label))
dim(normal.out)

##=====
## olk sample
##=====

#dt.dir <- paste (root, "/myGit/mixturemodel/cleanedData/OLK/", sep="")

## This is for phase II
dt.dir <- paste (root, "/myGit/workingWithYicheng/phase-I-data/cleanedData/OLK/", sep="")

lab <- "k"
files <- list.files (path = dt.dir, pattern=".rda")
reconed <- reconstruct(files, parameters)
olk.temp <- reconed[,-1]
dim(t(olk.temp))[1]
label <- rep(lab, dim(t(olk.temp))[1])
olk.out <- cbind(t(olk.temp), as.data.frame(label))
dim(olk.out)

##=====
## oscc sample
##=====

#dt.dir <- paste (root, "/myGit/mixturemodel/cleanedData/OSCC/", sep="")

## This is for phase II
dt.dir <- paste (root, "/myGit/workingWithYicheng/phase-I-data/cleanedData/OSCC/", sep="")

lab <- "c"
files <- list.files (path = dt.dir, pattern=".rda")
reconed <- reconstruct(files, parameters)
oscc.temp <- reconed[,-1]
dim(t(oscc.temp))[1]
label <- rep(lab, dim(t(oscc.temp))[1])
oscc.out <- cbind(t(oscc.temp), as.data.frame(label))
dim(oscc.out)

#setwd(paste (root, "/myGit/mixturemodel/reconData/para2/", sep=""))
#setwd(paste (root, "/myGit/mixturemodel/reconData/para3/", sep="")) #Newly tested May 9th, 2014

setwd(paste (root, "/myGit/workingWithYicheng/phase-I-data/reconData/", sep="")) #Newly tested Feb 5th, 2016
getwd()
combined.recon <- rbind (oscc.out, olk.out, normal.out)
str(combined.recon)
rownames(combined.recon)
colnames(combined.recon)
write.table (combined.recon, "recon_3classes_para4.txt", sep="\t", col.names = NA)

```

```

#####
##      reconstruction function
#####

reconstruct <- function (files, params)
{

dt.return <- ""

for (k in 1:length(files))
{

  load(paste(dt.dir, files[k], sep=""))
  popNum = 3
  if (cleanedSample$AneuLeft == "" || length(cleanedSample$AneuLeft) == 0)
  {
    popNum = 2
  }

  if (cleanedSample$SP_count == "")
  {
    popNum = 1
  }

  #Make it 8 if greater than 8
  if (popNum == 3)
  {
    cleanedSample$AneuLeft[which(cleanedSample$AneuLeft > 8)] <- 8
  }
  #if (is.na(cleanedSample$SP_count1))
  if (cleanedSample$SP_count >=5)
  {
    ratio <- cleanedSample$FP_count/cleanedSample$SP_count
  }else{
    ratio <- params$oneSampleRatio[1]/params$oneSampleRatio[2]
  }

  w.norm <- ratio/(1+ratio)
  w.mito <- 1/(1+ratio)
  x <- seq(0,2.3, by=(2.3/512))
  x <- x[-1]

  if (popNum == 3)
  {
    y1 <- w.norm*P(x, cleanedSample$FP_mean, cleanedSample$FP_std)
    y2 <- w.mito*P(x, cleanedSample$SP_mean, cleanedSample$SP_std)
    y = y1 + y2
  }else if (popNum ==2)
  {
    if (is.na(cleanedSample$SP_mean))
    {
      cleanedSample$SP_std = params$fakeSP_std
      cleanedSample$SP_mean = params$fakeSP_mean
    }else if (is.na(cleanedSample$SP_std))
    {
      cleanedSample$SP_std = params$fakeSP_std
    }

    y1 <- (w.norm*P(x, cleanedSample$FP_mean, cleanedSample$FP_std))*params$twoSampleRatio[1]
    y2 <- (w.mito*P(x, cleanedSample$SP_mean, cleanedSample$SP_std))*params$twoSampleRatio[1]
    y3 <- P(x, params$fake_aneu_mean, params$fake_aneu_std)*params$twoSampleRatio[2]
    y = y1 + y2 + y3
  }

  prob.y <- c()
  pdf.y <- y/sum(y)
  prob.y[1] <- pdf.y[1]

  #Gettig the empirical cdf, very important here!!
  for (i in 2:length(y))
  {
    temp.prob <- pdf.y[i]
    prob.y[i] <- prob.y[i-1] + temp.prob
  }

  if (popNum == 3)
  {
    numOfAneu <- length(cleanedSample$AneuLeft)
    num2Recontr <- 9*numOfAneu
  }else{

```

```

num2Recontr <- 1000
}

simDt <- c()
seed = 12345
for (i in 1:num2Recontr)
{
    x1 <- runif(1, 0, 1)
    index <- which(prob.y < (x1 + 0.002) & prob.y > (x1 - 0.002))
    if (length(index) < 1)
    {
        index <- which(prob.y < (x1 + 0.02) & prob.y > (x1 - 0.02))
    }
    if (length(index) < 1)
    {
        index <- which(prob.y < (x1 + 0.2) & prob.y > (x1 - 0.2))
    }
    temp <- sample( index, 1)
    simDt[i] <- x[temp]
}

if (popNum == 3)
{
    simDt <- c(simDt , cleanedSample$AneuLeft)
}

bk = floor(max(simDt))*2
den <- (hist(simDt, breaks=bk)$density)

for (m in length(den):16)
{
    den[m] <- params$filler
}
dt.temp <- as.data.frame(den)
colnames(dt.temp) <- cleanedSample$sample
dt.return <- cbind(dt.return, dt.temp)
}

return (dt.return)
}

```