



High Throughput Sequencing-Based Approaches for Gene Expression Analysis

R. Raja Sekhara Reddy and M. V. Ramanujam

Abstract

Next-generation sequencing has emerged as the method of choice to answer fundamental questions in biology. The massively parallel sequencing technology for RNA-Seq analysis enables better understanding of gene expression patterns in model and nonmodel organisms. Sequencing per se has reached the stage of commodity level while analyzing and interpreting huge amount of data has been a significant challenge. This chapter is aimed at discussing the complexities involved in sequencing and analysis, and tries to simplify sequencing based gene expression analysis. Biologists and experimental scientists were kept in mind while discussing the methods and analysis workflow.

Key words RNA, RNAseq, Transcriptome, NGS, Gene expression

1 Introduction

The next-generation sequencing (NGS) or high throughput DNA sequencing methods have emerged as central to answering fundamental biological questions on a genome wide scale, setting forth a revolution in biology. Since the invention of DNA sequencing, the technique has proven vital for studying the genome organization, stability and in turn molecular understanding of traits and diseases. The technical superiority of NGS makes it an excellent first step analysis choice to answer fundamental questions in modern biology [1]. Applications of NGS include genome sequencing, gene expression and epigenome analysis [2]. Molecular level comparison between species aided by NGS-based genome decoding facilitated better understanding of tree of life. The knowledge of gene conservation across species is providing insights to the molecular mechanisms of gene regulation.

RNA sequencing (RNA-Seq) is emerging as a standard research tool to address basic questions in biology such as cell cycle regulation, division, and divergence. RNA-Seq is superior to microarray

because it does not require prior sequence information of the organism and expression can be digitally quantified; it also detects the splicing patterns and posttranscriptional modifications [3]. RNA-Seq can be used to analyze the whole transcriptome, including mRNA, ncRNA, and smallRNA, and it has facilitated gene regulation profiling of nonmodel organisms like never before [4].

RNA-Seq can be performed using different NGS technologies such as pyrosequencing (Roche), sequencing by synthesis (Illumina), semiconductor sequencing (Ion torrent), single molecule real-time sequencing (Pacific Biosciences), and nanopore sequencing (Oxford nanopore). However, Illumina's sequencing by synthesis technique is widely used for RNA-Seq because of the data quantity requirements [5].

2 Methods

The NGS work flow can be broadly divided in to three major parts *viz.* (1) Library preparation, (2) Sequencing, and (3) Data analysis. Each of these parts includes different steps depending on the research questions to be answered.

2.1 RNA-Seq Library Preparation

Irrespective of the NGS technology, the first part in RNA-Seq experiment is library generation, which depends on the fraction of RNA (mRNA, smallRNA, transcriptome) to be investigated. Various commercial library preparation kits are available for different RNA-Seq applications. With the discovery of regulatory RNAs like small RNAs, microRNAs and long-noncoding RNAs (sRNA, miRNA, and lncRNA), recent RNA-Seq experiment are targeted towards whole transcriptomes; which can be used to quantitate mRNA and lncRNA transcripts.

First step of whole transcriptome library prep is depletion of rRNA since it constitutes 90–95% of cellular RNA content. Due to probe based design, efficiency of commercial rRNA depletion kits varies depending on the species. Irrespective of the kit used, 5–10% of the sequencing data would still contain rRNA reads. If an experiment's aim is to study the expression profile of protein coding genes/transcripts, enriching poly-A tail RNA molecules (in eukaryotes) is a flexible option.

The quality and quantity of the starting material is important to generate good sequencing libraries. Since library preparation is adapter ligation based, it is essential to quantify the sample accurately using a RNA binding dye and degradation of RNA should be checked on gel or using a fragment analysis method such as Caliper LabChip or Agilent Bioanalyzer.

Most of the commercially available kits include the following steps in RNA library preparation from mRNA or rRNA depleted samples.

1. Chemical fragmentation of RNA

RNA will be randomly fragmented by utilizing divalent cations (Mg^{2+}) and heat ($\sim 95^{\circ}C$).

2. First strand cDNA synthesis

Random oligos (hexamers) are used to prime the reverse transcription.

3. Second Strand synthesis

Depending on the requirement, the reaction mixture components would vary. For directional library preparation, dUTPs are used instead dTTPs, enabling quenching of second strand during library enrichment to retain strand information of RNA. (The strand information is utilized to discriminating transcript reads in genomic regions where both strands transcribe to generate different transcripts.)

4. Ligate adapters

The partial/full sequencing adapters are ligated in this step. When the full adapters are used, independent indexed (barcode) adapters are used for each sample.

5. Size selection

Depending on the sequencing technology to be used, the size of the library may vary. For example for Illumina sequencing the recommended size is (300–500 nt), whereas 454 platform is known to generate longer reads (800–1000 nt).

6. Enrichment PCR

Fragments with adapters on both the ends are enriched by PCR. The number of PCR cycles should be optimized depending on the input RNA quantity because increasing PCR cycles causes excessive PCR duplicates in the sequence data.

7. Quantity and quality check of library

It is essential to check the size distribution of the library using Bioanalyzer as it is vital to calculate the molarity of the library. The Library quantity should be measured with the use of fluorescent dye based assay.

The smallRNA library preparation requires a different approach because of the size of these molecules (20–32 nt). Total RNA (100 ng–1 μ g) or enriched small RNA (20–50 ng) would be used with protocols which leverage the presence of 5' phosphate and 3' OH in mature miRNA to ligate smallRNA specific adapters. The following steps are part of the small RNA library preparation

1. Ligate adapter(s)

The adapter(s) will be ligated to the smallRNA considering that only mature miRNAs contain 5' phosphate and 3' OH and hence adaptor(s) will only be ligated to those molecules.

2. Reverse transcription

The smallRNA ligated with adapter(s) will be reverse-transcribed using complementary adapter oligos.

3. Enrich library

Adapter ligated smallRNA molecules are enriched by PCR.

4. Size selection

It is essential to size select the smallRNA library because of its proximity with the adaptor dimers (~137 nt). Polyacrylamide gels are used to excise the smallRNA library (~145 nt) to avoid primer and primer dimer contamination.

5. Quality and quantity check

It is essential to check the size distribution of the library using Bioanalyzer as it is vital to calculate the molarity of the library. Library quantity should be measured with the use of fluorescent dye based assay.

The sequence data quality and experiment results are dependent on the quality of the library hence it is essential to generate and QC the sequencing library adhering to recommendations of the sequencing technology provider.

Some important considerations for library preparation

1. Target RNA fraction

Depending on the RNA fraction to be studied, a suitable protocol should be selected. If information of transcription strand is important, stranded/directional RNA library kits should be used. Similarly, if the aim is to study small or miRNA an appropriate kit for smallRNA library preparation should be used.

2. Quality of RNA

Samples with RNA integrity number (RIN) 7 or more, $OD_{260/280}$ and $OD_{260/230}$ close to 2 generate good quality libraries. In case of FFPE samples it is prudent to follow the kit recommendations for RNA QC.

3. Quantity check

Always use fluorescent dye based assay to quantitate RNA and library. Most of the commercially available library preparation methods require 10 ng–1 µg of RNA as starting material but if sample quantity is not a limitation it is better to start with at least 100 ng of RNA. Elute/dissolve RNA in 30–50 µL of low TE buffer.

4. Barcoding/Indexing

For multiplexing, libraries must be indexed with appropriate barcodes and while pooling compatibility of barcodes must be checked to avoid data loss or contamination.

5. Handling

Standard laboratory practices for RNA handling like use of dedicated RNA working area, RNase/nuclease-free plasticware, water, RNase inhibitor, barrier pipette tips, and enzyme aliquots to avoid freeze–thaw cycles should be followed. To avoid carryover contamination pre- and post-PCR areas should be separated.

2.2 Sequencing

To reap best results from NGS data, one should design the experiment appropriately. Any biological experiment without replicates could not deliver insightful results, and the same is applicable to RNA-Seq experiments as well. Most of the studies generate large number of reads per sample for an experimental condition, however, increasing number of reads may not provide articulate results [6, 7]. The ENCODE consortium in 2011 [8] set up guidelines for RNA-Seq experiments with regard to the number of reads, read length, biological replicates for differential gene expression analysis.

2.2.1 Important Considerations for Sequence Data Generation

1. Single or Paired end reads

With reduction in the cost of sequencing data generation, most researchers are opting for paired end reads because of superior mapping potential of paired end reads. However, single end reads can still be used in model organisms when cost is a limitation.

2. Read length

Using current sequence aligners, even short reads (<50 nt) can be mapped accurately for a model organism. For nonmodel organisms or organisms with repetitive regions in the genomes, 100 bp or longer reads will increase the mappability. Similarly, de novo RNA-Seq experiments require paired reads to generate longer contigs and scaffolds.

3. Number of replicates

At minimum, triplicates are recommended for each experimental condition. If high variability between replicates is expected, then increasing the number of replicates would provide better results.

4. Number of reads

In a differential expression analyses, RNA-Seq data is considered for tag-counting. If the experiment's goal is to identify differential expression of highly expressing transcripts in a model organism, 20–30 million paired reads would deliver the results [9, 10]. If the aim is to study the low expressed transcripts, then ~100 million reads or more would be needed. On the other hand, de novo RNA-Seq requires relatively more reads to generate an optimal transcriptome assembly. To determine the number of reads required, one can perform saturation analysis.

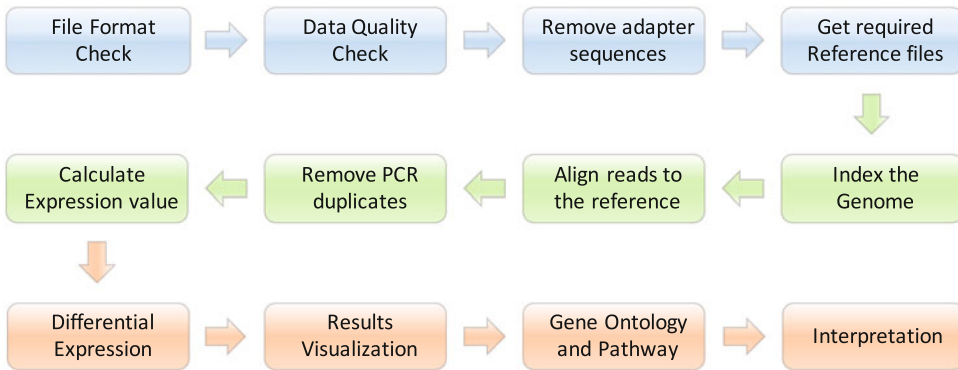


Fig. 1 Reference-based RNAseq data analysis workflow

2.3 Data Analysis

The sequence data should be filtered based on quality and further analyzed to generate interpretable results. Many biological researchers are not or vaguely familiar with the NGS data analysis workflows, causing delay in interpreting results. Since most of the analysis programs are command line applications, many researchers face difficulty in using them, and hence, in this chapter we discuss the data analysis process in a step-by-step manner (Fig. 1) using example data sets. To make the data analysis simple, we start with using Galaxy platform [11] and gradually migrate to command line tools.

2.4 Data Format

To analyze any data, one should be familiar with format of the data. In NGS, most widely used raw read data format is FASTQ; these files contain read name, read sequence, and quality value (of each base) of the read. A single fastq file can contain billions of reads. Once the sequencing run is completed the data will be demultiplexed to separate sample-specific reads and generate respective fastq files. If the data is not demultiplexed, we need to use some software utilities to demultiplex such files. A paired end sequence information is stored in two fastq files, one for forward (left) another for reverse (right). We will discuss about other formats of data as we proceed with the analysis.

2.5 Data Quality Check

Due to vast amounts of the data, it is practically impossible to check quality of each base present in fastq file(s). Summarized quality parameters like quality value, read length, base distribution across the reads, and presence of adapter sequences and duplicated sequences would provide overall information of data quality. To understand these parameters, we need to understand what they represent.

Quality value: The logarithmic probability of base calling error ($Q = -\log_{10} P$) [12]. To put it in perspective, Q value 30 means the probability of the base (nucleotide) being wrongly called is 0.001 and Q value 20 means probability of the base being wrongly called 0.01.

Read length distribution: percent of reads with their respective lengths.

Nucleotide distribution: It visualizes how A, T, G, C are distributed across all the reads at a nucleotide position. If all the reads have same nucleotide at a given position it could be a sequencing artifact. If all the reads have same sequence towards their 3' end it could be adapter sequence.

Read Duplication: Few sequences representing majority of the data indicates presence of rRNA contamination or PCR duplicates.

Since we know data format and which quality parameters to check, let us analyze an example data set containing four fastq files. This data is generated by sequencing a paired end stranded library using Illumina HiSeq platform.

2.5.1 Practice

To practice this data analysis, a computer with 4–8 GB RAM, 100 GB disk space and quad core processor is required. All the analyses mentioned in this chapter are performed on a MacBook Air. These analyses can also be performed using any computer with Unix-based Operating System (like Linux, Biolinux [13], Ubuntu, RHL, Fedora, and Linux Mint) or virtual drive with any Linux distribution. It is essential, to know the administrator/root password of the computer to install few software used in this chapter. We use Linuxbrew to install most of the applications utilized in this chapter. An Internet connection is a must for installing applications. Read every message shown at command prompt (terminal) as it will help in understanding the cause of error. Most of the time errors are caused due to spelling mistakes.

2.6 Install Few Helpful Software

Follow the links given below to install Git and Linuxbrew.

1. Installing Git

<https://git-scm.com/book/en/v2/Getting-Started-Installing-Git>

2. Brew or Linuxbrew

<http://linuxbrew.sh/> (For Linux)

<http://brew.sh/> (For Mac)

Check installation of brew by typing

```
brew install hisat2 to install hisat2 aligner.
```

3. Install Galaxy Platform (In Biolinux Galaxy is preinstalled)

Go to Galaxy project using the following link and check how to install latest version.

<https://new.Galaxyproject.org/admin/get-Galaxy/>

To get Galaxy, check for a command line like below.

```
git clone -b release_16.10 https://github.com/Galaxyproject/Galaxy.git
```

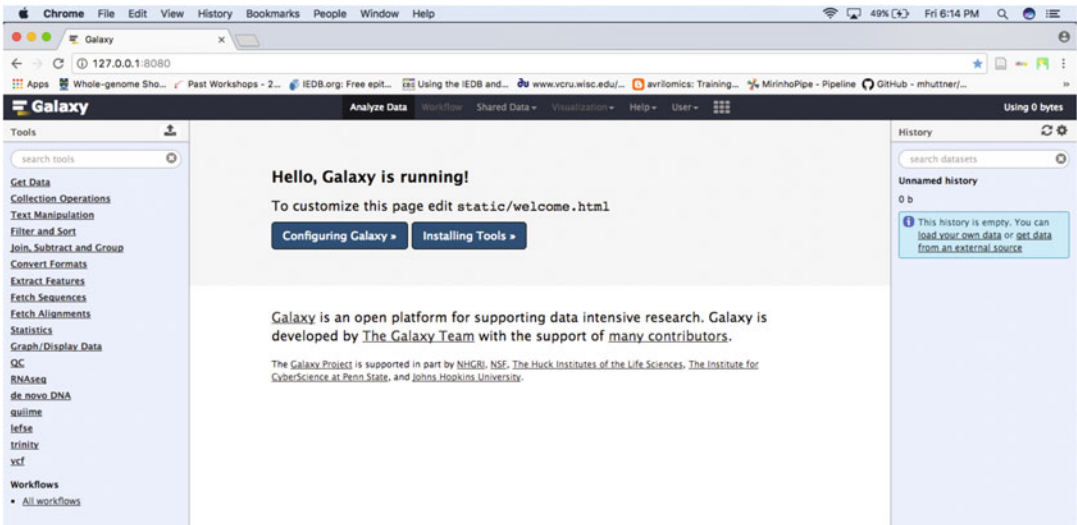


Fig. 2 Galaxy home page

Copy and paste the “git clone” command in terminal and press enter to download Galaxy into “galaxy” folder. Once the download is complete, from the same terminal window go to Galaxy folder by typing `cd galaxy` and start Galaxy by typing `sh run.sh`

Starting Galaxy for the first instance will take a bit long time as it requires to set up the environment and acquire some programs. Once the setup completes we can see an http link like below at the end of the terminal. The link is used to access locally installed Galaxy platform.

```
servng on http://127.0.0.1:8080
```

Copy the local Galaxy http link and paste it in chrome or Firefox web browser (Fig. 2). Create an account by clicking on “user” and providing email and password. Now we need to assign Galaxy administrator rights to the user we have registered.

Close the web browser and stop Galaxy by pressing “control+c” in the terminal where Galaxy is running. Now go to “config” folder located in galaxy folder and find “galaxy.ini” file. If the file does not exist, you can copy it from the sample “galaxy.ini.sample”. Open the file using a text editor and search for “#admin_users=” add the registered email after “=” and delete the “#” from the line. Save the “galaxy.ini” file after modifications. Now restart the Galaxy from the command line (terminal) by typing

```
sh run.sh
```

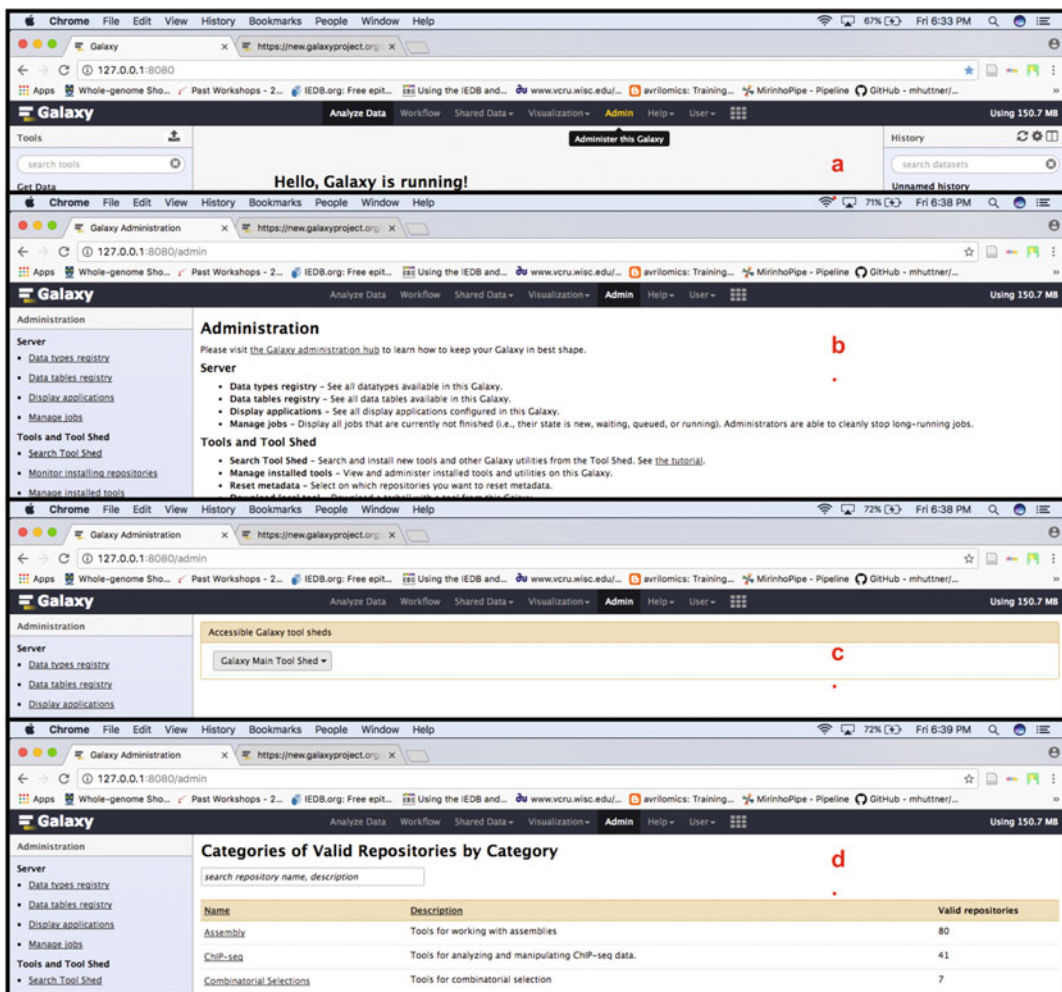



Fig. 3 Installing Galaxy tools

Open Galaxy in web browser and login with the email and password. Now we have an active “admin” option. This admin option is crucial to install any tool required for the analysis.




4. Install tools in Galaxy (Fig. 3)

- (a) Click on “admin.”
- (b) On the left click on “Search Tool Shed.”
- (c) Click on “Main Galaxy Tools Shed.”
- (d) Galaxy provides list of application categories, clicking on any of these categories will provide list of applications available for that category. To search a specific application/program we need to use name of that application in search bar.

For data QC, most widely used application is FastQC. Try to install it by your own (for help visit <https://wiki.galaxyproject.org/Learn>).

2.6.1 Quality Check Using FastQC

Since we have installed FastQC [14] tool in Galaxy, we will use it to check the quality of our data.

1. Open Galaxy in web browser.
2. Import data into Galaxy by clicking on “Get Data” or .
3. Click on FastQC from your installed tools (use top left search bar).
4. Select multiple datasets option () and select all the files.
5. Click execute and wait for the process to complete.
6. Once the analysis is complete please check the “webpage” output (in the history panel on left side) by clicking on eye icon (). Check quality parameters of all four fastq files.

2.6.2 Trimming and Filtering

Although the overall data looks acceptable we need to remove adapter sequences and low quality bases ($Q < 20$) from the ends of the reads using “Trim Galore!” [15]. Install Trim Galore in galaxy and quality-filter the data by using default parameters, paired end, and universal Illumina adaptor option (you may need to change the data type of files to “fastqsanger”). Once the trimming and filtering is complete, check the quality of the trimmed data using FastQC tool. Now our data is ready to be used for further analysis. Save these QC filtered fastq files to an analysis folder.

2.7 Getting Reference Sequence

Obtaining a genome version with proper annotations of exon, transcript, gene symbol, name, ontology, etc. is crucial for reference-based RNA-Seq data analysis. The annotation information of a genome is stored in a separate file in specific format. There are two widely used file formats for annotation: (1). GFF and (2) GTF; most of the applications used in RNA-Seq analysis would accept both of these formats. The Ensembl data base provides well annotated genomes for most of the sequenced organisms but for some organisms, dedicated web resources are available, which are more frequently updated as compared to ensemble database. For example, most widely used Rice genome (*Oryza sativa japonica*) is maintained by MSU Rice Genome Annotation Project team (<http://rice.plantbiology.msu.edu/>). Similarly PlasmoDB [16] maintains well-annotated genomes of different Plasmodium strains. A literature search before downloading a reference genome would help in obtaining a good annotated genome.

2.7.1 Practice

Since our data is from Human samples, check for different versions of genomes available and download GRCh 37 version genome (fasta file) and annotation file (.GTF).

2.8 Indexing the Genome

The RNA-Seq reads can be mapped to transcriptome; nonetheless, even for well-studied species such as human, we still do not know all transcripts, hence mapping the reads to the genome enables identification of novel transcripts and estimate their expression levels. Aligning reads to genome means, comparing the reads to the reference genome and finding a best match for each read but, NGS data contains billions of reads and mapping each read to the genome in a conventional manner (BLAST) requires humongous computational power and time. To answer this issue most NGS read mapping algorithms use Burrows–Wheeler transform (BWT), also known as block-sorting compression. In this approach the reference sequence will be transformed into small chunks of quick search compatible format which enable faster alignment of reads to the genome and this is know as “indexing”.

2.8.1 Practice

We have our quality filtered raw sequence data and well annotated genome, now we are ready to map (align) our sequence reads to the genome. Rename the genome and annotation files to “human_grch37.fasta” and “human_grch37.gtf” respectively and copy them to the analysis folder (the folder where our QC filtered fastq files were saved). Open a terminal window and change the directory to our analysis folder using command *cd*. The read mapping algorithm for RNA-Seq should be selected based on their ability to map reads generated from different splice forms of genes. In this chapter we use HISAT2 [17] due to its speed and low memory requirement.

Index the genome by typing the following command in terminal.

```
hisat2-build -t 2 human_grch37.fasta human_grch37
```

–t to specify number of processor cores to be used.

This command will generate multiple files with “.ht2” extension. The indexing process will take considerable time (~1–2 h) on a laptop depending on the configuration. We can generate reference index on any of the high end systems and copy it to any other computer. You may also try indexing the genome using Galaxy by installing HISAT2 tool.

2.9 Aligning Reads

Once genome is indexed, read mapping is straight forward, as most of the algorithms work fine with the default parameters. However, depending on the genome and NGS data, fine-tuning read mapping by altering few parameters may improve overall results. Sometimes comparing different algorithms would provide better insight into the results.

2.9.1 Practice

Rename the quality filtered fastq files for better understanding and tracking purpose to “trimmed_normal_1.fastq”, “trimmed_normal_2.fastq”, “trimmed_tumor_1.fastq,” and “trimmed_tumor_2.

fastq". To start alignment, type the following command in the terminal window.

```
hisat2 -p 4 --rna-strandness RF --dta -x human_grch37 -1
trimmed_normal_1.fastq -2 trimmerd_normal_2.fastq -S normal.
sam
```

-p is to specify number of processors to use

--rna-strandness specifies whether the library is strand specific or not since our data was generated using stranded library we use RF.

--dta enables reporting of alignments that can be used to identify novel transcripts.

-x to specify the base name of index files

-1 forward (left) reads file

-2 reverse (right) reads file

-S out put alignment file in SAM format.

Once the alignment is complete the following message will be shown in the terminal.

```
454369 reads; of these:
  454369 (100.00%) were paired; of these:
    5533 (1.22%) aligned concordantly 0 times
    403439 (88.79%) aligned concordantly exactly 1 time
    45397 (9.99%) aligned concordantly >1 times
    ----
    5533 pairs aligned concordantly 0 times; of these:
      452 (8.17%) aligned discordantly 1 time
      ----
    5081 pairs aligned 0 times concordantly or discordantly; of
these:
  10162 mates make up the pairs; of these:
    5239 (51.55%) aligned 0 times
    4213 (41.46%) aligned exactly 1 time
    710 (6.99%) aligned >1 times
99.42% overall alignment rate
```

This message denotes, our data contained 454,369 reads and all of them were paired. 88.79% of our reads mapped uniquely to the genome and 9.99% of reads mapped at more than one genomic location and 1.22% reads did not map to the genome. The Sequence Alignment/Map (SAM) file contains 11 mandatory fields for essential alignment information such as genome and read name, mapping position, sequence, and quality scores for each alignment.

Now try to map tumor sample reads and save the alignment to “tumor.sam” file.

The mapping quality can be assessed by different parameters like gene body coverage and proportion of reads mapped to features (exons). Use Qualimap2 [18] to assess the mapping quality.

2.10 Remove PCR Duplicates

We need to assess how many of the mapped reads originated from the same RNA molecule (PCR-duplicates) before calculating the expression levels. Computationally, read duplicates are defined by their mapping position, reads with same mapping position and length are considered as duplicates. There is no clear guideline on removing or retaining PCR duplicates [19], however if the PCR duplicates constitutes major fraction of the data, it is always good to compare the results with and without duplicates.

2.10.1 Practice

For this practice we use Picard tools [20] to estimate the fraction of PCR duplicates in the mapped reads. We need to sort SAM files by coordinate and convert them to Binary Alignment/Map (BAM) format as Picard takes sorted BAM format file as input. To work with SAM and BAM files we will use sambamba [21] and hence install both programs by typing the following commands in terminal.

```
brew install sambamba
brew install picard-tools
```

To convert SAM file to BAM use the following command.

```
sambamba view -f bam -S sam-o normal.bamnormal.sam
-f output file format
-S input file format
-o output file name
input file name
```

Now sort the bam file by coordinate

```
sambamba sort normal.bam
```

This command generates sorted bam file and index information of the bam file `normal.sorted.bam` and `normal.sorted.bam.bai` respectively. This bam file can be visualized using Integrative Genomics Viewer (IGV) [22].

Now use Picard tools to estimate duplicates percentage

```
picardMarkDuplicates I=normal.sorted.bam O=markdup.normal.sorted.bam M=markdup.normal.txt
```

I sorted BAM file
O duplicates marked BAM file
M metrics output file name

The metrics file contains details of number of duplicate read pairs, single tons, etc. In our “normal sample” ~13% of reads are duplicated. To check the file, open it with excel or any other spreadsheet app. In this practice we are not going to remove duplicates, you can compare these results with duplicates-removed results by yourselves.

2.11 Calculate Expression Values

In RNA-Seq experiments, gene/transcript expressions are measured by counting the reads mapped to its respective position in the genome. The expression values can be presented in different forms like read counts, RPM, RPKM, FPKM, and TPM. If relative expression of a transcript with respect to other transcripts in a sample is to be measured, then RPM, RPKM, FPKM, or TPM are used as the expression needs to be normalized. On the other hand, to compare expression of two samples, using read counts is the better option.

One frequently asked question is “Should we use total mapped reads or only uniquely mapped reads?” to estimate the expression levels. Some genomes, especially plants, contain high level repetitive regions and many of these repetitive regions contain genes or pseudo genes. In such genomes half of the reads may be mapped to multiple locations. In such instances, it is better to use algorithms which can assign counts to the multiple features. Similarly, if a read is mapped to two overlapping features it would be worth while to assign a count to both the features. For better understanding, one can always compare the results with and without multimapped and overlapped reads for expression analysis.

2.11.1 Practice

For this analysis we will use featureCounts [23] which is a part of subread application and can be downloaded from <https://sourceforge.net/projects/subread/files/subread-1.5.1/>. If you are not familiar with compiling source code, download binary distribution that suits your operating system. Once download is complete, unzip the archive and copy featureCounts executable file from “bin” folder and paste it in our working directory. Type the following command to get gene level summarized read counts.

```
featureCounts -p -T 4 -M -O -a human_grch37.gtf -o gene.
expression.txt normal.sorted.bamtumor.sorted.bam
```

- p for paired end reads
- T number of processors to use
- M to consider multimapped reads
- O to consider overlapping reads
- a genome annotation file
- o output filename

The output file `gene.expression.txt` is a tab delimited text file which can be opened with a spreadsheet application. Transcript level expression can be calculated by providing an extra option `--g "transcript_id"`.

2.12 Identification of Novel Transcripts

In the previous practice we used reference annotation to quantify expression of known genes and transcripts. Using `stringTie` [24] program we can assemble novel transcripts in genome guided or de novo mode. Install `stringTie` and identify novel transcripts in reference guided mode by the following steps.

```
brew install stringtie
stringtie -G human_grch37.gtf -o normal.transcripts.gtfnormal.sorted.bam
stringtie -G human_grch37.gtf -o tumor.transcripts.gftumor.sorted.bam
```

`-G` annotation file to be used as guide

`-o` output GTF file name

`stringTie` assigns arbitrary transcript IDs to each assembled transcript, therefore each GTF file (normal and tumor) may have different set of transcripts. There may be similarities between GTF files, but the number of transcripts and their exact structure will differ in the output files for each sample. One solution for this problem is to merge the GTF files and use it for expression quantification using `stringTie` merge option.

```
stringtie --merge -o merged.gtf -G huam_grch37.gtf normal.transcripts.gftumor.transcripts.gtf
```

The “merged.gtf” can be used with `featureCounts` to generate transcript level summarized read counts into “merged.transcripts.txt”. To annotate the novel transcripts use `gffcompare` program. (<https://ccb.jhu.edu/software/stringtie/gffcompare.shtml>).

2.13 Differential Expression

Various statistical methods are available for differential expression of RNA-Seq; they vary in data normalization and distribution model considerations. Algorithms which use negative binomial distribution like `DESeq` [25] and `edgeR` [26] are considered to be sensitive and specific. However, using more than one differential expression analysis method would provide better results in detecting true positives.

2.13.1 Practice

In this practice we will use `DESeq2` for differential expression analysis and we require R statistical program to use `DESeq2`. R is an open source language and environment for statistical computing and graphics. Availability of number of packages for different

applications makes R one of the widely used statistical programs in the field of genomics. Though R is a command line driven program, availability of Integrated Development Environments like RStudio makes use of R relatively. So we will install R and using brew and RStudio by downloading the installer.

```
brew install R
https://www.rstudio.com/products/rstudio/download/
```

Open RStudio and install Bioconductor Installer by typing the following command in the console panel.

```
source("https://bioconductor.org/biocLite.R")
biocLite()
```

Now install DESeq 2 package by typing the following command in console panel.

```
biocLite("DESeq2")
```

The installation should end with **Done (DESeq2)*. We can check if the package was installed or not by loading it into the R environment by typing the following command.

```
Require(DESeq2)
```

If a package is not installed, “there is no package called DESeq” will be returned. In such instance, one needs to check the warning or error messages at the end of the package installation.

Now import the tab delimited text file (merged.transcripts.txt) generated using featureCounts by clicking on “Import Dataset” in the Environment panel and selecting “From CSV” options from the dropdown list. In the import window change import options “Delimiter:” to Tab and “Comment:” to # (Fig. 4). Upon successful import, the data will be shown in a panel above the console panel and environment panel will have new dataset entry.

Check the dimension of the data frame “merged_transcripts” by typing

```
dim(merged_transcripts)
```

which returns

```
[1] 196678      8
```

There are total 196,678 rows and 8 columns in our data frame merged_transcripts.

We only require the counts data for differential expression analysis hence, let us create a new data frame “countdata” by retaining columns “Geneid”, and two more columns containing count data of our normal and tumor samples.

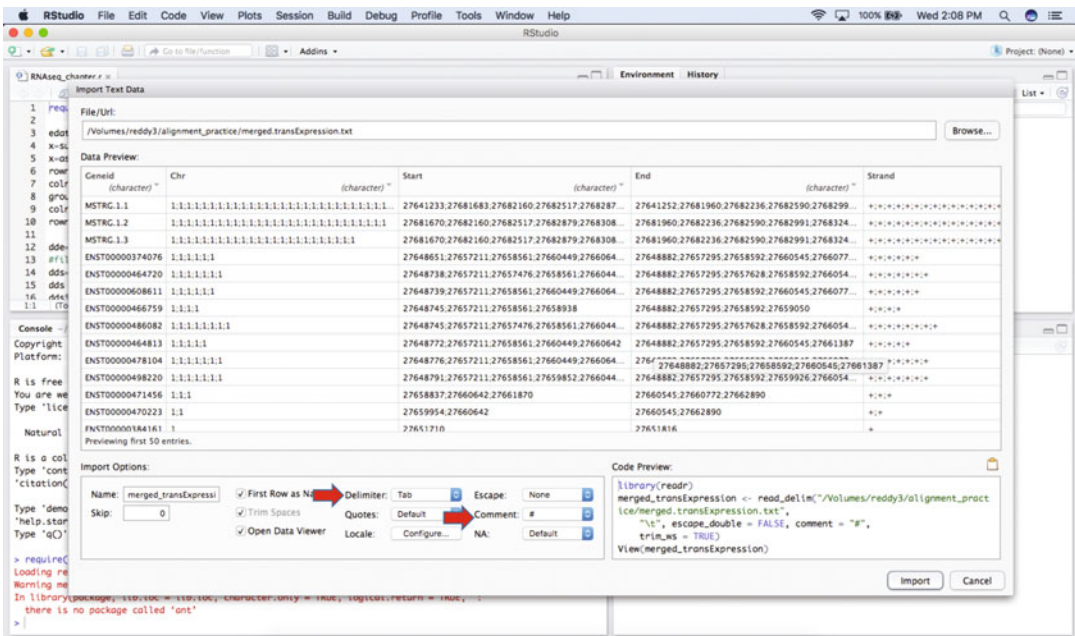


Fig. 4 Importing tab delimited data into R environment using Rstudio

```
countdata=merged_transcripts[,c(1,7,8)]
```

Above command instructs R to retain columns 1, 7, 8 in the merged_transcripts data frame and save them to new data frame “countdata”. Now check the dimension of countdata.

For DESeq analysis, we need a matrix as input so, transform countdata data frame into matrix of counts with transcript ids as its row names.

```
y=as.matrix(countdata[,c(2,3)])
rownames(y)=countdata$Geneid
```

Change the column name of y to sample names “normal” and “tumor”.

```
colnames(y)=c("normal", "tumor")
```

Create comparison groups and set the column name to condition and row names to sample id.

```
group=as.matrix(c("normal", "tumor"))
colnames(group)="condition"
rownames(group)=c("normal1", "tumor1")
```

If we have biological replicates then the above command will change to

```
group=as.matrix(c("normal", "normal", "normal", "tumor",
"tumor", "tumor"))
colnames(group)= "condition"
rownames(group)=c("normal1", "normal2", "normal3", "tumor1",
"tumor2", "tumor3")
```

Load DESeq2 if not already loaded and create DESeqDataSet.

```
require(DESeq2)
dde=DESeqDataSetFromMatrix(countData = y, colData = group,
design = ~condition)
```

Transcripts which have low read counts need to be filtered before differential expression analysis. Let us retain transcripts which have at least 1 read in any of the samples and store them to “dds” and compare dimensions of data before and after filtering.

```
dds=dde[ rowSums(counts(dde)) > 1, ]
dde
dds
```

We can see only 819 transcripts out of 196,678 have at least 1 read in at least one of the samples.

Now specify the condition that should be considered as reference for differential expression analysis.

```
dds$condition<- relevel(dds$condition, ref="normal")
```

Perform Differential expression (DE) analysis

```
dds=DESeq(dds)
```

DESeq2 will normalize the counts and parameters based on the number of samples, dispersion, variation between replicates, etc. Once the analysis is complete save the results in new variable “res”.

```
res=results(dds)
```

Visualize DE results by creating MA (Fig. 5) and Volcano plots.

```
plotMA(res, main="DESeq2", ylim=c(-10,10), alpha=0.1)
```

For generating volcano plot (Fig. 6) install and load Bioconductor package “a4Base”.

```
require(BiocInstaller)
biocLite("a4Base")
require(a4Base)
volcanoplotter(logRatio = res$log2FoldChange, pValue = res
$pvalue, pointLabels = rownames(res), topPValues = 3,
topLogRatios = 3)
```

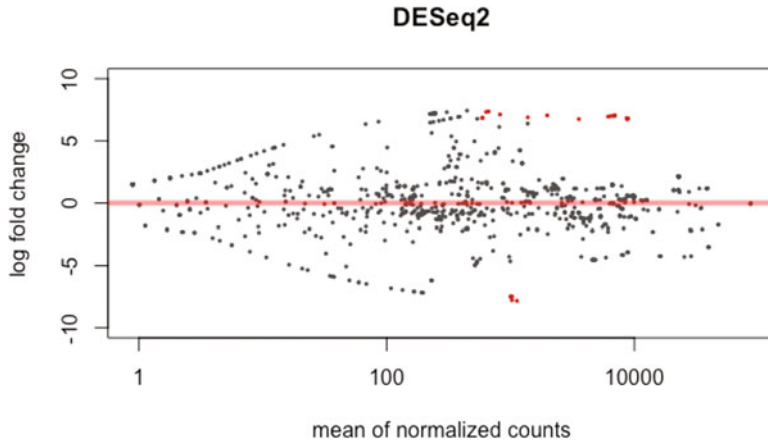


Fig. 5 Differential expression MA plot

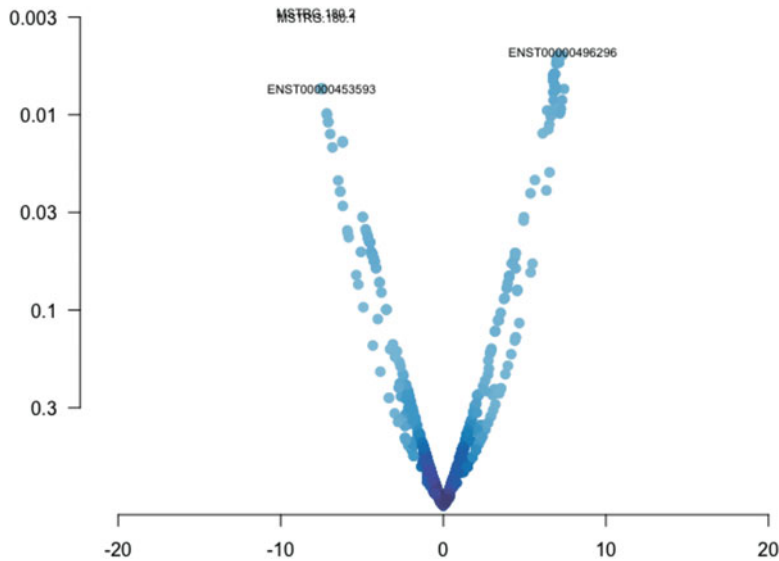


Fig. 6 Differential expression volcano plot

In the volcano plot we can see the top three transcripts with high \log_2 foldchange and low p -value.

We can export differential analysis results to a comma delimited file to Desktop.

```
write.csv(as.data.frame(res), file="~/Desktop/DE_results.csv")
```

If we want to separately save significantly under/over expressed transcripts, subset the results accordingly.

```

DResults=as.data.frame(res)
underEx=subset(DResults, DResults$log2FoldChange < -2
&DResults$pvalue< 0.05)
overEx=subset(DResults, DResults$log2FoldChange > 2
&DResults$pvalue< 0.05)
write.csv(underEx, file=~ /Desktop/under_expressed.csv")
write.csv(overEx, file=~ /Dekstop/over_expressed.csv")

```

2.14 Gene Ontology and Pathway Analysis

The Gene Ontology (GO) describes gene function and its classification based on their function. GO information is mainly used for enrichment analysis of gene sets that are up or down regulated under certain conditions. Gene set enrichment analysis will find which GO terms are overrepresented (or underrepresented). Similarly, pathway enrichment analysis will identify the major pathways in which these genes are involved.

2.14.1 Practice

Within R, there are many packages to perform gene set enrichment analysis but, due to ease, we will use Database for Annotation, Visualization, and Integrated Discovery (DAVID) [27]. Open <https://david.ncicrf.gov/> in web browser and select functional annotation.

1. Copy and paste ids of over expressed transcripts in upload gene list box.
2. Select the identifier “ENSEMBL_TRANSCRIPT_ID” from drop down menu. Select “Gene list” and submit the list (Fig. 7).
3. Some of our transcript ids will not be mapped because they were specific to our experiment hence in the next page click on “Continue to Submit the IDs That DAVID Could Map” (Fig. 8).
4. In next page (Fig. 9) select functional annotation chart.
5. Clear all default selections in next page (Fig. 10) and in Gene Ontology select GOTERM_BP_DIRECT, GOTERM_CC_DIRECT, and GOTERM_CC_DIRECT. Similarly select KEGG_PATHWAY in Pathways and click on Functional Annotation Chart button.
6. Results will open in a separate browser window (Fig. 11) and clicking on download file will open a text file in web browser. We can copy and paste the data in a text editor or spreadsheet application.

You may try performing gene set enrichment analysis for under expressed transcripts as well.

The screenshot displays the DAVID Functional Annotation Tool interface. At the top, there is a navigation bar with links: Home, Start Analysis, Shortcut to DAVID Tools, Technical Center, Downloads & APIs, Term of Service, Why DAVID?, and About Us. Below the navigation bar, a welcome message reads: "*** Welcome to DAVID 6.8 with updated Knowledgebase (more info). ***" and "*** If you are looking for DAVID 6.7, please visit our development site. ***".

The main content area is titled "Functional Annotation Tool". It features a blue sidebar on the left with the following sections:

- Upload Gene List**: Includes links for "Demolist 1", "Demolist 2", and "Upload Help".
- Step 1: Enter Gene List**:
 - A: Paste a list**: A text input field containing the following ENST transcript IDs: ENST00000357327, ENST00000460047, ENST00000488470, and ENST00000475336. A "Clear" button is next to the field.
 - Or**
 - B: Choose From a File**: A "Choose file" button with the text "No file chosen" and a "Multi-List File" checkbox.
- Step 2: Select Identifier**: A dropdown menu currently set to "ENSEMBL_TRANSCRIPT_ID".
- Step 3: List Type**: Radio buttons for "Gene List" (selected) and "Background".
- Step 4: Submit List**: A "Submit List" button.

The main content area includes a blue arrow pointing left with the text "Submit your gene list to start the tool!". To the right, there are links: "Tell us how you like the tool", "Read technical notes of the tool", and "Contact us for questions".

Below this, the "Key Concepts:" section lists three main features:

- Term/ Gene Co-Occurrence Probability**: Describes ranking functional categories based on co-occurrence with sets of genes in a gene list to aid in unraveling biological processes. It mentions that DAVID 6.8 allows sorting by the number of genes within each category or by the EASE-score. A "More" link is provided.
- Gene Similarity Search**: Explains that genes sharing a similar set of annotation terms are likely involved in similar biological mechanisms. The algorithm groups genes based on the agreement of sharing similar annotation terms by Kappa statistics. A "More" link is provided.
- Term Similarity Search**: States that typically, a biological process/term is done by a corporation of a set of genes. If two or more biological processes are done by a similar set of genes, they might be related in the biological network. The search function identifies related biological processes/terms by quantitatively measuring the degree of agreement in how terms share similar participating genes. A "More" link is provided.

At the bottom, the "Integrated Solutions" section highlights "Functional Annotation", noting that numerous public sources of protein and gene annotation have been parsed and integrated.

Fig. 7 Uploading Gene/Transcriptome list to DAVID

2.15 De Novo Transcriptome Analysis

One of the major advantages of RNA-Seq is its capability to study expression profile of organism for which reference genome or transcriptome are not available. Analyzing such RNA-Seq data consists two extra steps compared to the aforementioned reference-based analysis. (1) Assembling Transcriptome and (2) Annotating assembled transcripts. Once an annotated transcriptome is available, one can follow the reference based approach for differential expression analysis.

There are various de novo transcriptome assemblers [28–30], most of them work on the same principle “De Bruijn graph” [31]. According this method, the reads are broken into small K-mers and the overlapping K-mers are collapsed to make contigs. It is important to select the right K-mer length to get optimal assembly, hence generating assemblies with different K-mer length and comparing them is widely practiced [32]. The de novo transcriptome needs to be validated before using it as reference for differential expression analysis. Number of contigs, distribution of contig length and how many of them are annotated will provide

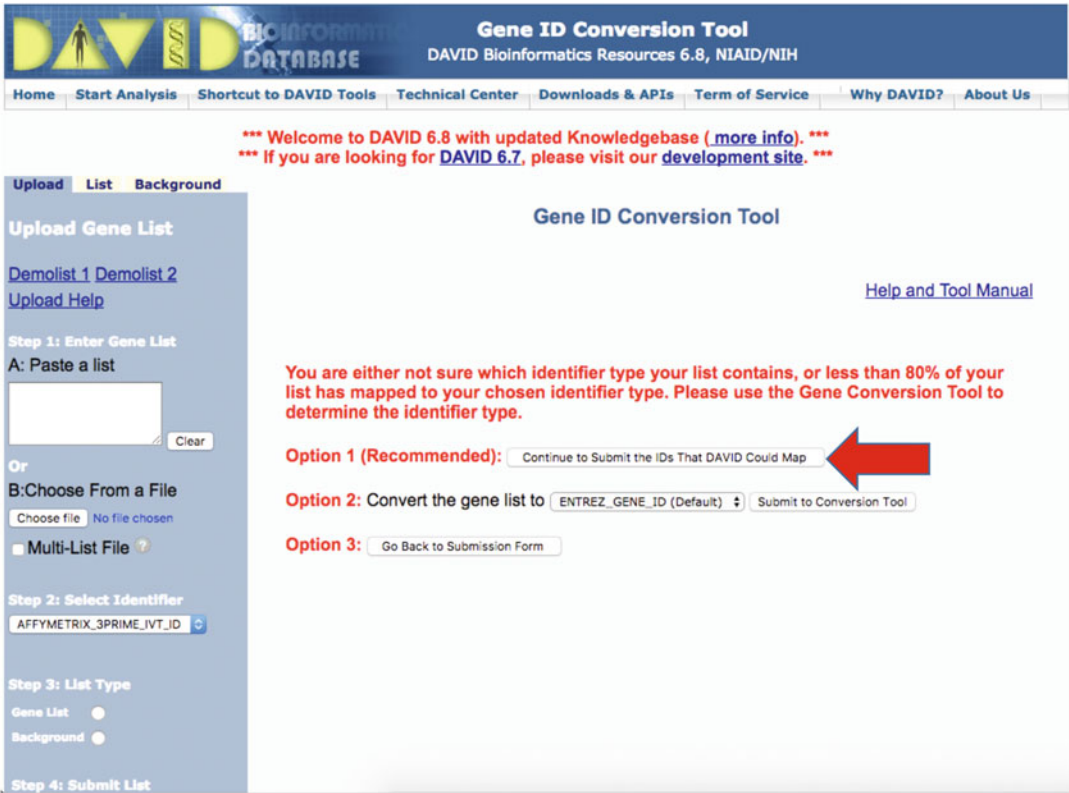


Fig. 8 Converting Gene/Transcript IDs

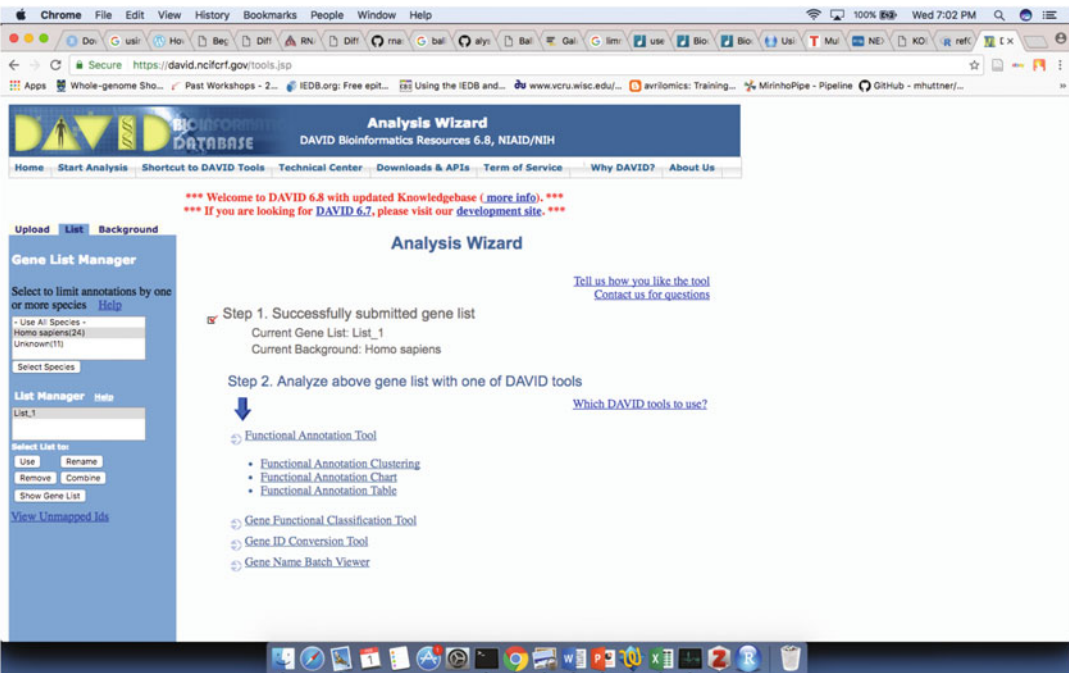


Fig. 9 Getting annotation chart data

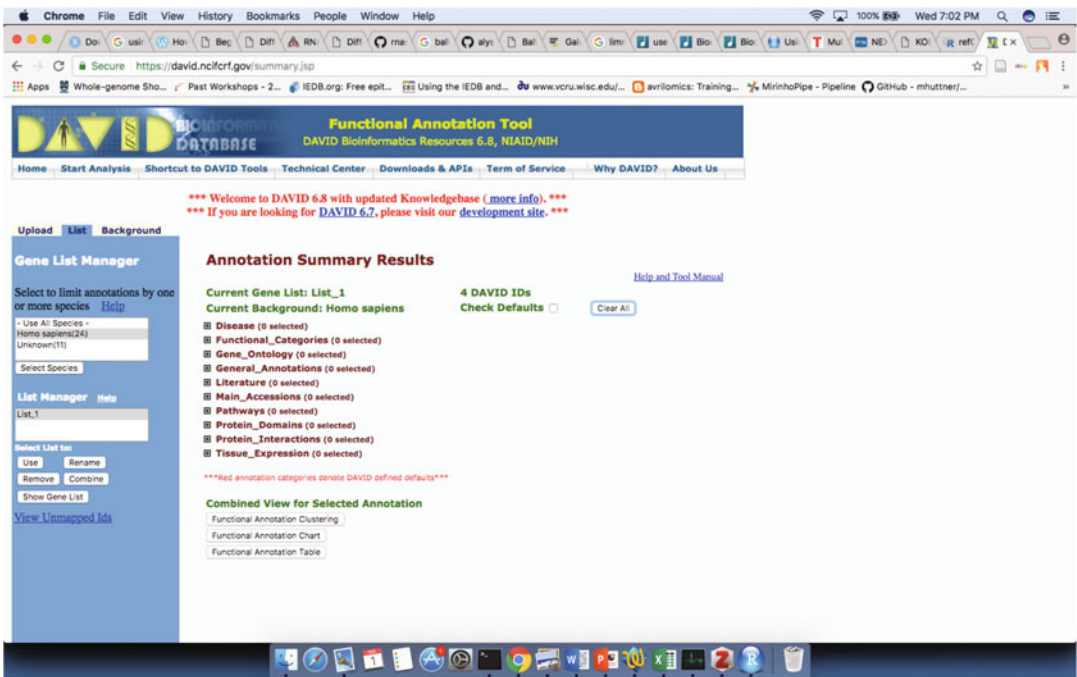


Fig. 10 Gene Ontology and Pathway parameter setup

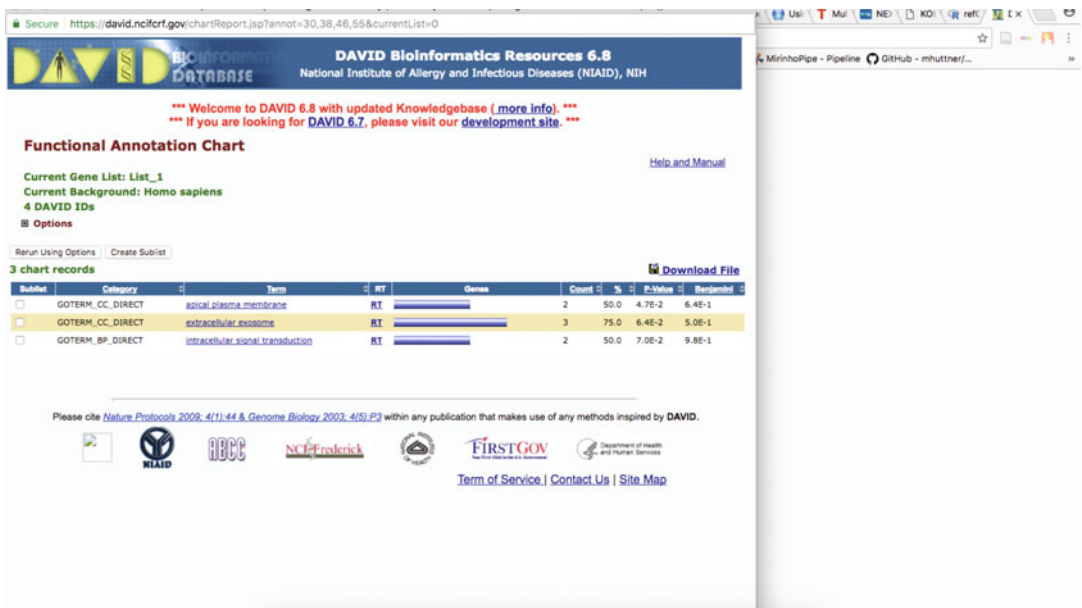


Fig. 11 Exporting GO and Pathway enrichment data

basic impression of how well the transcriptome is assembled [33]. In addition, what proportion of raw reads gets mapped onto the transcriptome also provides an idea of assembled transcriptome. The contigs are scaffolded with the paired end read information, as one of the paired reads are mapped to one contig and the other to another contig, these two contigs are stitched together. The scaffolds are annotated by BLASTing them against protein and noncoding RNA databases like Uniref90, NCBI non-redundant (nr) protein database using BLAST or similar tools, and the annotated contigs are termed Unigenes.

2.15.1 Practice

Using brew install “trinity” de novo transcriptome assembler using brew and generate de novo transcriptome from our practice data by default parameters. Scaffold the trinity contigs using SSPACE [34] default parameters. Check how many reads are mapping onto the scaffolds using HISAT2. Annotate scaffolds using Blast2GO [35] and export the annotation information into a tab delimited text file. Create a GTF file using the Blast2GO output and use it as reference annotation for scaffolds to perform differential expression analysis.

3 Conclusion

In conclusion, RNA-Seq is by far the best method available to study gene expression in model and nonmodel organisms. Success of any study depends on the experiment design and right amount of data. Careful selection of right tools from the library preparation to differential expression analysis would provide great insights into gene expression and functional profile of the study organism.

References

1. Mardis ER (2008) Next-generation DNA sequencing methods. *Annu Rev Genomics Hum Genet* 9:387–402
2. Buermans HPJ, den Dunnen JT (2014) Next generation sequencing technology: advances and applications. *Biochim Biophys Acta BBA* 1842:1932–1941
3. Koboldt DC, Steinberg KM, Larson DE, Wilson RK, Mardis E (2013) The next-generation sequencing revolution and its impact on genomics. *Cell* 155:27–38
4. Mutz K-O, Heilkenbrinker A, Lönne M, Walter J-G, Stahl F (2013) Transcriptome analysis using next-generation sequencing. *Curr Opin Biotechnol* 24:22–30
5. Mardis ER (2013) Next-generation sequencing platforms. *Annu Rev Anal Chem (Palo Alto Calif)* 6:287–303
6. Manga P et al (2016) Replicates, read numbers, and other important experimental design considerations for microbial RNA-seq identified using *Bacillus thuringiensis* datasets. *Front Microbiol* 7:794
7. Schurch NJ et al (2016) How many biological replicates are needed in an RNA-seq experiment and which differential expression tool should you use? *RNA* 22:839–851
8. Rosenbloom KR et al (2013) ENCODE data in the UCSC Genome Browser: year 5 update. *Nucleic Acids Res* 41:D56–D63
9. Sims D, Sudbery I, Illott NE, Heger A, Ponting CP (2014) Sequencing depth and coverage: key considerations in genomic analyses. *Nat Rev Genet* 15:121–132
10. Conesa A et al (2016) A survey of best practices for RNA-seq data analysis. *Genome Biol* 17:13

11. Afgan E et al (2016) The Galaxy platform for accessible, reproducible and collaborative biomedical analyses: 2016 update. *Nucleic Acids Res* 44:W3–W10
12. Ewing B, Green P (1998) Base-calling of automated sequencer traces using phred. II. Error probabilities. *Genome Res* 8:186–194
13. Field D et al (2006) Open software for biologists: from famine to feast. *Nat Biotechnol* 24:801–803
14. Andrews, S. FastQC A Quality control tool for high throughput sequence data. Available at: <http://www.bioinformatics.babraham.ac.uk/projects/fastqc/>. Accessed: 29th June 2016
15. Babraham Bioinformatics - Trim Galore! Available at: http://www.bioinformatics.babraham.ac.uk/projects/trim_galore/. Accessed: 30th January 2017
16. Bahl A et al (2003) PlasmoDB: the Plasmodium genome resource. A database integrating experimental and computational data. *Nucleic Acids Res* 31:212–215
17. Kim D, Langmead B, Salzberg SL (2015) HISAT: a fast spliced aligner with low memory requirements. *Nat Methods* 12:357–360
18. Okonechnikov K, Conesa A, García-Alcalde F (2016) Qualimap 2: advanced multi-sample quality control for high-throughput sequencing data. *Bioinformatics* 32:292–294
19. Parekh S, Ziegenhain C, Vieth B, Enard W, Hellmann I (2016) The impact of amplification on differential expression analyses by RNA-seq. *Sci Rep* 6:25533
20. Picard Tools - By Broad Institute. Available at: <http://broadinstitute.github.io/picard/>. Accessed: 31st January 2017
21. Tarasov A, Vilella AJ, Cuppen E, Nijman IJ, Prins P (2015) Sambamba: fast processing of NGS alignment formats. *Bioinformatics* 31:2032–2034
22. Thorvaldsdóttir H, Robinson JT, Mesirov JP (2013) Integrative Genomics Viewer (IGV): high-performance genomics data visualization and exploration. *Brief Bioinform* 14:178–192
23. Liao Y, Smyth GK, Shi W (2014) featureCounts: an efficient general purpose program for assigning sequence reads to genomic features. *Bioinformatics* 30:923–930
24. Pertea M et al (2015) StringTie enables improved reconstruction of a transcriptome from RNA-seq reads. *Nat Biotechnol* 33:290–295
25. Anders S, Huber W (2010) Differential expression analysis for sequence count data. *Genome Biol* 11:1–12
26. Robinson MD, McCarthy DJ, Smyth GK (2010) edgeR: a Bioconductor package for differential expression analysis of digital gene expression data. *Bioinformatics* 26:139–140
27. Huang DW, Sherman BT, Lempicki RA (2009) Systematic and integrative analysis of large gene lists using DAVID bioinformatics resources. *Nat Protoc* 4:44–57
28. Grabherr MG et al (2011) Full-length transcriptome assembly from RNA-Seq data without a reference genome. *Nat Biotechnol* 29:644–652
29. Xie Y et al (2014) SOAPdenovo-Trans: de novo transcriptome assembly with short RNA-Seq reads. *Bioinformatics* 30:1660–1666
30. Liu J et al (2016) BinPacker: packing-based de novo transcriptome assembly from RNA-seq data. *PLoS Comput Biol* 12:e1004772
31. Clarke K, Yang Y, Marsh R, Xie L, Zhang KK (2013) Comparative analysis of de novo transcriptome assembly. *Sci China Life Sci* 56:156–162
32. Durai DA, Schulz MH (2016) Informed kmer selection for de novo transcriptome assembly. *Bioinformatics* 32:1670–1677
33. Smith-Unna R, Boursnell C, Patro R, Hibberd JM, Kelly S (2016) TransRate: reference-free quality assessment of de novo transcriptome assemblies. *Genome Res* 26:1134–1144
34. Boetzer M, Henkel CV, Jansen HJ, Butler D, Pirovano W (2011) Scaffolding pre-assembled contigs using SSPACE. *Bioinformatics* 27:578–579
35. Conesa A et al (2005) Blast2GO: a universal tool for annotation, visualization and analysis in functional genomics research. *Bioinformatics* 21:3674–3676