GeoQuery

Manpreet S. Katari

GEOquery

GEOQuery is a handy R package that allows one to connect to the GEO database and retrieve different transcriptomice datasets. In this example we will focus on microarray experiments.

Install Bioconductor packages

Now we can download some packages to use. We will use bioconductor as the source for our packages.

```r
if (!requireNamespace("BiocManager", quietly = TRUE))
    install.packages("BiocManager")
BiocManager::install("GEOquery", version = "3.8")
```

Setting up the environment/Project

It is highly recommended that whenever you start analyzing a new dataset, you first create a fresh R-studio project for it. This way all files you will created, download, or need as input, will be (should be) in the same place.

Remember that once package is installed it needs to be loaded

```r
library(GEOquery)
```

```
## Warning: package 'GEOquery' was built under R version 3.5.2

## Loading required package: Biobase

## Loading required package: BiocGenerics

## Loading required package: parallel

##
## Attaching package: 'BiocGenerics'

## The following objects are masked from 'package:parallel':
##
##     clusterApply, clusterApplyLB, clusterCall, clusterEvalQ,
##     clusterExport, clusterMap, parApply, parCapply, parLapply,
##     parLapplyLB, parRapply, parSapply, parSapplyLB

## The following objects are masked from 'package:stats':
##
##     IQR, mad, sd, var, xtabs

## The following objects are masked from 'package:base':
##
##     anyDuplicated, append, as.data.frame, basename, cbind,
##     colMeans, colnames, colSums, dirname, do.call, duplicated,
##     eval, evalq, Filter, Find, get, grep, grepl, intersect,
##     is.unsorted, lapply, lengths, Map, mapply, match, mget, order,
##     paste, pmax, pmax.int, pmin, pmin.int, Position, rank, rbind,
##     Reduce, rowMeans, rownames, rowSums, sapply, setdiff, sort,
##     table, tapply, union, unique, unsplit, which, which.max,
##     which.min
```

```
## Welcome to Bioconductor
##
##     Vignettes contain introductory material; view with
##     'browseVignettes()'. To cite Bioconductor, see
##     'citation("Biobase")', and for packages 'citation("pkgname")'.
```

```
## Setting options('download.file.method.GEOquery'='auto')
```

```
## Setting options('GEOquery.inmemory.gpl'=FALSE)
```

The getGEO() function in the GEOquery package retrieves meta data from the GEO website with not only the meta data, but also the experimental data points. Below are some useful commands in the package.

```
gds<-getGEO("GDS2084")
```

```
## File stored at:
```

```
## /var/folders/q3/yzvt5fr95056x4frykp7p6kh0000gn/T//RtmpCWMFgW/GDS2084.soft.gz
```

```
## Parsed with column specification:
## cols(
##   ID_REF = col_character(),
##   IDENTIFIER = col_character(),
##   GSM114841 = col_double(),
##   GSM114844 = col_double(),
##   GSM114845 = col_double(),
##   GSM114849 = col_double(),
##   GSM114851 = col_double(),
##   GSM114854 = col_double(),
##   GSM114855 = col_double(),
##   GSM114834 = col_double(),
##   GSM114842 = col_double(),
##   GSM114843 = col_double(),
##   GSM114847 = col_double(),
##   GSM114848 = col_double(),
##   GSM114850 = col_double(),
##   GSM114852 = col_double(),
##   GSM114853 = col_double()
## )
```

```
Meta(gds)
```

```
## $channel_count
## [1] "1"
##
## $dataset_id
## [1] "GDS2084" "GDS2084"
##
## $description
## [1] "Analysis of omental adipose tissues of morbidly obese patients with polycystic ovary syndrome (
## [2] "control"
## [3] "polycystic ovary syndrome"
##
```

```
## $email
## [1] "geo@ncbi.nlm.nih.gov"
##
## $feature_count
## [1] "22283"
##
## $institute
## [1] "NCBI NLM NIH"
##
## $name
## [1] "Gene Expression Omnibus (GEO)"
##
## $order
## [1] "none"
##
## $platform
## [1] "GPL96"
##
## $platform_organism
## [1] "Homo sapiens"
##
## $platform_technology_type
## [1] "in situ oligonucleotide"
##
## $pubmed_id
## [1] "17062763"
##
## $ref
## [1] "Nucleic Acids Res. 2005 Jan 1;33 Database Issue:D562-6"
##
## $reference_series
## [1] "GSE5090"
##
## $sample_count
## [1] "15"
##
## $sample_id
## [1] "GSM114841,GSM114844,GSM114845,GSM114849,GSM114851,GSM114854,GSM114855"
## [2] "GSM114834,GSM114842,GSM114843,GSM114847,GSM114848,GSM114850,GSM114852,GSM114853"
##
## $sample_organism
## [1] "Homo sapiens"
##
## $sample_type
## [1] "RNA"
##
## $title
## [1] "Polycystic ovary syndrome: adipose tissue"
##
## $type
## [1] "Expression profiling by array" "disease state"
## [3] "disease state"
##
## $update_date
## [1] "Mar 21 2007"
##
## $value_type
## [1] "count"
##
## $web_link
## [1] "http://www.ncbi.nlm.nih.gov/geo"
```

```
head(Table(gds))
```

```
##      ID_REF IDENTIFIER GSM114841 GSM114844 GSM114845 GSM114849 GSM114851
## 1 1007_s_at    MIR4640     222.6     252.7     219.3     258.9     239.0
## 2   1053_at       RFC2      35.5      24.5      23.4      31.4      20.6
## 3    117_at      HSPA6      41.5      53.3      31.3      43.0      65.5
## 4    121_at       PAX8     229.8     419.6     274.5     227.1     271.6
## 5 1255_g_at     GUCA1A      14.3      13.0      29.6      16.3       4.6
## 6   1294_at    MIR5193     150.8     116.0      89.9     125.1      89.2
##   GSM114854 GSM114855 GSM114834 GSM114842 GSM114843 GSM114847 GSM114848
## 1     286.0     230.1     197.1     254.4     296.5     171.1     268.9
## 2      26.1      24.3      26.9      31.4      27.1      25.9      40.5
## 3      39.6      68.5      46.9      61.7      93.7      68.5      79.6
## 4     428.7     333.4     221.1     291.5     399.8     307.1     364.8
## 5      10.7       7.8       2.4      13.9      24.7       3.8      14.3
## 6      79.5      85.0      96.6     100.9     111.4      81.3     142.6
##   GSM114850 GSM114852 GSM114853
## 1     251.2     301.9     234.3
## 2      22.2      24.6      31.3
## 3      40.0      43.2      53.4
## 4     326.1     387.2     400.9
## 5       1.9      12.0      11.5
## 6      82.9     107.1     100.3
```

```
Columns(gds)
```

```
##       sample             disease.state
## 1  GSM114841                   control
## 2  GSM114844                   control
## 3  GSM114845                   control
## 4  GSM114849                   control
## 5  GSM114851                   control
## 6  GSM114854                   control
## 7  GSM114855                   control
## 8  GSM114834 polycystic ovary syndrome
## 9  GSM114842 polycystic ovary syndrome
## 10 GSM114843 polycystic ovary syndrome
## 11 GSM114847 polycystic ovary syndrome
## 12 GSM114848 polycystic ovary syndrome
## 13 GSM114850 polycystic ovary syndrome
## 14 GSM114852 polycystic ovary syndrome
## 15 GSM114853 polycystic ovary syndrome
##                                                                     description
## 1        Value for GSM114841: EP3_adipose_control; src: Omental adipose tissue
## 2       Value for GSM114844: EP23_adipose_control; src: Omental adipose tissue
## 3  Value for GSM114845: EP31_adipose_control_rep1; src: Omental adipose tissue
## 4       Value for GSM114849: EP37_adipose_control; src: Omental adipose tissue
## 5       Value for GSM114851: EP49_adipose_control; src: Omental adipose tissue
## 6       Value for GSM114854: EP69_adipose_control; src: Omental adipose tissue
## 7       Value for GSM114855: EP71_adipose_control; src: Omental adipose tissue
## 8      Value for GSM114834: EP1_adipose_pcos_rep1; src: Omental adipose tissue
## 9          Value for GSM114842: EP10_adipose_pcos; src: Omental adipose tissue
## 10         Value for GSM114843: EP18_adipose_pcos; src: Omental adipose tissue
## 11         Value for GSM114847: EP32_adipose_pcos; src: Omental adipose tissue
## 12         Value for GSM114848: EP34_adipose_pcos; src: Omental adipose tissue
## 13         Value for GSM114850: EP47_adipose_pcos; src: Omental adipose tissue
## 14         Value for GSM114852: EP55_adipose_pcos; src: Omental adipose tissue
## 15         Value for GSM114853: EP66_adipose_pcos; src: Omental adipose tissue
```

Notice that the Columns(gds) command provides information regarding the different factors and the leves involved in the dataset. Also the Table(gds) command retrieves all the normalized values. In the case where normalizing the data from the raw data files is too cumbersome, this is an easy alternative.

Below we download meta information about a specific sample and using the same commands, we retrieve similar type of information.

```
gsm<-getGEO("GSM114841")
```

```
## File stored at:
```

```
## /var/folders/q3/yzvt5fr95056x4frykp7p6kh0000gn/T//RtmpCWMFgW/GSM114841.soft
```

```
Meta(gsm)
```

```
## $biomaterial_provider_ch1
## [1] "Ramón y Cajal Hospital, Madrid, Spain"
##
## $channel_count
## [1] "1"
##
## $characteristics_ch1
## [1] "Morbidly obese control subject"
##
## $contact_address
## [1] "ARTURO DUPERIER"
##
## $contact_city
## [1] "MADRID"
##
## $contact_country
## [1] "Spain"
##
## $contact_email
## [1] "bperal@iib.uam.es"
##
## $contact_fax
## [1] "34 91 5854401"
##
## $contact_institute
## [1] "INSTITUTO DE INVESTIGACIONES BIOMEDICAS, CSIC-UAM"
##
## $contact_name
## [1] "BELEN,,PERAL"
##
```

```
## $contact_phone
## [1] "34 91 5854478"
##
## $contact_state
## [1] "MADRID"
##
## $`contact_zip/postal_code`
## [1] "28029"
##
## $data_processing
## [1] "MAS 5.0, scaled to 100 and RMA"
##
## $data_row_count
## [1] "22283"
##
## $description
## [1] "Total RNA was extracted from omental  adipose tissue from a control subject"
##
## $geo_accession
## [1] "GSM114841"
##
## $label_ch1
## [1] "Biotin"
##
## $last_update_date
## [1] "Jun 16 2006"
##
## $molecule_ch1
## [1] "total RNA"
##
## $organism_ch1
## [1] "Homo sapiens"
##
## $platform_id
## [1] "GPL96"
##
## $series_id
## [1] "GSE5090"
##
## $source_name_ch1
## [1] "Omental adipose tissue"
##
## $status
## [1] "Public on Jun 17 2006"
##
## $submission_date
## [1] "Jun 16 2006"
##
## $supplementary_file
## [1] "ftp://ftp.ncbi.nlm.nih.gov/geo/samples/GSM114nnn/GSM114841/suppl/GSM114841.CEL.gz"
## [2] "ftp://ftp.ncbi.nlm.nih.gov/geo/samples/GSM114nnn/GSM114841/suppl/GSM114841.EXP.gz"
##
## $taxid_ch1
## [1] "9606"
```

```
##
## $title
## [1] "EP3_adipose_control"
##
## $type
## [1] "RNA"
```

```r
Columns(gsm)
```

```
##              Column
## 1            ID_REF
## 2             VALUE
## 3          ABS_CALL
## 4 Detection p-value
##
## 1
## 2
## 3 Presence/absence of gene transcript in sample; the call in an absolute analysis that indicates if
## 4
```

```r
head(Table(gsm))
```

```
##            ID_REF VALUE ABS_CALL Detection p-value
## 1 AFFX-TrpnX-M_at   1.3        A          0.963431
## 2 AFFX-TrpnX-5_at   2.6        A          0.672921
## 3 AFFX-TrpnX-3_at   0.5        A          0.910522
## 4  AFFX-ThrX-M_at   4.3        A          0.631562
## 5  AFFX-ThrX-5_at   1.9        A          0.897835
## 6  AFFX-ThrX-3_at   3.4        A          0.852061
```

Differential Expression

Now that we can retrieve all the necessary information, we can use the functions we have been writing to perform a t-test and calculate fold change for all genes.

```r
# save all values to a dataframe
alldata = Table(gds)
allsamples = Columns(gds)

# retrieve only columns with sample names
expdata = alldata[,as.character(allsamples$sample)]

# if you run summary you would notice that each columns appears
# to be a character vector. let's make them numeric
expdata = apply(expdata, 2, as.numeric)

#let's put the probe/gene names back on the rownames
rownames(expdata) = alldata[,1]

genemean= apply(expdata, 1, tapply, allsamples$disease.state, mean)
generatio = log2(genemean["polycystic ovary syndrome",]/genemean["control",])

# create a function we can apply to each row, return only the p-value
dottest <- function(x, expgroups) {
  return (t.test(as.numeric(x) ~ expgroups)$p.value)
}

# apply the function to each row
ttestpvalues = apply(expdata, 1, dottest, allsamples$disease.state)

# perform fdr correction on the p-values
ttestpvaluesfdr = p.adjust(ttestpvalues, method="fdr")

#filter the dataset
DiffGenes=names(which( (abs(generatio) > log2(1.5)) & ttestpvaluesfdr < 0.05 ))

length(DiffGenes)
```

```
## [1] 0
```

None of the FDR corrected p-values pass the cutoff. So let's just take the original p-values and the logratio cutoffs

```
#filter the dataset
DiffGenes=names(which( (abs(generatio) > log2(1.5)) & ttestpvalues < 0.05 ))

length(DiffGenes)
```

```
## [1] 286
```

“‘

Compare the results to running the analysis using SimpleAffy package. What is the major difference between the methods? Why should this cause such a different in number of differentially expressed genes?