R Basics using Golub Data

Hopefully at this point you have all downloaded R from the following webpage: cran.r-project.org

Please also download RStudio, which is a user friendly interface for R. There are hundreds, if not thousands of packages that are freely available for you to use for various bioinformatics analyses. To view these packages, simply open the Packages tab on your RStudio sidebar.

Let's start using R by downloading the Golub data set. This is a great data set for learning how to use R. Let's load the Golub data set by using the following commands:

Use the command **library(multtest)** to load the library containing the data. I am leaving it out because the output is large. Note: You will see all of the packages loading when you open the multtest library.

```
data(golub)
```

You will be able to see the two data tables by looking at the **Data** section of the **Global Enviornment** tab.

We can take a look at the loaded data matrices by using the **head** command. This method will show the first 6 rows of each column in a data frame or matrix.

R uses methods to accomplish tasks, the best thing to do is to Google whatever task you are going to accomplish. Plently of people are using R and there is very helpful information available.

Let's look at the data. The golub table contains gene expression values from 3051 genes taken from 38 Leukemia patients. Twenty seven patients are diagnosed as acute lymphoblastic leukemia (ALL) and eleven as acute myeloid leukemia (AML). The golub.gnames table contains information on the gene, including gene index, manufacturing ID, and biological name.

```
head(golub)
```

```
##          [,1]     [,2]     [,3]     [,4]     [,5]     [,6]     [,7]
## [1,] -1.45769 -1.39420 -1.42779 -1.40715 -1.42668 -1.21719 -1.37386
## [2,] -0.75161 -1.26278 -0.09052 -0.99596 -1.24245 -0.69242 -1.37386
## [3,]  0.45695 -0.09654  0.90325 -0.07194  0.03232  0.09713 -0.11978
## [4,]  3.13533  0.21415  2.08754  2.23467  0.93811  2.24089  3.36576
## [5,]  2.76569 -1.27045  1.60433  1.53182  1.63728  1.85697  3.01847
## [6,]  2.64342  1.01416  1.70477  1.63845 -0.36075  1.73451  3.36576
##          [,8]     [,9]    [,10]    [,11]    [,12]    [,13]    [,14]
## [1,] -1.36832 -1.47649 -1.21583 -1.28137 -1.03209 -1.36149 -1.39979
## [2,] -0.50803 -1.04533 -0.81257 -1.28137 -1.03209 -0.74005 -0.83161
## [3,]  0.23381  0.23987  0.44201 -0.39560 -0.62533  0.45181  1.09519
## [4,]  1.97859  2.66468 -1.21583  0.59110  3.26050 -1.36149  0.64180
## [5,]  1.12853  2.17016 -1.21583 -1.10133  2.59982 -1.36149  0.22853
## [6,]  0.96870  2.72368 -1.21583  1.20192  2.83418 -1.36149  1.32744
##        [,15]    [,16]    [,17]    [,18]    [,19]    [,20]    [,21]
## [1,] 0.17628 -1.40095 -1.56783 -1.20466 -1.24482 -1.60767 -1.06221
## [2,] 0.41200 -1.27669 -0.74370 -1.20466 -1.02380 -0.38779 -1.06221
## [3,] 1.09318  0.34300  0.20010  0.38992  0.00641  1.10932  0.21952
## [4,] 2.32621 -1.40095 -1.56783  0.83502 -1.24482 -1.60767 -1.06221
## [5,] 2.34494 -1.40095 -1.56783  0.94532 -1.24482 -1.60767 -1.06221
## [6,] 1.52458 -1.40095 -1.56783 -0.23780 -1.24482 -1.60767 -1.06221
##         [,22]    [,23]    [,24]    [,25]    [,26]    [,27]    [,28]
## [1,] -1.12665 -1.20963 -1.48332 -1.25268 -1.27619 -1.23051 -1.43337
## [2,] -1.12665 -1.20963 -1.12185 -0.65264 -1.27619 -1.23051 -1.18065
## [3,] -0.72267  0.51690  0.28577  0.61937  0.20085  0.29278  0.26624
## [4,]  3.69445  3.70837 -1.48332  2.36698 -1.27619  2.89604  0.71990
## [5,]  3.52458  3.70837 -1.48332  1.79168 -1.27619  2.24892  0.02799
## [6,]  3.25470  2.73916 -1.48332  2.23430 -1.27619  1.83594  1.31110
##         [,29]    [,30]    [,31]    [,32]    [,33]    [,34]    [,35]
## [1,] -1.08902 -1.29865 -1.26183 -1.44434  1.10147 -1.34158 -1.22961
## [2,] -1.08902 -1.05094 -1.26183 -1.25918  0.97813 -0.79357 -1.22961
## [3,] -0.43377 -0.10823 -0.29385  0.05067  1.69430 -0.12472  0.04609
## [4,]  0.29598 -1.29865  2.76869  2.08960  0.70003  0.13854  1.75908
## [5,] -1.08902 -1.29865  2.00518  1.17454 -1.47218 -1.34158  1.55086
## [6,] -1.08902 -1.29865  1.73780  0.89347 -0.52883 -1.22168  0.90832
##         [,36]   [,37]    [,38]
## [1,] -0.75919 0.84905 -0.66465
## [2,] -0.71792 0.45127 -0.45804
## [3,]  0.24347 0.90774  0.46509
## [4,]  0.06151 1.30297  0.58186
## [5,] -1.18107 1.01596  0.15788
## [6,] -1.39906 0.51266  1.36249
```

```
head(golub.gnames)
```

```
##      [,1] [,2]
## [1,] "36" "AFFX-HUMISGF3A/M97935_MA_at (endogenous control)"
## [2,] "37" "AFFX-HUMISGF3A/M97935_MB_at (endogenous control)"
## [3,] "38" "AFFX-HUMISGF3A/M97935_3_at (endogenous control)"
## [4,] "39" "AFFX-HUMRGE/M10098_5_at (endogenous control)"
## [5,] "40" "AFFX-HUMRGE/M10098_M_at (endogenous control)"
## [6,] "41" "AFFX-HUMRGE/M10098_3_at (endogenous control)"
##      [,3]
## [1,] "AFFX-HUMISGF3A/M97935_MA_at"
## [2,] "AFFX-HUMISGF3A/M97935_MB_at"
## [3,] "AFFX-HUMISGF3A/M97935_3_at"
## [4,] "AFFX-HUMRGE/M10098_5_at"
## [5,] "AFFX-HUMRGE/M10098_M_at"
## [6,] "AFFX-HUMRGE/M10098_3_at"
```

You will also notice that there is a vector under the **values** tab called **golub.cl**, this is a vector consisting of a binary classification of ALL(0) vs. ALL(1) patients.

We should take look at some of the data in **golub**

Now we can check the number of rows(genes) and columns(patients):

```
nrow(golub)
```

```
## [1] 3051
```

```
ncol(golub)
```

```
## [1] 38
```

Suppose we are interested in the 1042 gene in the golub set, we can find out more information about that gene by checking the golub.gnames matrix:

```
golub.gnames[1042,]
```

```
## [1] "2354"           "CCND3 Cyclin D3" "M92287_at"
```

Maybe it's the other way around and we know the biological name, and we want to search the golub.gnames matrix to find the index number in the golub table:

```
grep("CCND3 Cyclin D3",golub.gnames[,2])
```

```
## [1] 1042
```

Suppose are interested in the second patients gene expression value for this gene:

```
golub[1042,2]
```

```
## [1] 1.52405
```

Now let us look at all of the patients:

```
golub[1042,]
```

```
##  [1]  2.10892  1.52405  1.96403  2.33597  1.85111  1.99391  2.06597
##  [8]  1.81649  2.17622  1.80861  2.44562  1.90496  2.76610  1.32551
## [15]  2.59385  1.92776  1.10546  1.27645  1.83051  1.78352  0.45827
## [22]  2.18119  2.31428  1.99927  1.36844  2.37351  1.83485  0.88941
## [29]  1.45014  0.42904  0.82667  0.63637  1.02250  0.12758 -0.74333
## [36]  0.73784  0.49470  1.12058
```

To make things easier, we should make a factor to discriminate the tumor type by using the golub.cl vector:

```
gol.fac <- factor(golub.cl, levels=0:1, labels = c("ALL","AML"))
```

Now we can print all of the gene expression values of CCND3 Cyclin D3 in only the ALL disease patients:

```
golub[1042,gol.fac=="ALL"]
```

```
##  [1] 2.10892 1.52405 1.96403 2.33597 1.85111 1.99391 2.06597 1.81649
##  [9] 2.17622 1.80861 2.44562 1.90496 2.76610 1.32551 2.59385 1.92776
## [17] 1.10546 1.27645 1.83051 1.78352 0.45827 2.18119 2.31428 1.99927
## [25] 1.36844 2.37351 1.83485
```

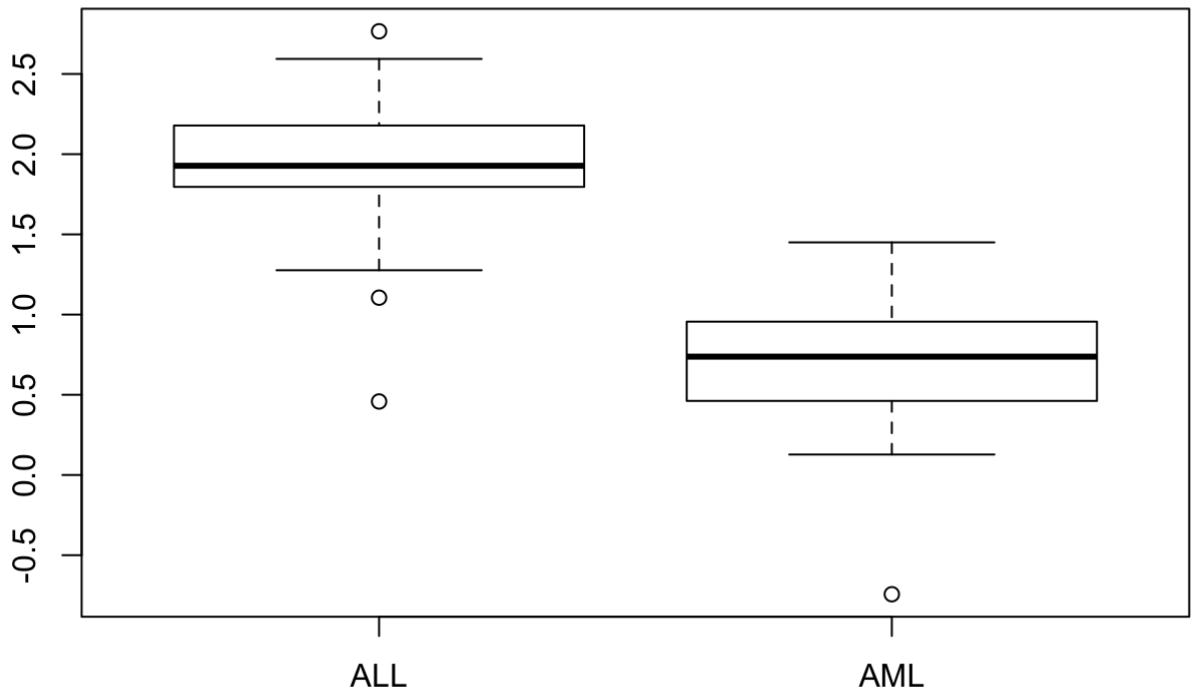
For many types of computations it is very useful to combine a factor with the **apply** functionality. For instance, to compute the mean gene expression over the ALL patients for each of the genes, we may use the following. We will show only the means of the first 6 genes:

```
meanALL <- apply(golub[,gol.fac=="ALL"], 1, mean)
head(meanALL)
```

```
## [1] -1.2715104 -0.9098137  0.2665778  0.9985141  0.6856785  0.7380300
```

Now we should make a plot to view the data, we will look at a boxplot of the gene expression values of CCND3 Cyclin D3 seperated by disease type:

```
boxplot(golub[1042,] ~ gol.fac, method="jitter")
```



Ok, we can definitely see that there is a difference in gene expression values of the CCND3 Cyclin D3 gene in ALL vs. AML patients. Let's do a two sample t-test to test the hypothesis that the means are different:

```
t.test(golub[1042,] ~ gol.fac, var.equal=FALSE)
```

```
##
##  Welch Two Sample t-test
##
## data:  golub[1042, ] by gol.fac
## t = 6.3186, df = 16.118, p-value = 9.871e-06
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  0.8363826 1.6802008
## sample estimates:
## mean in group ALL mean in group AML
##         1.8938826         0.6355909
```

You can return only the p-value of this test by adding the p.value command to the end:

```
t.test(golub[1042,] ~ gol.fac, var.equal=FALSE)$p.value
```

```
## [1] 9.870821e-06
```

The p-value is very low, much lower than the .05 threshold for statistical signifiance level. Therefore we can reject the null hypothesis the gene expression values of the CCND3 Cyclin D3 gene are the same in ALL vs. AML patients.

Exercise:

Use the apply method described above with the t.test, run the t.test on every gene in the golub data set and make a list (use the order() method) of the genes with the most differential expression between the ALL and AML disease state. What are the top 5 genes?

Author: Ben Glass

Reference: https://cran.r-project.org/doc/contrib/Krijnen-IntroBioInfStatistics.pdf (https://cran.r-project.org/doc/contrib/Krijnen-IntroBioInfStatistics.pdf)