



Introduction to R

Some slides adapted from :

Arin Basu MD MPH

DataAnalytics

dataanalytics@rediffmail.com

<http://dataanalytics.objectis.net>

What is R

- **Software for Statistical Data Analysis**
- **Based on S**
- **Programming Environment**
- **Interpreted Language**
- **Data Storage, Analysis, Graphing**
- **Free and Open Source Software**

Downloading R

- R (current version).

<http://cran.wustl.edu/>

- Rstudio, a user-friendly way to use R

[https://rstudio.com/products/rstudio/download/#do
wnload](https://rstudio.com/products/rstudio/download/#download)

Strengths and Weaknesses

- **Strengths**
 - Free and Open Source
 - Strong User Community
 - Highly extensible, flexible
 - Implementation of high end statistical methods
 - Flexible graphics and intelligent defaults
- **Weakness**
 - Steep learning curve
 - Slow for large datasets

In an R Session...

- First, read data from other sources
- Use packages, libraries, and functions
- Write functions wherever necessary
- Conduct Statistical Data Analysis
- Save outputs to files, write tables
- Save R workspace if necessary (exit prompt)

Specific Tasks

search()

To see which directories and data are loaded, type:

ls()

To see which objects are stored

getwd()

to see your current working directory

Reading data into R

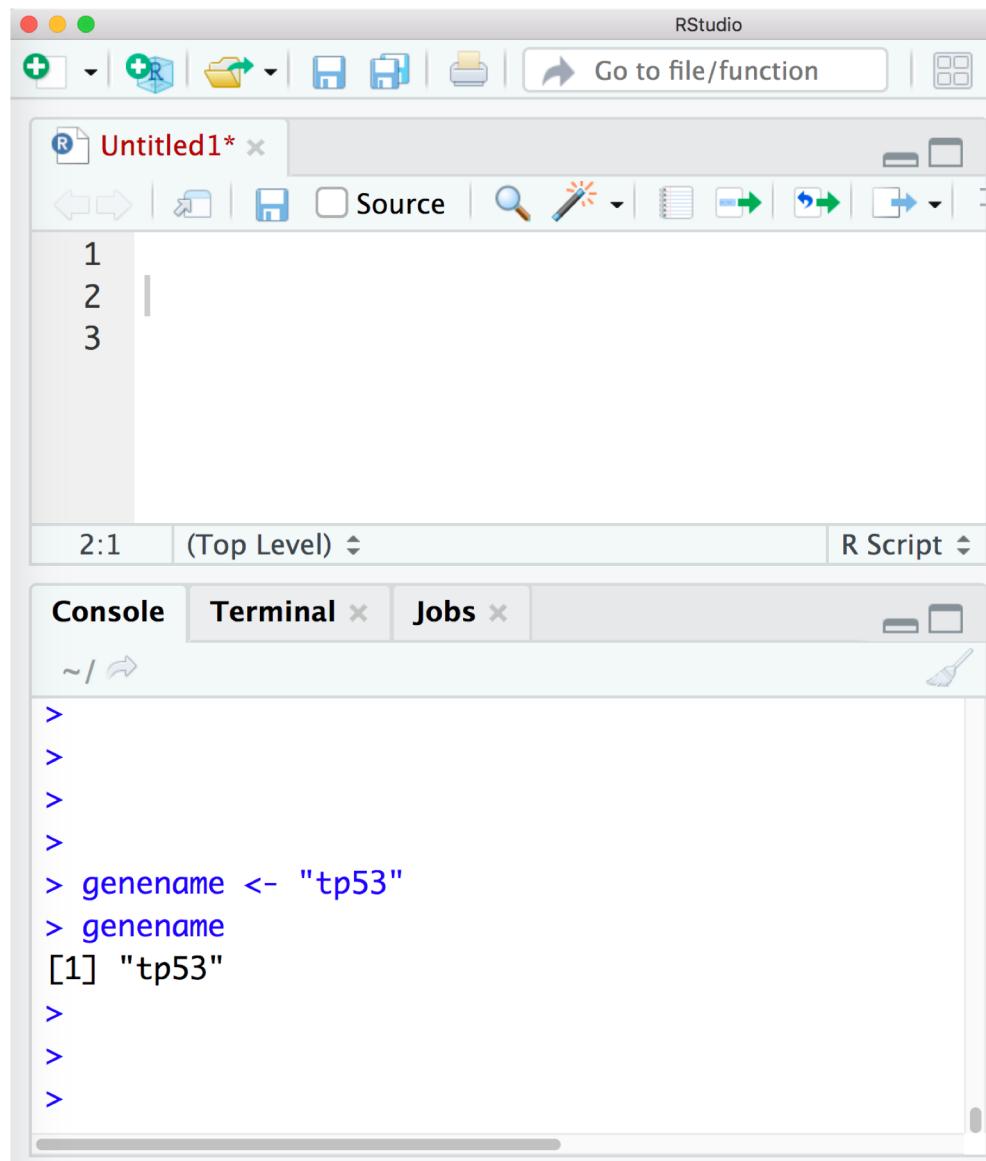
- R not well suited for data preprocessing
- Preprocess data elsewhere (Excel, etc...)
- Easiest form of data to input: .txt file
- Spreadsheet like data:
 - Small/medium size: use `read.table()`
 - Large data: use `scan()`

Reading Data: summary

- Directly using a vector e.g.: `x <- c(1,2,3...)`
- Using `scan` and `read.table` function
- Using `matrix` function to read data matrices
- Using `data.frame` to read mixed data
- `library(foreign)` for data from other programs

R

```
# this line creates the object 'a' and assigns it the value '1'  
a <- 1
```



Vectors

Vectors are probably the most used commonly used object type in R. A vector is a collection of values that are all of the same type (numbers, characters, etc.). One of the most common ways to create a vector is to use the `c()` function - the “concatenate” or “combine” function. Inside the function you may enter one or more values; for multiple values, separate each value with a comma:

R

```
# Create the SNP gene name vector  
snp_genes <- c("OXTR", "ACTN3", "AR", "OPRM1")
```

`setwd()`

Set working directory

`getwd()`

Get working directory

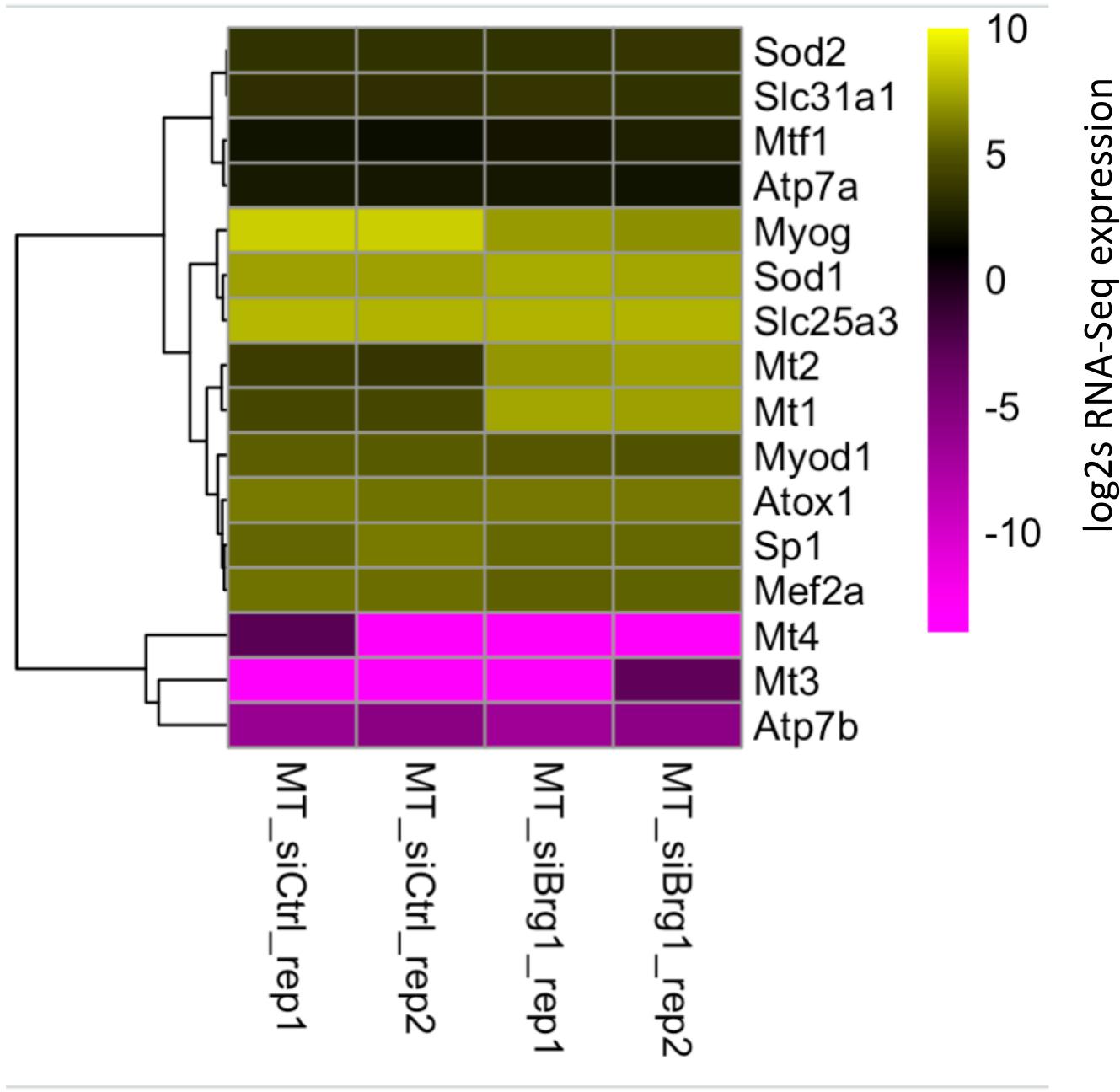
`library()`

load package of interest



R Packages for data analysis

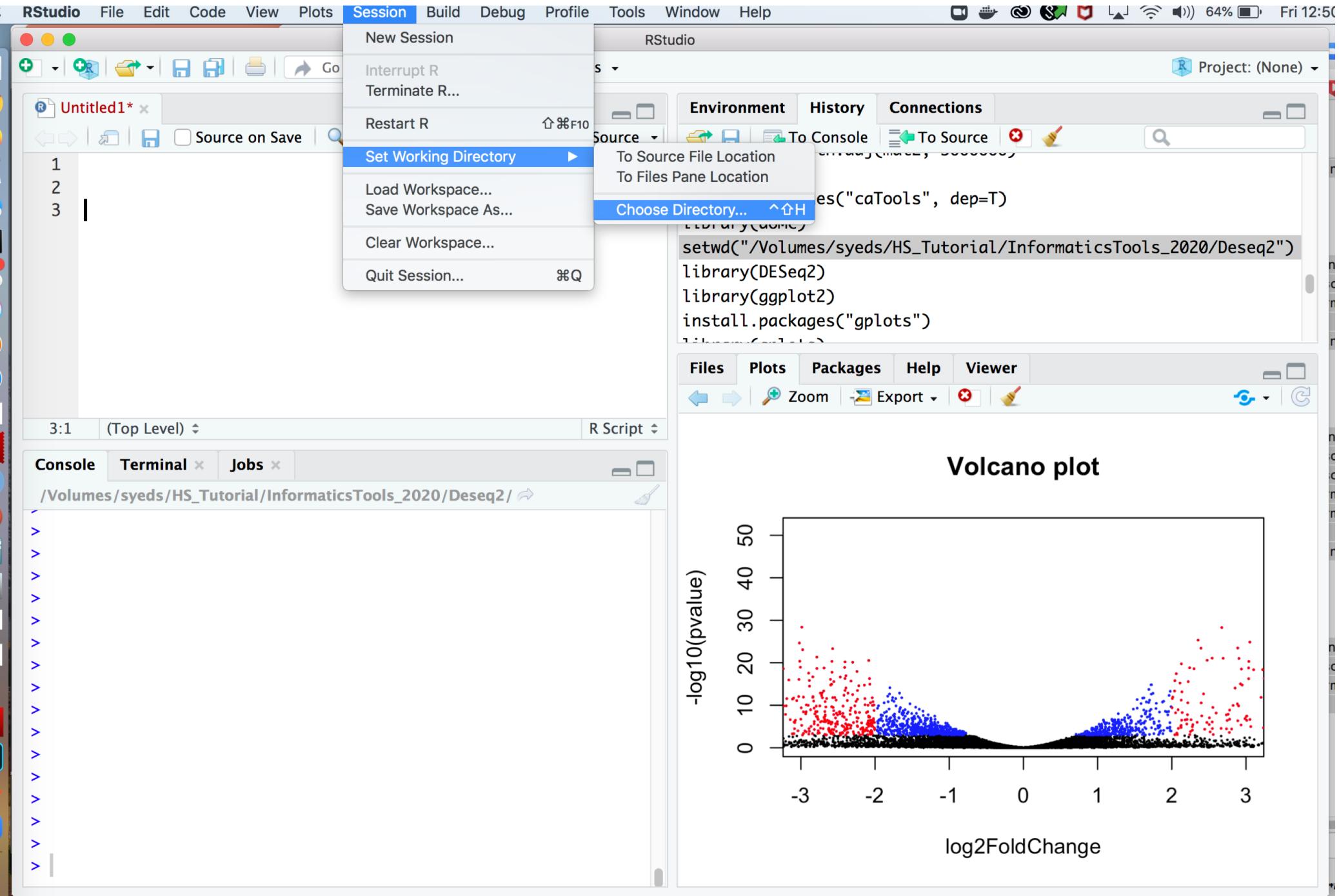
heatmaps



Prepare RSEM output files for heatmap creation

Copy TPM data into table

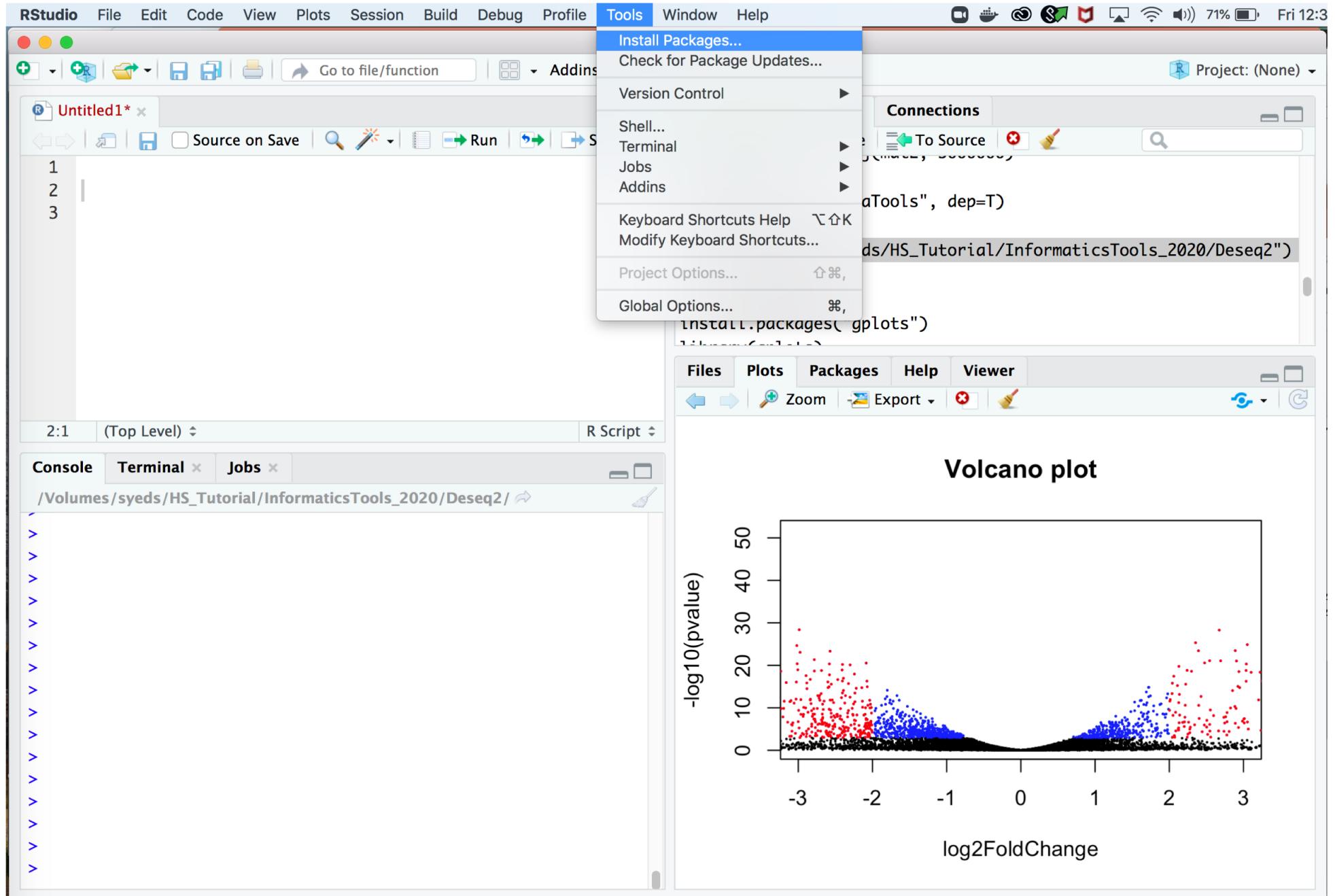
Calculate log2TPM of the table (if you have 0 values, convert them to a low decimal value (e.g. 0.0001)



```
> setwd("where file is located")
```

```
> install.packages("gplots")
```

Try installing RColorBrewer, colorRamps, and pheatmap using Rstudio buttons



```

> setwd("where file is located")

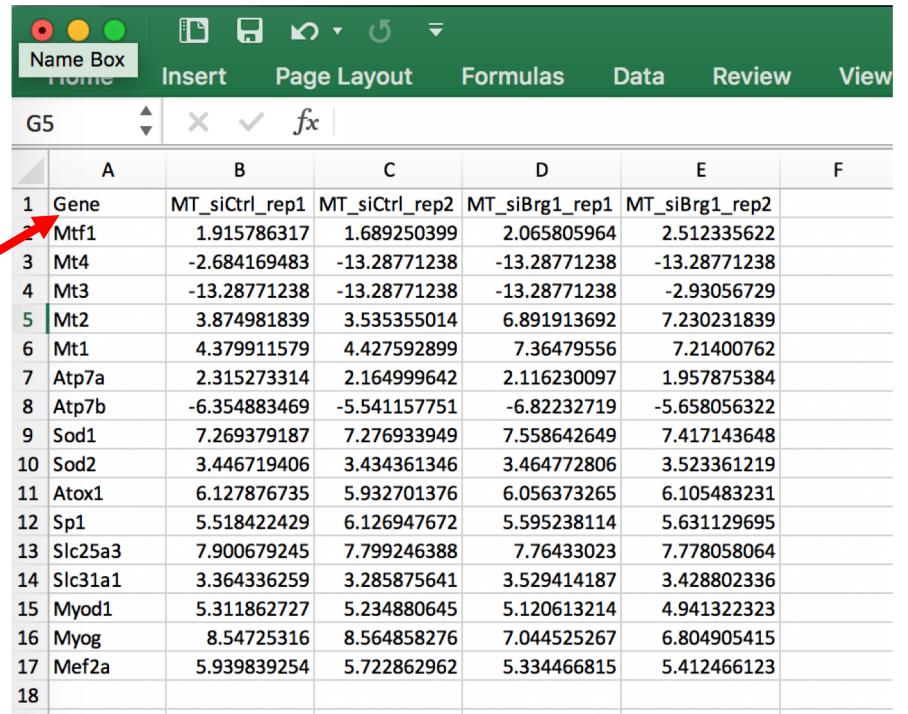
> install.packages("gplots")

> library(gplots)
> library(RColorBrewer)
> library(colorRamps)
> library(pheatmap)

> data=read.delim("Kan_genelist_1.txt", sep="\t", row.names="Gene", header=T)
> data_mat=as.matrix(data[,1:4])
> View(data_mat)

```

row.names = "Gene"



	A	B	C	D	E	F
1	Gene	MT_siCtrl_rep1	MT_siCtrl_rep2	MT_siBrg1_rep1	MT_siBrg1_rep2	
2	Mtf1	1.915786317	1.689250399	2.065805964	2.512335622	
3	Mt4	-2.684169483	-13.28771238	-13.28771238	-13.28771238	
4	Mt3	-13.28771238	-13.28771238	-13.28771238	-2.93056729	
5	Mt2	3.874981839	3.535355014	6.891913692	7.230231839	
6	Mt1	4.379911579	4.427592899	7.36479556	7.21400762	
7	Atp7a	2.315273314	2.164999642	2.116230097	1.957875384	
8	Atp7b	-6.354883469	-5.541157751	-6.82232719	-5.658056322	
9	Sod1	7.269379187	7.276933949	7.558642649	7.417143648	
10	Sod2	3.446719406	3.434361346	3.464772806	3.523361219	
11	Atox1	6.127876735	5.932701376	6.056373265	6.105483231	
12	Sp1	5.518422429	6.126947672	5.595238114	5.631129695	
13	Slc25a3	7.900679245	7.799246388	7.76433023	7.778058064	
14	Slc31a1	3.364336259	3.285875641	3.529414187	3.428802336	
15	Myod1	5.311862727	5.234880645	5.120613214	4.941322323	
16	Myog	8.54725316	8.564858276	7.044525267	6.804905415	
17	Mef2a	5.939839254	5.722862962	5.334466815	5.412466123	
18						

RStudio File Edit Code View Plots Session Build Debug Profile Tools Window Help

RStudio

Untitled1* data_mat

Go to file/function Addins

	MT_siCtrl_rep1	MT_siCtrl_rep2	MT_siBrg1_rep1	MT_siBrg1_rep2
Mtf1	1.915786	1.689250	2.065806	2.512336
Mt4	-2.684169	-13.287712	-13.287712	-13.287712
Mt3	-13.287712	-13.287712	-13.287712	-2.930567
Mt2	3.874982	3.535355	6.891914	7.230232
Mt1	4.379912	4.427593	7.364796	7.214008
Atp7a	2.315273	2.165000	2.116230	1.957875
Atp7b	-6.354883	-5.541158	-6.822327	-5.658056
Sod1	7.269379	7.276934	7.558643	7.417144
Sod2	3.446719	3.434361	3.464773	3.523361
Atox1	6.127877	5.932701	6.056373	6.105483
Sp1	5.518422	6.126948	5.595238	5.631130
Slc25a3	7.900679	7.799246	7.764330	7.778058

Showing 1 to 13 of 16 entries, 4 total columns

Console Terminal x Jobs x

/Volumes/syeds/HS_Tutorial/InformaticsTools_2020/projects/

```
Sod1      7.417144
Sod2      3.523361
Atox1     6.105483
Sp1       5.631130
Slc25a3   7.778058
Slc31a1   3.428802
Myod1     4.941322
Myog      6.804905
Mef2a     5.412466
> View(data_mat)
```

```
> setwd("where file is located")

> install.packages("gplots")

> library(gplots)
> library(RColorBrewer)
> library(colorRamps)
> library(pheatmap)

> data=read.delim("log2TPMtable.txt", sep="\t", row.names="Gene", header=T)
> data_mat=as.matrix(data[,1:4])
> View(data_mat)

> pdf("log2TPMtableheatmap_1.pdf", width=8, height=11)
```

Optional feature, turn on Pdf maker

```
> setwd("where file is located")

> install.packages("gplots")

> library(gplots)
> library(RColorBrewer)
> library(colorRamps)
> library(pheatmap)

> data=read.delim("log2TPMtable.txt", sep="\t", row.names="Gene", header=T)
> data_mat=as.matrix(data[1:4])
> View(data_mat)

> pdf("log2TPMtableheatmap_1.pdf", width=8, height=11)

> pheatmap(data_mat[1:16], breaks=c(seq(-14,-1.5, length.out=64),seq(-1.4,2,length.out=64),seq(2.1,10,length.out=64)), cluster_rows = TRUE, cluster_cols = FALSE, color =colorRampPalette(c("magenta", "black", "yellow"))(length(c(seq(-14,-1.5, length.out=64),seq(-1.4,2,length.out=64), seq(2.1,10,length.out=64)))))
```

Must identify how many rows in your dataset

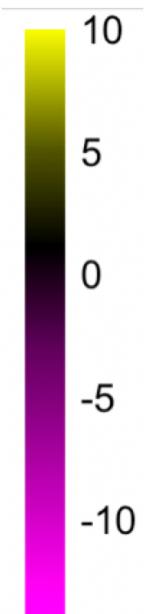
```
> pheatmap(data_mat[1:16], breaks=c(seq(-14,-1.5,
length.out=64),seq(-1.4,2,length.out=64),seq(2.1,10,length.out=64)),
cluster_rows = TRUE, cluster_cols = FALSE, color
=colorRampPalette(c("magenta", "black", "yellow"))(length(c(seq(-14,-
1.5, length.out=64),seq(-1.4,2,length.out=64),
seq(2.1,10,length.out=64)))))  

> dev.off()
```

Define the range of log2TPM values and when you want the colors to start transitioning

```
> pheatmap(data_mat[1:16], breaks=c(seq(-14,-1.5,  
length.out=64),seq(-1.4,2,length.out=64),seq(2.1,10,length.out=64)),  
cluster_rows = TRUE, cluster_cols = FALSE, color  
=colorRampPalette(c("magenta", "black", "yellow"))(length(c(seq(-14,-  
1.5, length.out=64),seq(-1.4,2,length.out=64),  
seq(2.1,10,length.out=64))))
```

```
> dev.off()
```



Clustering options

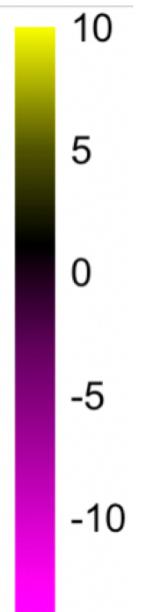
```
> pheatmap(data_mat[1:16], breaks=c(seq(-14,-1.5,
length.out=64),seq(-1.4,2,length.out=64),seq(2.1,10,length.out=64)),
cluster_rows = TRUE, cluster_cols = FALSE, color
=colorRampPalette(c("magenta", "black", "yellow"))(length(c(seq(-14,-
1.5, length.out=64),seq(-1.4,2,length.out=64),
seq(2.1,10,length.out=64)))))  

> dev.off()
```

```
> pheatmap(data_mat[1:16], breaks=c(seq(-14,-1.5,  
length.out=64),seq(-1.4,2,length.out=64),seq(2.1,10,length.out=64)),  
cluster_rows = TRUE, cluster_cols = FALSE, color  
=colorRampPalette(c("magenta", "black", "yellow"))(length(c(seq(-14,-  
1.5, length.out=64),seq(-1.4,2,length.out=64),  
seq(2.1,10,length.out=64))))
```

Define the colors you want for the legend

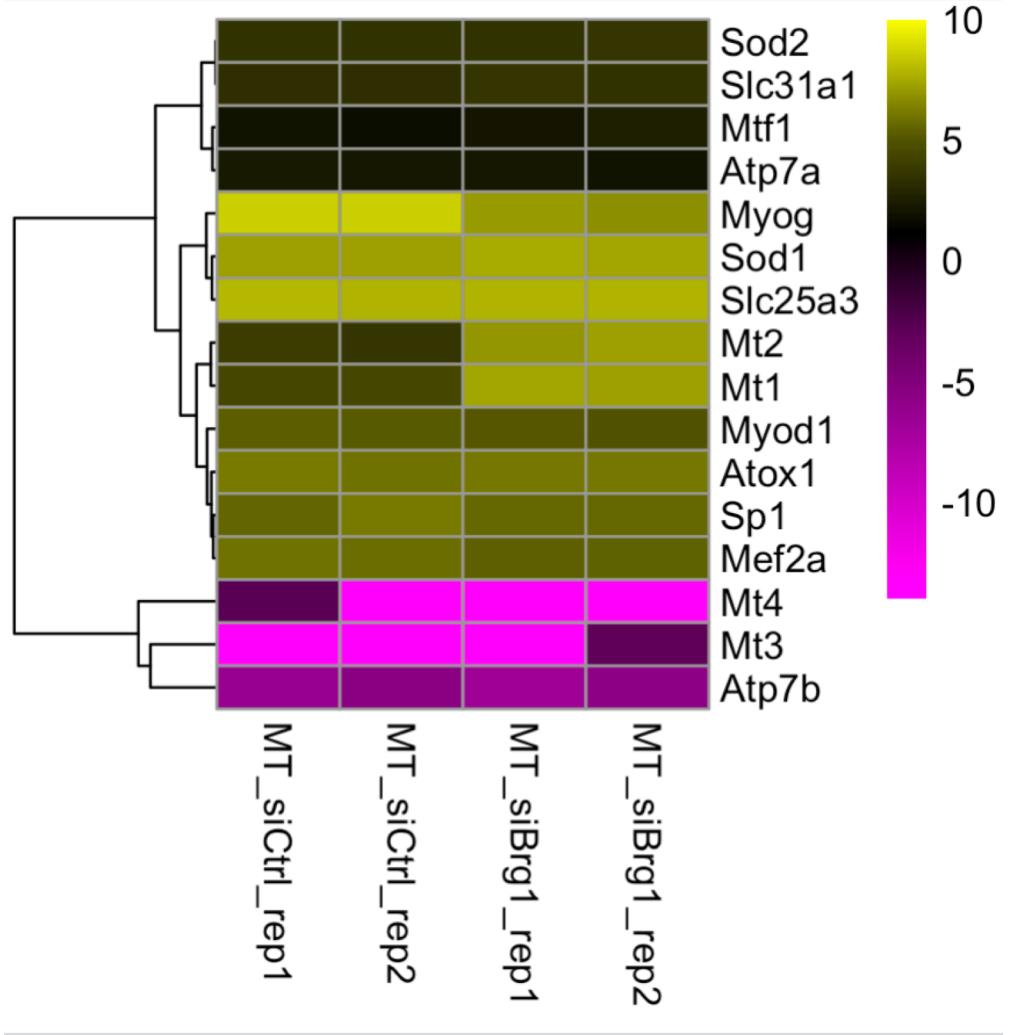
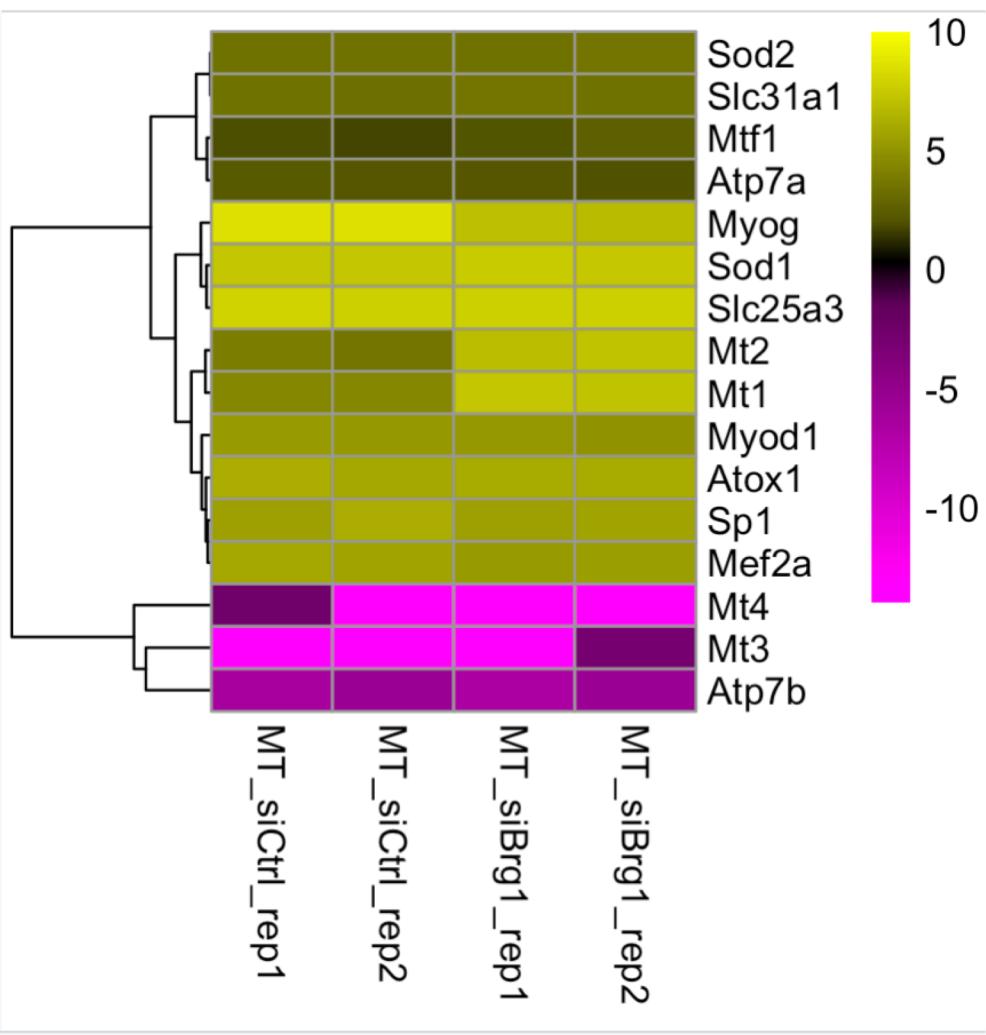
```
> dev.off()
```

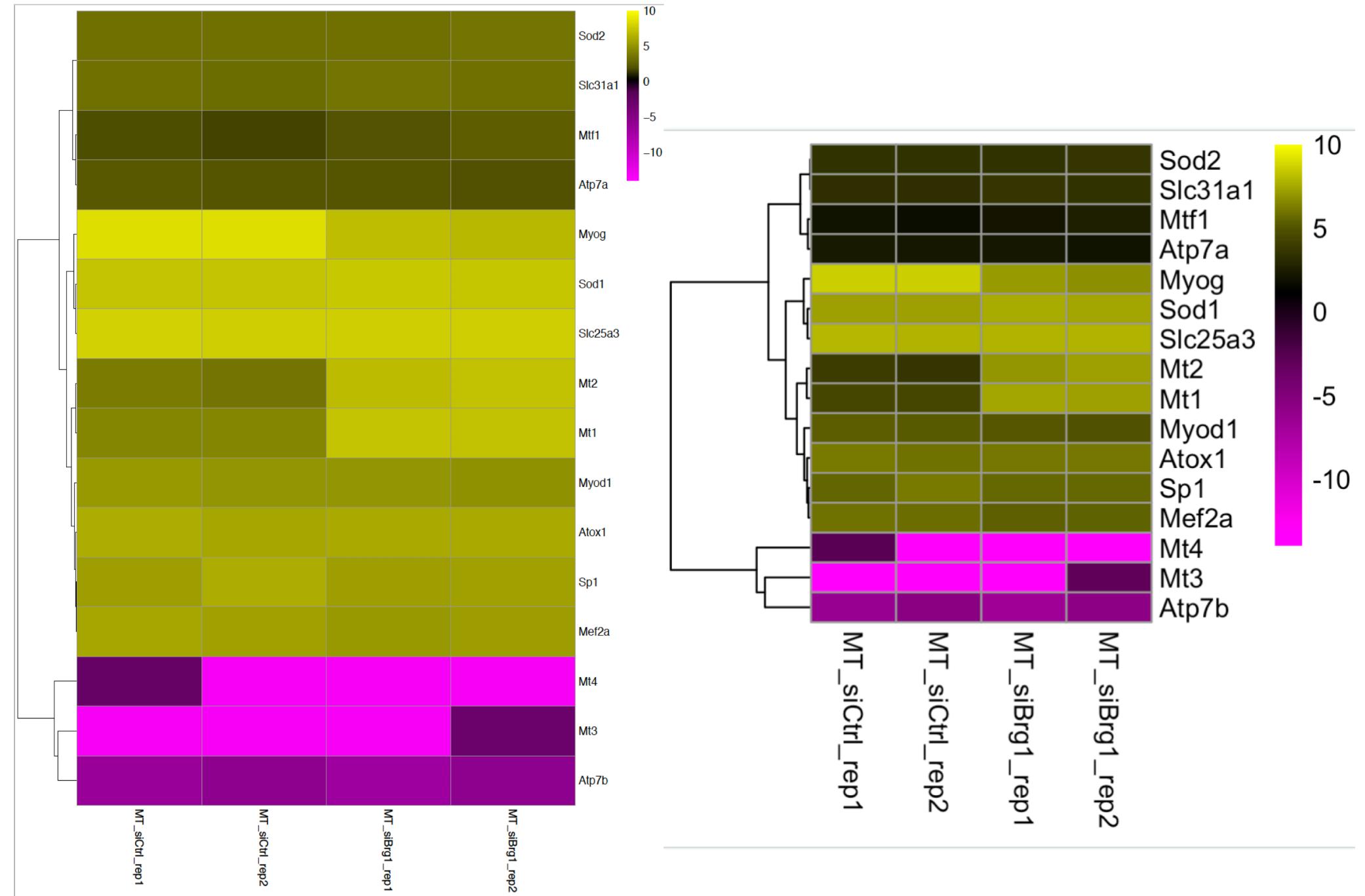


```
> pheatmap(data_mat[1:16], breaks=c(seq(-14,-1.5,  
length.out=64),seq(-1.4,2,length.out=64),seq(2.1,10,length.out=64)),  
cluster_rows = TRUE, cluster_cols = FALSE, color  
=colorRampPalette(c("magenta", "black", "yellow"))(length(c(seq(-14,-  
1.5, length.out=64),seq(-1.4,2,length.out=64),  
seq(2.1,10,length.out=64))))
```

```
> dev.off()
```

Turn off pdf maker





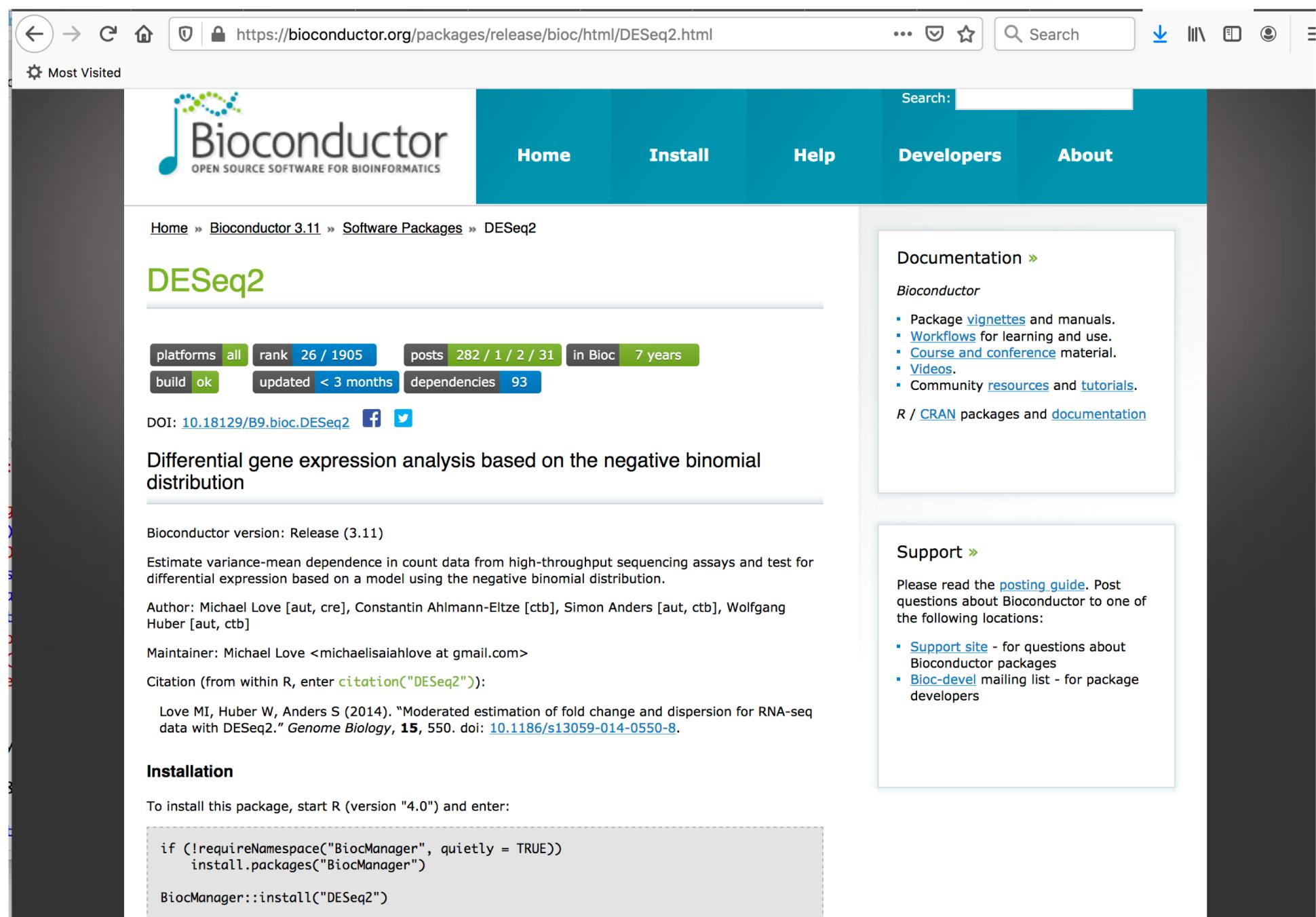
Made with Pdf maker option

Copied from Rstudio window



R Packages for data analysis

<https://bioconductor.org/packages/release/bioc/html/DESeq2.html>



The screenshot shows a web browser displaying the Bioconductor DESeq2 package page. The URL in the address bar is <https://bioconductor.org/packages/release/bioc/html/DESeq2.html>. The page features a dark teal header with the Bioconductor logo and navigation links for Home, Install, Help, Developers, and About. A search bar is also present in the header. The main content area includes a breadcrumb trail (Home > Bioconductor 3.11 > Software Packages > DESeq2), the title "DESeq2" in large green font, and various package statistics: platforms (all), rank (26 / 1905), posts (282 / 1 / 2 / 31), in Bioc (7 years), build (ok), updated (< 3 months), and dependencies (93). Below this, there's a DOI link ([10.18129/B9.bioc.DESeq2](https://doi.org/10.18129/B9.bioc.DESeq2)), social media links for Facebook and Twitter, and a brief description: "Differential gene expression analysis based on the negative binomial distribution". The page also lists Bioconductor version (Release 3.11), authors (Michael Love [aut, cre], Constantin Ahlmann-Eltze [ctb], Simon Anders [aut, ctb], Wolfgang Huber [aut, ctb]), maintainer (Michael Love <michaelisaiahlove@gmail.com>), citation information (Love MI, Huber W, Anders S (2014). "Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2." *Genome Biology*, **15**, 550. doi: [10.1186/s13059-014-0550-8](https://doi.org/10.1186/s13059-014-0550-8)), and an "Installation" section with R code for package installation. To the right, there are two sidebar boxes: "Documentation" (linking to vignettes, workflows, course material, videos, and community resources) and "Support" (linking to a posting guide and support sites).

Most Visited

Bioconductor
OPEN SOURCE SOFTWARE FOR BIOINFORMATICS

Home Install Help Developers About

Search:

Home » Bioconductor 3.11 » Software Packages » DESeq2

DESeq2

platforms all rank 26 / 1905 posts 282 / 1 / 2 / 31 in Bioc 7 years
build ok updated < 3 months dependencies 93

DOI: [10.18129/B9.bioc.DESeq2](https://doi.org/10.18129/B9.bioc.DESeq2) [f](#) [t](#)

Differential gene expression analysis based on the negative binomial distribution

Bioconductor version: Release (3.11)

Estimate variance-mean dependence in count data from high-throughput sequencing assays and test for differential expression based on a model using the negative binomial distribution.

Author: Michael Love [aut, cre], Constantin Ahlmann-Eltze [ctb], Simon Anders [aut, ctb], Wolfgang Huber [aut, ctb]

Maintainer: Michael Love <michaelisaiahlove@gmail.com>

Citation (from within R, enter `citation("DESeq2")`):

Love MI, Huber W, Anders S (2014). "Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2." *Genome Biology*, **15**, 550. doi: [10.1186/s13059-014-0550-8](https://doi.org/10.1186/s13059-014-0550-8).

Installation

To install this package, start R (version "4.0") and enter:

```
if (!requireNamespace("BiocManager", quietly = TRUE))
  install.packages("BiocManager")

BiocManager::install("DESeq2")
```

Documentation »

Bioconductor

- Package [vignettes](#) and manuals.
- [Workflows](#) for learning and use.
- [Course and conference](#) material.
- [Videos](#).
- Community [resources](#) and [tutorials](#).

R / [CRAN](#) packages and [documentation](#)

Support »

Please read the [posting guide](#). Post questions about Bioconductor to one of the following locations:

- [Support site](#) - for questions about Bioconductor packages
- [Bioc-devel](#) mailing list - for package developers

<https://bioconductor.org/packages/release/bioc/vignettes/DESeq2/inst/doc/DESeq2.html>

Analyzing RNA-seq data with DESeq2

Michael I. Love, Simon Anders, and Wolfgang Huber

05/12/2020

Abstract

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, and mass spectrometry. An important analysis question is the quantification 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 vignette explains the use of the package and demonstrates typical workflows. [An RNA-seq workflow](#) on the Bioconductor website covers similar material to this vignette but at a slower pace, including the generation of count matrices from FASTQ files. DESeq2 package version: 1.28.1

- Standard workflow
 - [Quick start](#)
 - [How to get help for DESeq2](#)
 - [Acknowledgments](#)
 - [Input data](#)
 - [Why un-normalized counts?](#)
 - [The DESeqDataSet](#)
 - [Transcript abundance files and *tximport / tximeta*](#)
 - [Tximeta for import with automatic metadata](#)
 - [Count matrix input](#)
 - [htseq-count input](#)
 - [SummarizedExperiment input](#)
 - [Pre-filtering](#)
 - [Note on factor levels](#)

<https://bioconductor.org/packages/release/bioc/vignettes/DESeq2/inst/doc/DESeq2.html>

Input data

Why un-normalized counts?

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 can be assigned 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 (with quantitative mass spectrometry). We will list method for obtaining count matrices in sections below.

The values in the matrix should be un-normalized counts or estimated counts of sequencing reads (for single-end RNA-seq) or fragments (for paired-end RNA-seq). The [RNA-seq workflow](#) describes multiple techniques for preparing such count matrices. It is important to provide count matrices as input for DESeq2's statistical model (Love, Huber, and Anders 2014) to hold, as only the count values allow assessing the measurement precision correctly. The DESeq2 model internally corrects for library size, so transformed or normalized values such as counts scaled by library size should not be used as input.

RSEM output

mAdiposeMSC_rsem

Search

Name	Date Modified	Size	Kind
AdiposeMSC_1_R1_ucsc_RSEM.genes.results.txt	Jan 14, 2019 at 1:05 PM	1.3 MB	Plain Text
AdiposeMSC_1_R1_ucsc_RSEM.isoforms.results	Jan 14, 2019 at 1:06 PM	1.8 MB	Document
AdiposeMSC_1_R1_ucsc_RSEM.transcript.bam	Jan 14, 2019 at 1:08 PM	1.96 GB	Document
AdiposeMSC_2_R1_ucsc_RSEM.genes.results.txt	Jan 14, 2019 at 1:06 PM	1.3 MB	Plain Text
AdiposeMSC_2_R1_ucsc_RSEM.isoforms.results	Jan 14, 2019 at 1:06 PM	1.8 MB	Document
AdiposeMSC_2_R1_ucsc_RSEM.transcript.bam	Jan 14, 2019 at 1:10 PM	2.44 GB	Document

syeds > Imbalzano_Lab > TranscriptomeComparisons > mAdiposeMSC_rsem

6 items, 69.26 TB available

Gene results

A	B	C	D	E	F	G
gene_id	transcript_id(s)	length	effective_length	expected_count	TPM	FPKM
0610005C13Rik	NR_038165,NR_038166	1018.5	983.5	0	0	0
0610007P14Rik	NM_021446	1185	1150	1296	58.24	31.74
0610009B22Rik	NM_025319	795	760	826.71	56.21	30.63
0610009L18Rik	NR_038126	619	584	87	7.7	4.2
0610009O20Rik	NM_024179	2404	2369	391.08	8.53	4.65
0610010B08Rik	NM_001177543	4539	4504	195.99	2.25	1.23
0610010F05Rik	NM_027860	4140	4105	1023	12.88	7.02
0610010K14Rik	NM_001177601,NM_001177603,NM_00117	806.29	771.29	710	47.57	25.92
0610011F06Rik	NM_026686	811	776	318	21.18	11.54
0610012G03Rik	NR_027897	1471	1436	1547.11	55.68	30.34
0610030E20Rik	NM_026696	4634	4599	172.12	1.93	1.05
0610031O16Rik	NR_045760	824	789	0	0	0
0610037L13Rik	NM_028754	1551	1516	981	33.44	18.22
0610038B21Rik	NR_028125	1523	1488	0	0	0
0610039K10Rik	NR_028113	951	916	0	0	0
0610040B10Rik	NR_027874	440	405	14	1.79	0.97
0610040F04Rik	NR_040757,NR_104577	784	749	4	0.28	0.15

Transcript variant results

transcript_id	gene_id	length	effective_length	expected_count	TPM	FPKM	IsoPct
NR_038165	0610005C13Rik	915	880	0	0	0	0
NR_038166	0610005C13Rik	1122	1087	0	0	0	0
NM_021446	0610007P14Rik	1185	1150	1296	58.24	31.74	100
NM_025319	0610009B22Rik	795	760	826.71	56.21	30.63	100
NM_038126	0610009L18Rik	619	584	87	7.7	4.2	100
NM_024179	0610009O20Rik	2404	2369	391.08	8.53	4.65	100
NM_001177543	0610010B08Rik	4539	4504	195.99	2.25	1.23	100
NM_027860	0610010F05Rik	4140	4105	1023	12.88	7.02	100
NM_001177601	0610010K14Rik	886	851	83.3	5.06	2.76	10.63
NM_001177603	0610010K14Rik	781	746	310.33	21.5	11.71	45.19
NM_001177606	0610010K14Rik	679	644	0	0	0	0
NM_001177607	0610010K14Rik	706	671	107.02	8.24	4.49	17.33
NM_026757	0610010K14Rik	780	745	0	0	0	0
NM_145758	0610010K14Rik	882	847	209.36	12.77	6.96	26.85
NM_026686	0610011F06Rik	811	776	318	21.18	11.54	100
NR_027897	0610012G03Rik	1471	1436	1547.11	55.68	30.34	100
NM_026696	0610030E20Rik	4634	4599	172.12	1.93	1.05	100
NR_045760	0610031O16Rik	824	789	0	0	0	0
NM_028754	0610037L13Rik	1551	1516	981	33.44	18.22	100

Prepare RSEM output files for Differential Gene Expression analysis

Copy counts data into table

***NOTE, excel alters genes Marc1, Marc2 and March1, March2 to 03-01-20 so keep any eye on these gene names if you receive an error about duplicates.

```
head(countData)
```

```
##          ensgene SRR1039508 SRR1039509 SRR1039512 SRR1039513 SRR1039516
## 1 ENSG000000000003      723      486      904      445     1170
## 2 ENSG000000000005      0        0        0        0        0
## 3 ENSG00000000419     467      523      616      371      582
## 4 ENSG00000000457     347      258      364      237      318
## 5 ENSG00000000460      96       81       73       66      118
## 6 ENSG00000000938      0        0        1        0        2
##          SRR1039517 SRR1039520 SRR1039521
## 1      1097      806      604
## 2        0        0        0
## 3      781      417      509
## 4      447      330      324
## 5       94      102       74
## 6        0        0        0
```

DESeq2 Tutorial

https://lashlock.github.io/compbio/R_presentation.html

Install packages and load libraries

```
#install.packages("htmltools")
#library(htmltools)
#source("https://bioconductor.org/biocLite.R")
#biocLite("DESeq2")

library("DESeq2")
library(ggplot2)
```

Example of a metadata file that describes the experiment and replicates

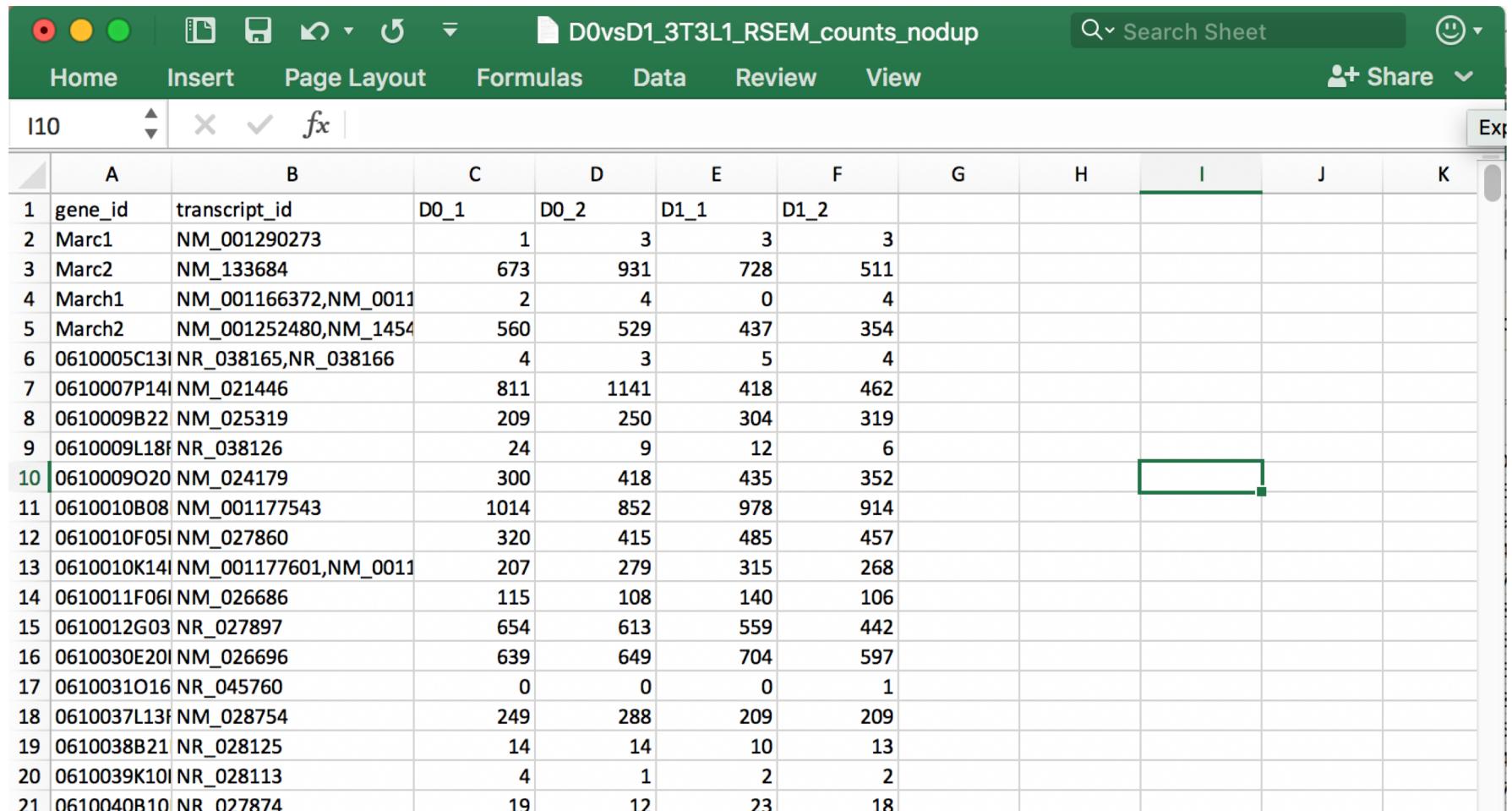
```
metaData
```

```
##           id      dex celltype      geo_id
## 1 SRR1039508 control  N61311 GSM1275862
## 2 SRR1039509 treated  N61311 GSM1275863
## 3 SRR1039512 control  N052611 GSM1275866
## 4 SRR1039513 treated  N052611 GSM1275867
## 5 SRR1039516 control  N080611 GSM1275870
## 6 SRR1039517 treated  N080611 GSM1275871
## 7 SRR1039520 control  N061011 GSM1275874
## 8 SRR1039521 treated  N061011 GSM1275875
```

RSEM Script

```
> file <- "D0vsD1_3T3L1_RSEM_counts_nodup.txt"  
> file2 <- "metadata.txt"
```

```
file <- "D0vsD1_3T3L1_RSEM_counts_nodup.txt"
```



	A	B	C	D	E	F	G	H	I	J	K
1	gene_id	transcript_id	D0_1	D0_2	D1_1	D1_2					
2	Marc1	NM_001290273		1	3	3	3				
3	Marc2	NM_133684	673	931	728	511					
4	March1	NM_001166372,NM_0011	2	4	0	4					
5	March2	NM_001252480,NM_1454	560	529	437	354					
6	0610005C13	NR_038165,NR_038166	4	3	5	4					
7	0610007P14	NR_021446	811	1141	418	462					
8	0610009B22	NR_025319	209	250	304	319					
9	0610009L18	NR_038126	24	9	12	6					
10	0610009O20	NR_024179	300	418	435	352					
11	0610010B08	NR_001177543	1014	852	978	914					
12	0610010F05	NR_027860	320	415	485	457					
13	0610010K14	NR_001177601,NM_0011	207	279	315	268					
14	0610011F06	NR_026686	115	108	140	106					
15	0610012G03	NR_027897	654	613	559	442					
16	0610030E20	NR_026696	639	649	704	597					
17	0610031O16	NR_045760	0	0	0	1					
18	0610037L13	NR_028754	249	288	209	209					
19	0610038B21	NR_028125	14	14	10	13					
20	0610039K10	NR_028113	4	1	2	2					
21	0610040B10	NR_027874	19	12	23	18					

***NOTE, excel alters genes Marc1, Marc2 and March1, March2 to 03-01-20 so keep an eye on these gene names if you receive an error about duplicates.

```
file2 <- "metadata.txt"
```

RSEM Script

Load the RSEM_counts and the experimental metadata files

```
> file <- "D0vsD1_3T3L1_RSEM_counts_nodup.txt"
```

```
> file2 <- "metadata.txt"
```

Convert those files into R tables

```
> rsem <- read.table(file , sep="\t", header=TRUE, row.names=1)
```

```
> metadata <- read.table(file2 , sep="\t", header=TRUE, row.names=1)
```

Construct DESEQDataSet Object

```
> dds <- DESeqDataSetFromMatrix(countData=rsem, colData=metaData,  
design=~group, tidy = TRUE)
```

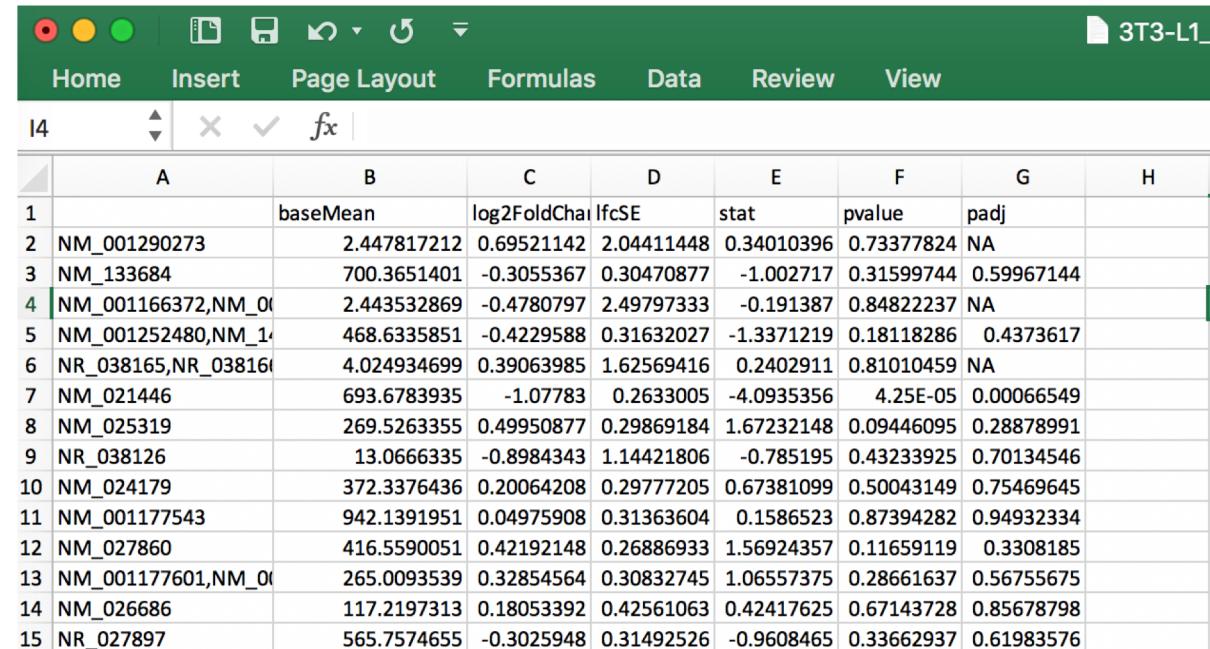
Run the DESEQ function

```
> dds <- DESeq(dds)  
      > dds <- DESeq(dds)  
      estimating size factors  
      estimating dispersions  
      gene-wise dispersion estimates  
      mean-dispersion relationship  
      final dispersion estimates  
      fitting model and testing  
      > |
```

Export Results

```
> res <- results(dds)  
> write.table(res, file="3T3-L1_D0vsD1_DESeq_FCanalysis.txt", sep="\t")
```

DESeq2 output: log2FoldChange calculation and statistics



The screenshot shows a Microsoft Excel spreadsheet titled "3T3-L1_". The table has columns labeled A through H. Column A contains sample identifiers, and columns B through H contain statistical parameters: baseMean, log2FoldChange, IfcSE, stat, pvalue, and padj. The data includes various gene and transcript identifiers like NM_001290273, NM_133684, and NR_038165.

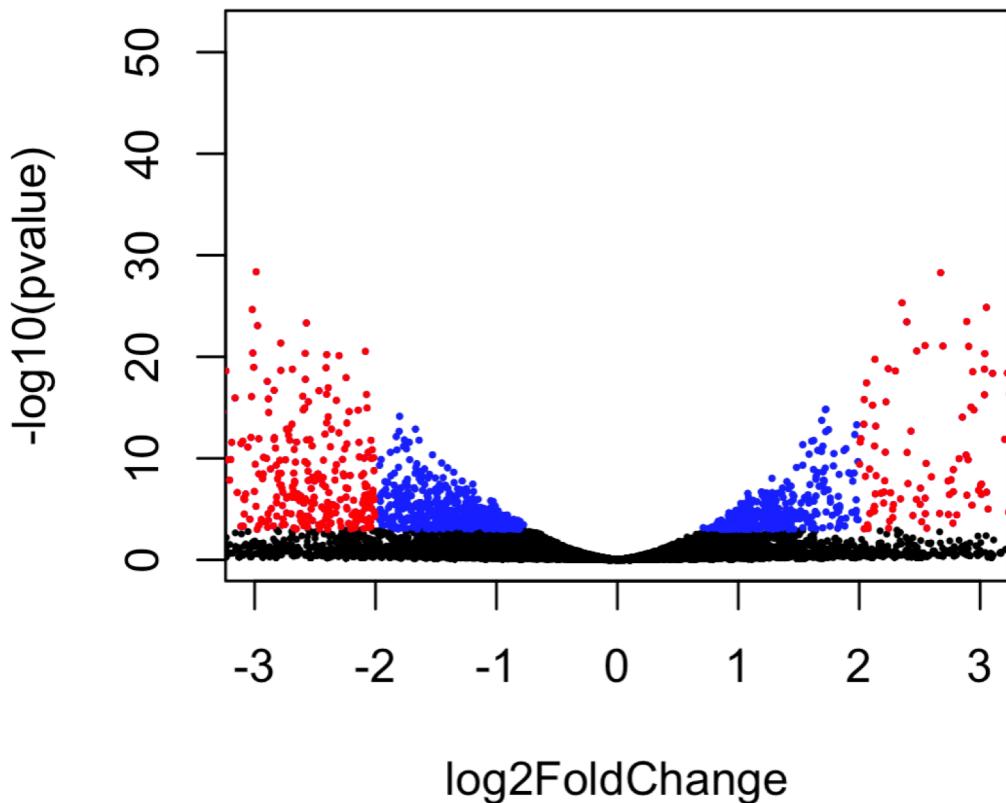
A	B	C	D	E	F	G	H
	baseMean	log2FoldChange	IfcSE	stat	pvalue	padj	
NM_001290273	2.447817212	0.69521142	2.04411448	0.34010396	0.73377824	NA	
NM_133684	700.3651401	-0.3055367	0.30470877	-1.002717	0.31599744	0.59967144	
NM_001166372,NM_001252480,NM_133684	2.443532869	-0.4780797	2.49797333	-0.191387	0.84822237	NA	
NM_001252480,NM_133684	468.6335851	-0.4229588	0.31632027	-1.3371219	0.18118286	0.4373617	
NR_038165,NR_038166	4.024934699	0.39063985	1.62569416	0.2402911	0.81010459	NA	
NM_021446	693.6783935	-1.07783	0.2633005	-4.0935356	4.25E-05	0.00066549	
NM_025319	269.5263355	0.49950877	0.29869184	1.67232148	0.09446095	0.28878991	
NR_038126	13.0666335	-0.8984343	1.14421806	-0.785195	0.43233925	0.70134546	
NM_024179	372.3376436	0.20064208	0.29777205	0.67381099	0.50043149	0.75469645	
NM_001177543	942.1391951	0.04975908	0.31363604	0.1586523	0.87394282	0.94932334	
NM_027860	416.5590051	0.42192148	0.26886933	1.56924357	0.11659119	0.3308185	
NM_001177601,NM_001252480	265.0093539	0.32854564	0.30832745	1.06557375	0.28661637	0.56755675	
NM_026686	117.2197313	0.18053392	0.42561063	0.42417625	0.67143728	0.85678798	
NR_027897	565.7574655	-0.3025948	0.31492526	-0.9608465	0.33662937	0.61983576	

- **baseMean** = normalized average read count.
- **log2foldchange**
- **IfcSE** = the standard error estimate for the log2 fold change estimate.
- **stat** = wald statistic (the Wald test is the default used for hypothesis testing when comparing two groups. The Wald test is a test of hypothesis usually performed on parameters that have been estimated by maximum likelihood).
- **p-value** = non adjusted **pvalue**.
- ***p-value adjusted* (padj)** column contains the p-values, adjusted for multiple testing with the Benjamini-Hochberg procedure (see the standard R function `p.adjust`), which controls *false discovery rate* (FDR) . It's possible to restrict the result for the ones which are under a fixed FDR cut-off. *Example : A FDR adjusted p-value (or q-value) of 0.05 implies that 5% of significant tests will result in false positives.*

Volcano Plot

```
#reset par  
par(mfrow=c(1,1))  
  
# Make a basic volcano plot  
  
with(res, plot(log2FoldChange, -log10(pvalue), pch=20, cex=0.3, main="Volcano  
plot", xlim=c(-3,3)))  
  
# Add colored points: blue if padj<0.01, red if log2FC>1 and padj<0.05  
  
with(subset(res, padj<.01 ), points(log2FoldChange, -log10(pvalue), pch=20, cex=0.3,  
col="blue"))  
  
with(subset(res, padj<.01 & abs(log2FoldChange)>2), points(log2FoldChange, -  
log10(pvalue), pch=20, cex=0.3, col="red"))
```

Volcano plot



For labeling genes on a volcano plot

```
> library(dplyr)  
> library(ggplot2)  
> library(ggrepel)
```

Read blogs: <https://gettinggeneticsdone.blogspot.com/2014/05/r-volcano-plots-to-visualize-rnaseq-microarray.html>

<https://gettinggeneticsdone.blogspot.com/2016/01/repel-overlapping-text-labels-in-ggplot2.html>