Chapter 18

Bioinformatic Analysis of Next-Generation Sequencing Data to Identify WT1-Associated Differential Gene and Isoform Expression

Stuart Aitken and Ruthrothaselvi Bharathavikru

Abstract

Differential gene expression analysis has been conventionally performed by microarray techniques; however with the recent advent of next-generation sequencing (NGS) approaches, it has become easier to analyze the coding as well as the noncoding components. Additionally, NGS data analysis also provides information regarding the expression changes of specific isoforms. There are several bioinformatics tools available to analyze NGS data but with different parameters. This chapter provides a comparative insight into these tools by utilizing NGS datasets available from Wt1 knockout and embryonic stem cell line model.

Key words Next-generation sequencing (NGS), Cuffdiff2, DESeq2, edgeR

1 Next-Generation Sequencing Data Analysis: Current Challenges

High-throughput sequencing is a rapidly developing technology with diverse applications including de novo DNA sequence assembly, SNP detection, and the detection of differentially expressed genes. In contrast with earlier techniques, there is no need to specify probe sequences or any restriction to a reference genome assembly [1]. Sequencing costs are reducing and this is another factor contributing to the increased use of this technique.

However, estimating the abundance of RNA transcripts from sequencing data is not without difficulty. Early approaches treated the read data simply as count data—read counts per transcript have been shown to be linearly related to transcript abundance—and used the Poisson distribution as the underlying statistical model [2]. Problems have subsequently been identified with this assumption as counts typically show a variance that is greater than the mean (mean and variance are the same in the Poisson distribution, which has a single parameter λ) [2]. The assumed distribution plays a role in testing for differential expression, hence impacts on the assignment of differential expression.

Nicholas Hastie (ed.), *The Wilms' Tumor (WT1) Gene: Methods and Protocols*, Methods in Molecular Biology, vol. 1467, DOI 10.1007/978-1-4939-4023-3_18, © Springer Science+Business Media New York 2016

It has been noted that the variance in sequencing read counts varies with the mean; hence many attempts have been made to estimate dispersion (the disparity between the variance and the mean) from the available data, usually as a function of the mean, and use the negative binomial distribution (which has two parameters, the mean and variance) when testing for differential expression.

The state-of-the-art tools for differential expression testing include DESeq2 [3], Cuffdiff2 [4], and edgeR [5]. DESeq2 adopts the negative binomial distribution and applies sophisticated techniques to estimate dispersion on a per-gene basis, detect outliers, and prevent type I errors (false positives). DESeq2 calculates an estimate of the fold change that is moderated, that is, reduced in absolute value in comparison with a simple estimate from raw read counts. Cuffdiff2 estimates the read counts for each isoform of each gene, rather than treating each gene as a single entity, adopting the beta-negative binomial distribution for testing differential expression. Trapnell et al. note that for genes with multiple isoforms, a change in fragment count for a gene does not necessarily mean a change in expression but may indicate a change in isoform abundance [4]. Distinguishing the expression of alternative isoforms is of interest in many situations, for example, to distinguish isoforms of Wt1 with and without the KTS sequence as described below.

A recent comparison of differential expression tools [1] concluded that the number of biological replicates was a major factor: where two or more replicates were available the tools made similarly good predictions. In the absence of replicates, differences in calls of significant genes were more notable.

In this chapter, we present protocols for running Cuffdiff2 and DESeq2. Following is the protocol for generating expression data from cell line models.

2 Materials

2.1 Cell Lines

Mouse ES cell line E14 and the Wt1 knockout ES line (KO1A) were cultured as a monolayer with retinoic acid (1 μM) for 5 days in ES cell media without LIF [6].

2.2 RNA Isolation

These cell lines were processed for RNA isolation using the Qiagen RNAeasy mini columns as per the manufacturer's protocol.

2.3 Library Preparation

The isolated total RNA was subjected to Poly A selection and subjected to library preparation with the NEBnext Ultra RNA library kit for Illumina for performing NGS.

3 Methods

3.1 RNA Isolation

1. Cells were harvested by trypsinization and collected in PBS followed by lysis in RLT buffer + β-mercaptoethanol as recommended. Centrifuged at 8049 × *g* for 3 min in a microfuge.
2. The supernatant was mixed with equal volume of absolute ethanol and added to the RNeasy Qiagen columns (700 μl at a time). Centrifuged at 13792 × *g* for 1 min in a microfuge.
3. The columns were washed with RW1 buffer (700 μl). Centrifuged at 13792 × *g* for 1 min in a microfuge.
4. The columns were washed with RPE buffer with ethanol (500 μl), twice. Centrifuge at 13792 × *g* for 1 min. Centrifuge again at 13792 g for 2 min in a microfuge.
5. To the columns, 30 μl of RNase-free water was added to elute RNA. The columns were centrifuged at 13792 × *g* for 1 min to collect the RNA sample in a microfuge tube.
6. RNA concentration was estimated by nanodrop and stored in −80 °C till further use.

3.2 Samples for RNA Sequencing

1. mRNA was polyA+ enriched from the total RNA sample of 1 μg.
2. cDNA synthesis was performed by random hexamer priming and subjected to enrichment.
3. The above samples were barcoded and multiplexed, and subjected to sequencing on the Illumina platform to obtain 50 bp single reads.

3.3 Data Analysis Protocols

Here we present the essential steps in the computational analysis of the unpaired 50 bp reads generated by Illumina sequencing described above. The following protocols are easily adapted to the situation where sequencing data for multiple biological replicates is available.

3.3.1 Cuffdiff2 Protocol

The Cuffdiff2 analysis requires the following tools to be installed and run at the command line: bowtie2 (v2.2.3), tophat2 (v2.0.13), cufflinks (v2.2.0), and samtools (0.1.18). Files from the Ensembl mouse genome assembly mm9/mm10 must also be installed (available from http://support.illumina.com/sequencing/sequencing_software/igenome.html). The following protocol is based on [7]. The steps in the protocol are organized into bash shell scripts that specify the resource files and command arguments needed. These scripts are designed to be run at the command line (full paths to files are omitted for brevity, they should be substituted for <path>).

Each replicate sequencing data set for each condition should first be aligned to the genome (**step 1**). Note that the label "X" should be replaced by a meaningful term such as wild type (WT) or knockout (KO) (which could be read from the command line). When using a gtf annotation file (tophat2 -G option), the chromosome names, i.e., 1, 2, 3 or chr1, chr2 chr3, in the gtf file must match those in the bowtie2 index (use bowtie2-inspect –names <index-file> to check). Cufflinks can also be run with a gtf file as input when attempting to identify novel transcripts in the context of an established reference [8] but this option is not essential [7].

Cuffdiff2 Step 1. Script to run tophat2 and cufflinks on sequencing data X (bowtie2 is used for alignment). Note that the results of the alignment and cufflinks results are written to the directories tophat_X and cufflinks_X, respectively. The samtools commands sort and index the bam file for use in genome browsers such as IGV.

```
#!/bin/sh
bowtie2index="<path>/Mus_musculus/Ensembl/NCBIM37/
Sequence/Bowtie2Index/genome"
gtffile="<path>/Mus_musculus/Ensembl/NCBIM37/
Annotation/Genes/genes.gtf"

tophat2 -p 4 -o tophat_X -G $gtffile $bowtie2index
sequencing_data_X.fastq
cd tophat_X
if test -f accepted_hits.bam
then {
          samtools sort accepted_hits.bam accepted_
          hitsSorted;
          samtools index accepted_hitsSorted.bam;
          mv accepted_hitsSorted.bam accepted_hits.bam;
          mv accepted_hitsSorted.bam.bai accepted_hits.
          bam.bai; }
fi
cd ..
cufflinks -p 4 -o cufflinks_X ./tophat_X/accepted_hits.bam
```

Once all data has been aligned (X and Y in the present example), a merged assembly of transcripts found in all conditions can be created by listing the cufflinks transcript outputs in a file called assemblies.txt (**step 2**), and running cuffmerge (**step 3**).

Cuffdiff2 Step 2. Create the assemblies.txt file that identifies the cufflinks transcripts to be merged in **step 3**.

```
<path>/cufflinks_X/transcripts.gtf
<path>/cufflinks_Y/transcripts.gtf
```

Cuffdiff2 Step 3. Script to run cuffmerge on the set of transcripts in the file assemblies.txt created in **step 2**. Note that the results are written to the directory merged_XY.

```
#!/bin/sh
gtffile="<path>/Mus_musculus/Ensembl/NCBIM37/Annotation/
Genes/genes.gtf"
```

```
fastafile="<path>/Mus_musculus/Ensembl/NCBIM37/Sequence/
WholeGenomeFasta/genome.fa"

cuffmerge -o merged_XY -g $gtffile -s $fastafile -p 4
assemblies.txt
```

The merged X–Y assembly and the aligned reads in X and Y are inputs to cuffdiff which performs the differential expression analysis (**step 4**). This step is very computationally intensive, and can be time consuming even when using 4 cores (-p 4 option).

Cuffdiff2 Step 4. Script to run cuffdiff on the mapped reads in data sets X and Y using the merged transcript file created in **step 3**. Note that the labels X and Y in –L X,Y should be replaced by something more meaningful such as wild type and knockout (–L WT,KO), and that the results are written to the directory cuffdiff_XY.

```
#!/bin/sh
fastafile="<path>/Mus_musculus/Ensembl/NCBIM37/Sequence/
WholeGenomeFasta/genome.fa"

cuffdiff -o cuffdiff_XY -b $fastafile -p 4 -L X,Y -u
merged_XY/merged.gtf \      ./tophat_X/accepted_hits.bam
./tophat_Y/accepted_hits.bam
```

The final step of the Cuffdiff analysis is performed in R using the Cummerbund library. **Step 5** shows the creation of the cuffdiff database object and the extraction of significant genes and isoforms from it. R version 3.2.1 was used here; note that earlier versions of R will not load Cummerbund v2.8.2. Additional details can be found in [7].

Cuffdiff2 Step 5. R code calling methods in Cummerbund to create the cuffmerge database and to extract significant genes and isoforms.

```
cuffData <- readCufflinks(dir="<path>/cuffdiff_XY",
gtfFile="<path>/merged_XY/merged.gtf",
genome="mm9", rebuild=TRUE);

sigGenes <- getSig(cuffData,level='genes',al
pha=0.05);
sigIsoforms <- getSig(cuffData,level='isoforms',al
pha=0.05);
```

3.3.2 DESeq2 Protocol

The DESeq2 [3] analysis also requires the reads in each dataset to be aligned to the genome. The alignment performed in **step 1** of the Cuffdiff protocol can be used. It is necessary to count the reads assigned to each gene to generate a table of raw counts, and this can be performed with htseq-count (v0.6.1p1) as shown in **step 1**.

DESeq2 Step 1. Command to generate counts per gene using htseq-count using the reads mapped by bowtie2 in **step 1** of the

Cuffdiff2 protocol. Note that the final lines in X_counts.tsv contain run information.

```
$htseq-count -s no -f bam ./tophat_X/accepted_hits.bam   \
<path>/Mus_musculus/Ensembl/NCBIM37/Annotation/Genes/
genes.gtf > X_counts.tsv
```

DESeq2 Step 2. R code calling methods in DESeq2 to create a data table from the htseq-count output files, run the DESeq analysis, and order the results by *p* value.

```
sampleTable <- data.frame(sampleName = c("WT","KO"),
                          fileName  =  c("X_counts.tsv",
                          "Y_counts.tsv"),
                          condition   =   c("untreated",
                          "treated"));
htseq <- DESeqDataSetFromHTSeqCount(sampleTable = sampleTable,
                                    directory = <path>,
                                    design= ~ condition);
colData(htseq)$condition <- factor(colData(htseq)$condition,
                                   levels=c("untreated","treated"));
htseq <- DESeq(htseq);
result <- results(htseq);
result <- result[order(result$pvalue),];
```

The remainder of the analysis is performed in R using the DESeq2 library. **Step 2** shows the creation of the count object

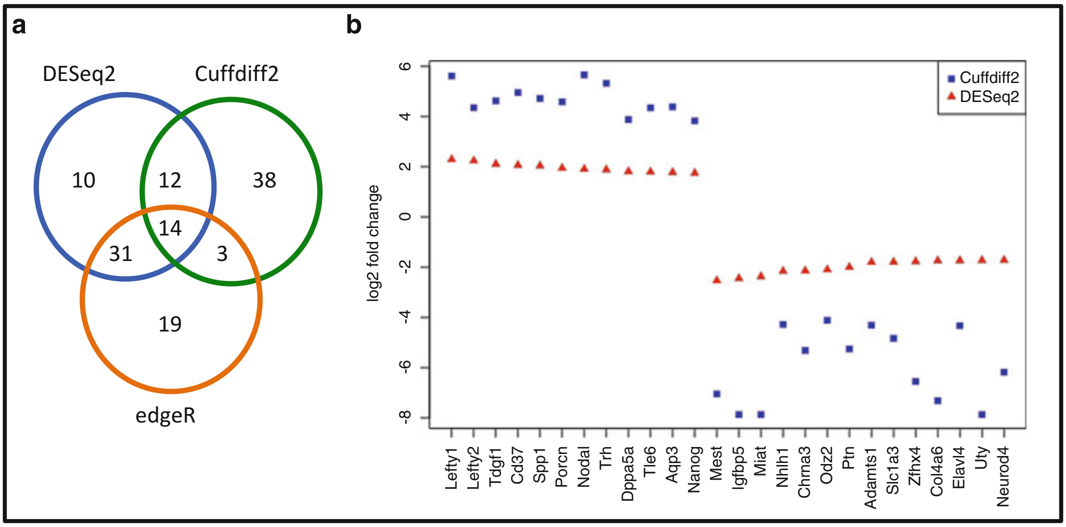
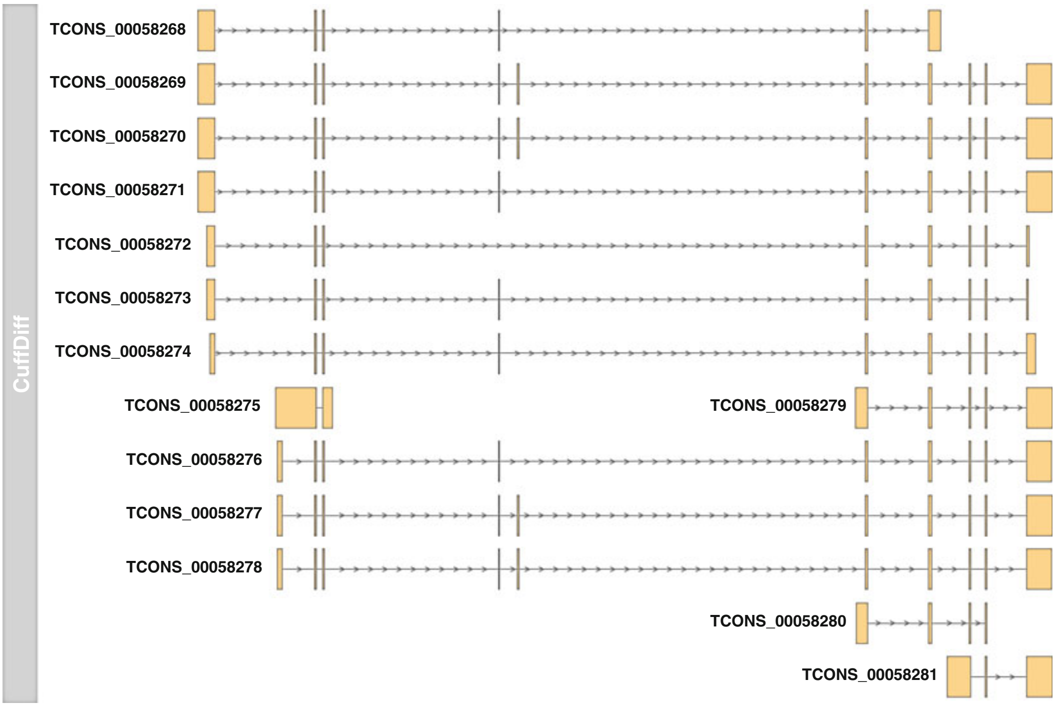


Fig. 1 Transcriptome changes in ES cells upon Wt1 knockout: (**a**) Venn diagram representation of the number of differentially regulated genes identified by the three different tools used for analysis, DESeq2, Cuffdiff2, and EdgeR. (**b**) Differential regulation of gene expression in Wt1 knockout cells compared to the ES cells represented as log2 fold change. Data points represent analysis by DESeq2 (*red triangles*) and Cuffdiff2 (*blue squares*)



No.	Isoform Id	KTS
1	TCONS_00058269	Present
2	TCONS_00058270	Present
3	TCONS_00058272	Present
4	TCONS_00058273	Present
5	TCONS_00058276	Present
6	TCONS_00058278	Present
7	TCONS_00058279	Present
8	TCONS_00058271	Not present
9	TCONS_00058274	Not present
10	TCONS_00058277	Not present
11	TCONS_00058281	Not present
12	TCONS_00058268	No exon 9
13	TCONS_00058275	No exon 9
14	TCONS_00058280	No exon 9

Fig. 2 RNA sequencing approach identifies Wt1 isoforms in ES cells: Different isoforms of Wt1 identified in the ES cells are represented with their identification numbers. The table represents information of the presence or absence of KTS in the above isoforms

from the output files of htseq-count, running the analysis, and extracting the results.

3.4 Results

Cuffdiff2 identified 67 regulated genes, including 24 upregulated and 43 downregulated genes (using a generous alpha value of 0.2). DESeq2 did not identify any significantly changed genes (all adjusted *p* values were >0.9) and Cuffdiff2 did not identify any isoforms with significant changes in the E14 data. To compare DESeq2 with Cuffdiff2, a set of the highest confidence genes (those with the highest regularized log 2 fold change calculated by DESeq2) of the same size as the set calculated by Cuffdiff2 was created. As a further comparison, edgeR [5] was run using a single value for dispersion estimated from the two samples available. The results of edgeR were filtered by *p* value to create a gene set of size 67: the intersection of the three sets of results is shown in Fig. 1a.

The differences in estimates of fold change calculated by DESeq2 and Cuffdiff2 are illustrated in Fig. 1b, where it can be seen that DESeq2 has reduced the fold changes to moderated values of approximately +2 or −2 from the greater estimates that follow from the read counts more directly. The alternative isoforms of Wt1 identified by Cuffdiff2 are shown in Fig. 2. There is sufficient information in the isoform annotation to identify those isoforms that contain the KTS sequence, those that do not, and those that lack exon 9. Wt1 isoforms are typically reduced in expression in the KO condition, some considerably; however, the Cuffdiff2 statistical model does not assign a significant adjusted *p* value.

4 Notes

1. Good-quality RNA is absolutely essential for an informative sequencing experiment. Although most sequencing experiments have now been modified so as to use starting material of very low nanogram concentration as well as to include formalin-fixed, paraffin-embedded (FFPE) samples, a good coverage can be guaranteed only from reasonably well-concentrated samples with a good RNA integrity number (RIN) value.
2. The agreement between DESeq2 and Cuffdiff2 is 34% for the data set analysed. Given the small number of genes identified, a practical strategy would be to consider the union of genes called as having (more) significant changes, and to consider the wider set called by edgeR. Considering the analysis in [1] we can conclude that the discrepancy (and lack of significant genes called by DESeq2) is most likely due to the lack of biological replicates, a common situation in exploratory studies. Hence, if possible, it is always advised to sequence replicates.

References

1. Zhang ZH, Jhaveri DJ, Marshall VM et al (2014) A comparative study of techniques for differential expression analysis on RNA-Seq data. PLoS One. doi:10.1371/journal.pone.0103207
2. Anders S, Huber W (2010) Differential expression analysis for sequence count data. Genome Biol 11:R106
3. Love IM, Anders S, Huber W (2014) Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2. Genome Biol 15:550
4. Trapnell C, Hendrickson DG, Sauvageau M et al (2013) Differential analysis of gene regulation at transcript resolution with RNA-seq. Nat Biotechnol 31(1):46–54
5. Robinson MD, McCarthy DJ, Smyth GK (2010) edgeR: a Bioconductor package for differential expression analysis of digital gene expression data. Bioinformatics 26:139–140
6. Spraggon L, Dudnakova T, Slight J et al (2007) hnRNP-U directly interacts with WT1 and modulates WT1 transcriptional activation. Oncogene 26(10):1484–1491
7. Trapnell C, Roberts A, Goff L et al (2012) Differential gene expression and transcript expression analysis of RNA-seq experiments with TopHat and Cufflinks. Nat Protoc 7(3):562–578
8. Roberts A, Pimentel H, Trapnell C et al (2011) Identification of novel transcripts in annotated genomes using RNA-Seq. Bioinformatics 27(17):2325–2329