**Authors**: Aigul Sharip, Askhat Molkenov, Ulykbek Kairov

**Topic**: Differential expression analysis and pathway analysis of RNA-Seq data

**Content**

1. Purpose

2. Introduction

2.1 High-performance genomic platforms

3. Comprehensive RNA-Seq transcriptome analysis with DESeq2

4. RNA-Seq workflow: DeSeq2 analysis

4.1 Input data

4.1.1 Airway package

4.2 Starting from *SummarizedExperiment*

4.3 Starting from count matrices

4.4 Exploratory analysis and visualization

4.4.1 Pre-filtering the dataset

4.4.2 The rlog transformation

4.5 Differential expression analysis

4.5.1 Running the differential expression pipeline

4.5.2 Building the results table

4.6 Visualization of the results

4.7 Gene clustering

4.8 Annotating and exporting results

4.9 Functional gene enrichment analysis

4.9.1 KEGG pathways

4.9.2 Gene Ontology (GO)

4.9.3 Reactome pathways

5. References

1. **Purpose**

The purpose of this tutorial is to describe the approach for analysing transcriptome data from high-performance genomic platforms. This guide introduces modern bioinformatics and mathematical methods for analysing transcriptome data from high-performance genomic platforms and RNA-seq data from four human airway smooth muscle cell lines treated with dexamethasone will be analysed as example.

1. **Introduction**

**High-performance genomic platforms**

The decoding of first human genome in 2001 inaugurates the genomic era (Lander, Linton et al. 2001). This remarkable achievement has changed scientists’ approach for solving biological issues related to human by looking human genome sequence and this transforms the practice of medicine (Venter, Adams et al. 2001). The ability to decipher the code of life linked the genomic discovery with many fields such as medicine, molecular and synthetic biology, computational science and informational technology. Over the past years, we have seen NGS technologies being applied in variety of contexts and fields (Auffray, Caulfield et al. 2011).

Genomic science depends heavily on the advancements of DNA sequencing platforms, computational analysis and information technologies. The demand for more affordable and higher throughput genomic sequencing drives the remarkable achievements in DNA sequencing technologies. Thus, DNA sequencing technologies undergone significant transformation since completion of first human genome sequencing. Sanger sequencing technologies have been replaced by next generation sequencing (NGS) for large-scale genomic application (Sanger, Nicklen et al. 1977). The first NGS platform become available in 2004 (Margulies, Egholm et al. 2005). The advent of NGS brought substantial improvements in efficiency and speed of DNA sequencing as well as significant cost reduction, making genome sequencing more affordable for researchers and wide-spread use of NGS technologies for genomic studies. As NGS platforms continue to improve and new platforms become available, the cost and efficiency of DNA sequencing improve as well. For example, Illumina HiSeq X10 sequences the draft of human genome for approximately $1000 in less than 2 weeks in 2015 (Levy and Myers 2016)

Although the development of NGS technologies substantially improve the efficiency and cost of DNA sequencing, it created new challenges for computational assembly and analysis of the genomic data. Sequencing technologies are only capable of sequencing the short fragments up to several hundred base pairs, then resulting short sequence reads should be reconstructed to original genome by mapping to the reference genome using sophisticated computer algorithms (Auffray, Caulfield et al. 2011).

In fact, the analysis of sequencing data rather than sequencing and generation of data is the biggest challenge for genomic studies, because the analysis is often complex requiring multiple steps and programs. Even through a large number of tools have been developed and most of them are readily accessible to scientists, selecting the appropriate tools and being able to use them effectively for specific application can be difficult for untrained and inexperienced users (Pop and Salzberg 2008). Therefore, the competent bioinformatics resources and training is necessary for analysing NGS sequencing data.

Sequencing technologies have potential for making breakthrough in health-care system, especially for preventive, predictive and personalized medicine. With advancement of genomic and computational technologies, our understanding of molecular mechanisms of diseases greatly enhance our ability in discovering and developing better and more precise diagnostics and treatment for diseases. The knowledge of individual’s genome and other multi-omics data will facilitate the development of the most effective treatment options.

1. **Comprehensive transcriptome analysis with RNA-Seq**

Transcriptomics is the study of the complete set of transcripts in a cell that are produced by the genome. Understanding the transcriptome is crucial for interpretation of functional elements of genome, uncovering the molecular components of cells and molecular mechanisms of development and diseases (Zhou, Ren et al. 2010). Studying transcriptome provides a more comprehensive understanding of the complexity of eukaryotic transcriptomes, identification of all transcripts in the context of cell types for a species and determination the transcriptional structure of genes. Also, it allows for the precise quantification of exon expression and splicing variants (Cloonan, Forrest et al. 2008, Nagalakshmi, Waern et al. 2010) and quantify the expression levels of each transcript during development or under different physiological and pathological conditions (Nagalakshmi, Waern et al. 2010).

Transcription is the process when RNA transcripts ae produced by genome. It is the first step of gene expression in which a particular segment of DNA is copied into RNA by the enzyme RNA polymerase. In the process of transcription the two DNA may be labeled as antisense strand that serves for the production of the RNA transcript and the sense strand which includes the DNA version of the transcript sequence. The antisense strand is identical to the sense strand with the exception that thymines (T) are replaced with uraciles (U) in the RNA (Figure 1) (Pineda, Gomez-Rubio et al. 2015).

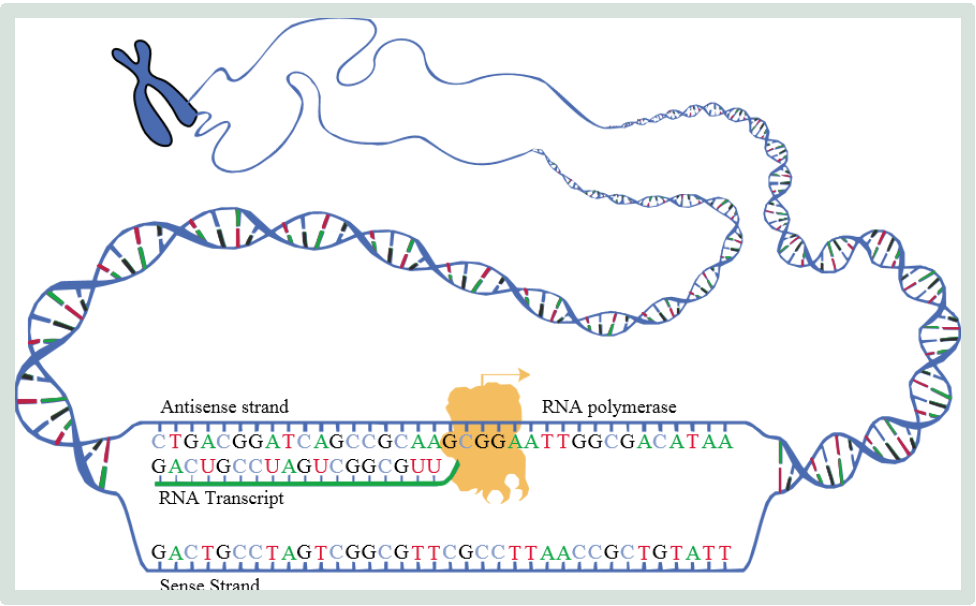


Figure 1. Synthesis of mRNA copied from the DNA base sequences by RNA polymerase

Various technologies have been developed to characterize the transcriptome of a population of cells, including hybridization-based microarrays and Sanger sequencing-based methods (Zhou, Ren et al. 2010). The advent of high-throughput sequencing-based methods has changed the way in which transcriptomes are studied. With the availability of more affordable and faster NGS platforms, more transcriptomics analysis are performed using deep sequencing approach, RNA-Seq. Studies using RNA-Seq technologies already have contributed on our understanding of the complexity of eukaryotic transcriptomes (Zhou, Ren et al. 2010)

Sequencing of RNA, or RNA-Seq, is now common approach for analysing the gene expression and identifying novel RNA species. Compared to DNA microarray method, RNA-Seq offers less background noise and greater detection opportunity (Zhou, Ren et al. 2010). Moreover, RNA-Seq directly indicated the sequence identify, which is crucial for analysis of unknown genes and novel transcripts discovery. Since its first inception in 2008, RNA-Seq has become a widely used and indispensable tool for studying the gene expression (Mortazavi, Williams et al. 2008).

RNA-Seq involves direct sequencing of complementary DNAs (cDNAs) using high-throughput DNA sequencing technologies followed by the mapping of the sequencing reads to the genome. The reads obtained from this can then be aligned to a reference genome in order to construct a whole- genome transcriptome map (Nagalakshmi, Waern et al. 2010).

The actual sequencing step is generally the same for RNA and DNA sequencing, but library preparation and the analysis are quite different. RNA-seq library preparation includes reverse transcription. Data analysis of RNA-seq data may include transcript assembly, novel transcript discovery and transcript quantification. Before starting RNA-seq experiments, sequencing platform should be chosen (Chu and Corey 2012). The data obtained from different sequencing platforms vary and this variation can effect on the interpretation of experiment. Therefore, appropriate protocols for sample preparation and sequencing at right platforms is the premise for productive experiment. As sequencing technologies continue to advance, several different technologies and methods have been developed for RNA-Seq and expected to emerge in the future (Pepke, Wold et al. 2009).

Dealing with the large volume of RNA-seq data generated during experiments is time-consuming and challenging. For example, Hiseq2000 (Illumina) can produce up to 200 million 100-nt reads (approximately 50 GB) of data in one lane in one sequencer run. These data must be processed to not only identify matches to the transcriptome, but also for assembly into transcripts and quantitated before insights can be made into biological meaning. Duplicate or triplicate experimental datasets improve data variability and facilitate optimal interpretation of data (Chu and Corey 2012).

1. **RNA-Seq workflow: DeSeq2 package**

A basic task in the analysis of count data from RNA-seq is the detection of differentially expressed genes. The count data are presented as a table which reports, for each sample, the number of sequence fragments that have been assigned to each gene. Analogous data also arise for other assay types, including comparative ChIP-Seq, HiC, shRNA screening, mass spectrometry. An important analysis question is the quantitation and statistical inference of systematic changes between conditions, as compared to within condition variability.

The package DESeq2 provides methods to test for differential expression by use of negative binomial generalized linear models; the estimates of dispersion and logarithmic fold changes incorporate data-driven prior distributions. This tutorial have been adapted from original package author’s manual (Love, Anders et al. 2015) and explains the use of the DeSeq2 package and demonstrates typical workflows.

**3.1 Input data**

As input, the DESeq2 package expects count data as obtained, e. g., from RNA-seq or another high-throughput sequencing experiment, in the form of a matrix of integer values. The value in the i-th row and the j-th column of the matrix tells how many reads have been mapped to gene i in sample j. Analogously, for other types of assays, the rows of the matrix might correspond e. g. to binding regions (with ChIP-Seq) or peptide sequences.

The count values must be raw counts of sequencing reads (for single-end RNA-seq) or fragments (for paired-end RNA-seq). The RNA-seq work how describes multiple techniques for preparing such count matrices. It is important to provide count matrices as input for DESeq2's statistical model (Love, Huber et al. 2014) to hold, as only the count values allow assessing the measurement precision correctly. The DESeq2 model internally corrects for library size, so transformed values such as counts scaled by library size should never be used as input.

There are several methods for obtaining count matrices:

1. *SummarizedExperiemnt*
2. Count matrices
   * 1. **Airway package**

This package provides a *RangedSummarizedExperiment* object of read counts in genes for an RNA-Seq experiment on four human airway smooth muscle cell lines treated with dexamethasone. Details on the gene model and read counting procedure are provided in the package vignette.

The data used in this workflow is stored in the airway package that summarizes an RNA-seq experiment wherein airway smooth muscle cells were treated with dexamethasone, a synthetic glucocorticoid steroid with anti-inflammatory effects. Glucocorticoids are used, for example, by people with asthma to reduce inflammation of the airways. In the experiment, four primary human airway smooth muscle cell lines were treated with 1 micromolar dexamethasone for 18 hours. For each of the four cell lines, we have a treated and an untreated sample.

**4.2 Starting from SummarizedExperiment**

The class used by the DESeq2 package to store the read counts is *DESeqDataSet* which extends the *RangedSummarizedExperiment* class of the *SummarizedExperiment* package. This facilitates preparation steps and also downstream exploration of results.

The *SummarizedExperiment* container is diagrammed in Figure 2 and discussed in the latest Bioconductor paper. In our case we have created a single matrix named “counts” that contains the fragment counts for each gene and sample, which is stored in assay. It is also possible to store multiple matrices, accessed with assays. The component parts of the *SummarizedExperiment* are accessed with an R function of the same name: assay (or assays), *rowRanges* and *colData*.

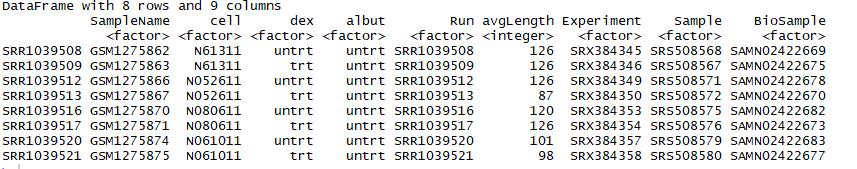
This example code above actually only counted a small subset of fragments from the original experiment. Nevertheless, we can still investigate the resulting *SummarizedExperiment* by looking at the counts in the assay slot, the phenotypic data about the samples in *colData* slot, and the data about the genes in the *rowRanges* slot. Other methods for obtaining count matrices are described in the next section.



Figure 2. The component parts of a *SummarizedExperiment* object**.** The assay (pink block) contains the matrix of counts, the *rowRanges* (blue block) contains information about the genomic ranges and the *colData* (green block) contains information about the samples. The highlighted line in each block represents the first row (note that the first row of *colData* lines up with the first column of the assay).

We now use R’s *data* command to load a prepared *SummarizedExperiment* that was generated from the publicly available sequencing data files. The steps we used to produce this object were equivalent to those you worked through in the previous sections, except that we used all the reads and all the genes. For more details on the exact steps used to create this object, type vignette(“airway”) into your R session.

|  |
| --- |
| > library("airway")  > data ("airway")  > airway  class: RangedSummarizedExperiment  dim: 64102 8  metadata(1): ''  assays(1): counts  rownames(64102): ENSG00000000003 ENSG00000000005 ... LRG\_98 LRG\_99  rowData names(0):  colnames(8): SRR1039508 SRR1039509 ... SRR1039520 SRR1039521  colData names(9): SampleName cell ... Sample BioSample  > airway\_se<- airway  > colData(airway\_se) |



Once we have our fully annotated *SummarizedExperiment* object, we can construct a *DESeqDataSet* object from it that will then form the starting point of the analysis. We add an appropriate design for the analysis:

> library("DESeq2")

> dds <- DESeqDataSet(airway\_se, design = ~ cell + dex)

> dds

class: DESeqDataSet

dim: 64102 8

metadata(2): '' version

assays(1): counts

rownames(64102): ENSG00000000003 ENSG00000000005 ... LRG\_98 LRG\_99

rowData names(0):

colnames(8): SRR1039508 SRR1039509 ... SRR1039520 SRR1039521

colData names(9): SampleName cell ... Sample BioSample

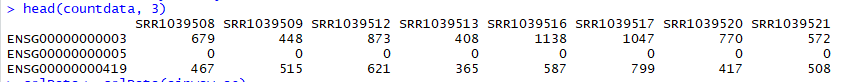
* 1. **Starting from count matrices**

In this section, we will show how to build an *DESeqDataSet* supposing we only hace a count matrix and a table of sample information.

The function *DESeqDataSetFromMatrix* can be used if you already have a matrix of read counts prepared from another source. Another method for quickly producing count matrices from alignment files is the featureCounts function in the Rsubread package. To use DESeqDataSetFromMatrix, the user should provide the counts matrix, the information about the samples (the columns of the count matrix) as a DataFrame or data.frame, and the design formula.

> countdata <- assay(airway\_se)

> head(countdata, 3)



|  |
| --- |
| > coldata<- colData(airway\_se)  > coldata |
|  |
| We now have all the ingredients to prepare our data object in a form that is suitable for analysis, namely:  • countdata: a table with the fragment counts  • coldata: a table with information about the samples |

To now construct the DESeqDataSet object from the matrix of counts and the sample information table, we use:

> ddsMat <- DESeqDataSetFromMatrix(countData = countdata,

+ colData = coldata,

+ design = ~ cell + dex)

> ddsMat

class: DESeqDataSet

dim: 64102 8

metadata(1): version

assays(1): counts

rownames(64102): ENSG00000000003 ENSG00000000005 ... LRG\_98 LRG\_99

rowData names(0):

colnames(8): SRR1039508 SRR1039509 ... SRR1039520 SRR1039521

colData names(9): SampleName cell ... Sample BioSample

**4.4 Exploratory analysis and visualization**

There are two separate paths in this workflow; the one we will see first involves transformations of the counts in order to visually explore sample relationships. In the second part, we will go back to the original raw counts for statistical testing. This is critical because the statistical testing methods rely on original count data (not scaled or transformed) for calculating the precision of measurements.

**4.4.1 Pre-filtering the dataset**

Our count matrix with our DESeqDataSet contains many rows with only zeros, and additionally many rows with only a few fragments total. In order to reduce the size of the object, and to increase the speed of our functions, we can remove the rows that have no or nearly no information about the amount of gene expression. Here we remove rows of the DESeqDataSet that have no counts, or only a single count across all samples:

> nrow(ddsMat)

[1] 64102

> ddsMat <- ddsMat[ rowSums(counts(ddsMat)) > 1, ]

> nrow(ddsMat)

[1] 29391

**4.4.2 The rlog transformation**

Many common statistical methods for exploratory analysis of multidimensional data, for example clustering and principal components analysis (PCA), work best for data that generally has the same range of variance at different ranges of the mean values. When the expected amount of variance is approximately the same across different mean values, the data is said to be homoscedastic. For RNA-seq raw counts, however, the variance grows with the mean. For example, if one performs PCA directly on a matrix of size-factor-normalized read counts, the result typically depends only on the few most strongly expressed genes because they show the largest absolute differences between samples. A simple and often used strategy to avoid this is to take the logarithm of the normalized count values plus a small pseudocount; however, now the genes with the very lowest counts will tend to dominate the results because, due to the strong Poisson noise inherent to small count values, and the fact that the logarithm amplifies differences for the smallest values, these low count genes will show the strongest relative differences between samples.

As a solution, DESeq2 offers transformations for count data that stabilize the variance across the mean. One such transformation is the regularized-logarithm transformation or rlog2. For genes with high counts, the *rlog* transformation will give similar result to the ordinary log2 transformation of normalized counts. For genes with lower counts, however, the values are shrunken towards the genes’ averages across all samples. Using an empirical Bayesian prior on inter-sample differences in the form of a ridge penalty, the *rlog*-transformed data then becomes approximately homoskedastic, and can be used directly for computing distances between samples and making PCA plots.

The function *rlog* returns a *SummarizedExperiment* object that contains the rlog-transformed values in its assay slot.

|  |
| --- |
| > rld <- rlog(dds)  > head(assay(rld), 3)  SRR1039508 SRR1039509 SRR1039512 SRR1039513 SRR1039516 SRR1039517 SRR1039520 SRR1039521  ENSG00000000003 9.399155 9.142506 9.501691 9.320807 9.757189 9.512179 9.617365 9.315321  ENSG00000000005 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000  ENSG00000000419 8.901297 9.113966 9.032566 9.063920 8.981935 9.108522 8.894845 9.052299 |
|  |

**4.5 Differential expression analysis**

The standard differential expression analysis steps are wrapped into a single function, *DESeq*. The estimation steps performed by this function are described in in the manual page for ?DESeq and in the Methods section of the DESeq2 publication (Love et.al 2014).

**4.5.1 Running the differential expression pipeline**

As we have already specified an experimental design when we created the DESeqDataSet, we can run the differential expression pipeline on the raw counts with a single call to the function DESeq:

> dds <- DESeq(ddsMat)

estimating size factors

estimating dispersions

gene-wise dispersion estimates

mean-dispersion relationship

final dispersion estimates

fitting model and testing

This function will print out a message for the various steps it performs. Briefly these are: the estimation of size factors (controlling for differences in the sequencing depth of the samples), the estimation of dispersion values for each gene, and fitting a generalized linear model. A *DESeqDataSet* is returned that contains all the fitted parameters within it, and the following section describes how to extract out results tables of interest from this object.

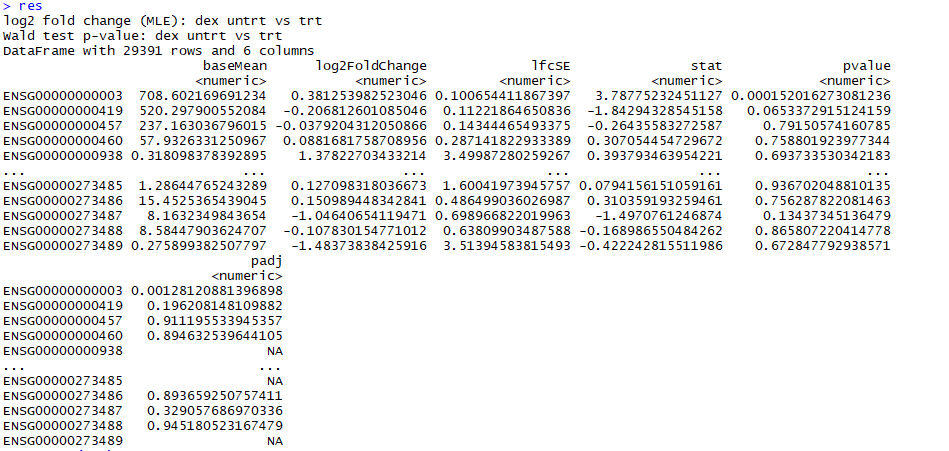
**4.5.2 Building the results table**

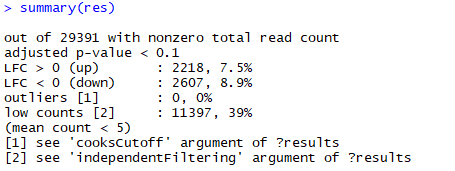
Results tables are generated using the function *results*, which extracts a results table with log2 fold changes, p values and adjusted p values. If there are more than 2 levels for this variable and no arguments are specified to results, the results will be done for the last variable in the design formula, and the comparison will be the last level of this variable over the first level. The text tells you that the estimates are of the logarithmic fold change between two conditions.

Calling results without any arguments will extract the estimated log2 fold changes and p values for the last variable in the design formula. If there are more than 2 levels for this variable, results will extract the results table for a comparison of the last level over the first level. This comparison is printed at the top of the output: dex trt vs untr.

> res<- results(dds)

> res





There are many genes with differential expression due to dexamethasone treatment at the false discovery rate (FDR) level of 10%. This makes sense, as the smooth muscle cells of the airway are known to react to glucocorticoid steroids. However, there are two ways to be more stricter about which set of genes are considered significant:

• lower the false discovery rate threshold (the threshold on padj in the results table)

• raise the log2 fold change threshold from 0 using the *lfcThreshold* argument of results

If we lower the false discovery rate threshold, we should also tell this value to results (), so that the function will use an alternative threshold for the optimal independent filtering step:

> res.05 <- results(dds, alpha=.05)

> table(res.05$padj < .05)

FALSE TRUE

12829 4026

If we want to raise the log2 fold change threshold, so that we test for genes that show more substantial changes due to treatment, we simply supply a value on the log2 scale. For example, by specifying *lfcThreshold*=1, we test for genes that show significant effects of treatment on gene counts more than doubling or less than halving, because 21 = 2.

> resLFC1 <- results(dds, lfcThreshold=1)

> table(resLFC1$padj < 0.1)

FALSE TRUE

20034 240

* 1. **Plotting the results**

A quick way to visualize the counts for a particular gene is to use the *plotCounts* function that takes as arguments the *DESeqDataSet*, a gene name, and the group over which to plot the counts(Figure 3).

> topGene <- rownames(res)[which.min(res$padj)]

> plotCounts(dds, gene=topGene, intgroup=c("dex"))

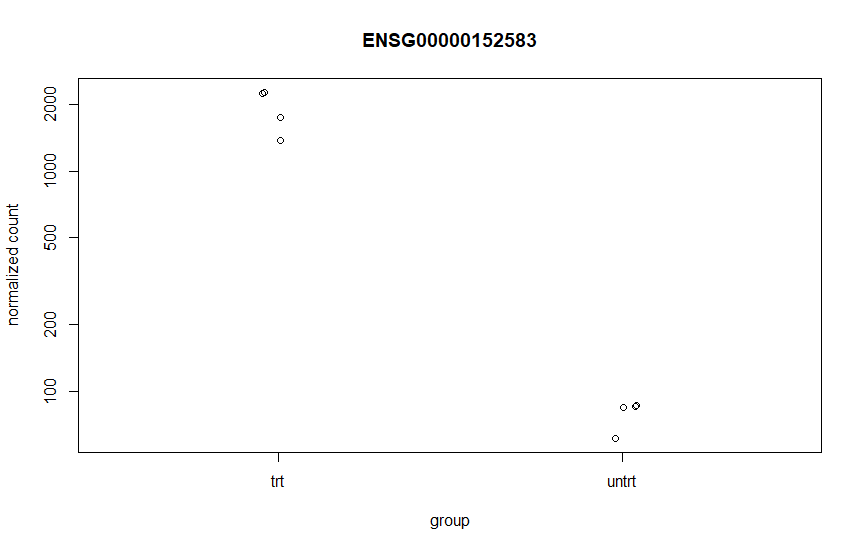


Figure 3. Normalized counts for a single gene over treatment group.

An *MA-plot* provides a useful overview for an experiment with a two-group comparison (Figure 4). The log2 fold change for a particular comparison is plotted on the y-axis and the average of the counts normalized by size factor is shown on the x-axis.

> plotMA(res, ylim=c(-2,2), main="DESeq2")

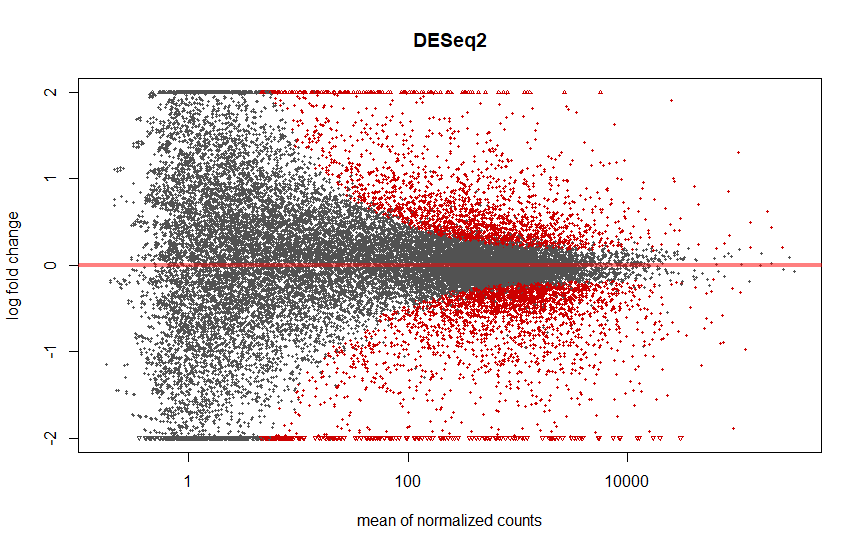


Figure 4. An MA-plot of changes induced by treatment. The log2 fold change for a particular comparison is plotted on the y-axis and the average of the counts normalized by size factor is shown on the x-axis (“M” for minus, because a log ratio is equal to log minus log, and “A” for average). Each gene is represented with a dot. Genes with an adjusted p value below a threshold (here 0.1, the default) are shown in red.

**4.7 Gene clustering**

Gene clustering is only relevant for genes that actually carry a signal, one usually would only cluster a subset of the most highly variable genes. Here, for demonstration, we will select the 100 genes with the highest variance across samples. We will work with the *rlog* transformed counts

|  |
| --- |
| > library(pheatmap)  > topVarGenes <- head(order(rowVars(assay(rld)),decreasing=TRUE),20)  > mat <- assay(rld)[ topVarGenes, ]  > mat <- mat - rowMeans(mat)  > df <- as.data.frame(colData(rld)[,c("cell","dex")])  > pheatmap(mat, annotation\_col=df) |

The heatmap becomes more interesting if we do not look at absolute expression strength but rather at the amount by which each gene deviates in a specific sample from the gene’s average across all samples. Hence, we center each genes’ values across samples, and plot a heatmap (Figure 5). We provide a data.frame that instructs the pheatmap function how to label the columns.

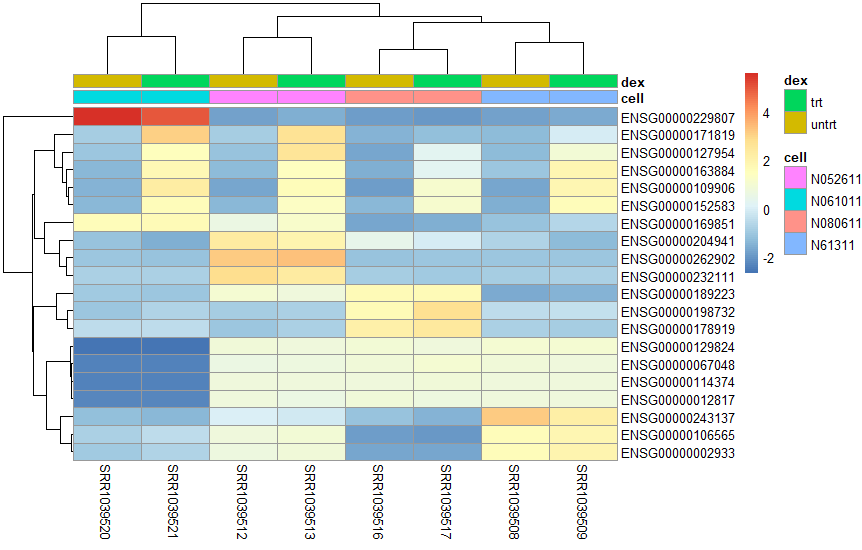


Figure 5. Heatmap of relative rlog-transformed values across samples. Treatment status and cell line information are shown with colored bars at the top of the heatmap. Note that a set of genes at the top of the heatmap are separating the N061011 cell line from the others. In the center of the heatmap, we see a set of genes for which the dexamethasone treated samples have higher gene expression.

**4.8 Annotating and exporting results**

Our result table so far only contains information about Ensembl gene IDs, but alternative gene names may be more informative for further analysis. *Bioconductor’s* annotation packages help with mapping various ID schemes to each other. We load the *AnnotationDbi* package and the annotation package *org.Hs.eg.db*:

> library("AnnotationDbi")

> library("org.Hs.eg.db")

This is the organism annotation package (“org”) for *Homo sapiens* (“Hs”), organized as an *AnnotationDbi* database package (“db”), using Entrez Gene IDs (“eg”) as primary key. To get a list of all available key types, use:

> columns(org.Hs.eg.db)

## [1] "ENTREZID" "PFAM" "IPI" "PROSITE" "ACCNUM" "ALIAS"

## [7] "CHR" "CHRLOC" "CHRLOCEND" "ENZYME" "MAP" "PATH"

## [13] "PMID" "REFSEQ" "SYMBOL" "UNIGENE" "ENSEMBL" "ENSEMBLPROT"

## [19] "ENSEMBLTRANS" "GENENAME" "UNIPROT" "GO" "EVIDENCE" "ONTOLOGY"

## [25] "GOALL" "EVIDENCEALL" "ONTOLOGYALL" "OMIM" "UCSCKG"

We can use the *mapIds* function to add individual columns to our results table. We provide the row names of our results table as a key, and specify that *keytype*=ENSEMBL. The column argument tells the *mapIds* function which information we want, and the *multiVals* argument tells the function what to do if there are multiple possible values for a single input value. Here we ask to just give us back the first one that occurs in the database. To add the gene symbol and Entrez ID, we call *mapIds* twice.

> res$symbol <- mapIds(org.Hs.eg.db,

+ keys=row.names(res),

+ column="SYMBOL",

+ keytype="ENSEMBL",

+ multiVals="first")

> res$entrez <- mapIds(org.Hs.eg.db,

+ keys=row.names(res),

+ column="ENTREZID",

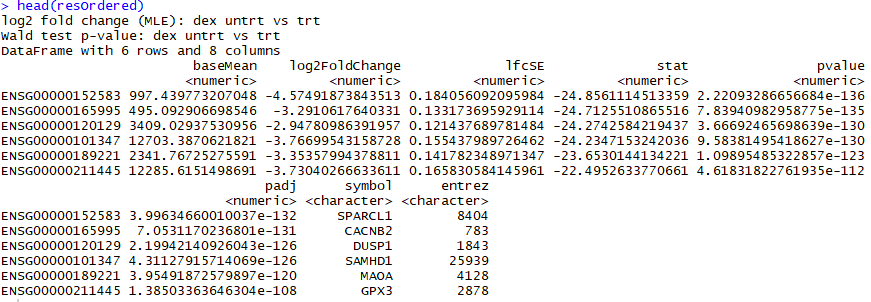
+ keytype="ENSEMBL",

+ multiVals="first")

Now the results have the desired external gene IDs. We can order genes according to their adjusted p value:

> resOrdered <- res[order(res$padj),]

> head(resOrdered)



**Exporting results**

You can easily save the results table in a CSV file, that you can then share or load with a spreadsheet program such as Excel. The call to *as.data.frame* is necessary to convert the *DataFrame* object (*IRanges* package) to a *data.frame* object that can be processed by *write.csv*.

> write.csv(as.data.frame(resOrdered),

+ file=" results.csv")

**4.9 Gene enrichment analysis**

**4.9. 1 KEGG pathways**

In this part, we are going to show KEGG pathway analysis using GAGE package (Luo, Friedman et al. 2009). The *gage* package implements the GAGE method. GAGE is generally applicable independent of microarray and RNA-Seq data attributes including sample sizes, experimental designs, assay platforms, and other types of heterogeneity, and consistently achieves superior performance over other frequently used methods. We introduce functions and data for routine and advanced gene set (enrichment) analysis, as well as results presentation and interpretation.

The gageData package has pre-compiled databases mapping genes to KEGG pathways and GO terms for common organisms. kegg.sets.hs is a named list of 229 elements. Each element is a character vector of member gene Entrez IDs for a single KEGG pathway. sigmet.idx.hs is an index of numbers of signalling and metabolic pathways in kegg.set.gs. In other words, KEGG pathway include other types of pathway definitions, which may be undesirable in pathway analysis. Therefore, kegg.sets.hs[sigmet.idx.hs] gives you the “cleaner” gene sets of signalling and metabolic pathways only.

> library(dplyr)

> library(pathview)

> library(gage)

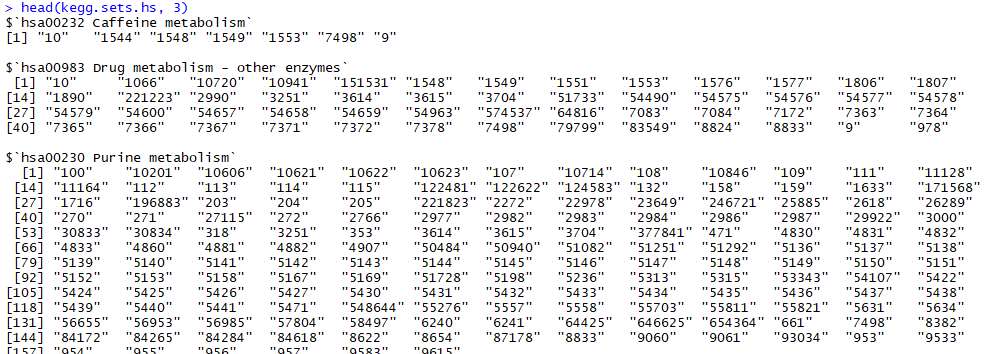
> library(gageData)

> data(kegg.sets.hs)

> data(sigmet.idx.hs)

> kegg.sets.hs <- kegg.sets.hs[sigmet.idx.hs]

> head(kegg.sets.hs, 3)



The gage() function requires a named vector of fold changes, where the names of the values are the Entrez

gene IDs.

> foldchanges <- res$log2FoldChange

> names(foldchanges) <- res$entrez

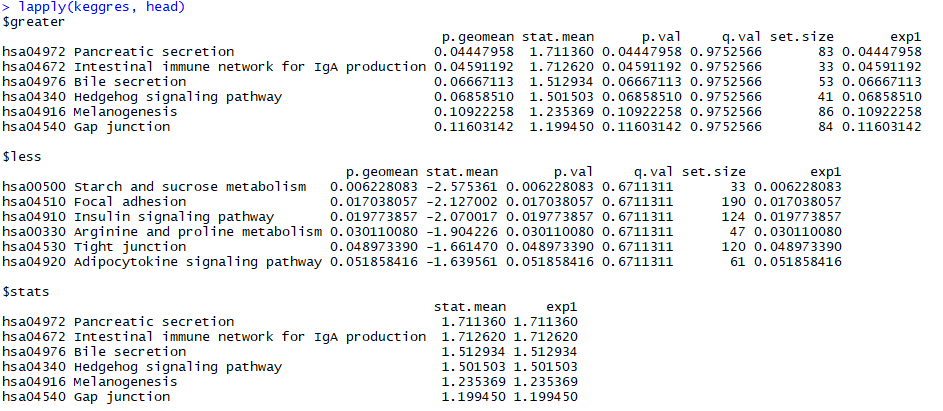
> head(foldchanges)



Next, we will display the results and show upregulated and downregulated pathways with statistics.

> keggres <- gage(foldchanges, gsets=kegg.sets.hs, same.dir=TRUE)

> lapply(keggres, head)



Now, let’s process the results to pull out the top 5 upregulated pathways, then further process that just to get the IDs. We’ll use these KEGG pathway IDs downstream for plotting.

> keggrespathways <- data.frame(id=rownames(keggres$greater), keggres$greater) %>%

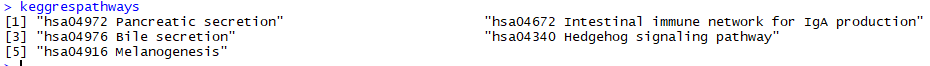
+ tbl\_df() %>%

+ filter(row\_number()<=5) %>%

+ .$id %>%

+ as.character()

> keggrespathways



> # Get the IDs.

> keggresids <- substr(keggrespathways, start=1, stop=8)

> keggresids

[1] "hsa00190" "hsa04260" "hsa03010" "hsa04640" "hsa04672"

> # Define plotting function for applying later

> plot\_pathway <- function(pid) pathview(gene.data=foldchanges, pathway.id=pid, species="hsa", new.signature=FALSE)

> # plot multiple pathways (plots saved to disk and returns a throwaway list object)

> tmp <- sapply(keggresids, function(pid) pathview(gene.data=foldchanges, pathway.id=pid, species="hsa"))

Once we have a list of enriched pathways, we’re going to use the *pathview* package to draw pathway diagrams, shading the molecules in the pathway by their degree of up/down-regulation. the pathview() function in the pathview package makes the plots (Luo and Brouwer 2013). Let’s write a function so we can loop through and draw plots for the top 5 pathways we created above (Figure 6).

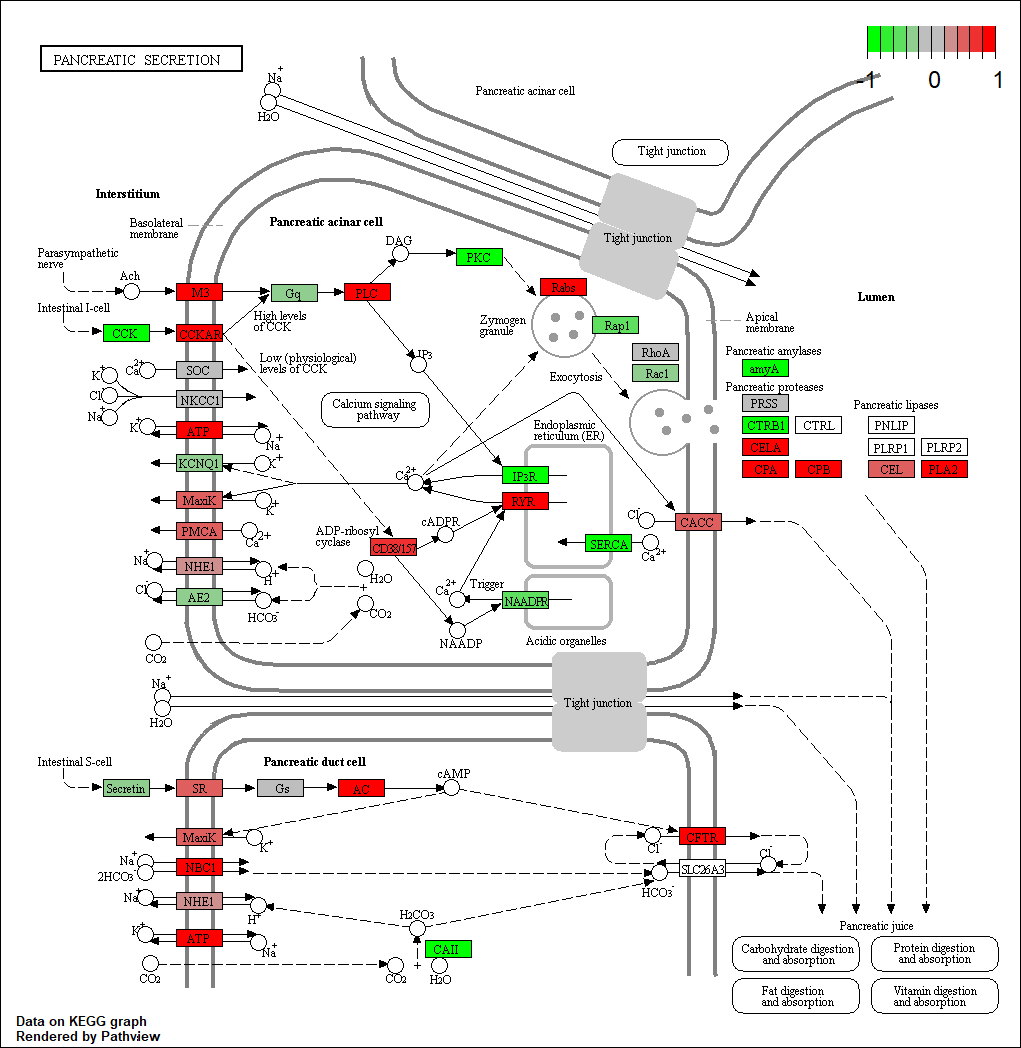


Figure 6. Visualization of KEGG view of Pancreatic section resulting using *pathview* package

**4.9.2 Gene Ontology (GO)**

We can also do a similar procedure with gene ontology. Similar to above, *go.sets.hs* has all GO terms. *go.subs.hs* is a named list containing indexes for the BP, CC, and MF ontologies. Let’s only do Biological Process.

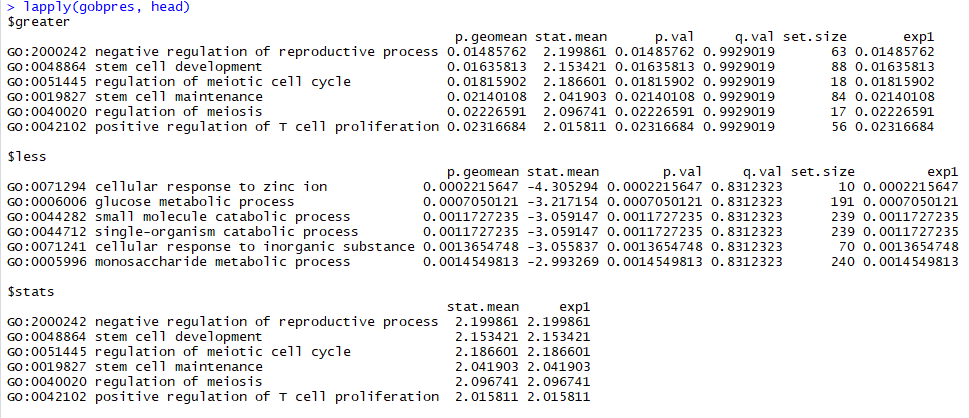
> data(go.sets.hs)

> data(go.subs.hs)

> gobpsets <- go.sets.hs[go.subs.hs$BP]

> gobpres <- gage(foldchanges, gsets=gobpsets, same.dir=TRUE)

> lapply(gobpres, head)



**4.9.3 ReactomePA analysis**

This package is designed for reactome pathway-based analysis (Yu and He 2016). Reactome is an open-source, open access, manually curated and peer-reviewed pathway database. Enrichment analysis is a widely used approach to identify biological themes. Here, we implement hypergeometric model to assess whether the number of selected genes associated with reactome pathway is larger than expected. The p values were calculated based the hypergeometric model. In order to consider the potentially biological complexities in which a gene may belong to multiple annotation categories, we developed cnetplot function to extract the complex association between genes and diseases.

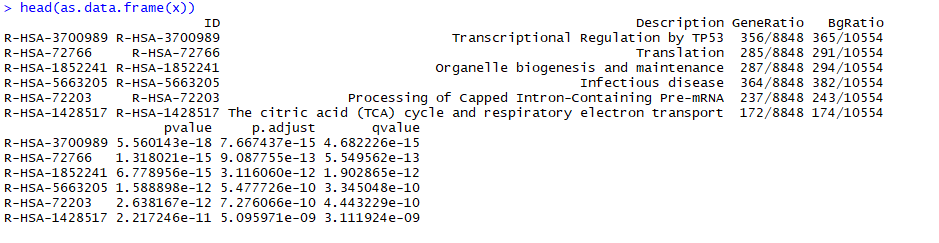
> library(ReactomePA)

> geneList<- res$entrez

> head(geneList)



> x<-enrichPathway(gene=geneList, pvalueCutoff = 0.05, readable = TRUE)



We implement barplot, dotplot enrichment map and category-gene-network for visualization. It is very common to visualize the enrichment result in bar or pie chart.

> barplot(x, showCategory = 8)

> dotplot(x, showCategory = 8)

Enrichment map can be visualized by enrichMap:

> emapplot(x)

In order to consider the potentially biological complexities in which a gene may belong to multiple annotation categories, we developed *cnetplot* function to extract the complex association between genes and diseases.

> cnetplot(x, categorySize="pvalue", foldChange = geneList)

|  |  |
| --- | --- |
| |  | | --- | |  | |

Figure 7. Visualization of of Reactome pathways using *ReactomePA* package. (A) Barplot. (B) Dotplot. (C) Emaplot

1. **References**

Auffray, C., T. Caulfield, M. J. Khoury, J. R. Lupski, M. Schwab and T. Veenstra (2011). "Genome Medicine: past, present and future." Genome medicine **3**(1): 6-6.

Chu, Y. and D. R. Corey (2012). "RNA sequencing: platform selection, experimental design, and data interpretation." Nucleic Acid Ther **22**(4): 271-274.

Cloonan, N., A. R. Forrest, G. Kolle, B. B. Gardiner, G. J. Faulkner, M. K. Brown, D. F. Taylor, A. L. Steptoe, S. Wani, G. Bethel, A. J. Robertson, A. C. Perkins, S. J. Bruce, C. C. Lee, S. S. Ranade, H. E. Peckham, J. M. Manning, K. J. McKernan and S. M. Grimmond (2008). "Stem cell transcriptome profiling via massive-scale mRNA sequencing." Nat Methods **5**(7): 613-619.

Lander, E. S., L. M. Linton, B. Birren, C. Nusbaum, M. C. Zody, J. Baldwin, K. Devon, K. Dewar, M. Doyle, W. FitzHugh, R. Funke, D. Gage, K. Harris, A. Heaford, J. Howland, L. Kann, J. Lehoczky, R. LeVine, P. McEwan, K. McKernan, J. Meldrim, J. P. Mesirov, C. Miranda, W. Morris, J. Naylor, C. Raymond, M. Rosetti, R. Santos, A. Sheridan, C. Sougnez, Y. Stange-Thomann, N. Stojanovic, A. Subramanian, D. Wyman, J. Rogers, J. Sulston, R. Ainscough, S. Beck, D. Bentley, J. Burton, C. Clee, N. Carter, A. Coulson, R. Deadman, P. Deloukas, A. Dunham, I. Dunham, R. Durbin, L. French, D. Grafham, S. Gregory, T. Hubbard, S. Humphray, A. Hunt, M. Jones, C. Lloyd, A. McMurray, L. Matthews, S. Mercer, S. Milne, J. C. Mullikin, A. Mungall, R. Plumb, M. Ross, R. Shownkeen, S. Sims, R. H. Waterston, R. K. Wilson, L. W. Hillier, J. D. McPherson, M. A. Marra, E. R. Mardis, L. A. Fulton, A. T. Chinwalla, K. H. Pepin, W. R. Gish, S. L. Chissoe, M. C. Wendl, K. D. Delehaunty, T. L. Miner, A. Delehaunty, J. B. Kramer, L. L. Cook, R. S. Fulton, D. L. Johnson, P. J. Minx, S. W. Clifton, T. Hawkins, E. Branscomb, P. Predki, P. Richardson, S. Wenning, T. Slezak, N. Doggett, J. F. Cheng, A. Olsen, S. Lucas, C. Elkin, E. Uberbacher, M. Frazier, R. A. Gibbs, D. M. Muzny, S. E. Scherer, J. B. Bouck, E. J. Sodergren, K. C. Worley, C. M. Rives, J. H. Gorrell, M. L. Metzker, S. L. Naylor, R. S. Kucherlapati, D. L. Nelson, G. M. Weinstock, Y. Sakaki, A. Fujiyama, M. Hattori, T. Yada, A. Toyoda, T. Itoh, C. Kawagoe, H. Watanabe, Y. Totoki, T. Taylor, J. Weissenbach, R. Heilig, W. Saurin, F. Artiguenave, P. Brottier, T. Bruls, E. Pelletier, C. Robert, P. Wincker, D. R. Smith, L. Doucette-Stamm, M. Rubenfield, K. Weinstock, H. M. Lee, J. Dubois, A. Rosenthal, M. Platzer, G. Nyakatura, S. Taudien, A. Rump, H. Yang, J. Yu, J. Wang, G. Huang, J. Gu, L. Hood, L. Rowen, A. Madan, S. Qin, R. W. Davis, N. A. Federspiel, A. P. Abola, M. J. Proctor, R. M. Myers, J. Schmutz, M. Dickson, J. Grimwood, D. R. Cox, M. V. Olson, R. Kaul, C. Raymond, N. Shimizu, K. Kawasaki, S. Minoshima, G. A. Evans, M. Athanasiou, R. Schultz, B. A. Roe, F. Chen, H. Pan, J. Ramser, H. Lehrach, R. Reinhardt, W. R. McCombie, M. de la Bastide, N. Dedhia, H. Blocker, K. Hornischer, G. Nordsiek, R. Agarwala, L. Aravind, J. A. Bailey, A. Bateman, S. Batzoglou, E. Birney, P. Bork, D. G. Brown, C. B. Burge, L. Cerutti, H. C. Chen, D. Church, M. Clamp, R. R. Copley, T. Doerks, S. R. Eddy, E. E. Eichler, T. S. Furey, J. Galagan, J. G. Gilbert, C. Harmon, Y. Hayashizaki, D. Haussler, H. Hermjakob, K. Hokamp, W. Jang, L. S. Johnson, T. A. Jones, S. Kasif, A. Kaspryzk, S. Kennedy, W. J. Kent, P. Kitts, E. V. Koonin, I. Korf, D. Kulp, D. Lancet, T. M. Lowe, A. McLysaght, T. Mikkelsen, J. V. Moran, N. Mulder, V. J. Pollara, C. P. Ponting, G. Schuler, J. Schultz, G. Slater, A. F. Smit, E. Stupka, J. Szustakowki, D. Thierry-Mieg, J. Thierry-Mieg, L. Wagner, J. Wallis, R. Wheeler, A. Williams, Y. I. Wolf, K. H. Wolfe, S. P. Yang, R. F. Yeh, F. Collins, M. S. Guyer, J. Peterson, A. Felsenfeld, K. A. Wetterstrand, A. Patrinos, M. J. Morgan, P. de Jong, J. J. Catanese, K. Osoegawa, H. Shizuya, S. Choi, Y. J. Chen, J. Szustakowki and C. International Human Genome Sequencing (2001). "Initial sequencing and analysis of the human genome." Nature **409**(6822): 860-921.

Levy, S. E. and R. M. Myers (2016). "Advancements in Next-Generation Sequencing." Annu Rev Genomics Hum Genet **17**: 95-115.

Love, M. I., S. Anders, V. Kim and W. Huber (2015). "RNA-Seq workflow: gene-level exploratory analysis and differential expression." F1000Research **4**: 1070-1070.

Love, M. I., W. Huber and S. Anders (2014). "Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2." Genome Biol **15**(12): 550.

Luo, W. and C. Brouwer (2013). "Pathview: an R/Bioconductor package for pathway-based data integration and visualization." Bioinformatics **29**(14): 1830-1831.

Luo, W., M. S. Friedman, K. Shedden, K. D. Hankenson and P. J. Woolf (2009). "GAGE: generally applicable gene set enrichment for pathway analysis." BMC Bioinformatics **10**: 161.

Margulies, M., M. Egholm, W. E. Altman, S. Attiya, J. S. Bader, L. A. Bemben, J. Berka, M. S. Braverman, Y. J. Chen, Z. Chen, S. B. Dewell, L. Du, J. M. Fierro, X. V. Gomes, B. C. Godwin, W. He, S. Helgesen, C. H. Ho, G. P. Irzyk, S. C. Jando, M. L. Alenquer, T. P. Jarvie, K. B. Jirage, J. B. Kim, J. R. Knight, J. R. Lanza, J. H. Leamon, S. M. Lefkowitz, M. Lei, J. Li, K. L. Lohman, H. Lu, V. B. Makhijani, K. E. McDade, M. P. McKenna, E. W. Myers, E. Nickerson, J. R. Nobile, R. Plant, B. P. Puc, M. T. Ronan, G. T. Roth, G. J. Sarkis, J. F. Simons, J. W. Simpson, M. Srinivasan, K. R. Tartaro, A. Tomasz, K. A. Vogt, G. A. Volkmer, S. H. Wang, Y. Wang, M. P. Weiner, P. Yu, R. F. Begley and J. M. Rothberg (2005). "Genome sequencing in microfabricated high-density picolitre reactors." Nature **437**(7057): 376-380.

Mortazavi, A., B. A. Williams, K. McCue, L. Schaeffer and B. Wold (2008). "Mapping and quantifying mammalian transcriptomes by RNA-Seq." Nat Methods **5**(7): 621-628.

Nagalakshmi, U., K. Waern and M. Snyder (2010). "RNA-Seq: a method for comprehensive transcriptome analysis." Curr Protoc Mol Biol **Chapter 4**: Unit 4 11 11-13.

Pepke, S., B. Wold and A. Mortazavi (2009). "Computation for ChIP-seq and RNA-seq studies." Nat Methods **6**(11 Suppl): S22-32.

Pineda, S., P. Gomez-Rubio, A. Picornell, K. Bessonov, M. Marquez, M. Kogevinas, F. X. Real, K. Van Steen and N. Malats (2015). "Framework for the Integration of Genomics, Epigenomics and Transcriptomics in Complex Diseases." Human Heredity **79**(3-4): 124-136.

Pop, M. and S. L. Salzberg (2008). "Bioinformatics challenges of new sequencing technology." Trends Genet **24**(3): 142-149.

Sanger, F., S. Nicklen and A. R. Coulson (1977). "DNA sequencing with chain-terminating inhibitors." Proc Natl Acad Sci U S A **74**(12): 5463-5467.

Venter, J. C., M. D. Adams, E. W. Myers, P. W. Li, R. J. Mural, G. G. Sutton, H. O. Smith, M. Yandell, C. A. Evans, R. A. Holt, J. D. Gocayne, P. Amanatides, R. M. Ballew, D. H. Huson, J. R. Wortman, Q. Zhang, C. D. Kodira, X. H. Zheng, L. Chen, M. Skupski, G. Subramanian, P. D. Thomas, J. Zhang, G. L. Gabor Miklos, C. Nelson, S. Broder, A. G. Clark, J. Nadeau, V. A. McKusick, N. Zinder, A. J. Levine, R. J. Roberts, M. Simon, C. Slayman, M. Hunkapiller, R. Bolanos, A. Delcher, I. Dew, D. Fasulo, M. Flanigan, L. Florea, A. Halpern, S. Hannenhalli, S. Kravitz, S. Levy, C. Mobarry, K. Reinert, K. Remington, J. Abu-Threideh, E. Beasley, K. Biddick, V. Bonazzi, R. Brandon, M. Cargill, I. Chandramouliswaran, R. Charlab, K. Chaturvedi, Z. Deng, V. Di Francesco, P. Dunn, K. Eilbeck, C. Evangelista, A. E. Gabrielian, W. Gan, W. Ge, F. Gong, Z. Gu, P. Guan, T. J. Heiman, M. E. Higgins, R. R. Ji, Z. Ke, K. A. Ketchum, Z. Lai, Y. Lei, Z. Li, J. Li, Y. Liang, X. Lin, F. Lu, G. V. Merkulov, N. Milshina, H. M. Moore, A. K. Naik, V. A. Narayan, B. Neelam, D. Nusskern, D. B. Rusch, S. Salzberg, W. Shao, B. Shue, J. Sun, Z. Wang, A. Wang, X. Wang, J. Wang, M. Wei, R. Wides, C. Xiao, C. Yan, A. Yao, J. Ye, M. Zhan, W. Zhang, H. Zhang, Q. Zhao, L. Zheng, F. Zhong, W. Zhong, S. Zhu, S. Zhao, D. Gilbert, S. Baumhueter, G. Spier, C. Carter, A. Cravchik, T. Woodage, F. Ali, H. An, A. Awe, D. Baldwin, H. Baden, M. Barnstead, I. Barrow, K. Beeson, D. Busam, A. Carver, A. Center, M. L. Cheng, L. Curry, S. Danaher, L. Davenport, R. Desilets, S. Dietz, K. Dodson, L. Doup, S. Ferriera, N. Garg, A. Gluecksmann, B. Hart, J. Haynes, C. Haynes, C. Heiner, S. Hladun, D. Hostin, J. Houck, T. Howland, C. Ibegwam, J. Johnson, F. Kalush, L. Kline, S. Koduru, A. Love, F. Mann, D. May, S. McCawley, T. McIntosh, I. McMullen, M. Moy, L. Moy, B. Murphy, K. Nelson, C. Pfannkoch, E. Pratts, V. Puri, H. Qureshi, M. Reardon, R. Rodriguez, Y. H. Rogers, D. Romblad, B. Ruhfel, R. Scott, C. Sitter, M. Smallwood, E. Stewart, R. Strong, E. Suh, R. Thomas, N. N. Tint, S. Tse, C. Vech, G. Wang, J. Wetter, S. Williams, M. Williams, S. Windsor, E. Winn-Deen, K. Wolfe, J. Zaveri, K. Zaveri, J. F. Abril, R. Guigo, M. J. Campbell, K. V. Sjolander, B. Karlak, A. Kejariwal, H. Mi, B. Lazareva, T. Hatton, A. Narechania, K. Diemer, A. Muruganujan, N. Guo, S. Sato, V. Bafna, S. Istrail, R. Lippert, R. Schwartz, B. Walenz, S. Yooseph, D. Allen, A. Basu, J. Baxendale, L. Blick, M. Caminha, J. Carnes-Stine, P. Caulk, Y. H. Chiang, M. Coyne, C. Dahlke, A. Mays, M. Dombroski, M. Donnelly, D. Ely, S. Esparham, C. Fosler, H. Gire, S. Glanowski, K. Glasser, A. Glodek, M. Gorokhov, K. Graham, B. Gropman, M. Harris, J. Heil, S. Henderson, J. Hoover, D. Jennings, C. Jordan, J. Jordan, J. Kasha, L. Kagan, C. Kraft, A. Levitsky, M. Lewis, X. Liu, J. Lopez, D. Ma, W. Majoros, J. McDaniel, S. Murphy, M. Newman, T. Nguyen, N. Nguyen, M. Nodell, S. Pan, J. Peck, M. Peterson, W. Rowe, R. Sanders, J. Scott, M. Simpson, T. Smith, A. Sprague, T. Stockwell, R. Turner, E. Venter, M. Wang, M. Wen, D. Wu, M. Wu, A. Xia, A. Zandieh and X. Zhu (2001). "The sequence of the human genome." Science **291**(5507): 1304-1351.

Yu, G. and Q. Y. He (2016). "ReactomePA: an R/Bioconductor package for reactome pathway analysis and visualization." Mol Biosyst **12**(2): 477-479.

Zhou, X., L. Ren, Q. Meng, Y. Li, Y. Yu and J. Yu (2010). "The next-generation sequencing technology and application." Protein Cell **1**(6): 520-536.