Single cell Project 2

Guanxun Li & Honggang Wang

4/26/2021

Introduction

The gene expression matrix of scRNA-seq data is a sparse matrix with a large portion of zeros in it. There are two reasons why this is the case. One is the presence of technical drop out due to the low concentration of mRNAs in a given individual cell. The other reason is a biological reason—that is, the cell actively switches off the transcription of a certain set of genes at any time. Statistics are needed here to distinguish between the causes of zeros in gene expression values in scRNA-seq data.

The problem proposed here concerns the computing correlation between two genes' expression levels. The correlation analysis using Spearman or Pearson correlation tests works with scRNA-seq data but zeros in the data cause the problem. There is a zero-inflated version of correlation analysis that may solve the problem, but we have not yet tested it with real data. Moreover, if two genes have a positive correlation, it means that two genes may be positively regulated by each other and if two genes have a negative correlation, it means that two genes may be negatively regulated by each other.

Furthermore, we can also treat scRNA-seq data as the binary data and calculate the binary version correlation between each pair of genes. In this case, if two genes have a positive correlation, it means that two genes will be more likely to express at the same cells and if two genes have a negative correlation, it means that two genes won't express at the same cells. It is natural to think these two correlations should be consistent, showing the same directionality—for example, a negative correlation between two genes should produce fewer overlapped cells co-expressing the two genes.

In this project, we wanted to see if there are cases two kinds of correlations give the opposite direction and how pronounced these opposite patterns can be overserved in real data sets.

Correlation Calculation

We load the data first.

```
library(Matrix)

## load data
dta <- as.matrix(readRDS("data set/dta.rds"))
p <- ncol(dta)
n <- nrow(dta)
gene_name <- rownames(dta)
```

Next is to calculate the adjusted Spearman's correlation matrix and the sign of the Phi coefficients for each pair of genes.

```
index0 <- apply(dta, 1, function(x){which(as.numeric(x) == 0)})
index1 <- lapply(index0, function(x){setdiff(seq_len(p), x)})

dta_01 <- matrix(0, nrow = n, ncol = p)
```

```r
dta_01[which(abs(dta) > 0)] <- 1
rownames(dta_01) <- rownames(dta)

## spearman correlation
spearman_mat <- matrix(NA, nrow = n, ncol = n)
cor_01 <- matrix(NA, nrow = n, ncol = n)
cor_common <- matrix(NA, nrow = n, ncol = n)

## begin calculation
for (i in seq_len(n)) {
  for (j in i:n) {
    index11 <- intersect(index1[[i]], index1[[j]])
    p11 <- length(intersect(index1[[i]],  index1[[j]])) / p
    p00 <- length(intersect(index0[[i]], index0[[j]])) / p
    p10 <- length(intersect(index1[[i]], index0[[j]])) / p
    p01 <- length(intersect(index0[[i]], index1[[j]])) / p
    px1 <- length(index1[[i]]) / p
    py1 <- length(index1[[j]]) / p
    if (length(index11) == 0) {
      rho <- 0
    } else {
      x <- as.numeric(dta[i, index11])
      y <- as.numeric(dta[j, index11])
      if (identical(x, y)) {
        rho <- 1
      } else {
        rho <- suppressWarnings(cor(x, y, method = "spearman"))
        if (is.na(rho)) {
          rho <- 0
        }
      }
    }
    tmp <- p00 * p11 - p01 * p10
    cor_common[i, j] <- cor_common[j, i] <- rho
    cor_01[i, j] <- cor_01[j, i] <- tmp
    spearman_mat[i, j] <- spearman_mat[j, i] <- p11 * px1 * py1 * rho + 3 * tmp
  }
}
cor_01[which(cor_01 > 0)] <- 1
cor_01[which(cor_01 < 0)] <- -1

spearman_01 <- matrix(NA, nrow = n, ncol = n)
spearman_01[which(spearman_mat > 0)] <- 1
spearman_01[which(spearman_mat < 0)] <- -1

rownames(cor_01) <- colnames(cor_01) <- gene_name
rownames(cor_common) <- colnames(cor_common) <- gene_name
rownames(spearman_mat) <- colnames(spearman_mat) <- gene_name
rownames(spearman_01) <- colnames(spearman_mat) <- gene_name
```

After getting two correlation sign matrices, next is to find the "contradiction pairs".

```r
## calculate the difference pair with adjusted spearman
index <- which(abs(cor_01 - spearman_01) > 0)
```

```
index0 <- which(cor_01 == 0)
index <- setdiff(index, index0)

pair_mat <- matrix(NA, nrow = 2, ncol = length(index))
pair_mat[1, ] <- index %/% n + 1
pair_mat[2, ] <- index %% n
pair_mat[2, ][which(pair_mat[2, ] == 0)] <- n

gene_pair <- matrix(NA, nrow = length(index) / 2, ncol = 2)
iter <- 1
for (i in seq_len(length(index))) {
  if (pair_mat[1, i] < pair_mat[2, i]) {
    gene_pair[iter, ] <- gene_name[c(pair_mat[1, i],  pair_mat[2, i])]
    iter <- iter + 1
  }
}
```

The last step is to use Fisher's exact test to keep gene pairs whose correlation is significiant.

```
## For each pair do fisher exact test
pair_p <- rep(NA, nrow(gene_pair))
for (i in seq_len(length(pair_p))) {
  tmp <- table(dta_01[gene_pair[i, 1], ], dta_01[gene_pair[i, 2], ])
  if (identical(as.numeric(dim(tmp)), c(2,2))) {
    pair_p[i] <- fisher.test(dta_01[gene_pair[i, 1], ], dta_01[gene_pair[i, 2], ])$p.value
  } else{
    pair_p[i] <- 1
  }
}
gene_pair <- gene_pair[which(pair_p < 0.05), ]
```

The top 6 pairs of genes are following:

```
head(gene_pair)
```

```
##      [,1]     [,2]
## [1,] "Acadl"  "Rps23"
## [2,] "Aldoa"  "Postn"
## [3,] "Anp32b" "Postn"
## [4,] "Anxa7"  "Rplp0"
## [5,] "Apoe"   "Col1a2"
## [6,] "Apoe"   "Rpl12"
```

There are totally 142 pairs of genes we found.

```
nrow(gene_pair)
```

```
## [1] 142
```

Adjusted Correlation for Binary data based on Zero Inflated Negative Binomial Model

In this case, we need to fit a ZINB model for our scRNA-seq data.

```
## Train ZINB model
library(zinbwave)
```

```
set.seed(1)
dta_zinb <- zinbFit(dta, K = 2, stop.epsilon.optimize = 1e-07, verbose = TRUE,
                    maxiter.optimize = 25, commondispersion = FALSE)
res$dta_zinb <- dta_zinb
```

Next is to get the adjusted zero inflated probability.

```
## calculate adjusted probability
mu <- exp(t(dta_zinb@X %*% dta_zinb@beta_mu + t(dta_zinb@V %*% dta_zinb@gamma_mu) +
             dta_zinb@W %*% dta_zinb@alpha_mu + dta_zinb@O_mu))
pi_zinb <- t(dta_zinb@X %*% dta_zinb@beta_pi + t(dta_zinb@V %*% dta_zinb@gamma_pi) +
               dta_zinb@W %*% dta_zinb@alpha_pi + dta_zinb@O_pi)
pi_zinb <- exp(pi_zinb) / (1 + exp(pi_zinb))
theta <- exp(matrix(dta_zinb@zeta, nrow = n, ncol = p, byrow = TRUE))
pi0 <- exp(theta * (log(theta) - log(theta + mu)))
pi_adj <- pi_zinb / (pi_zinb + pi0)
```

Based on our adjusted probability, we can calculate the new sign of the Phi coefficient matrix.

```
## begin calculation
for (i in seq_len(n)) {
  for (j in i:n) {
    ## both 1
    index11 <- intersect(index1[[i]], index1[[j]])
    n11 <- length(index11)
    ## consider modified 0
    index00 <- intersect(index0[[i]],  index0[[j]])
    tmp1 <- pi_zinb[i, index00]
    tmp2 <- pi_zinb[j, index00]
    n00 <- sum((1 - tmp1) * (1 - tmp2))
    ## consider only one 0
    index10 <- intersect(index1[[i]],  index0[[j]])
    tmp1 <- pi_zinb[j, index10]
    n10 <- sum(1 - tmp1)
    index01 <- intersect(index0[[i]],  index1[[j]])
    tmp2 <- pi_zinb[i, index01]
    n01 <- sum(1 - tmp2)
    ## joint probability
    n_new <- n11 + n00 + n10 + n01
    p11 <- n11 / n_new
    p10 <- n10 / n_new
    p01 <- n01 / n_new
    p00 <- n00 / n_new
    ## marginal probability
    px1 <- (n11 + n10) / n_new
    py1 <- (n01 + n11) / n_new
    if (length(index11) == 0) {
      rho <- 0
    } else{
      x <- as.numeric(dta[i, index11])
      y <- as.numeric(dta[j, index11])
      if (identical(x, y)) {
        rho <- 1
      } else{
        rho <- suppressWarnings(cor(x, y, method = "spearman"))
```

```
        if (is.na(rho)) {
          rho <- 0
        }
      }
    }
    tmp <- p00 * p11 - p01 * p10
    cor_common[i, j] <- cor_common[j, i] <- rho
    cor_01[i, j] <- cor_01[j, i] <- tmp
    spearman_mat[i, j] <- spearman_mat[j, i] <- p11 * px1 * py1 * rho + 3 * tmp
  }
}
cor_01[which(cor_01 > 0)] <- 1
cor_01[which(cor_01 < 0)] <- -1

spearman_01 <- matrix(NA, nrow = n, ncol = n)
spearman_01[which(spearman_mat > 0)] <- 1
spearman_01[which(spearman_mat < 0)] <- -1

rownames(cor_01) <- colnames(cor_01) <- gene_name
rownames(cor_common) <- colnames(cor_common) <- gene_name
rownames(spearman_mat) <- colnames(spearman_mat) <- gene_name
rownames(spearman_01) <- colnames(spearman_mat) <- gene_name
```

After getting two correlation sign matrices, the left thing is same as before.

```
## calculate the difference pair with adjusted spearman
index <- which(abs(cor_01 - spearman_01) > 0)
index0 <- which(cor_01 == 0)
index <- setdiff(index, index0)

pair_mat <- matrix(NA, nrow = 2, ncol = length(index))
pair_mat[1, ] <- index %/% n + 1
pair_mat[2, ] <- index %% n
pair_mat[2, ][which(pair_mat[2, ] == 0)] <- n

gene_pair <- matrix(NA, nrow = length(index) / 2, ncol = 2)
iter <- 1
for (i in seq_len(length(index))) {
  if (pair_mat[1, i] < pair_mat[2, i]) {
    gene_pair[iter, ] <- gene_name[c(pair_mat[1, i],  pair_mat[2, i])]
    iter <- iter + 1
  }
}

## For each pair do fisher exact test
pair_p <- rep(NA, nrow(gene_pair))
for (i in seq_len(length(pair_p))) {
  tmp <- table(dta_01[gene_pair[i, 1], ], dta_01[gene_pair[i, 2], ])
  if (identical(as.numeric(dim(tmp)), c(2,2))) {
    pair_p[i] <- fisher.test(dta_01[gene_pair[i, 1], ], dta_01[gene_pair[i, 2], ])$p.value
  } else{
    pair_p[i] <- 1
  }
}
```

```
gene_pair <- gene_pair[which(pair_p < 0.05), ]
```

The top 6 pairs of genes in this case are following:

```
head(gene_pair)
```

```
##      [,1]             [,2]
## [1,] "1810055G02Rik" "Top2a"
## [2,] "Aaed1"         "Tmsb10"
## [3,] "Abca8a"        "Frzb"
## [4,] "Abca8a"        "Spon2"
## [5,] "Acadl"         "Rps23"
## [6,] "Ace"           "Top2a"
```

There are totally 608 pairs of genes we found.

```
nrow(gene_pair)
```

```
## [1] 608
```

These pairs of genes include 141 of 142 pairs we found in the original method.

Summary

We use two methods in this project to calculate the association between pairs of genes and find anti-correlation pairs. The standard correlation analysis for scRNA-seq gene expression data is Spearman or Pearson correlation, but zeros in the data cause problems. As a result, for the zero-inflated data, we try an adjusted Spearman correlation. Then, by treating scRNA-seq data as binary data, we calculate the binary version similarity between each pair of genes. In this case, the positive and negative relationship is described by the sign of the *Phi* coefficients. We choose pairs of genes whose signs differ between two correlations after obtaining two association relationships. Finally, we perform the Fisher's exact test on selected pairs of genes, keeping only those with p-values less than 0.05.

In the future, we'll need to see if there are any biological implications for these pairs of genes.