GENE EXPRESSION DATA ANALYSIS PROJECT

Alireza Doustmohammadi

JULAY 2019
TARBIAT MODARES UNIVERSITY
Nasr, Jalal al Ahmad, Tehran

Abstract:

MicroRNAs (miRNAs) are a class of non-coding RNAs that play important roles in regulating gene expression. The majority of miRNAs are transcribed from DNA sequences into primary miRNAs and processed into precursor miRNAs, and finally mature miRNAs. In most cases, miRNAs interact with the 3′ untranslated region (3′ UTR) of target mRNAs to induce mRNA degradation and translational repression. Under certain conditions, miRNAs can also activate translation or regulate transcription. The interaction of miRNAs with their target genes is dynamic and dependent on many factors, such as subcellular location of miRNAs, the abundancy of miRNAs and target mRNAs, and the affinity of miRNA-mRNA interactions. [1]

Keywords: microRNA,miRNA, gene regulation

Introduction

Micro RNAs (miRNAs) are a class of highly conserved (20–29 nucleotide) small non-coding
RNAs that play an important part in the post-transcriptional regulation of gene expression, making them essential to many fundamental pathological and biological processes.
MiRNAs have an important role in the manifestation of a wide range of diseases such as cancer. Some of these miRNAs uses as biomarkers for disease. Specific miRNA signatures have been identified in cancers such as lung and breast. [2]
In this study I present a pipeline to transcript-level expression analysis of RNA-seq experiments with HISAT, StringTie and Ballgown and identified a differentially expressed miRNA (miR-320a) in circulating blood among a cohort of healthy females (n = 6) compared to a cohort of female breast cancer patients (n = 6) based on Illumina 2000 sequencing data (Table one). My dataset is single end.

Table 1. Sample cohort set analyzed using small RNA sequencing

Accession Number	sample type	type	Age
SRR2201536	cancer	Ductal	50 - 60
SRR2201537	cancer	Lobular	80 - 90
SRR2201538	cancer	Lobular	80 - 90
SRR2201539	cancer	Ductal	50 - 60
SRR2201541	cancer	Ductal	70 - 80
SRR2201542	cancer	Ductal	40 - 50
SRR2201546	Pre-Menopausal Healthy	Healthy	30 - 40
SRR2201547	Pre-Menopausal Healthy	Healthy	20 - 30
SRR2201548	Pre-Menopausal Healthy	Healthy	20 - 30
SRR2201549	Pre-Menopausal Healthy	Healthy	20 - 30
SRR2201560	Pre-Menopausal Healthy	Healthy	30 - 40
SRR2201562	Pre-Menopausal Healthy	Healthy	50 - 60

Implementation:

1. **Downloading Data:**
   Downloading RNAseq data from:
   <https://www.ncbi.nlm.nih.gov/sra?linkname=bioproject_sra_all&from_uid=294226>
   12 sample number : 6 for case and 6 for control
   This dataset has been downloaded from SRA data base of NCBI that published by an article.
   The SRA created a toolkit to fast downloading data and I used this to download my dataset
   For all accession numbers we repeated this in sraToolkit.2.9.1/bin:

>>./prefetch *accessionNumber*

2. Converting to fastq

After downloading we had to convert all downloaded file to fastq format with this command:

>>./fastq-dump *accessionNumber*

3. Trime adapters and improve Quality:

In this stage I want to check quality of the data and adapter existence. If data has adapter or low quality should be trim or be delete. 123FaASTQ program visualize reads quality and show some alert if there is low quality. I check adapter existence by 123FASTQ but this program is not suitable to trime miRNA adapters so I use Bbduck program to remove adapters and after that use 123FASTQ to improve quality and remove reads with: $len(read) \leq 19$ *or* $len(read) \geq 30$

>>./bbduck.sh –Xmx1g in=*'fastq files'* out=*'fastq files'* ref=resources/adapters.fa ktrim=r k=23 mink=11 hdist=1 tpe tbo

4. Alignment:

I use hisat2 for aligning reads with reference genome

>>./hisat2 -x hg38/genome –p 4 -U *accessionNumber.fastq* -S *accessionNumber.sam*

Table 2. Samples alignment with homo sapience genome

Accession Number	sample type	type	Aligned exactly one time	Aligned >1 time	Total aligned
SRR2201536	cancer	Ductal	55.96%	28.32%	84.28%
SRR2201537	cancer	Lobular	77.70%	12.91%	90.61%
SRR2201538	cancer	Lobular	51.24%	44.39%	95.63%
SRR2201539	cancer	Ductal	55.01%	21%	76.01%
SRR2201541	cancer	Ductal	77.52%	16.93%	94.45%
SRR2201542	cancer	Ductal	46.74%	45.07%	91.81%
SRR2201546	Pre-Menopausal Healthy	Healthy	44.42%	51.90%	96.32%
SRR2201547	Pre-Menopausal Healthy	Healthy	45.47%	51.40%	96.87%
SRR2201548	Pre-Menopausal Healthy	Healthy	40.83%	55.14%	95.97%
SRR2201549	Pre-Menopausal Healthy	Healthy	52.13%	43.89%	96.02%
SRR2201560	Pre-Menopausal Healthy	Healthy	35.21%	61.26%	96.47%
SRR2201562	Pre-Menopausal Healthy	Healthy	41.69%	53.28%	94.97%

5. Convert SAM files to BAM files and sort BAM files:

For next stage we need BAM files as result we have to convert sam to bam and sort all bam files with samTools [3]

>> samtools sort -@ 4 –o AccessionNumber.bam AccessionNumber.sam

6. Assemble expressed genes and transcripts:

Assemble transcripts for each sample: [3]

>>stringtie -p 4 -G *ref.gtf* –o *AccessionNumber.gtf* –l *AccessionNumber.bam*

Run StringTie over all my assemblies to create a single merged transcriptome annotation: [3]

>> stringtie --merge -p 4 -G *ref.gtf* -o stringtie_merged.gtf chrX_data/mergelist.txt

In mergelist.txt there are address of bam files.

Ballgown also supports reading of data from Cufflinks.

7. Examine how the transcripts compare with the reference annotation:

>> gffcompare -R -r *ref.gtf* -o strtcmp stringtie_merged.gtf

The command above will generate multiple files and 'strtcmp.stats' is this stage result:

```
#= Summary for dataset: ../SRR22015.gtf
#     Query mRNAs :  167510 in   42010 loci  (162014 multi-exon transcripts)
#           (21728 multi-transcript loci, ~4.0 transcripts per locus)
# Reference mRNAs :  166982 in   41969 loci  (161593 multi-exon)
# Super-loci w/ reference transcripts:    41969
#-----------------| Sensitivity | Precision  |
        Base level:   100.0     |   100.0    |
        Exon level:   100.0     |   100.0    |
      Intron level:   100.0     |   100.0    |
Intron chain level:   100.0     |    99.7    |
  Transcript level:   100.0     |    99.7    |
       Locus level:   100.0     |    99.9    |

     Matching intron chains:  161593
       Matching transcripts:  166982
              Matching loci:   41969

          Missed exons:      0/419366     (  0.0%)
           Novel exons:     41/419509     (  0.0%)
        Missed introns:      0/358801     (  0.0%)
         Novel introns:      0/358801     (  0.0%)
           Missed loci:       0/41969     (  0.0%)
            Novel loci:      41/42010     (  0.1%)

Total union super-loci across all input datasets: 42010
167510 out of 167510 consensus transcripts written in strtcmp.annotated.gtf (0 discarded as redundant)
```

The gffcompare command shown here will also compute sensitivity and precision statistics for different gene features in the strtcmp**.**stats output file. Sensitivity is defined as the proportion of genes from the annotation that are correctly reconstructed, whereas positive predictive value captures the proportion of the output that overlaps the annotation.

8. **Estimate transcript abundances and create table counts for Ballgown:** [3]
   >> stringtie –e –B -p 4 -G stringtie_merged.gtf –o ballgown/*AccessionNumber*/ *AccessionNumber.bam*

9. **Run the differential expression analysis protocol:**

I use the Ballgown package for performing analyses, RSkittleBrewer for setting up colors, geneflter for calculation of means and variances, dplyr for sorting and arranging results and devtools for reproducibility.

After Ballgown object is created, we should normalize dates and filter to remove low-abundance genes.I use FPKM to normalize RNA-seq data (Table 3). Note that in single end dates FPKM is equal to RPKM. In RNA-seq data genes often have very few or zero counts. Another approach that has been used for gene expression analysis is to apply a variance filter. I remove all transcripts with a variance across samples less than one.

Table 3. Filter dates

Results	Number of gene / transcript before remove low-abundance gene/ transcripts with a variance less than one	Number of gene / transcript after remove low-abundance gene/ transcripts with a variance less than one
Gene – Results	160038	41
Transcript - Results	167510	41

One thing that I want to do is make sure that I account for variation in expression due to other variables. I look for transcripts that are differentially expressed between sample types, while correcting for any differences in expression due to the population variable. I can do this using the 'stattest' function from Ballgown. I set the 'getFC=TRUE' parameter so that we can look at the confounder-adjusted fold change between the two groups. Ballgown's statistical test is a standard linear model-based comparison. For small sample sizes ($n < 4$ per group), it is often better to perform regularization. This can be done using the 'limma package in Bioconductor'. After this step. I filter only genes and transcripts have lower than 0.05 p-value (Table 4).

Table 4. Filter genes and transcripts have lower than 0.05 p-value

geneNames	geneIDs	feature	id	fc	pval	qval
	MSTRG.7	gene	MSTRG.7	2.576855	0.010448	0.428386
	MSTRG.9	gene	MSTRG.9	2.11502	0.025793	0.483758
	MSTRG.33	gene	MSTRG.33	0.556827	0.051548	0.483758
.	MSTRG.7	transcript	7	2.576855	0.010448	0.428386
.	MSTRG.9	transcript	9	2.11502	0.025793	0.483758
.	MSTRG.33	transcript	33	0.556827	0.051548	0.483758

As shown in the table 4, We have three transcripts that are differentially expressed between the sample types. At the gene level, we have three differentially expressed genes at the same *p* value cutoff.

After this step, I use Ballgown to visualize RNA-seq results. (Figure 1)

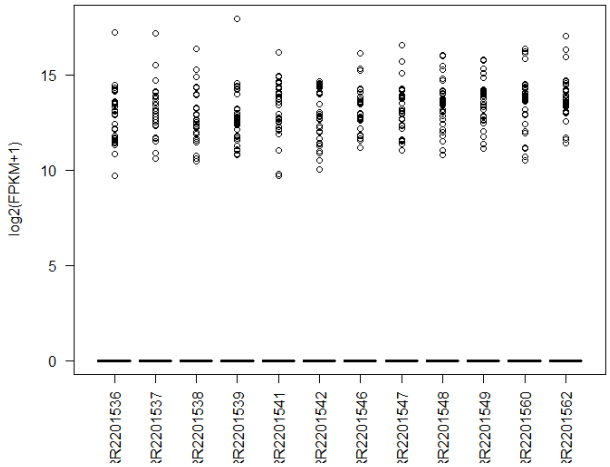

Figure 1 - Distribution of FPKM values across the 12 samples.

I Make plots of individual transcripts across samples:

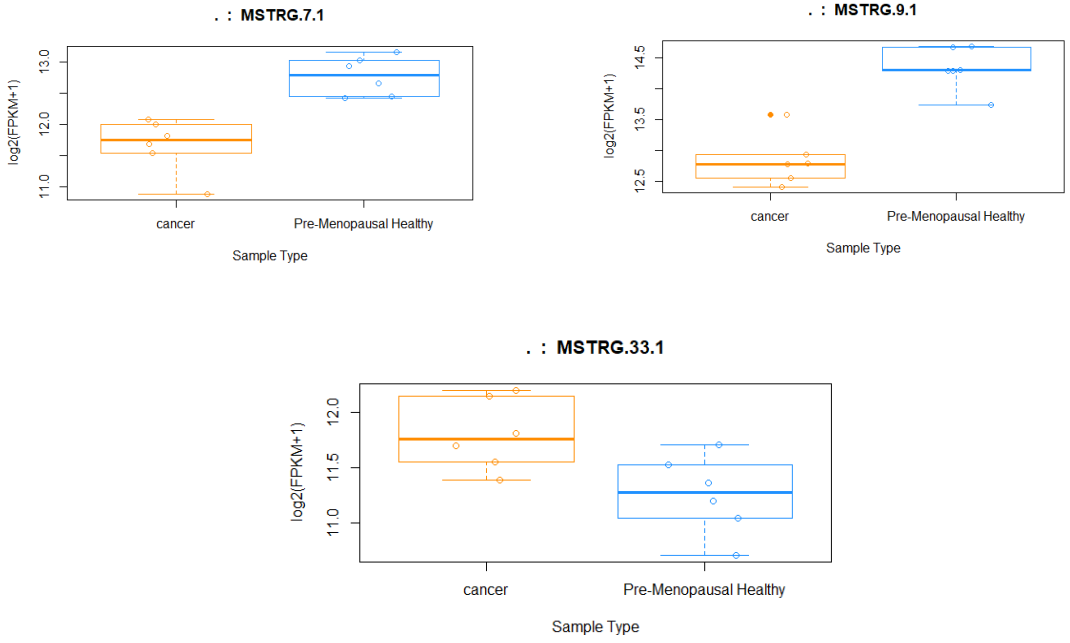

Figure 5 - FPKM distributions in sample types for transcript 7

from gene *MSTRG7*, transcript 9

from gene *MSTRG9 and* transcript 33

from gene *MSTRG33*

Source Code:

```
library(ballgown)
library(devtools)
library(RSkittleBrewer)
library(genefilter)
library(dplyr)
library(limma)
#Import Phenotype Data
pheno_data = read.csv("phenodata.csv")


blg= ballgown(dataDir = "ballgown/", samplePattern = "SRR22015", pData=pheno_data)
#remove low-abundance genes
blg_filt = subset(blg,"rowVars(texpr(blg)) >1",genomesubset=TRUE)

#statistically signifcant differences between groups
results_transcripts = stattest(blg_filt,feature="transcript",covariate="sample.type",getFC=TRUE, meas="FPKM")
results_genes = stattest(blg_filt, feature="gene",covariate="sample.type", getFC=TRUE,meas="FPKM")
#Add gene names and gene IDs
results_transcripts =data.frame(geneNames=ballgown::geneNames(blg_filt),geneIDs=ballgown::geneIDs(blg_filt), results_transcripts)
#sort
results_transcripts = arrange(results_transcripts,pval)
results_genes = arrange(results_genes,pval)

write.csv(results_transcripts, "transcript_results.csv",row.names=FALSE)
write.csv(results_genes, "gene_results.csv",row.names=FALSE)

subset(results_transcripts,results_transcripts$qval<0.05)
subset(results_genes,results_genes$qval<0.05)

tropical= c('darkorange', 'dodgerblue','hotpink', 'limegreen', 'yellow')
palette(tropical)

fpkm = texpr(blg,meas="FPKM")
fpkm = log2(fpkm+1)
boxplot(fpkm,col=as.numeric(pheno_data$sex),las=2,ylab='log2(FPKM+1)')

#transcript 7
plot(fpkm[7,] ~ pheno_data$sample.type, border=c(1,2),main=paste(ballgown::geneNames(blg)[7],' :
',ballgown::transcriptNames(blg)[7]),pch=19, xlab="Sample Type",ylab='log2(FPKM+1)')
points(fpkm[7,] ~ jitter(as.numeric(pheno_data$sample.type)),col=as.numeric(pheno_data$sample.type))

#transcript 9
plot(fpkm[9,] ~ pheno_data$sample.type, border=c(1,2),main=paste(ballgown::geneNames(blg)[9],' :
',ballgown::transcriptNames(blg)[9]),pch=19, xlab="Sample Type",ylab='log2(FPKM+1)')
points(fpkm[9,] ~ jitter(as.numeric(pheno_data$sample.type)),col=as.numeric(pheno_data$sample.type))

#transcript 33
plot(fpkm[33,] ~ pheno_data$sample.type, border=c(1,2),main=paste(ballgown::geneNames(blg)[33],' :
',ballgown::transcriptNames(blg)[33]),pch=19, xlab="Sample Type",ylab='log2(FPKM+1)')
points(fpkm[33,] ~ jitter(as.numeric(pheno_data$sample.type)),col=as.numeric(pheno_data$sample.type))


plotMeans('MSTRG.7', blg_filt,groupvar="sample.type",legend=FALSE)
plotMeans('MSTRG.9', blg_filt,groupvar="sample.type",legend=FALSE)
plotMeans('MSTRG.33', blg_filt,groupvar="sample.type",legend=FALSE)
```

References

[1] Jacob O'Brien, Heyam Hayder, Yara Zayed, and Chun Peng, "Overview of MicroRNA Biogenesis, Mechanisms of Actions, and Circulation," *Front. Endocrinol,* 2018.

[2] Helena Kelly, Tim Downing, Nina L. Tuite, Terry J. Smith, Michael J. Kerin5, Róisín, "Cross Platform Standardisation of an Experimental Pipeline for Use in the Identification of Dysregulated Human Circulating MiRNAs," *PLOS ONE,* 2015.

[3] Mihaela Pertea, Daehwan Kim, Geo M Pertea, Jeffrey T Leek,Steven L Salzberg, "Transcript-level expression analysis of RNA-seq experiments with HISAT, StringTie and Ballgown," *Nature America,* 2016.