BIOS 7659 Homework 1

Tim Vigers

20 September 2020

# Case Studies

## Case Study 3

The aim of case study 3 is to identify subgroups of patients using cluster analysis and then build a model to differentiate between the clusters. Samples are taken from 60 patients with B-cell lymphomas.

### Experimental Design 1

Half of the samples are labeled with Cy3 and the other half with Cy5. Then one Cy3-labeled sample and one Cy5-labeled sample are hybridized to 30 different two-color arrays.

The major advantage of this design is reduced cost, because it only requires 30 arrays compared to the other two designs’ 60. However, there are serious issues with this design that outweigh the cost benefits. First, it is not clear how samples will be selected for each label or how they will be distributed across the arrays. There is a possibility for selection-based dye bias here, for example if the investigator tends to favor labeling cases with Cy3 and controls with Cy5.

Assuming that each array has a sample from one case and one control, and that the Cy3-labeled samples are even split between cases and controls, this is a balanced block design. Balanced block designs work well for performing direct comparisons between two groups, but do not work for cluster analyses. Because there is no reference sample, the investigator will not be able to make comparisons between samples on different arrays, due to natural variation between arrays. Without the balanced block design assumption, it might not even be possible to compare cases to controls.

### Experimental Design 2

Samples from each patient are labeled with Cy3 and hybridized to 60 two-color arrays. A universal reference sample labeled with Cy5 is also hybridized to each array.

This is design is obviously more expensive than Design 1, because it require twice the number of arrays and preparation of the global reference sample. However, it has some significant advantages compared to Design 1.

By including the same reference sample on each array, any sample can be compared to any other sample, and the analysis does not depend on comparing two specific groups. This makes the reference design particularly advantageous for cluster analysis and developing models that include information aside from group membership (e.g. disease stage of the patient).

### Experimental Design 3

Samples from each patient are hybridized to 60 different Affymetrix arrays.

This design has many of the same positives as Design 2, although it could potentially be more expensive than using “in-house” arrays made by the lab. Affymetrix arrays are generally comparable between samples, particularly if it is assumed that all the samples were taken at the same timepoint. Any differential hybridization can be dealt with using standard normalization techniques. One potential advantage over Design 2 is that it would be easier to compare results across labs if you want external validation of the model. This could potentially be done using Design 2, but only if both labs use the same reference sample. Finally, this design requires less sample because there is only one aliquot required per plate.

## Case Study 4

Yeast was allowed to reproduce across seven timepoints, and the investigators are interested in identifying genes with similar expression profiles across time.

### Experimental Design 1

Each sample is hybridized to a single Affymetrix array.

Affymetrix chips are generally less favorable for time-series designs, especially with only one sample per plate. For example, if one of the arrays fluoresces more than the others, then it would be difficult to tell whether this is due to time-related changes in gene expression or is an artifact of differential hybridization. It’s possible that adjusting the data for background fluorescence would help address this problem, but it would be better to have more replicates if possible.

### Experimental Design 2

Samples from timepoints 1 - 6 are labeled with Cy3 and each is hybridized an array along with a Cy5-labeled reference sample from time 0.

This design is essentially the same as the reference design from the previous section, and would allow for valid comparisons between timepoints. Using time 0 as a reference should also allow for adjustment for differential hybridization (“bright” or “dark” chips). However, there could be problems with dye bias because time 0 is always labeled with Cy5.

An additional benefit to this design is that it uses 1 less chip than the other two and will therefore be less expensive. However, using time 0 as a reference an all the arrays requires a large sample at baseline.

### Experimental Design 3

Each sample is labeled once with Cy3 and once with Cy5, and are hybridized to 7 arrays in a loop design.

A major benefit of the loop design is that it only requires n arrays where n = the number of samples, although it does require two aliquots per sample. It allows each sample to be directly compared while adjusting for sample distributions, array effects, etc. However, comparisons between samples that are not adjacent in the loop require modeling the indirect effects of arrays that are in between the arrays of interest, which can be complex.

Compared to Design 2, this approach has the benefit that each sample is measured twice on two different arrays, which helps account for variability between arrays. The major disadvantages are that it requires an additional array (more expensive), more sample, and the loop could be broken if one array doesn’t work correctly.

# Sample Size Calculations

The sample size calculations described here assume that gene expression levels are approximately normally distributed within both the treated and non-treated groups. The null hypothesis that a gene is expressed equally in the two groups will be rejected based on a t test at significance level . This level represents the probability of a type 1 error (false positive), and is more strict than the common in order to account for the large number of genes that will be compared. Thus with a 20,000 probe set we would expect approximately 2 genes to be incorrectly identified as differentially expressed.

Statistical power is the probability of correctly rejecting the null hypothesis when the mean expression level of a gene is different between the two groups by . For these calculations we assume that a meaningful difference in standardized expression is on the base 2 logarithmic scale (a twofold difference), and consider two reasonable estimates of the standard deviation (SD) of gene expression for this study, and .

Based on these assumptions, we can solve the equation

for n. Calculations were done using the pwr package (Champely 2020) in the R programming language version 4.0.2. Cost reflects a per-array price of $1,000 and does not include reagents, technician salary, etc.

# 4 levels of power  
power\_levels <- seq(0.8,0.95,by = 0.05)  
sigmas <- c(0.5,0.25)  
# Apply  
ns <- apply(expand.grid(power\_levels,sigmas),1,function(x){  
 p\_level <- as.numeric(x[1])  
 sig <- as.numeric(x[2])  
 p <- pwr.t.test(d=1/sig,sig.level = 0.0001,power = p\_level)  
 group\_n <- ceiling(p$n)  
 tot\_n <- group\_n\*2  
 cost <- paste0("$",format(tot\_n\*1000,big.mark = ","))  
 return(c(p\_level,sig,tot\_n,group\_n,cost))  
})  
ns <- t(ns)  
colnames(ns) <- c("Power","SD","Total n","n per Group","Cost")  
kable(ns)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Power | SD | Total n | n per Group | Cost |
| 0.8 | 0.5 | 30 | 15 | $30,000 |
| 0.85 | 0.5 | 32 | 16 | $32,000 |
| 0.9 | 0.5 | 36 | 18 | $36,000 |
| 0.95 | 0.5 | 40 | 20 | $40,000 |
| 0.8 | 0.25 | 14 | 7 | $14,000 |
| 0.85 | 0.25 | 14 | 7 | $14,000 |
| 0.9 | 0.25 | 16 | 8 | $16,000 |
| 0.95 | 0.25 | 16 | 8 | $16,000 |

# Sample Size Comparisons

## a)

Using pwr.t.test, what is the sample size needed based on , fold change of 2 () and standard deviation of to achieve power of at least 0.8 or 0.95?

a <- 0.001  
delta <- 1  
sd <- 0.5  
pwr.t.test(d=delta/sd,sig.level = a,power = 0.8)

##   
## Two-sample t test power calculation   
##   
## n = 11.3292  
## d = 2  
## sig.level = 0.001  
## power = 0.8  
## alternative = two.sided  
##   
## NOTE: n is number in \*each\* group

pwr.t.test(d=delta/sd,sig.level = a,power = 0.95)

##   
## Two-sample t test power calculation   
##   
## n = 14.99323  
## d = 2  
## sig.level = 0.001  
## power = 0.95  
## alternative = two.sided  
##   
## NOTE: n is number in \*each\* group

Under the assumptions above, one would need 12 samples per group to achieve power of at least 0.8, and 15 per group to achieve power of at least 0.95.

## b)

As in part a), determine the sample size needed, but with a FDR of 0.05 instead. This requires an additional parameter, , which is the proportion of true null hypotheses. Here we assume that 97.5% of the null hypotheses are true.

power.t.test.FDR(sd=sd,delta = delta,FDR.level = 0.05,  
 power = 0.8,pi0 = 0.975)

##   
## Two-sample t test power calculation   
##   
## n = 11.20745  
## delta = 1  
## sd = 0.5  
## FDR.level = 0.05  
## power = 0.8  
## alternative = two.sided  
##   
## NOTE: n is number in \*each\* group

power.t.test.FDR(sd=sd,delta = delta,FDR.level = 0.05,  
 power = 0.95,pi0 = 0.975)

##   
## Two-sample t test power calculation   
##   
## n = 14.53779  
## delta = 1  
## sd = 0.5  
## FDR.level = 0.05  
## power = 0.95  
## alternative = two.sided  
##   
## NOTE: n is number in \*each\* group

For a false discovery rate of 0.05 (assuming 97.5% of the null hypotheses are true), we would need 12 in each group to achieve power of at least 0.8. To achieve power of at least 0.95 we would need 15 in each group. So with the assumption of , this calculation produces the same results as the simple t test. However, if we increase the number of genes we expect to be differentially expressed to 10% (), then we get less conservative estimates for sample size:

power.t.test.FDR(sd=sd,delta = delta,FDR.level = 0.05,  
 power = 0.8,pi0 = 0.90)

##   
## Two-sample t test power calculation   
##   
## n = 8.872857  
## delta = 1  
## sd = 0.5  
## FDR.level = 0.05  
## power = 0.8  
## alternative = two.sided  
##   
## NOTE: n is number in \*each\* group

power.t.test.FDR(sd=sd,delta = delta,FDR.level = 0.05,  
 power = 0.95,pi0 = 0.90)

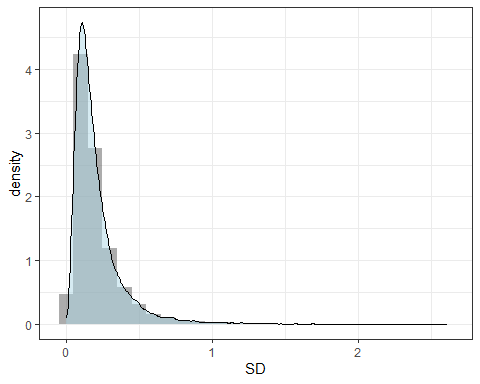
##   
## Two-sample t test power calculation   
##   
## n = 11.81863  
## delta = 1  
## sd = 0.5  
## FDR.level = 0.05  
## power = 0.95  
## alternative = two.sided  
##   
## NOTE: n is number in \*each\* group

These calculations suggest that only 9 people are needed in each group for power of 0.8, and 12 in each group for at least 0.95 power. So the more genes that are different between the two groups, the lower sample size you need to keep FDR at a constant level. If you wanted to be more conservative in your sample size estimate, you could simply pick a relatively high .

## c)

### Plot the pooled standard deviation for each gene:

sds <- read.delim("./sdvalues.txt",sep = " ",header = F)  
ggplot(sds,aes(x=V2)) +   
 geom\_histogram(aes(y=..density..),binwidth = 0.1,alpha = 0.5)+  
 geom\_density(alpha=.5, fill="light blue") +  
 theme\_bw() + xlab("SD")



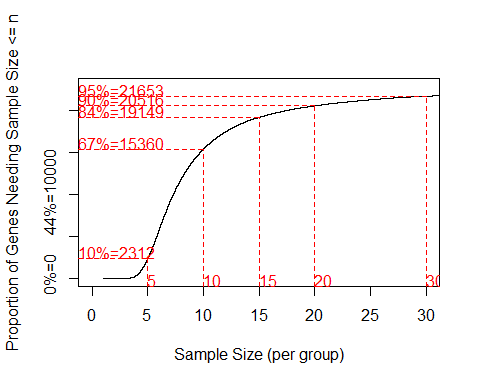
The SD for these genes tends to be around 0.25, although the distribution is highly right-skewed.

### Examine the sample size based on these standard deviations:

#### 80% Power

s <- ssize(sds$V2,delta,a,0.8)

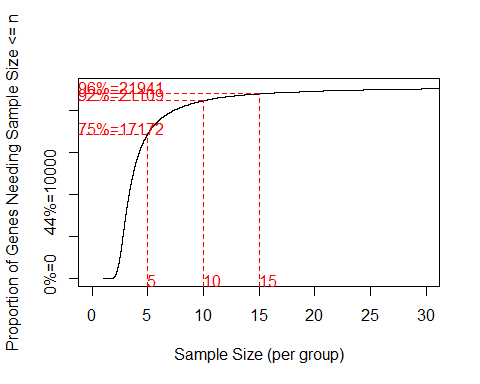
ssize.plot(s,marks = c(5,10,15,20,30),xlim=c(0,30))



The above calculations are still assuming that SD () is equal between the two groups. Based on this pilot data, it appears that a sample size of 20 per group would provide sufficient power to detect a two-fold change for 90% of the genes. However, the ssize() function implements a Bonferonni correction by default, which feels particularly conservative given our level of 0.001.

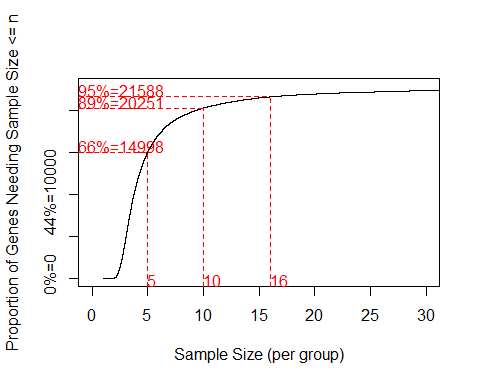
Assuming that this relatively low already accounts for multiple comparisons, we get:

s <- ssize(sds$V2,delta,a,0.8,alpha.correct = "None")



As expected, with no additional p value correction, the required sample size for power of at least 0.8 is lower. To detect a difference in 95% of genes you would only need 15 per group.

#### 95% Power



Again with no additional p value correction, the required sample size for power of at least 0.95 to detect a difference in 95% of genes is 16 per group.

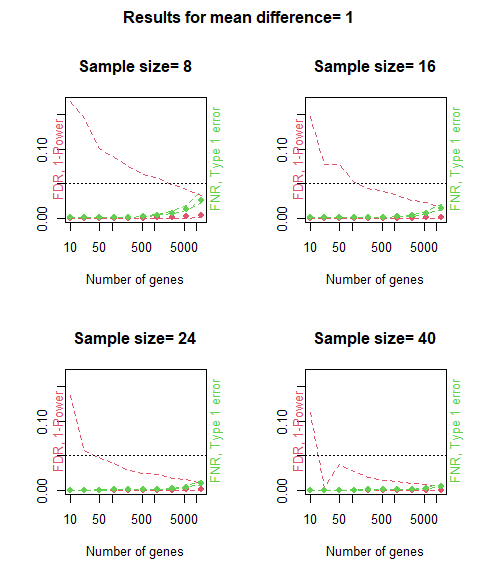
## d)

First create a samr object:

arrays <- read.table("./arraydata.txt",row.names = 1)  
x <- as.matrix(arrays)  
data = list(x=x,y=c(rep(1,4),rep(2,4)), geneid=row.names(x),   
 genenames = row.names(x), logged2 = TRUE)  
samr.obj = samr(data, resp.type="Two class unpaired", nperms=500,  
 assay.type="array")

Next assess sample size using samr.assess.samplesize.plot():

ss\_obj <- samr.assess.samplesize(samr.obj,data,dif = 1)  
invisible(samr.assess.samplesize.plot(ss\_obj))



These plots show how false discovery rate (FDR) and false negative rate change as the number of differentially expressed genes increases. Based on the plot it appears that for approximately 20 different genes, a sample size of 40 (20 per group) would be appropriate for this study (to achieve FDR < 0.05).

The plots produced by the samr package are a little more confusing than the others. The left vertical axis represents both FDR and 1-power, so to evaluate power of 0.8 you have to accept an FDR of 0.2. So when using this package it makes more sense to evaluate at power 0.95 and FDR of 0.05.

## e)

Generally speaking I would say that these methods agree with one another fairly well. There are minor differences and the samr package approach appears to be slightly more conservative, but the sample size estimates were at least in the same general range of 15 - 20 per group. Also, aside from the pwr.t.test() function, all of these methods require additional assumptions or decisions.

power.t.test.FDR() requires the number of truly null hypotheses, which may be difficult estimate particularly in populations that have not been studied yet. Assuming you are able to make a resonable guess though, the sample size output is straightforward.

The ssize() function requires preliminary data about the standard deviation of each gene’s expression. If you have this information the package produces useful plots showing what percentage of the genes you would be able to detect a difference in given the other sample size parameters. If you have pilot data on every gene this is a nice approach because it doesn’t require you to guess how many genes actually will be differentially expressed.

samr.assess.samplesize() requires the most pilot data out of any of these methods, and also produces the most confusing output. It does not allow the user to the select a desired level of power and FDR separately, but instead sets FDR equal to 1-power by design. Also, like power.t.test.FDR() it requires some knowledge about the number of truly non-null (or truly null) genes. Perhaps the most problematic aspect of this package is that the plots change drastically depending on the random seed, which makes them difficult to evaluate. This method would likely be my last choice for all of the reasons above.

|  |  |  |
| --- | --- | --- |
| Method | 80% Power | 95% Power |
| pwr.t.test() | 12 | 15 |
| power.t.test.FDR() | 9-12 | 12-15 |
| ssize() | 15 | 16 |
| samr.assess.samplesize() | NA | 20 |