A general workflow for differential expression analysis of RNA-seq and introductions on related tools

Zhong Zhang[1, a]

[1]School of Government, Beijing Normal University, Beijing 100875, China.

[a]1043581073@qq.com

Keywords: RNA-seq, differential expression analysis, TopHat2, Cufflinks

Abstract. RNA-seq technology have been used widely in many biological research field, but analyzing the enormous RNA-seq data is still a big challenge to biologist without sufficient knowledge of bioinformatics. Considering that the differential expression analysis is a crucial and tricky application of RNA-seq, we introduce a general workflow in this paper. In order to make our instruction more practical, we also provide the basic usage of some famous analysis tools for each step of the workflow.

Introduction

The high throughout sequencing technology, also known as next-generation sequencing (NGS), refers to methods using similar sequencing by synthesis chemistry of individual nucleotides, but performed in a massively parallel format, so that the number of sequencing reactions in a single run can be in millions. [1] Thus the application and development of high throughout sequencing bring the genomics research to a new generation. Molecular biologists can perform their research based on the numerous data provided by those new technologies. In the upgrade of the sequencing technology, RNA-seq methods played an important role. RNA-seq can be used to reveal new genes and splice variants or quantify expression genome-wide in a single assay. [2] Compared with earlier methods such as microarrays, RNA-seq could capture almost all of the expressed transcripts for a snapshot of cells in theory so that it can detect novel splicing variants, novel genes and novel transcripts. In addition, the incredibly high throughput of current RNA-seq platforms, the sensitivity afforded by newer technologies and the ability to discover novel transcripts, gene models, and small noncoding RNA species together make RNA-seq a powerful technology for transcriptomics studies.

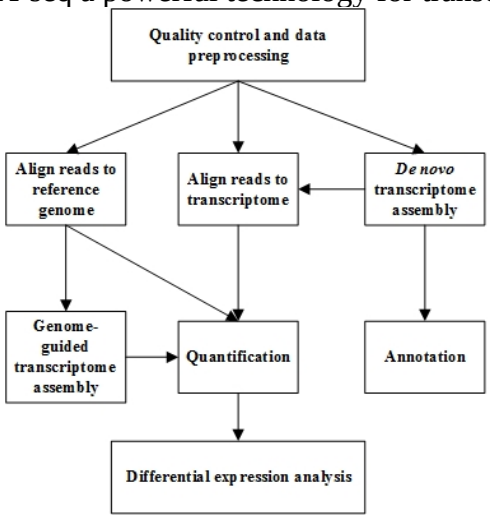

Fig. 1 Some possible analysis procedures for RNA-seq data.

Depending on whether a reference genome is available for the species, researchers can choose to use De novo workflow to discover new genes or apply differential expression analysis.

Because of the high quality and quantity of RNA-seq data, RNA-seq methods have many various applications, ranging from gene and splice variant discovery to differential expression analysis and

detection of fusion genes, variants, and RNA editing. Figure 1 shows the major steps in some routine analyses. Researchers can choose different analysis paths depending on the purposes of their projects and the priori information of the spices they studied like whether a reference genome or transcriptome is available. As RNA-seq data analysis is an active field, many alternative programs and softwares, with or without a graphical user interface, are available for each analysis steps. Although many of those analysis tools are well designed for users to master them quickly, analyzing and processing the numerous RNA-seq data may still be troublesome to some biologists without bioinformatics background and sufficient computer skills. Thus in this paper, we introduce a general workflow about the differential expression analysis based on the RNA-seq data and include the instructions on the usage of related softwares. It shows the steps of differential expression analysis workflow and the corresponding analysis tools we are going to introduce (Figure 2). Note that the tools and softwares we will use in this paper may not be the best nor the most suitable for all researchers. Many optional tools are available for each steps and luckily many thorough tool comparisons have been published. In this paper we just introduce the tools that are able to satisfy most requirements in the differential expression analysis field, so choosing other tools that may suitable for your specific purposes and data features is also a good idea. The demo data (SRR1944261.sra and SRR1944253.sra) we used in this paper is from Ann-Jay Tong and Xin Liu's work. [3]

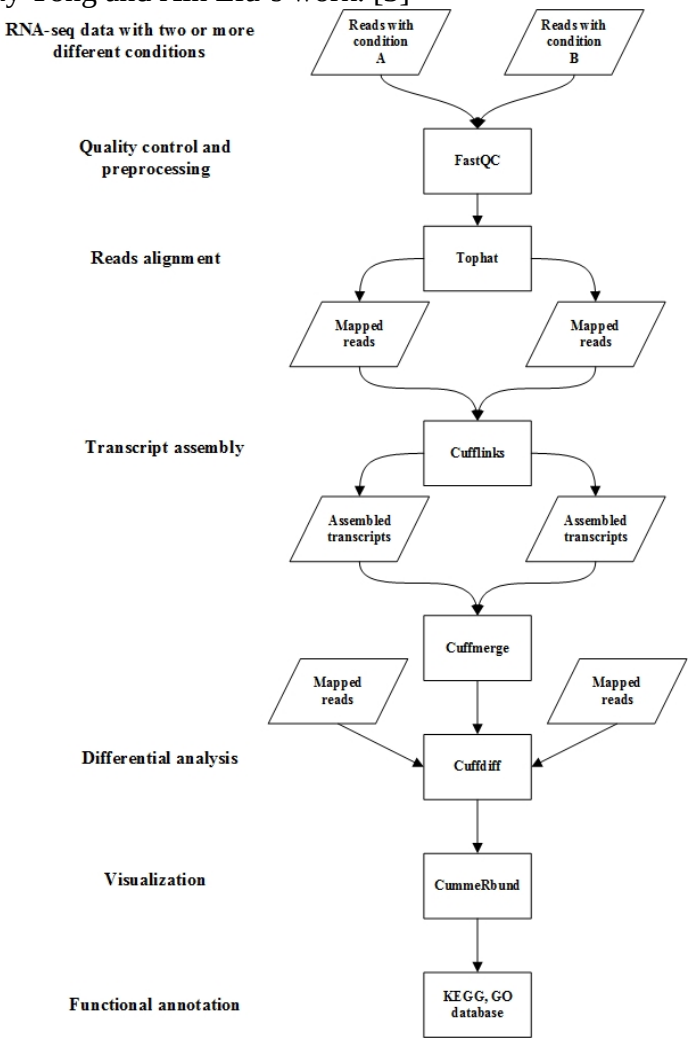

Fig. 1 A general differential analysis workflow.

The terms in the left side show the main steps of our workflow while the frame in the right side shows the software or databases we need to use and the relationship between them.

Quality Control

Although the RNA-seq methods can provide data with considerably high quality, some quality problems may still originate inevitably. Those quality problems, such as low-confidence bases, sequence-specific bias and sequence contamination, can affect the subsequent analysis steps more or less. Thus it is necessary to check the quality of raw reads, and then correct the problems or at least be aware of them during the following analyses.

The analysis tools we will introduce are based on the reads files with the FASTQ format, which is a text-based format for storing both a biological sequence and its corresponding quality scores. But reads downloaded from NCBI may be stored with SRA format, we can use the fastq-dump tool in the SRA toolkit (http://trace.ncbi.nlm.nih.gov/Traces/sra/) to conveniently convert the SRA format to the FASTQ format. Once we get the raw reads data in the FASTQ format, we can use FASTQC to do quality checking (Figure 3). FASTQC can work with a graphical user interface and will output a thorough quality report for each selected fastq file. In the quality checking report, FASTQC summarizes and visualizes information on base quality, sequence content, read length and so on. Also FASTQC reports a judgment for each term shown as the traffic light (green, yellow and red respectively mean pass, warn and fail), but note that the judgment is based on general thresholds and may not applicable for all data. Now we want to talk about some quality issues that can be detected by the quality report and provide some suggestions on how to deal with them.

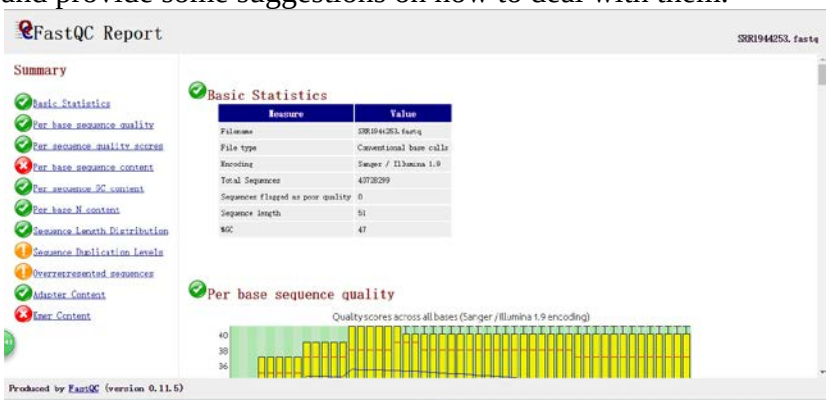

Fig. 3 The beginning part of the FASTQC quality report.

FastQC can produce a thorough report for quality checking of the RNA-seq and also provide a judgement for each term based on a general threshold.

(a) Base quality problem

Base quality indicates the confidence in the base call and is expressed by the Phred scale. It can be calculated by Q= -10 log10 P, where Q and P represent the quality scores and base-calling error probabilities respectively. The quality values typically range from 0 to 40 (higher values mean high quality). The term per base sequence quality in the quality report shows the box plots for the base quality along reads (Figure 4). Through the box plots we can check the base qualities per base position, and then we can have a general idea about the quality of the whole data. But it is still necessary to check the distribution of the reads' mean quality for there may be a subset of reads having an overall bad quality. To fulfill such requirement, the quality report by FASTQC also provides a distribution plot for quality scores, shown as Figure 5. If the RNA-seq data fails in the base quality test, we may have to do some preprocessing to it. Filtering and trimming are good ways to process the reads containing low-quality bases. The difference is that filtering removes the entire reads while trimming can remove only the low-quality ends of reads [1]. Many programs have been developed to do these jobs, such as trimmomatic, FASTX and PRINSEQ (PReprocessing and INformation of SEQuence data). They can filter or trim reads based on their qualities and learning how to use, they is not a big challenge.

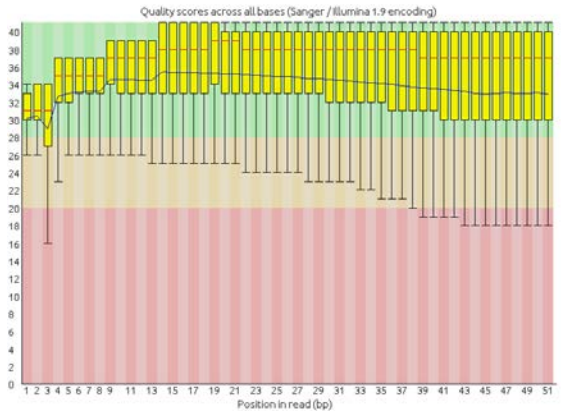

Fig. 4 The box plot for the base quality along reads.

The figure indicates that most of the reads have a relatively high quality score (>28).

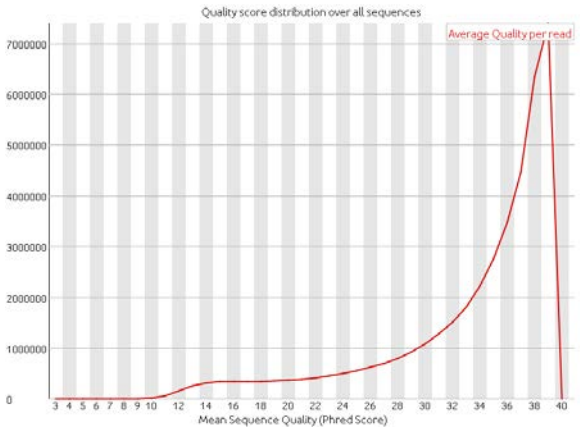

Fig.5 The distribution plot of mean sequence quality.

The peak at the right side shows that most of the sequences have a high quality score.

(b) Read length

Whether the reads have enough length is also an important question to the quality control, because mapping short reads to genome unambiguously is much more difficult than longer ones. As shown in Figure 6, the FASTQC quality report offers the sequence length distribution plot. In order to fix the reads length problem, we can simply use tools mentioned above to filter reads based on the read length rather than the base quality.

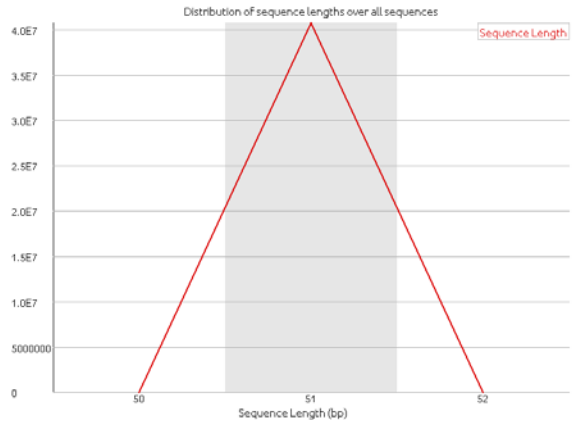

Fig. 6 The Sequence length distribution plot.

Almost all the sequences have a proper length (51bp).

(c) Adapters

An adapter is a short, chemically synthesized, double stranded DNA molecule which is used to link the ends of two other DNA molecules. It was artificially added during the sequencing and needed to be trimmed away before data analysis. Using those trimming tools we introduced above can easily clip adapter sequences. If we do not know the adapter sequence, the terms in the quality report by FASTQC, such as overrepresented sequences, adapter content and K-mer content, can help us to detect the adapter sequence or other abnormal sequences we may consider to check again or trim away.

Reads Alignment

After quality checking and preprocessing for the raw data, we then get the FASTQ files of high-quality RNA-seq data. We can proceed our analysis workflows based on those high-quality FASTQ files. The next step, the reads alignment, is crucial to the differential expression analysis for RNA-seq. The purpose of reads alignment or reads mapping is to match the reads with a reference genome to find out the origin of each read. If a reference genome is not available, some other approaches such as de novo assembly can be applied to identify novel genes.

Because of the great importance of reads alignment, many different programs have been developed to handle this task. Bowtie (http://bowtie-bio.sourceforge.net/index.shtml) is one of the most efficient aligners for its extremely high mapping speed and low memory requirement. Then we introduces its newer version Bowtie2 which has a better behavior when deal with long reads. However, Bowtie is not capable of making spliced alignments so it does allow large gaps. Hence, Bowtie cannot be used to map reads that span introns while using TopHat (https://ccb.jhu.edu/software/tophat/index.shtml) can break this limitation. Still, we concentrate on its newer version TopHat2 which uses Bowtie2 as its alignment engine. Usually, the following steps can be applied to use Bowtie2 or TopHat2 when doing differential expression analysis.

(a) preparing reference files

Since we focus on the differential expression analysis rather than the discovery of novel genes, it is necessary to prepare several reference files before we do reads mapping. Firstly, we need to prepare Bowtie2 reference genome indexes which allow Bowtie2 to work so rapidly. Usually, we can download the index directly from Bowtie2 website or the Illumina iGenomes website (http://support.illumina.com/sequencing/sequencing_software/igenome.html). Also if a ready-made index is not available for the organism you research on, we can just build an index by ourselves using the bowtie2-build command. In order to build a Bowtie2 index, we have to prepare genome FASTA files which can be downloaded from many sources such as Ensemble (http://www.ensembl.org/info/data/ftp/index.html). Once we get the genome FASTA files, we can use the following command to build our index:

```
bowtie2 -build –f <FASTA files> <index name>
```

<FASTA files> is the directory of your FASTA files and <index name> the index basename used to build the index files. The FASTA files will be used again when using TopHat2, so do not delete them after building the Bowtie2 index. In addition, If possible, you'd better also prepare an annotation in GTF/GFF file format and choose the same source for GTF files as FASTA files so that they share the same style of chromosome names (1 or chr1).

(b) aligning the reads

When all the reference files have been well prepared, we can start aligning the reads to reference using TopHat2. We use the following code to align our demo reads:

```
tophat -p 8 –G ~/genome.gtf -o ~/tophat_output/ ~/Bowtie2Index/genome ~/WT_0h/SRR1944253.fastq, ~/WT_lipidA_120/SRR1944261.fastq
```

The –p option determines the number of threads TopHat2 will use. The –G option follows the directory of the GTF/GFF file means TopHat2 will use the annotation file as guide while aligning reads. The –o option determines the directory of TopHat2 outputs. Bowtie2Index is the folder with Bowtie2 index files and genome is the index basename. Note that you should include the FASTA file in the Bowtie2Index folder, otherwise TopHat2 will waste time to rebuild it from the index files. At the end of the command is the directory of the FASTQ files we want to align. Here we use commas to separate them means they will be treated as single-end reads and if your reads are pair-end, you can use space to separate them.

(c) checking the results

TopHat2 produces several result files such as accept_hits.bam, which we will use in the following analysis steps, junctions.bed, insertions.bed, deletions.bed and align_summary.txt. The align_summary.txt file reports the mapped rate and how many reads have multiple alignment which helps you to judge if the reads mapped to the reference effectively.

Transcript Assembly

In order to determine the expression levels of genes from RNA-seq reads, we have to firstly identify which isoform of a gene provided each reads. So we should assemble a full-length transcript sequences from aligned reads before we do expression analysis. Cufflinks (http://cole-trapnell-lab.github.io/cufflinks/) is just a powerful program that can assemble transcripts, estimate their abundances, and test for differential expression and regulation in RNA-Seq samples. Using the following code, we can assemble transcripts from the aligned reads we got in the prior step:

```
cufflinks -p 4 –o ~/cufflinks_output ~tophat_output/accepted_hits.bam
```

Here the –p option also means the number of threads used. The –o option means the directory of output folder and at the end of the code is the BAM file we got from TopHat2. The output files of Cufflinks include transcripts.gtf, skipped.gtf, isoforms.fpkm_tracking and genes.fpkm_tracking. Among them, the transcripts.gtf records the transcripts with the exon information and will be used in the following step.

When doing differential expression analysis, our purpose is to quantify the expression level of each gene of several samples with different conditions and then compare them with each other to identify the differentially expressed genes. An efficient strategy to achieve this goal is to assemble a transcript with the information from all samples and the compare reads from each sample with it individually. So using Cufflinks we already have the transcript of each sample, the next step is to merge them together. Cuffmerge can help us to do like this:

```
cuffmerge -g ~/genome.gtf -s ~/genome.fa -p 8 -o ~/cuffmerge_output ~/assemblies.txt
```

The –g and –s options means using the GTF and FASTA files we prepared in the prior step to guide the transcript assembly. Each line of the assemblies.txt file should be the directory of one of the transcripts.gtf file we want to merge together. The output of Cuffmerge is merged.gtf, the file of the merged transcript.

Differential Analysis

After we obtained the aligned reads from TopHat2 and merged assembly from Cufflinks and Cuffmerge, then we can start the differential analysis. To determine the differentially expressed genes, we should calculate the expression for several samples with different conditions and tests the statistical significance of the difference between them. We can use Cuffdiff, one of the programs in the Cufflinks package, to do those jobs:

```
cuffdiff -o ~/cuffdiff -p 8 -labels WT_0h,WT_lipidA_120 ~/merged.gtf ~/WT_0h/tophat_output/accepted_hits.bam ~/WT_lipidA_120/tophat_output/accepted_hits.bam
```

Here the –labels option determines the names used for each sample in the output files. At the end of the code is the directory of the merged assembly GTF file and the BAM file for each sample. Cuffdiff produces many output files [4]:

1. Transcript FPKM (+count) expression tracking.
2. Gene FPKM (+count) expression tracking; tracks the summed FPKM of transcripts sharing each gene_id
3. Primary transcript FPKM (+count) tracking; tracks the summed FPKM of transcripts sharing each tss_id
4. Coding sequence FPKM (+count) tracking; tracks the summed FPKM of transcripts sharing each p_id, independent of tss_id
5. Transcript differential FPKM.
6. Gene differential FPKM. Tests difference sin the summed FPKM of transcripts sharing each gene_id
7. Primary transcript differential FPKM. Tests difference sin the summed FPKM of transcripts sharing each tss_id
8. Coding sequence differential FPKM. Tests difference sin the summed FPKM of transcripts sharing each p_id independent of tss_id
9. Differential splicing tests: this tab delimited file lists, for each primary transcript, the amount of overloading detected among its isoforms, i.e. how much differential splicing exists between isoforms processed from a single primary transcript. Only primary transcripts from which two or more isoforms are spliced are listed in this file.
10. Differential promoter tests: this tab delimited file lists, for each gene, the amount of overloading detected among its primary transcripts, i.e. how much differential promoter use exists between samples. Only genes producing two or more distinct primary transcripts (i.e. multi-promoter genes) are listed here.
11. Differential CDS tests: this tab delimited file lists, for each gene, the amount of overloading detected among its coding sequences, i.e. how much differential CDS output exists between samples. Only genes producing two or more distinct CDS (i.e. multi-protein genes) are listed here.

Visualization

The outputs of Cuffdiff are recorded as tables in text files and before we start the tough data mining procedure, it is always a good idea to first visualize those data to observe their properties directly. CummeRbund is just such a tool to visualize and manipulate the Cuffdiff output files. It is a R package and can be downloaded easily from the Bioconductor website http://bioconductor.org/packages/release/bioc/html/cummeRbund.html. The following code shows several visualization examples:

```
#Load the cummeRbund package

library('cummeRbund')

#Build the cummeRbund database form Cuffdiff output files

diff<-readCufflinks('E:/code/R/cummeRbund/cuffdiff_out')
```

#Plot the distribution of expression levels for each sample (Fig. 7)

csDensity(genes(diff))

#Use a volcano figure to inspect differentially expressed genes (Fig. 8)

csVolcano(genes(diff),"WT_0h","WT_lipidA_120",alpha=0.05,T)

#Observe the correlation level of gene expression in two samples (Fig. 9)

csScatter(genes(diff),"WT_0h","WT_lipidA_120",smooth=TRUE)

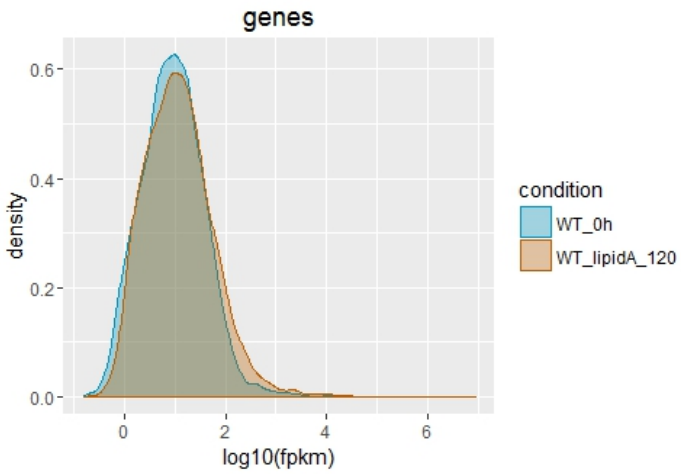

Fig. 7 The distribution of expression levels of each sample.
The blue stands for wild type cells, and the yellow stands for cells treated with lipid A.

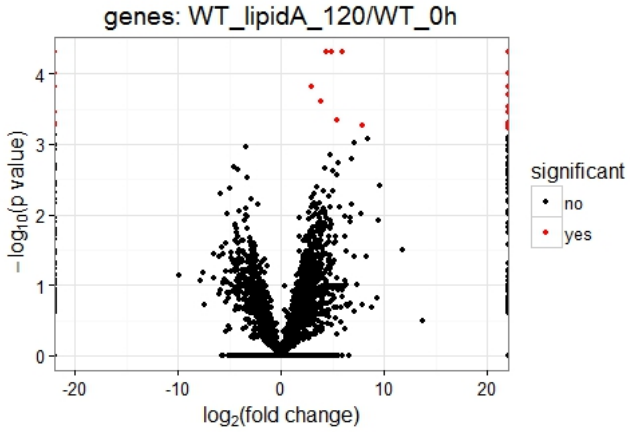

Fig. 8 The volcano figure that can be used to visualize the differential expressed genes.
The black dots stand for the control group and the treated group having no significant difference. The red dots show the two groups have significant difference.

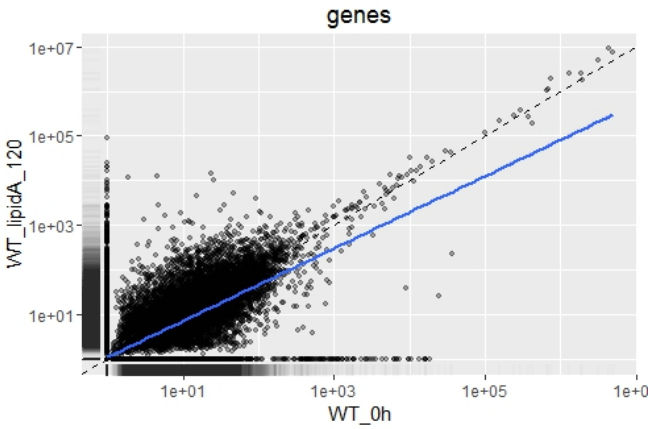

Fig. 9 The scatter plot that can be used to visualize the correlation level of gene expression between two samples. The farther from the center line of the points represent the smaller the correlation. On the contrary, the greater the correlation.

Besides the figures we introduced above, cummeRbund also supports many other kinds of plots, such as the barplot, to visualize the Cuffdiff output data. And those plot functions are all very friendly to use and have several additional options to suit your specific needs. Furthermore, cummeRbund can also be used to filter the data or make some simple queries. For example, the function getSig(diff) returns the gene id of significant genes in the differential expression test and the function getGenes(diff,GeneIds) makes a query and returns the related information for genes which match the GeneIds.

Functional Annotation

From the previous analysis, we may already know which genes expressed differentially between different conditions and their expression levels, but sometimes it is still not enough because, usually, the final goal of our data analysis is to reveal the biological meanings behind the data, only the gene ids or gene names obviously cannot tell much biological content. Making a functional annotation for the significant genes is a good method to solve this problem. By functional annotation, you can use your gene ids to search corresponding terms in some pre-built databases which provide the functional information of genes. There are many databases containing different kinds of functional information, as well as many tools or websites that can help you to do functional annotation, enrichment analysis and even results visualization. We will introduce several frequently used databases and tools as follows:

GO (http://www.geneontology.org/): The Gene Ontology (GO) project is a collaborative effort to develop and use ontologies to support biologically meaningful annotation of genes and their products in a wide variety of organisms. The GO ontologies provide a systematic language, or ontology, for the consistent description of attributes of genes and gene products, in three key biological domains that are shared by all organisms: molecular function, biological process and cellular component.[7]

KEGG (http://www.genome.jp/kegg/): The Kyoto Encyclopedia of Genes and Genomes (KEGG) is a project to link genes with higher order of functional information by computerizing current knowledge on cellular process and by standardizing gene annotations. There are three databases included in KEGG: PATHWAY for terms of the network of interacting molecules, GENES for collections of gene catalogs and LIGAND for the collection of chemical compounds in the cell, enzyme molecules and enzymatic reactions.[8]

DAVID (https://david.ncifcrf.gov/): The Database for Annotation, Visualization and Integrated Discovery (DAVID) is a free online bioinformatics resource developed by the Laboratory of Immunopathogenesis and Bioinformatics (LIB). To use it, you only need to upload your gene list and make choices of some simple options, then it can finish some data mining and functional annotation works automatically.[9-10]

KOBAS (http://kobas.cbi.pku.edu.cn/): KEGG Orthology-Based Annotation System (KOBAS) annotates sequences with KEGG Orthology terms and identifies the frequently occurring (or significantly enriched) pathways among the queried sequences compared against a background distribution. 5 pathway databases are used (KEGG Pathway, PID, BioCyc, Reactome, Panther) and 5 human databases (OMIM, KEGG Disease, FunDO, GAD, NHGRI GWAS).

Conclusion

In this paper, we have introduced a general workflow for differential expression analysis based on the RNA-seq data. We divided this workflow into six parts: (1) quality control and preprocessing, (2) reads alignment, (3) transcript assembly, (4) differential analysis, (5) visualization and (6) functional annotation. We not only introduced the theoretical meaning of each step but also included instructions on several useful tools, so that following our instruction readers can stride across the gap between theories and practice.

The tools we used in this paper such as FastQC, TopHat and Cufflinks are famous and credible. They have been used widely to manipulate and analyze the RNA-seq data and have stable performance in many different situations. But that does not mean that they are the best tools, there are still many other tools and they all have different features and advantages. Also, there are many studies about the comparison between those tools [12-14]. Choosing tools that have the best performance in your workflow can help you get the more effective results.

Finally, the differential expression analysis is just one of the most important applications of RNA-seq data. RNA-seq data, because of the high quality and amount of the information it carried, can be used in many different research directions. And many instructions and tools are available for biologists to make the best use of their RNA-seq data.

References

[1] Korpelainen, E., Tuimala, J. and Somervuo, P. (2014) RNA-seq data analysis: A practical approach. Boca Raton, FL, United States: CRC Press.

[2] Trapnell, C., Roberts, A., Goff, L., Pertea, G., Kim, D., Kelley, D.R., Pimentel, H., Salzberg, S.L., Rinn, J.L. and Pachter, L. (2012) 'Differential gene and transcript expression analysis of RNA-seq experiments with TopHat and cufflinks', Nature Protocols, 7(3), pp. 562–578.

[3] Tong, A.-J., Liu, X., Thomas, B.J., Lissner, M.M., Baker, M.R., Senagolage, M.D., Allred, A.L., Barish, G.D. and Smale, S.T. (2016) 'A stringent systems approach Uncovers gene-specific mechanisms regulating inflammation', Cell, 165(1), pp. 165–179. doi: 10.1016/j.cell.2016.01.020.

[4] Galaxy Available at: https://usegalaxy.org/ (Accessed: 7 September 2016).

[5] Kim, Daehwan and Pertea, Geo and Trapnell, Cole and Pimentel, Harold and Kelley, Ryan and Salzberg, Steven L (2013). TopHat2: accurate alignment of transcriptomes in the presence of insertions, deletions and gene fusions. In Genome Biology, 14 (4), pp. R36.

[6] Trapnell, Cole and Williams, Brian A and Pertea, Geo and Mortazavi, Ali and Kwan, Gordon and van Baren, Marijke J and Salzberg, Steven L and Wold, Barbara J and Pachter, Lior (2010). Transcript assembly and quantification by RNA-Seq reveals unannotated transcripts and isoform switching during cell differentiation. In Nature Biotechnology, 28 (5), pp. 511–515.

[7] The Gene Ontology project in 2008 (2007) Nucleic Acids Research, 36(Database), pp. D440–D444.

[8] Kanehisa, M. (2000) 'KEGG: Kyoto encyclopedia of genes and Genomes', Nucleic Acids Research, 28(1), pp. 27–30.

[9] Huang DW, Sherman BT, Lempicki RA. Systematic and integrative analysis of large gene lists using DAVID Bioinformatics Resources. Nature Protoc. 2009;4(1):44-57.

[10] Huang DW, Sherman BT, Lempicki RA. Bioinformatics enrichment tools: paths toward the comprehensive functional analysis of large gene lists. Nucleic Acids Res. 2009;37(1):1-13.

[11] Wu, J., Mao, X., Cai, T., Luo, J. and Wei, L. (2006) 'KOBAS server: A web-based platform for automated annotation and pathway identification', Nucleic Acids Research, 34(Web Server), pp. W720–W724. doi: 10.1093/nar/gkl167.

[12] Engström, P.G., Steijger, T., Sipos, B., Grant, G.R., Kahles, A., Alioto, T., Behr, J., Bertone, P., Bohnert, R., Campagna, D., Davis, C.A., Dobin, A., Gingeras, T.R., Goldman, N., Guigó, R., Harrow, J., Hubbard, T.J., Jean, G., Kosarev, P., Li, S., Liu, J., Mason, C.E., Molodtsov, V., Ning, Z., Ponstingl, H., Prins, J.F., Rätsch, G., Ribeca, P., Seledtsov, I., Solovyev, V., Valle, G., Vitulo, N., Wang, K., Wu, T.D. and Zeller, G. (2013) 'Systematic evaluation of spliced alignment programs for RNA-seq data', Nature Methods, 10(12), pp. 1185–1191.

[13] Fonseca, N.A., Marioni, J. and Brazma, A. (2014) 'RNA-Seq gene profiling - A systematic empirical comparison', PLoS ONE, 9(9), p. e107026.

[14] Seyednasrollah, F., Laiho, A. and Elo, L.L. (2013) 'Comparison of software packages for detecting differential expression in RNA-seq studies', Briefings in Bioinformatics, 16(1), pp. 59–70.