在您键入时，此键盘是同步的，以便每个查看此页面的人都能看到相同的文本。这使您可以在文档上进行无缝协作。

介绍

我们的研讨会包括：
R 대체 RNAseq
差异表达和基因组测试-,v
对齐和特征计数
应用 RNAseq / BYO 数据

我们研讨会的页面（包括安装说明）：
http://combine-australia.github.io/2016-05-11-RNAseq/

您可以 in 这里找到本研讨会的所有课程笔记：
http://combine-australia.github.io/RNAseq-R/

反馈： http ： //goo.gl/forms/unKN4MNgQG

无线上网：

에듀로암
用户名（通常）：<机构用户名> @ <机构>，例如用于 UoM login@student.unimelb.edu.au

글리마下载：
https://www.bioconductor.org/packages/release/bioc/html/Glimma.html
您将首先需要 DESeq2
install.packages(“ https://www.bioconductor.org/packages/release/bioc/bin/macosx/mavericks/contrib/3.3/Glimma_1.0.0.tgz ”, repos = NULL)

紧急 **RStudio** 实例（写您的名字以声明实例）：
https://docs.google.com/spreadsheets/d/1wDkSgVL-RqXyjosHNBQPTUxrUyOGU1qXk_b9HyXmy-M/edit?usp=sharing

에서 此页面上写下您的名字!
彼得·侯韦林
劳伦·布朗
伊丽莎白·克里斯蒂(엘리자베스 크리스티)
카산드라 고(莫纳什代表!)
그레첸 푸팅가(格雷琴·波廷加)
杰西卡海滩
建康
列夫·卡茨
米里亚姆·范朱尔(미리암 판줄)
汉森·科萨西(한센 코사시)
弥弥
安德里亚·沃尔特曼

考特尼·范·盖伦
오메르 길란(欧默·吉兰)
妮可·范·卑尔根(Nicole Van Bergen)*
阿什莉·杜(애슐리 두)
西莉亚
克莱尔·斯莱尼
雷切尔·基亚罗尼-克拉克
埃里克·库斯纳迪(에릭 쿠스나디)
瓦娜·萨尼斯拉夫(오아나 사니슬라브)
隋莱
莎莎·阿布·哈马德
克莱尔·安斯特德（Clare Anstead）
特伦特·佩里
萨珊娜·杜申安（Sathana Dushyanthen）
莎拉·普雷斯顿
山姆·曼纳
王敏雨
卡拉·布里特
卢卡斯·邓特
汤姆·塞里
洛伦佐·德·马可（Lorenzo De Marco）

coat,weight,likes_string
calico,2.1,TRUE
black,5.0,FALSE
tabby,3.2,TRUE
---
title: "Pathway analysis"
author: "Asela Wijeratne"
output: html_document

###During this session you will learn about:

1. Use of GAGE package to do pathway analysis

   a. "Native workflow"

   b. Combined workflow with RNAseq

2. Interpret the GAGE output

3. Use of Pathview to visualize the perturbed KEGG pathways

######For more details, please go to relevent reference manauls:

GAGE package and tutorial:

http://bioconductor.org/packages/devel/bioc/html/gage.html

pathview and gageData:

http://bioconductor.org/packages/devel/bioc/html/pathview.html

http://bioconductor.org/packages/devel/data/experiment/html/gageData.html
EdgeR user guide:

http://bioconductor.org/packages/release/bioc/vignettes/edgeR/inst/doc/edgeRUsersGuide.pdf

###Installing necessary packages

Before you start the tutorial following packages needs to be installed.
Install required packages:

```
source("http://bioconductor.org/biocLite.R")
biocLite(c("pathview", "edgeR", "gage"))
```

Load required packages

```
library(edgeR)
library(gage)
library(pathview)


```

###Getting and preprocessing data

First, let's get some RNAseq data. This data comes from a study described in Zhang et al., 2011, Genome-wide mapping of the HY5-mediated gene-networks in Arabidopsis that involve both transcriptional and post-transcriptional regulation (http://onlinelibrary.wiley.com/doi/10.1111/j.1365-313X.2010.04426.x/full). This a comparison between a Arabidopsis mutant called hy5 and wild-type. For each genotype, there are two replicates. Vince Buffalo has re-analyzed this data as described here: (https://github.com/vsbuffalo/rna-seq-example). We are going to grab the count data file from this analysis.

```
rnaseq_URL <- "https://github.com/vsbuffalo/rna-seq-example/raw/master/results/raw-counts.txt"
download.file(rnaseq_URL, destfile = "rnaseq_data.txt")
```

```
rnaseq_data <- read.table("rnaseq_data.txt",
                          header=TRUE)
```

#####If above code work, ignore the rest of the next part of the code.

You can also get the data from the following link and save it in the ```Desktop/pathway_analysis``` folder.

http://bit.ly/150826_rnaseqdata

Read the data into R using the following code:

```
rnaseq_data <- read.table("rnaseq_data.txt",
                          header=TRUE)

```

Let's look at the head of the data set.

```

head(rnaseq_data)
````


This is how they are labelled:

```
GEO:GSM613465: wild type samples
      SRR070570: replicate 1
      SRR070571: replicate 2
GEO:GSM613466: hy5 mutant samples
      SRR070572: replicate 1
      SRR070573: replicate 2

```

Rename the columns so we can identify them easily.

```
colnames(rnaseq_data) <- c("WT_1", "WT_2", "hy_1", "hy_2")

```

###Getting KEGG data

#####1. Finding your species in KEGG

First, we need to make sure the species we work with is found in KEGG database. You can go the following webpage and see the organism is list:

http://rest.kegg.jp/list/organism

In addition, ```pathview``` also carry a data matrix with KEGG supported species. You can explore this matrix as well.

```
data(korg)
head(korg)

```

You can get the short names for Arabidopsis by searching the ```korg``` data.

```
org <- "arabidopsis thaliana"
species <- unlist(sapply(1:ncol(korg), function(i) {
   agrep(org, korg[, i])
}))
korg[species, 1, drop = F]

```

#####2. Creating the KEGG dataset for GAGE analysis
```kegg.gsets``` can be used to get KEGG data for any species present in the KEGG database. GAGE manual recommends that you save this data as a .Rdata file. This way you don't need to download this each time you need to use and also increase the reproducibility.

```
kegg_arab <- kegg.gsets("ath")
kegg.gs <- kegg_arab$kg.sets[kegg_arab$sigmet.idx]

```

###GAGE "native" workflow

#####1. Remove zero counts in all samples.

```
rnaseq_counts <- rnaseq_data # rename the dataset

dim(rnaseq_counts)
non_zero <- rowSums(rnaseq_counts) != 0
rnaseq_counts <- rnaseq_counts[non_zero,]
# number of genes left after removing zero counts
```
```

dim(rnaseq_counts)
```

#####2. Normalize library sizes

```
libsizes <- colSums(rnaseq_counts)
size_factor <- libsizes / exp(mean(log(libsizes)))
norm_counts <- t(t(rnaseq_counts) / size_factor)
range(norm_counts)
```

#####3. Variance stabilizing transformation
We need to add small (positive) value to avoid getting ```-inf``` during log transformation. We will add 8 as per GAGE manual.

```
norm_counts <- log2(norm_counts + 8)
range(norm_counts)
```

You can do MA and PCA plots to see the processed data variances and overall variances and similarity between samples, respectively.

#####4. Define reference and experiment samples

```
ref_idx <- 1:2 #wt
samp_idx <- 3:4 #hy mutant
```

#####5. Enrichment analysis
Here we are going to perform the enrichment analysis using fold change (default) as the gene set statistics. In addition, control (wt) and experiment (hy) are not paired. Therefore, we going to use ```compare ="unpaired"```.

```
native_kegg_fc <- gage(norm_counts,
                       gsets = kegg.gs,
                       ref = ref_idx,
                       samp = samp_idx,
                       compare ="unpaired")
```

#####6. Some outputs

First four pathways that are up-regulated.
```

```
head(native_kegg_fc$greater[,1:5], 4)
```

######Meaning of the output:

```p.geomean``` geometric mean of the individual p-values from multiple single array based gene set tests

```stat.mean``` mean of the individual statistics from multiple single array based gene set tests

```p.val``` gloal p-value or summary of the individual p-values from multiple single array based gene set tests. This is the default p-value being used.

```q.val``` FDR q-value adjustment of the global p-value using the Benjamini & Hochberg procedure implemented in multtest package. This is the default q-value being used.

```set.size``` the effective gene set size, i.e. the number of genes included in the gene set test

First four pathways that are down-regulated.

```
head(native_kegg_fc$less[,1:5], 4)

```

We can also do significance test to find the pathways that are up-regulated and down-regulated using a given cutoff (default is 0.1). ```sigGeneSet``` function will also create a heat map, showing the average fold change of expression in pathways that are perturbed.

ㅁㄴㅇㄹ
```
wt_hy_sig_kegg <-sigGeneSet(native_kegg_fc, outname="wt_hy.kegg")
```

Write the significant pathway sets to a text file.

```
write.table(rbind(wt_hy_sig_kegg$greater,
                  wt_hy_sig_kegg$less),
            file = "wt_hy_sig_kegg.txt", sep = "\t")
```

###Combine GAGE and EdgeR analysis

First, we will do a differential gene expression analysis using the ```edgeR``` package.
```

#####1. Make the study design as follows:

```
targets <-
      data.frame(sample_name=colnames(rnaseq_data),Group=rep(c("WT","hy"),each=2))

targets
```

#####2. Create DGElist object.

This is a R list object that can be easily manipulated.

```

d <- DGEList(counts=rnaseq_data,
           group=targets$Group) # Constructs DGEList object
```

#####3. Remove lowly expressed genes
Keep genes with more than 10 counts per million (calculated with ```cpm()```) at least in two samples.

```
#before filtering:
dim(d)
keep <- rowSums(cpm(d)> 10) >= 2
d <- d[keep,]
#after filtering:
dim(d)

```

#####4. Normalize the data

```
d$samples$lib.size <- colSums(d$counts)
d <- calcNormFactors(d)
d

```

#####5. multidimentioanl scaling plot
Quick sanity check to see how samples are related to each other using multidimensional scaling plot (MSD).

```

plotMDS(d, labels = targets$Group,
       col = c("darkgreen","blue")[factor(targets$Group)])
```

#####6. Estimate the dispersion

```
d1 <- estimateCommonDisp(d, verbose=T)

d1 <- estimateTagwiseDisp(d1)

plotBCV(d1)
```

#####7. Differentail gene expression

```
de.com <- exactTest(d1, pair = c("WT", "hy"))
topTags(de.com, n = 5)

```

```
FDR <- p.adjust(de.com$table$PValue, method="BH")
de1 <- decideTestsDGE(de.com, adjust.method="BH", p.value=0.05)
summary(de1)

```

#####8. Making the data suitable for pathway analysis

```
isDE <- as.logical(de1) # covert to DE set to true/false set

DEnames <- rownames(d)[isDE] # get the DE gene names
edger.fc <- de.com$table$logFC # get the log fold change
names(edger.fc) <- rownames(de.com$table) # assign row names to fold change
exp.fc <- edger.fc
head(exp.fc)

```

Getting DE gene set and fold change

```
DE_foldchange <- edger.fc[DEnames]
```

length(DE_foldchange)
```
#####9. GAGE analysis

```
fc.kegg.p <- gage(exp.fc, gsets = kegg.gs, ref = NULL, samp = NULL)

head(fc.kegg.p$greater[,1:5], 4)

```

Significance test

```
wt_hy_sig_kegg <-sigGeneSet(fc.kegg.p, outname="wt_hy.kegg")
```




###Visualize the data using Pathview


```
log_fc= norm_counts[, samp_idx]-rowMeans(norm_counts[, ref_idx])

```
#####1. Selecting the path ids for the upregulated set.

```
greater_set <- native_kegg_fc$greater[, "q.val"] < 0.1 &
      !is.na(native_kegg_fc$greater[, "q.val"])
greater_ids <- rownames(native_kegg_fc$greater)[greater_set]
head(greater_ids)
```

#####2. Selecting path ids for down-regulated set

```
less_set <- native_kegg_fc$less[, "q.val"] < 0.1 &
      !is.na(native_kegg_fc$less[,"q.val"])
less_ids <- rownames(native_kegg_fc$less)[less_set]
```

#####3. Combine up and down-regulated path ids.

```
combine_ids <- substr(c(greater_ids, less_ids), 1, 8)
```
```

```
head(combine_ids)
```

#####4. Visualization

Here we are going to get the first three pathways and visualize using ```pathview```.

```
pv.out.list <- sapply(combine_ids[1:3], function(pid) pathview(
      gene.data =  exp.fc, pathway.id = pid,
      species = "ath", out.suffix="Wt_hy",
      gene.idtype="KEGG"))
```

#####5. Mapping DE genes to individual pathways

We can map genes that show statistically significant changes to individual pathways. As an example, we can take DNA replication pathway and see which of the DE genes from EdgeR is present.

```
pv_replication <- pathview(gene.data = DE_foldchange,
                           gene.idtype = "KEGG",
                           pathway.id = combine_ids[3],
                           species = "ath",
                           out.suffix = "DNA_replication",
                           keys.align = "y",
                           kegg.native = T,
                           match.data = T,
                           key.pos = "topright")
```

See how the output look like.

```
head(pv_replication)
```

Gapminder data set
http://combine-australia.github.io/RNAseq-R/data/gapminder-FiveYearData.csv
Gretchen Poortinga
Clare Anstead
Oana Sanislav
Sarah Preston
Sui Lay
Ashley Dou
Sam Manna
Lev Kats
Celia Vandestadt
Lauren Brown

Trent Perry
Elizabeth Christie
Lucas Dent
Sathana Dushyanthen
Minyu Wang
Rachel Chiaroni-Clarke
Hansen Kosasih
Jessica Beach
Courtney van Geelen
clare slaney
Shatha AbuHammad
Kara Britt
Nicole Van Berge
Eric Kusnadi
jian kang
Namitha Mohandas
Miriam Fanjul
Andreea Waltmann
Cassandra Koh

- cont.matrix <- makeContrasts(B.PregVsLac = basal.pregnant - basal.lactate,
- L.PregVsLac = luminal.pregnant - luminal.lactate,
- levels=design)

```
fit <- lmFit(v)

fit.cont <- contrasts.fit(fit,cont.matrix)

fit.cont <- eBayes(fit.cont)
fit.cont
summary(decideTests(fit.cont))

library(org.Mm.eg.db)
ann <-
select(org.Mm.eg.db,keys=rownames(fit.cont),columns=c("ENTREZID","SYMBOL","GENENAME"))
# Have a look at the annotation
head(ann)

table(ann$ENTREZID==rownames(fit.cont))

fit.cont$genes <- ann
```