20151208
Scripts summary for Pengda #1
Questions to: **hyjin@scripps.edu**, Hyun Yong Jin, Xiao Lab, Scripps Research
using miR-17~92 TG, WT, TKO ribosome profiling analysis (R138)
Ribosome footprint quantification

Overview

*Note: this methods are largely similar to RNA-seq analysis but some are different. The major differences are

1.
RNA-seq pipline: Cufflink --> Cuffmerge --> Cuffdiff
Ribosome profling pipeline: Directly to Cuffdiff
(Cuffmerge is to build de novo transcript discovery and build, but you cannot build de novo transcripts from results in ribosome profiling)

2.
After genome alignment (tophat) the perfect matched hit will be selected from the results.
Genome alignment usually allows one or a few nucleotide mismatches depending on scores of sequence quality of individual nucleotides and length of reads, but ribosome footprints are short, so we only allow perfectly matched reads for downstream analysis to increase sensitivity.

Step01: Clip and Trim
from fastq raw files,
clip the adaptor sequences and trim a nucleotide at the end (final nucleotide frequently introduced with a mutation)

Shell command, use FastX toolkit

```
$gzcat ~/TheShell/SeqResults/R138_Jin_RiPr/TG1_*.fastq.gz| \
fastx_clipper -Q33 -a CTGTAGGCACCATCAAT -l 5 -c -n -v| \
fastx_trimmer -Q33 -f 2 -l 80 >TG1_cltr_5-80.fq
```

*for 9 samples, repeat 9 samples individually.
*To conduct the multiple samples in a single script, use "&&" operator to consecutive commend.
For example to do two samples together,

```
$ gzcat ~/TheShell/SeqResults/R138_Jin_RiPr/WT2_*.fastq.gz| fastx_clipper -Q33 -a
CTGTAGGCACCATCAAT -l 5 -c -n -v| fastx_trimmer -Q33 -f 2 -l 80 >WT2_cltr_5-80.fq && gzcat
~/TheShell/SeqResults/R138_Jin_RiPr/WT3_*.fastq.gz| fastx_clipper -Q33 -a CTGTAGGCACCATCAAT
-l 5 -c -n -v| fastx_trimmer -Q33 -f 2 -l 80 >WT3_cltr_5-80.fq
```

This wil generate results as follow.
If your library is correctly generated, majority of your imput must contains adaptor sequences

Clipping Adapter: CTGTAGGCACCATCAAT
Min. Length: 5
Non-Clipped reads - discarded.

Input: 52027012 reads.
Output: **48119180 reads.**
discarded 33462 too-short reads.
discarded 128793 adapter-only reads.
discarded 3745577 non-clipped reads.

Step02-1: Download reference genome.
There are several reference genome you can use, but I will recommend iGenome from illumine, which is annotated in a way that cufflink smoothly works.
If you have generate your local environment using homebrew, use you local account (/usr/loca/) as the destined folder.

Cufflink provides direct link to illumine iGenome
http://cole-trapnell-lab.github.io/cufflinks//igenome_table/index.html

or you can directly go like this from Shell
wget --ftp-user=igenome --ftp-password=G3nom3s4u
ftp://ussdftp.illumina.com/Homo_sapiens/UCSC/hg19/Homo_sapiens_UCSC_hg19.tar.gz

I used mm10 as reference.

Step02-2: Build rRNA reference ebwt file from the iGenome

```
$ mkdir -p /usr/local/iGenomes/contam/mm10rRNA
$ cd ~/TheShell/iGenomes/contam/mm10rRNA
$ bowtie-build
/usr/local/iGenomes/Mus_musculus/UCSC/mm10/Sequence/AbundantSequences/musRibosomal.fa
mm10rRNA
```

output directory is /contam/mm10rRNA and new index name is mm10rRNA

Step02-3: Remove rRNA sequences using Bowtie

```
$ bowtie -l 23 -t -p 4 --un=TG1_norrna.fq /usr/local/iGenomes/contam/mm10rRNA/mm10rRNA
~/TheShell/SeqResults/R138_jin_RiPr/TG1_cltr_5-80.fq 2>> TG1_stats.txt > TG1_rrnaAlignments.aln &&
\
> bowtie -l 23 -t -p 4 --un=TG2_norrna.fq /usr/local/iGenomes/contam/mm10rRNA/mm10rRNA
~/TheShell/SeqResults/R138_jin_RiPr/TG2_cltr_5-80.fq 2>> TG2_stats.txt > TG2_rrnaAlignments.aln &&
\
> bowtie -l 23 -t -p 4 --un=TG3_norrna.fq /usr/local/iGenomes/contam/mm10rRNA/mm10rRNA
```

```
~/TheShell/SeqResults/R138_jin_RiPr/TG3_cltr_5-80.fq 2>> TG3_stats.txt > TG3_rrnaAlignments.aln &&
\
> bowtie -l 23 -t -p 4 --un=tkO1_norrna.fq /usr/local/iGenomes/contam/mm10rRNA/mm10rRNA
~/TheShell/SeqResults/R138_jin_RiPr/tKO1_cltr_5-80.fq 2>> tKO1_stats.txt > tKO1_rrnaAlignments.aln
&& \
> bowtie -l 23 -t -p 4 --un=tkO2_norrna.fq /usr/local/iGenomes/contam/mm10rRNA/mm10rRNA
~/TheShell/SeqResults/R138_jin_RiPr/tKO2_cltr_5-80.fq 2>> tKO2_stats.txt > tKO2_rrnaAlignments.aln
&& \
> bowtie -l 23 -t -p 4 --un=tkO3_norrna.fq /usr/local/iGenomes/contam/mm10rRNA/mm10rRNA
~/TheShell/SeqResults/R138_jin_RiPr/tKO3_cltr_5-80.fq 2>> tKO3_stats.txt > tKO3_rrnaAlignments.aln
&& \
> bowtie -l 23 -t -p 4 --un=WT1_norrna.fq /usr/local/iGenomes/contam/mm10rRNA/mm10rRNA
~/TheShell/SeqResults/R138_jin_RiPr/WT1_cltr_5-80.fq 2>> WT1_stats.txt > WT1_rrnaAlignments.aln
&& \
> bowtie -l 23 -t -p 4 --un=WT2_norrna.fq /usr/local/iGenomes/contam/mm10rRNA/mm10rRNA
~/TheShell/SeqResults/R138_jin_RiPr/WT2_cltr_5-80.fq 2>> WT2_stats.txt > WT2_rrnaAlignments.aln
&& \
> bowtie -l 23 -t -p 4 --un=WT3_norrna.fq /usr/local/iGenomes/contam/mm10rRNA/mm10rRNA
~/TheShell/SeqResults/R138_jin_RiPr/WT3_cltr_5-80.fq 2>> WT3_stats.txt > WT3_rrnaAlignments.aln
```

bowtie -23 -t -p 4 --un=<unaligned fastq file name (results that we are interested in)> <Path to ebwt rRNA folder>/<index name> <previously trimmed and clipped fastq file> 2>> stats.txt > <name of rRNA sequence aligned file>
-l 23 indicates seed length is 23hr, -t is for knowing running time, -p 4 to enable multithreading (recommended by Gareth)

*Note: You can remove rRNA at cuffdiff steps using mask.gtf file, and this is actually standard for RNA-seq analysis.
http://onetipperday.blogspot.com/2012/08/how-to-get-trnarrnamitochondrial-gene.html
But our current method was from Ingolia's 2012 protocol paper, and I assume the reasons are (1) you can check quality of reads using FastQC and (2) bowtie is specifically good for "short read" alignments

Step02-4: Quality check with FastQC
http://www.bioinformatics.babraham.ac.uk/projects/fastqc/
Download this and import your resulting fastq file.
It's very strait forward to use, and especially length distribution visualization is default output.

Step 03-1 Tophat alignment

This is the rate limiting steps, so could take more than 24h in my computer, and some RNA-seq takes even 3 days.
But your's must be faster.

no-novel-junc flag will tell tophat not to deal with novel splice discovery, which will significantly reduce running time.

Depending on the numbers of your cpu cores, you can increase thread numbers allowing multi-treaded calculation. My mac is dual core, so maximum threads I can use is 2, but I use 1 as I need to use my computer for other purpose at the same time.
You can run with n-1 threads on an n-core machine. For examples, 15 threads on a 16 core computer.

Scripts for individual sample.

```
$tophat --no-novel-juncs \
--GTF /usr/local/iGenomes/Mus_musculus/UCSC/mm10/Annotation/Genes/genes.gtf \
--num-threads 1 \
--output-dir ~/TheShell/SeqResults/R138_Jin_RiPr/04_topHat_no_rtrna/ \
  /usr/local/iGenomes/Mus_musculus/UCSC/mm10/Sequence/BowtieIndex/genome
~/TheShell/SeqResults/R138_Jin_RiPr/03_bowtie_no_trna/R138_no_trna.fq
```

#no-novel-juncs flag will save time if you are not interested in doing novel splice site discovery
#num-threads 1 --> if workflow is run on a machine with multiple cores, this number may be increased to reflect the number of cores present
#GTF file is for known junction used for analysis.
#this will sue Bowtie 1.0.0.0. as I don't have bowtie2 installed. maybe it is as intended.

If you have 9 sample analysis together,

```
$tophat --no-novel-juncs --GTF
/usr/local/iGenomes/Mus_musculus/UCSC/mm10/Annotation/Genes/genes.gtf --num-threads 1
--output-dir ~/TheShell/SeqResults/R138_Jin_RiPr/TG1_topHat/
/usr/local/iGenomes/Mus_musculus/UCSC/mm10/Sequence/BowtieIndex/genome
~/TheShell/SeqResults/R138_Jin_RiPr/TG1_norrna.fq && \
tophat --no-novel-juncs --GTF
/usr/local/iGenomes/Mus_musculus/UCSC/mm10/Annotation/Genes/genes.gtf --num-threads 1
--output-dir ~/TheShell/SeqResults/R138_Jin_RiPr/TG2_topHat/
/usr/local/iGenomes/Mus_musculus/UCSC/mm10/Sequence/BowtieIndex/genome
~/TheShell/SeqResults/R138_Jin_RiPr/TG2_norrna.fq && \
tophat --no-novel-juncs --GTF
/usr/local/iGenomes/Mus_musculus/UCSC/mm10/Annotation/Genes/genes.gtf --num-threads 1
--output-dir ~/TheShell/SeqResults/R138_Jin_RiPr/TG3_topHat/
/usr/local/iGenomes/Mus_musculus/UCSC/mm10/Sequence/BowtieIndex/genome
~/TheShell/SeqResults/R138_Jin_RiPr/TG3_norrna.fq && \
tophat --no-novel-juncs --GTF
/usr/local/iGenomes/Mus_musculus/UCSC/mm10/Annotation/Genes/genes.gtf --num-threads 1
--output-dir ~/TheShell/SeqResults/R138_Jin_RiPr/tKO1_topHat/
/usr/local/iGenomes/Mus_musculus/UCSC/mm10/Sequence/BowtieIndex/genome
~/TheShell/SeqResults/R138_Jin_RiPr/tKO1_norrna.fq && \
tophat --no-novel-juncs --GTF
/usr/local/iGenomes/Mus_musculus/UCSC/mm10/Annotation/Genes/genes.gtf --num-threads 1
--output-dir ~/TheShell/SeqResults/R138_Jin_RiPr/tKO2_topHat/
/usr/local/iGenomes/Mus_musculus/UCSC/mm10/Sequence/BowtieIndex/genome
~/TheShell/SeqResults/R138_Jin_RiPr/tKO2_norrna.fq && \
tophat --no-novel-juncs --GTF
/usr/local/iGenomes/Mus_musculus/UCSC/mm10/Annotation/Genes/genes.gtf --num-threads 1
--output-dir ~/TheShell/SeqResults/R138_Jin_RiPr/tKO3_topHat/
/usr/local/iGenomes/Mus_musculus/UCSC/mm10/Sequence/BowtieIndex/genome
```

```
~/TheShell/SeqResults/R138_Jin_RiPr/tKO3_norrna.fq && \
tophat --no-novel-juncs --GTF
/usr/local/iGenomes/Mus_musculus/UCSC/mm10/Annotation/Genes/genes.gtf --num-threads 1
--output-dir ~/TheShell/SeqResults/R138_Jin_RiPr/WT1_topHat/
/usr/local/iGenomes/Mus_musculus/UCSC/mm10/Sequence/BowtieIndex/genome
~/TheShell/SeqResults/R138_Jin_RiPr/WT1_norrna.fq && \
tophat --no-novel-juncs --GTF
/usr/local/iGenomes/Mus_musculus/UCSC/mm10/Annotation/Genes/genes.gtf --num-threads 1
--output-dir ~/TheShell/SeqResults/R138_Jin_RiPr/WT2_topHat/
/usr/local/iGenomes/Mus_musculus/UCSC/mm10/Sequence/BowtieIndex/genome
~/TheShell/SeqResults/R138_Jin_RiPr/WT2_norrna.fq && \
tophat --no-novel-juncs --GTF
/usr/local/iGenomes/Mus_musculus/UCSC/mm10/Annotation/Genes/genes.gtf --num-threads 1
--output-dir ~/TheShell/SeqResults/R138_Jin_RiPr/WT3_topHat/
/usr/local/iGenomes/Mus_musculus/UCSC/mm10/Sequence/BowtieIndex/genome
~/TheShell/SeqResults/R138_Jin_RiPr/WT3_norrna.fq
```

Step 03-2 Select perfect matched hits using Samtool

This is unique to ribosome profiling that are not usually done with RNA-seq analysis (from Ingolia's 2012 paper)

```
$ cd <output folder>

$ samtools view -h ~/TheShell/SeqResults/R138_Jin_RiPr/03_topHat/TG1_topHat/accepted_hits.bam|
grep -E '(NM:i:0)|(^@)'| samtools view -S -b ->perfect_hits_TG1.bam &&\
samtools view -h ~/TheShell/SeqResults/R138_Jin_RiPr/03_topHat/TG2_topHat/accepted_hits.bam| grep
-E '(NM:i:0)|(^@)'| samtools view -S -b ->perfect_hits_TG2.bam &&\
samtools view -h ~/TheShell/SeqResults/R138_Jin_RiPr/03_topHat/TG3_topHat/accepted_hits.bam| grep
-E '(NM:i:0)|(^@)'| samtools view -S -b ->perfect_hits_TG3.bam &&\
samtools view -h ~/TheShell/SeqResults/R138_Jin_RiPr/03_topHat/tKO1_topHat/accepted_hits.bam|
grep -E '(NM:i:0)|(^@)'| samtools view -S -b ->perfect_hits_tKO1.bam &&\
samtools view -h ~/TheShell/SeqResults/R138_Jin_RiPr/03_topHat/tKO2_topHat/accepted_hits.bam|
grep -E '(NM:i:0)|(^@)'| samtools view -S -b ->perfect_hits_tKO2.bam &&\
samtools view -h ~/TheShell/SeqResults/R138_Jin_RiPr/03_topHat/tKO3_topHat/accepted_hits.bam|
grep -E '(NM:i:0)|(^@)'| samtools view -S -b ->perfect_hits_tKO3.bam &&\
samtools view -h ~/TheShell/SeqResults/R138_Jin_RiPr/03_topHat/WT1_topHat/accepted_hits.bam|
grep -E '(NM:i:0)|(^@)'| samtools view -S -b ->perfect_hits_WT1.bam &&\
samtools view -h ~/TheShell/SeqResults/R138_Jin_RiPr/03_topHat/WT2_topHat/accepted_hits.bam|
grep -E '(NM:i:0)|(^@)'| samtools view -S -b ->perfect_hits_WT2.bam &&\
samtools view -h ~/TheShell/SeqResults/R138_Jin_RiPr/03_topHat/WT3_topHat/accepted_hits.bam|
grep -E '(NM:i:0)|(^@)'| samtools view -S -b ->perfect_hits_WT3.bam
```

Step04 Cuffdiff for differential gene expression analysis

```
$ cuffdiff -L TG,tKO,WT \
-o cuffdiffOutput /usr/local/iGenomes/Mus_musculus/UCSC/mm10/Annotation/Genes/genes.gtf \
accepted_hits_TG1.bam,accepted_hits_TG2.bam,accepted_hits_TG3.bam \
accepted_hits_tKO1.bam,accepted_hits_tKO2.bam,accepted_hits_tKO3.bam \
accepted_hits_WT1.bam,accepted_hits_WT2.bam,accepted_hits_WT3.bam
```

This results are now in the final folde "cuffdiffOutput"

Step05 CummeRbund analysis for gene experession quantification

Results are already calculated in "cuffdiffOutput" folder, you can manually extract data from it. But bioconductor package in R called CummeRbund will make this extraction steps easy.

Now move on to R,

```
library(cummeRbund)

#set up defalt directory as cuffdiff output folder containing genes.fpkm_tracking etc.
# this will generate cuffData.db file

setwd("~/TheShell/SeqResults/R138_Jin_RiPr/05_cuffdiff/perfect_hits_analysis/cuffdiffOutput/")
cuff <-readCufflinks() #Now all your results are under "cuff"

cuff
#output
CuffSet instance with:
    3 samples
        23980 genes
        33295 isoforms
        27067 TSS
        26408 CDS
        71772 promoters
        81201 splicing
        61419 relCDS


# You can do many analysis in this program, so please visit vignette.

#for FPKM results in csv table, do it as follow

# mean value
gene.matrix <-fpkmMatrix(genes(cuff))
head(gene.matrix)
write.csv(gene.matrix, file="fpkm_mean.csv") # export to the base dir

#Individual replicates
gene.rep.matrix<-repFpkmMatrix(genes(cuff))
head(gene.rep.matrix)
write.csv(gene.rep.matrix, file="fpkm_replicate.csv") # export to the base dir
```