Check batch effect

Zhixiang Lin

3/6/2017

How bad is the batch effect? Think about GC content, etc. Need the batch information from Mahdi.

```
library(readr)
library(WeightedCluster)
```

```
## Loading required package: TraMineR

##
## TraMineR stable version 1.8-13 (Built: 2016-10-06)

## Website: http://traminer.unige.ch

## Please type 'citation("TraMineR")' for citation information.

## Loading required package: cluster

## This is WeightedCluster stable version 1.2 (Built: 2016-05-05)

##
## To get the manuals, please run:

##    vignette("WeightedCluster") ## Complete manual in English

##    vignette("WeightedClusterFR") ## Complete manual in French

##    vignette("WeightedClusterPreview") ## Short preview in English

##
## To cite WeightedCluster in publications please use:

## Studer, Matthias (2013). WeightedCluster Library Manual: A practical guide to

##    creating typologies of trajectories in the social sciences with R.

##    LIVES Working Papers, 24. doi: 10.12682/lives.2296-1658.2013.24
```

load data, K562

data matrix

```
ForeGround <- read_csv("/Users/zhixianglin/Documents/collaboration/mahdi/scATAC/K562/416Cells/ForeGroun
```

```
## Parsed with column specification:
## cols(
##   .default = col_integer()
## )

## See spec(...) for full column specifications.
```

```
ForeGround <- as.matrix(ForeGround)

BackGround <- read_csv("/Users/zhixianglin/Documents/collaboration/mahdi/scATAC/K562/416Cells/BackGroun
## Parsed with column specification:
## cols(
##   .default = col_integer()
## )
## See spec(...) for full column specifications.
```

```
BackGround <- as.matrix(BackGround)
```

batch information

```
batch <- read_delim("/Users/zhixianglin/Documents/collaboration/mahdi/scATAC/K562/416Cells/SampleOrderB
```

```
## Parsed with column specification:
## cols(
##   X1 = col_character(),
##   X2 = col_character(),
##   X3 = col_character()
## )
```

```
batch <- batch[,2][[1]]
print(table(batch))
```

```
## batch
##    CDKi Imat1hr    JNKi    rep1    rep2    rep3
##      45      69      68     115      75      44
```

run weighted K-medoid

Calculate 1 - Spearman

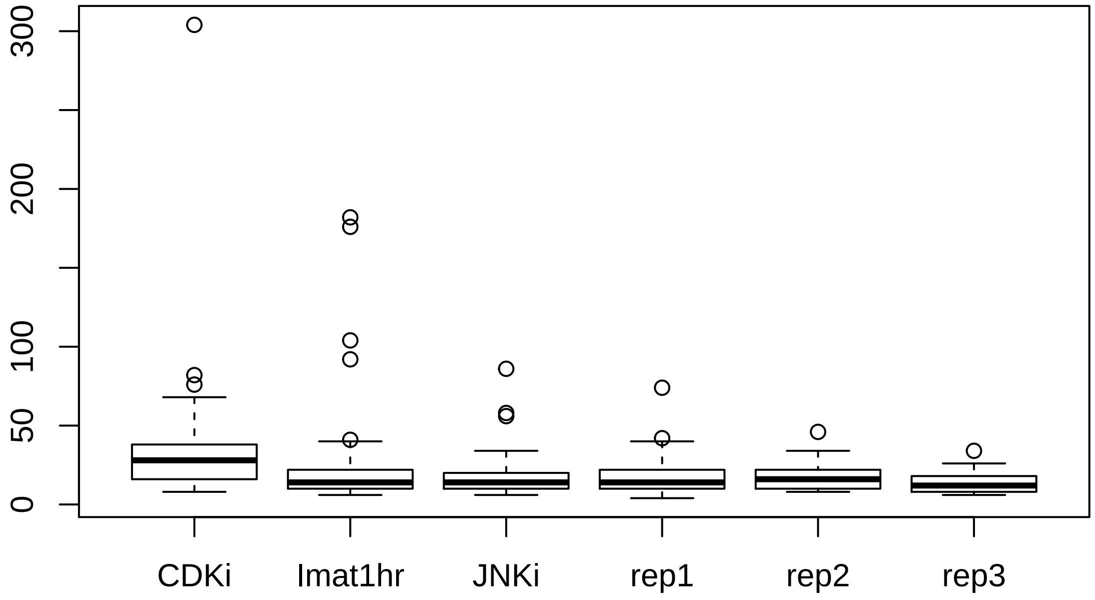
```
distS <- 1-cor(ForeGround, method="spearman")
```

Calculate the median of BackGround for each sample

```
BackGroundMedian <- apply(BackGround, 2, median)
```

Boxplot of BackGroundMedian stratified by batch

```
boxplot(BackGroundMedian~batch)
```



```
nCluster <- 6
lambda <- 1
a <- quantile(BackGroundMedian, 0.5)
W <- 1/(1+exp(-lambda*(BackGroundMedian-a)))
resultW <- wcKMedoids(distS, k=nCluster, weights=W)
clusterW <- resultW$clustering
clusterW <- as.numeric(factor(clusterW))
```

get landmark

$landmark$ is $numofpeaks \times numoflanmarks$

```
landmarks <- c()
  for (i in 1:nCluster){
    tmp <- which(clusterW==i)
    if (length(tmp)==1){
        landmarks <- cbind(landmarks, ForeGround[,tmp]  )
    } else {
        landmarks <- cbind(landmarks, rowSums(ForeGround[,tmp])  )
    }
  }
```

```
selectTop <- function(x, top){
  thres <- sort(x, decreasing=T)[top]
  x[x<thres] <- 0
  return(x)
}
```

pick the top peaks in the landmarks

```
top <- 2000
landmarksTop <- apply(landmarks, 2, selectTop, top)
```

run KNN

```
scor <- cor(ForeGround, landmarksTop, method="spearman")
clusterWKNN <- apply(scor, 1, which.max)
```

check the clustering result.

```
cells <- unique(batch)
```

```
getClusterCount <- function(cluster, samples, cells){
  ### input
  #cluster: clustering result for each sample
  #samples: vector with the cell types for each sample
  #cells: order of the cell types
  clusterCount <- matrix(0, nrow=length(unique(cluster)), ncol=length(cells))
  for (i in 1:length(unique(cluster))){
    tmp <- samples[which(cluster==i)]
    for (j in 1:length(cells)){
      cell <- cells[j]
      clusterCount[i, j] <- sum(tmp==cell)
    }
  }
  tmp <- apply(clusterCount, 2, which.max)
  if (max(table(tmp))>1){
    seqs <- c()
    for (i in 1:nrow(clusterCount)){
      if (i==1){
        seqs <- c(seqs, which.max(clusterCount[,i])[1])
      }
      if (i>1 & i<nrow(clusterCount)){
        seqs <- c(seqs, c(c(1:nrow(clusterCount))[-seqs])[which.max(clusterCount[-seqs,i])[1]])
      }
      if (i==nrow(clusterCount)){
        seqs <- c(seqs, c(1:nrow(clusterCount))[-seqs])
      }
    }
  } else {
    seqs <- tmp
  }
  clusterCount <- clusterCount[seqs,]
  row.names(clusterCount) <- paste("cluster", 1:length(unique(cluster)))
  colnames(clusterCount) <- cells
  return(clusterCount)
}
```

```
getCorrectCount <- function(clusterCount){
  ### input
  #c lusterCount: output from getClusterCount
  ### output, we assign the cell type of each cluster by the majority
  # c(the number of correctly clusterd cells, the percentage of correctly clustered cells)
  return(c( sum(apply(clusterCount, 1, max)),  sum(apply(clusterCount, 1, max))/sum(clusterCount)) )
}
```

weighted K-medoids

```
ClusterCountW <- getClusterCount(cluster=clusterW, samples=batch, cells=cells)
#CorrectCountW <- getCorrectCount(ClusterCountW)
print(ClusterCountW)
```

```
##           rep1 JNKi Imat1hr rep3 rep2 CDKi
## cluster 1   98   68      66   39   67   45
## cluster 2    4    0       0    2    1    0
## cluster 3    6    0       2    1    7    0
## cluster 4    2    0       0    2    0    0
## cluster 5    2    0       0    0    0    0
## cluster 6    3    0       1    0    0    0
```

```
#print(CorrectCountW)
```

weighted K-medoids + KNN

```
ClusterCountWKNN <- getClusterCount(cluster=clusterWKNN, samples=batch, cells=cells)
#CorrectCountWKNN <- getCorrectCount(ClusterCountWKNN)
print(ClusterCountWKNN)
```

```
##           rep1 JNKi Imat1hr rep3 rep2 CDKi
## cluster 1  100   68      66   39   68   45
## cluster 2    4    0       0    2    2    0
## cluster 3    4    0       2    1    5    0
## cluster 4    2    0       0    2    0    0
## cluster 5    2    0       0    0    0    0
## cluster 6    3    0       1    0    0    0
```

change number of clusters to 2

```
nCluster <- 2
lambda <- 1
a <- quantile(BackGroundMedian, 0.5)
W <- 1/(1+exp(-lambda*(BackGroundMedian-a)))
resultW <- wcKMedoids(distS, k=nCluster, weights=W)
clusterW <- resultW$clustering
clusterW <- as.numeric(factor(clusterW))
landmarks <- c()
for (i in 1:nCluster){
    tmp <- which(clusterW==i)
    if (length(tmp)==1){
        landmarks <- cbind(landmarks, ForeGround[,tmp]  )
    } else {
        landmarks <- cbind(landmarks, rowSums(ForeGround[,tmp])  )
    }
}
top <- 2000
landmarksTop <- apply(landmarks, 2, selectTop, top)
scor <- cor(ForeGround, landmarksTop, method="spearman")
clusterWKNN <- apply(scor, 1, which.max)
ClusterCountW <- getClusterCount(cluster=clusterW, samples=batch, cells=cells)
print(ClusterCountW)
```

```
##           rep1 JNKi Imat1hr rep3 rep2 CDKi
## cluster 1  112   68      68   44   75   45
## cluster 2    3    0       1    0    0    0
```

```
ClusterCountWKNN <- getClusterCount(cluster=clusterWKNN, samples=batch, cells=cells)
print(ClusterCountWKNN)
```

```
##           rep1 JNKi Imat1hr rep3 rep2 CDKi
## cluster 1  112   68      68   44   75   45
## cluster 2    3    0       1    0    0    0
```

```
nCluster <- 3
lambda <- 1
a <- quantile(BackGroundMedian, 0.5)
W <- 1/(1+exp(-lambda*(BackGroundMedian-a)))
resultW <- wcKMedoids(distS, k=nCluster, weights=W)
clusterW <- resultW$clustering
clusterW <- as.numeric(factor(clusterW))
landmarks <- c()
for (i in 1:nCluster){
    tmp <- which(clusterW==i)
    if (length(tmp)==1){
        landmarks <- cbind(landmarks, ForeGround[,tmp]  )
    } else {
        landmarks <- cbind(landmarks, rowSums(ForeGround[,tmp])  )
    }
}
top <- 2000
landmarksTop <- apply(landmarks, 2, selectTop, top)
scor <- cor(ForeGround, landmarksTop, method="spearman")
clusterWKNN <- apply(scor, 1, which.max)
ClusterCountW <- getClusterCount(cluster=clusterW, samples=batch, cells=cells)
print(ClusterCountW)
```

```
##           rep1 JNKi Imat1hr rep3 rep2 CDKi
## cluster 1  110   68      68   44   75   45
## cluster 2    2    0       0    0    0    0
## cluster 3    3    0       1    0    0    0
```

```
ClusterCountWKNN <- getClusterCount(cluster=clusterWKNN, samples=batch, cells=cells)
print(ClusterCountWKNN)
```

```
##           rep1 JNKi Imat1hr rep3 rep2 CDKi
## cluster 1  110   68      68   44   75   45
## cluster 2    2    0       0    0    0    0
## cluster 3    3    0       1    0    0    0
```