

<https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE4170>  
 (https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE4170)

GEO has an integrated R script generator, so I installed R for Jupyter on my Mac (I already had R and Jupyter Notebooks, for Python, so it was pretty quick to do so:

- <http://www.storybench.org/install-r-jupyter-notebook/> (<http://www.storybench.org/install-r-jupyter-notebook/>)
- <https://medium.com/@kyleake/how-to-install-r-in-jupyter-with-irkernel-in-3-steps-917519326e41> (<https://medium.com/@kyleake/how-to-install-r-in-jupyter-with-irkernel-in-3-steps-917519326e41>)
- <https://stackoverflow.com/questions/39008069/r-and-python-in-one-jupyter-notebook> (<https://stackoverflow.com/questions/39008069/r-and-python-in-one-jupyter-notebook>)
- <https://bioconductor.org/packages/release/bioc/html/Biobase.html> (<https://bioconductor.org/packages/release/bioc/html/Biobase.html>)
- <https://www.bioconductor.org/packages/release/bioc/html/GEOquery.html> (<https://www.bioconductor.org/packages/release/bioc/html/GEOquery.html>)

First few lines of the GEO2R script, from GEO, involve loading R packages. In terminal, I had to install these packages in my conda environment before they would load here:

```
In [1]: suppressMessages( library( limma ) )
```

```
In [17]: system( "conda env list", intern=TRUE )
```

```
'# conda environments:  ' #  'base                /miniconda3'
'sbmi5330              /miniconda3/envs/sbmi5330'
'test-cf-r             * /miniconda3/envs/test-cf-r' ''
```

```
In [19]: system("conda list", intern=TRUE)
```

```

'# packages in environment at /miniconda3/envs/test-cf-r: ' '#
'# Name          Version          Build Channel'
'_r-mutex        1.0.0            anacondar_1 '
'appnope         0.1.0            py37_0 '
'attrs          19.1.0           py37_1 ' 'backcall      0.1.0          py37_0 '
'bleach         3.1.0            py37_0 ' 'bwidget        1.9.11         1 '
'bzip2          1.0.6            h1de35cc_1002 conda-forge'
'ca-certificates 2019.1.23        0 '
'cairo          1.14.12          h9d4d9ac_1005 conda-forge'
'cctools        895              h7512d6f_0 conda-forge'
'certifi        2019.3.9         py37_0 '
'clang          4.0.1            h662ec87_0 conda-forge'
'clang_osx-64   4.0.1            h1ce6c1d_11 '
'clangxx        4.0.1            hc9b4283_0 '
'clangxx_osx-64 4.0.1            h22b1bf0_11 '
'compiler-rt    4.0.1            h5487866_0 conda-forge'
'curl           7.64.1           h22ea746_0 conda-forge'
'dbus           1.13.6           h90a0687_0 '
'decorator      4.4.0            py37_1 '
'defusedxml     0.5.0            py37_1 '
'entrypoints    0.3              py37_0 '
'expat          2.2.6            h0a44026_0 '
'fontconfig     2.13.1           h1027ab8_1000 conda-forge'
'freetype       2.10.0           h24853df_0 conda-forge'
'gettext        0.19.8.1         h46ab8bc_1002 conda-forge'
'gfortran_osx-64 4.8.5            h22b1bf0_8 conda-forge'
'glib           2.56.2           h67dad55_1001 conda-forge'
'graphite2      1.3.13           h2098e52_1000 conda-forge'
'harfbuzz       1.9.0            h9889186_1001 conda-forge'
'icu            58.2             h0a44026_1000 conda-forge'
'ipykernel      5.1.0            py37h39e3cac_0 '
'ipython        7.4.0            py37h39e3cac_0 '
'ipython_genutils 0.2.0           py37_0 '
'ipywidgets     7.4.2            py37_0 ' 'jedi            0.13.3         py37_0 '
'jinja2         2.10             py37_0 '
'jpeg           9c               h1de35cc_1001 conda-forge'
'jsonschema     3.0.1            py37_0 '
'jupyter        1.0.0            py37_7 '
'jupyter_client 5.2.4            py37_0 '
'jupyter_console 6.0.0           py37_0 '
'jupyter_core   4.4.0            py37_0 '
'krb5           1.16.3           hcfa6398_1001 conda-forge'
'ld64           274.2            h7c2db76_0 conda-forge'
'libcurl        7.64.1           h16faf7d_0 conda-forge'
'libcxx         4.0.1            h579ed51_0 '
'libcxxabi      4.0.1            hebd6815_0 '
'libedit        3.1.20170329     hcfe32e1_1001 conda-forge'
'libffi         3.2.1            h6de7cb9_1006 conda-forge'

```

'libgfortran	3.0.1	0 conda-forge'
'libiconv	1.15	h01d97ff_1005 conda-forge'
'libpng	1.6.36	ha441bb4_1000 conda-forge'
'libsodium	1.0.16	h3efe00b_0 '
'libssh2	1.8.2	hcdc9a53_1 conda-forge'
'libtiff	4.0.10	h79f4b77_1001 conda-forge'
'libxml2	2.9.8	hf14e9c8_1005 conda-forge'
'llvm	4.0.1	hc748206_0 conda-forge'
'llvm-lto-tapi	4.0.1	h6701bc3_0 conda-forge'
'llvm-openmp	4.0.1	hcfea43d_1 '
'make	4.2.1	h1de35cc_2004 conda-forge'
'markupsafe	1.1.1	py37h1de35cc_0 '
'mistune	0.8.4	py37h1de35cc_0 '
'nbconvert	5.4.1	py37_3 '
'nbformat	4.4.0	py37_0 '
'ncurses	6.1	h0a44026_1002 conda-forge'
'notebook	5.7.6	py37_0 '
'openssl	1.1.1b	h1de35cc_1 '
'pandoc	2.2.3.2	0 '
'pandocfilters	1.4.2	py37_1 '
'pango	1.40.14	h57a785b_1003 conda-forge'
'parso	0.3.4	py37_0 '
'pcre	8.41	h0a44026_1003 conda-forge'
'pexpect	4.6.0	py37_0 '
'pickleshare	0.7.5	py37_0 '
'pip	19.0.3	py37_0 '
'pixman	0.34.0	h1de35cc_1003 conda-forge'
'prometheus_client	0.6.0	py37_0 '
'prompt_toolkit	2.0.9	py37_0 '
'ptyprocess	0.6.0	py37_0 '
'pygments	2.3.1	py37_0 '
'pyqt	5.9.2	py37h655552a_2 '
'pysistent	0.14.11	py37h1de35cc_0 '
'python	3.7.2	haf84260_0 '
'python-dateutil	2.8.0	py37_0 '
'pyzmq	17.1.2	py37h1de35cc_0 '
'qt	5.9.7	h468cd18_1 '
'qtconsole	4.4.3	py37_0 '
'r-assertthat	0.2.0	r351hf348343_0 '
'r-base	3.5.1	he1ce5cd_1005 conda-forge'
'r-base64enc	0.1_3	r351h6402f54_4 '
'r-bh	1.66.0_1	r351hf348343_0 '
'r-bindr	0.1.1	r351hf348343_0 '
'r-bindrcpp	0.2.2	r351h32998d9_0 '
'r-cli	1.0.0	r351h6115d3f_0 '
'r-crayon	1.3.4	r351hf348343_0 '
'r-curl	3.2	r351h6402f54_0 '
'r-digest	0.6.15	r351h6402f54_0 '
'r-dplyr	0.7.6	r351h32998d9_0 '

'r-evaluate	0.11	r351hf348343_0	'
'r-fansi	0.2.3	r351h6402f54_0	'
'r-glue	1.3.0	r351h6402f54_0	'
'r-htmltools	0.3.6	r351h32998d9_0	'
'r-httr	1.3.1	r351hf348343_0	'
'r-irdisplay	0.5.0	r351hf348343_0	'
'r-irkernel	0.8.12	r351_0	'
'r-jsonlite	1.5	r351h6402f54_0	'
'r-magrittr	1.5	r351hf348343_4	'
'r-mime	0.5	r351h6402f54_0	'
'r-openssl	1.0.2	r351h46e59ec_1	'
'r-pbdzmq	0.3_3	r351h32998d9_0	'
'r-pillar	1.3.0	r351hf348343_0	'
'r-pkgconfig	2.0.1	r351hf348343_0	'
'r-plogr	0.2.0	r351hf348343_0	'
'r-purrr	0.2.5	r351h6402f54_0	'
'r-r6	2.2.2	r351hf348343_0	'
'r-rcpp	0.12.18	r351h32998d9_0	'
'r-repr	0.15.0	r351hf348343_0	'
'r-rlang	0.2.1	r351h6402f54_0	'
'r-stringi	1.2.4	r351h32998d9_0	'
'r-stringr	1.3.1	r351hf348343_0	'
'r-tibble	1.4.2	r351h6402f54_0	'
'r-tidyr	0.8.1	r351h32998d9_0	'
'r-tidysselect	0.2.4	r351h32998d9_0	'
'r-utf8	1.1.4	r351h6402f54_0	'
'r-uuid	0.1_2	r351h6402f54_4	'
'readline	7.0	hcfe32e1_1001	conda-forge'
'send2trash	1.5.0	py37_0	'
'setuptools	40.8.0	py37_0	'
'sip	4.19.8	py37h0a44026_0	'
'six	1.12.0	py37_0	'
'sqlite	3.26.0	ha441bb4_0	'
'terminado	0.8.1	py37_1	'
'testpath	0.4.2	py37_0	'
'tk	8.6.9	ha441bb4_1001	conda-forge'
'tktable	2.10	h1de35cc_0	'
'tornado	6.0.2	py37h1de35cc_0	'
'traitlets	4.3.2	py37_0	' 'wcwidth 0.1.7 py37_0 '
'webencodings	0.5.1	py37_1	'
'wheel	0.33.1	py37_0	'
'widgetsnbextension	3.4.2	py37_0	'
'xz	5.2.4	h1de35cc_1001	conda-forge'
'zeromq	4.2.5	h0a44026_1	'
'zlib	1.2.11	h1de35cc_1004	conda-forge'

```
In [22]: %R installed.packages()
```

```
Error in parse(text = x, srcfile = src): <text>:1:1: unexpected input
1: %R installed.packages()
  ^
Traceback:
```

```
In [18]: suppressMessages(library(GEOquery))
```

```
Error in library(GEOquery): there is no package called 'GEOquery'
Traceback:
```

```
1. suppressMessages(library(GEOquery))
2. withCallingHandlers(expr, message = function(c) invokeRestart("muffleMessage"))
3. library(GEOquery)
4. stop(txt, domain = NA)
```

```
In [6]: # Version info: R 3.2.3, Biobase 2.30.0, GEOquery 2.40.0, limma 3.26.8
# R scripts generated Thu Mar 28 01:28:39 EDT 2019
```

```
#####
# Differential expression analysis with limma
suppressMessages(library(Biobase))
suppressMessages(library("GEOquery"))
suppressMessages(library(limma))
```

```
Error in library("GEOquery"): there is no package called 'GEOquery'
Traceback:
```

```
1. suppressMessages(library("GEOquery"))
2. withCallingHandlers(expr, message = function(c) invokeRestart("muffleMessage"))
3. library("GEOquery")
4. stop(txt, domain = NA)
```

```
In [ ]: # load series and platform data from GEO
gset <- getGEO("GSE4170", GSEMatrix = TRUE, AnnotGPL = TRUE) #Radich 2006
if (length(gset) > 1) idx <- grep("GPL2029", attr(gset, "names")) else i
dx <- 1
gset <- gset[[idx]]
```

In the web tool, marked the first 21 of 119 samples. The first 6 were labeled "cd" group, the next 15 labeled "ap" group. A third group, "bc" was created by not used for any samples.

The tool limits users to only 10 groups, because it encodes them as single digit integers:



```

In [ ]: # log2 transform
ex <- exprs(gset)
qx <- as.numeric(quantile(ex, c(0., 0.25, 0.5, 0.75, 0.99, 1.0), na.rm=T
))
LogC <- (qx[5] > 100) ||
        (qx[6]-qx[1] > 50 && qx[2] > 0) ||
        (qx[2] > 0 && qx[2] < 1 && qx[4] > 1 && qx[4] < 2)
if (LogC) { ex[which(ex <= 0)] <- NaN
  exprs(gset) <- log2(ex) }

# set up the data and proceed with analysis
sml <- paste("G", sml, sep="") # set group names
fl <- as.factor(sml)
gset$description <- fl
design <- model.matrix(~ description + 0, gset)
colnames(design) <- levels(fl)
fit <- lmFit(gset, design)
cont.matrix <- makeContrasts(G1-G0, levels=design)
fit2 <- contrasts.fit(fit, cont.matrix)
fit2 <- eBayes(fit2, 0.01)
tT <- topTable(fit2, adjust="fdr", sort.by="B", number=250)

#tT <- subset(tT, select=c("ID", "adj.P.Val", "P.Value", "F", "Gene.symbol",
# "Gene.title"))
#reselected columns, to include gene id and GO function...
tT <- subset(tT, select=c("ID", "adj.P.Val", "P.Value", "F", "Gene.symbol",
"Gene.title", "Gene.ID", "GO.Function"))
write.table(tT, file=stdout(), row.names=F, sep="\t")

```

Used a 2nd tab on the GEO2R tool to create a box plot. I think it added a 2nd half to the script for this plot as a result:



```
In [ ]: #####  
#   Boxplot for selected GEO samples  
library(Biobase)  
library(GEOquery)  
  
# load series and platform data from GEO  
  
gset <- getGEO("GSE4170", GSEMatrix = TRUE, getGPL = FALSE)  
if (length(gset) > 1) idx <- grep("GPL2029", attr(gset, "names")) else i  
dx <- 1  
gset <- gset[[idx]]  
  
# group names for all samples in a series  
gsms <- paste0("0000001111111111111111XXXXXXXXXXXXXXXXXXXXXXXXXXXXX",  
               "XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX",  
               "XXXXXXXXXXXXXXXXXXXX")  
sml <- c()  
for (i in 1:nchar(gsms)) { sml[i] <- substr(gsms,i,i) }  
sml <- paste("G", sml, sep=" ") set group names  
  
# eliminate samples marked as "X"  
sel <- which(sml != "X")  
sml <- sml[sel]  
gset <- gset[,sel]  
  
# order samples by group  
ex <- exprs(gset)[ , order(sml)]  
sml <- sml[order(sml)]  
fl <- as.factor(sml)  
labels <- c("cd","ap","bc")  
  
# set parameters and draw the plot  
palette(c("#dfeaf4","#f4dfdf","#f2cb98", "#AABBCC"))  
dev.new(width=4+dim(gset)[[2]]/5, height=6)  
par(mar=c(2+round(max(nchar(sampleNames(gset)))/2),4,2,1))  
title <- paste ("GSE4170", '/', annotation(gset), " selected samples", s  
ep='')  
boxplot(ex, boxwex=0.6, notch=T, main=title, outline=FALSE, las=2, col=f  
l)  
legend("topleft", labels, fill=palette(), bty="n")
```