mixIN: A meta-analysis method for RNA-seq count data

Birbal Prasad[1] and Xinzhong Li[1*]
[1] National Horizons Centre, School of Health and Life Sciences, Teesside University, Darlington, DL1 1HG, UK.

*Correspondence: X.Li@tees.ac.uk

Contents

1 Introduction

This document provides the way to perform meta-analysis of RNA-seq count data using the mixture inverse-normal (mixIN) method in R. A working implementation of the steps involved and described here for three different Glioblastoma (GBM) RNA-seq studies (GSE123892, GSE151352 and TCGA-GBM) can be found in *mixIN_meta_analysis.R*. Combination of data or results from multiple independent but related gene expression studies (referred to as meta-analysis) have been widely used to increase available sample size and consequently the statistical power to obtain a precise estimate of gene expression differentials. In context of integrated differential expression analysis of RNA-seq data, mixIN approach accounts for both the sample size and direction of gene regulation of a gene in each individual study. The raw p-value for per-gene and per-study obtained using the individual differential expression analysis is used in this meta-analysis method.

2 Getting started

At first, install R version 3.6.0 or above and load the required R packages *org.Hs.eg.db* [1] and *annotate* [2] which can be achieved using another *pacman* R package [3].

```
> if(!require("pacman")) install.packages("pacman")
> pacman::p_load(org.Hs.eg.db, annotate)
```

Next, we need to prepare the results from per-study differential analysis for RNA-seq data. Popular methods such as DESeq [4], edgeR [5], etc. can be used for this step. Each study results should at least contain the gene id, raw p-values and log_2(FC) (logFC) from the individual differential analaysis.

```
> head(study_1)
entrez_id    logFC     PValue
   7153     9.666761   1.12e-37
  51555    -9.087178   1.09e-36
   6241     8.980119   2.84e-35
  26289    -8.663425   2.50e-29
   9928     7.244726   2.69e-28
```

Assessment of the underlying assumption that p-values for all genes obtained from per-study differential analysis are uniformly distributed under the null hypothesis needs to be carried out .

```
 #check 1:  distribution of raw p-values from a study.  Needs to be roughly uniform under
the null hypothesis.
> h <- hist(study_1$PValue)
> plot( h, col= "red", xlim=c(0,1), main = "Study_1", xlab = "p-value")
```

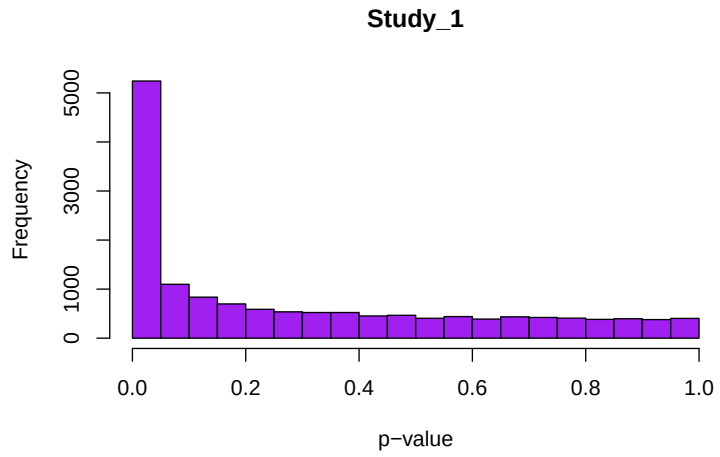

Figure 1: Histogram of the raw p-values obtained from differential expression analysis of GSE123892 using edgeR.

Usually this assumption is not satisfied in case of RNA-seq data but filtering of weakly expressed genes using the method described in Rau et al. (2014)[6] or using the counts per million criteria in Raithel et al. (2016)[7] circumvents this difficulty to a significant extent.

3 Meta-analysis by mixIN method

Once we have the results from per-study differential analysis for all the studies that are under consideration for the meta-analysis, following steps are carried out during the mixIN meta-analysis.

a. Input: Load individual study differential analysis results and sample information for each example dataset.

```
 > datasets <- c("GSE123892", "GSE151352", "TCGA_GBM")
> n_samp <- c(7, 24, 165) # sample size of each dataset
```

b. Assessment of direction of expression for each gene in each study based on the sign of the logFC obtained during individual differential expression analysis. Append this information as a new column to the results of individual analysis.

```
 > study_1$dir <- sign(study_1$logFC)
> head(study_1)
 entrez_id     logFC      PValue   dir
    7153     9.666761   1.12e-37    1
   51555    -9.087178   1.09e-36   -1
    6241     8.980119   2.84e-35    1
   26289    -8.663425   2.50e-29   -1
    9928     7.244726   2.69e-28    1
```

c. Get all unique genes among all the studies considered.

```
 > unique_genes <- Reduce(union, list(study_1$entrez_id, study_2$entrez_id, study_3$entrez_id))
```

d. For all the unique genes create a matrix with columns representing direction of expression of a gene in each study and if they are conflicting or not.

```
 > sign <- matrix(data = 0, nrow = length(unique_genes), ncol = length(n_samp)+1)
> row.names(sign) <- unique_genes
> colnames(sign) <- c("study_1", "study_2", "study_3", "conflict_status")
> sign[, 1] <- study_1$dir[match(unique_genes, study_1$entrez_id)]
> sign[, 2] <- study_2$dir[match(unique_genes, study_2$entrez_id)]
> sign[, 3] <- study_3$dir[match(unique_genes, study_3$entrez_id)]
> for (l in 1:length(unique_genes)){
> if (1 %in% sign[l, c(1:length(n_samp))] & -1%in% sign[l, c(1:length(n_samp))]){
> sign[l, (length(n_samp)+1)] <- 1}}
> head(sign)
        study_1  study_2  study_3  conflict_status
  7153      1        1        1            0
 51555     -1       -1       -1            0
  6241      1        1        1            0
386618     -1       NA       -1            0
 26289     -1       -1       -1            0
```

e. Computation of N_g statistic: To compute N_g as defined in the mixIN method, we compute the weights w_s and then each term of N_g corresponding to each gene and study. Finally, we sum all the terms of N_g for each gene g together for all the studies.

```
 # 1.  estimation of weights.

# initialize an empty weights matrix with the number of rows equal to the number of unique_gene
and columns equal to the number of studies considered

> weights <- matrix(0, nrow = length(unique_genes), ncol = length(n_samp))

# each element in the weights matrix corresponds to a unique gene in a study.
# the numerator term of w_s for each gene

> weights[which(unique_genes %in% study_1$entrez_id == TRUE), 1] <- n_samp[1]
> weights[which(unique_genes %in% study_2$entrez_id == TRUE), 2] <- n_samp[2]
> weights[which(unique_genes %in% study_3$entrez_id == TRUE), 3] <- n_samp[3]

# demominator term of w_s

> denom <- apply(weights, 1, sum)

# divide the numerator by denominator and take the square root to get the final weights

> weights <- weights/denom
> weights <- sqrt(weights)
> row.names(weights) <- as.character(unique_genes)
> colnames(weights) <- c("study_1", "study_2", "study_3")
> weights <- as.data.frame(weights, stringsAsFactors = FALSE)
> head(weights)
          study_1    study_2    study_3
  7153  0.1889822  0.3499271  0.9175166
 51555  0.1889822  0.3499271  0.9175166
  6241  0.1889822  0.3499271  0.9175166
386618  0.2017366  0.0000000  0.9794398
 26289  0.1889822  0.3499271  0.9175166
  9928  0.1889822  0.3499271  0.9175166

# 2.  calculation of each term in Ng for a gene g

> ng_terms <- matrix(0, nrow = nrow(weights), ncol = ncol(weights))
```

```
> for(j in 1:nrow(ng_terms)){

# check if a gene has the same direction of expression across studies
# if yes, then use the first case definition for N_g

> if (sign[j, ncol(sign)] == 0){
> if (unique_genes[j] %in% study_1$entrez_id){
> k = which(study_1$entrez_id == unique_genes[j])
> p_val=min(max(study_1$PValue[k],1e-16),1-1e-16)
> ng_terms[j, 1] <- weights$study_1[j] * qnorm((1-p_val), mean = 0, sd = 1)}
> if (unique_genes[j] %in% study_2$entrez_id){
> k = which(study_2$entrez_id == unique_genes[j])
> p_val=min(max(study_2$PValue[k],1e-16),1-1e-16)
> ng_terms[j, 2] <- weights$study_2[j] * qnorm((1-p_val), mean = 0, sd = 1)}
> if (unique_genes[j] %in% study_3$entrez_id){
> k = which(study_3$entrez_id == unique_genes[j])
> p_val=min(max(study_3$PValue[k],1e-16),1-1e-16)
> ng_terms[j, 3] <- weights$study_3[j] * qnorm((1-p_val), mean = 0, sd = 1)}}


# check if a gene has conflicting direction of expression across studies
# if yes, then use the second case definition for N_g

> if (sign[j, ncol(sign)] == 1){
> if (unique_genes[j] %in% study_1$entrez_id){
> k = which(study_1$entrez_id == unique_genes[j])
> p_val=min(max(study_1$PValue[k],1e-16),1-1e-16)
> ind_sign <- sign(study_1$logFC[k])
> ng_terms[j, 1] <- weights$study_1[j]* ind_sign * abs(qnorm((1-p_val), mean = 0, sd =
1))}
> if (unique_genes[j] %in% study_2$entrez_id){
> k = which(study_2$entrez_id == unique_genes[j])
> p_val=min(max(study_2$PValue[k],1e-16),1-1e-16)
> ind_sign <- sign(study_2$logFC[k])
> ng_terms[j, 2] <- weights$study_2[j]* ind_sign *abs(qnorm((1-p_val), mean = 0, sd = 1))}
> if (unique_genes[j] %in% study_3$entrez_id){
> k = which(study_3$entrez_id == unique_genes[j])
> p_val=min(max(study_3$PValue[k],1e-16),1-1e-16)
> ind_sign <- sign(study_3$logFC[k])
> ng_terms[j, 3] <- weights$study_3[j]* ind_sign *abs(qnorm((1-p_val), mean = 0, sd = 1))}}}

> colnames(ng_terms) <- datasets
> ng_terms <- as.data.frame(ng_terms, stringsAsFactors = FALSE)
> row.names(ng_terms) <- row.names(weights)

# sum all ng_terms row-wise

> ng <- as.data.frame(rowSums(ng_terms))
> colnames(ng) <- c("ng")
> row.names(ng) <- row.names(ng_terms)
```

f. Hypothesis testing:

```
 # first do one-sided for all and then replace with two sided for conflicting direction
```

```
genes

> ng$mix_in_p_val <- 1-pnorm(ng$ng)

# index of conflicting direction genes

> conf_ind <- which(sign[, ncol(sign)] == 1)
> ng$mix_in_p_val[conf_ind] <- 2 * (1-pnorm(abs(ng$ng[conf_ind])))

# mutiple-testing correction

> ng$BH_p_value <- p.adjust(ng$mix_in_p_val, method = "BH", n = length(ng$mix_in_p_val))
> head(ng)
```

	ng	mix_in_p_val	BH_p_value	entrez_id	symbol
7153	9.909582	0.000000e+00	0.000000e+00	7153	TOP2A
51555	10.415685	0.000000e+00	0.000000e+00	51555	PEX5L
6241	9.793367	0.000000e+00	0.000000e+00	6241	RRM2
386618	8.045723	4.440892e-16	6.166409e-15	386618	KCTD4
26289	10.787492	0.000000e+00	0.000000e+00	26289	AK5
9928	9.648872	0.000000e+00	0.000000e+00	9928	KIF14

g. Annotations: In case not already annotated and entrez ids mapped to gene symbols

```
 > ng$entrez_id <- as.numeric(as.character(row.names(ng)))

# now use the entrez ids to get the symbols

> egSYMBOL <- toTable(org.Hs.egSYMBOL)
> match_SY <- match(row.names(ng), as.character(egSYMBOL$gene_id))
> ng$symbol <- as.character(egSYMBOL$symbol[match_SY])
```

For genes with conflicting direction of expression across different studies, the effective direction of expression from the meta-analysis is then determined by the sign of N_g.

4 References

1. Carlson M. org. Hs. eg. db: Genome Wide Annotation for Human. R package version 3.8. 2.
2. Gentleman R. annotate: Annotation for microarrays. R package version 1.68.0, 2020.
3. Tyler Rinker and Dason Kurkiewicz. pacman: Package Management Tool. R package version 0.2.0, 2012. `https://github.com/trinker/pacman`
4. Simon Anders and Wolfgang Huber. Differential expression analysis for sequence count data. Genome Biology. 2010; 11:R106.
5. Robinson MD, McCarthy DJ, Smyth GK. edgeR: a Bioconductor package for differential expression analysis of digital gene expression data. Bioinformatics. 2010 Jan 1;26(1):139-40.
6. Rau A, Marot G, Jaffrézic F. Differential meta-analysis of RNA-seq data from multiple studies. BMC bioinformatics. 2014 Dec 1;15(1):91.
7. Raithel S, Johnson L, Galliart M, Brown S, Shelton J, Herndon N, Bello NM. Inferential considerations for low-count RNA-seq transcripts: a case study on the dominant prairie grass Andropogon gerardii. BMC genomics. 2016 Dec;17(1):1-6.