Quizzes due May 13, 2021 23:01 +03

Assignment Setup

The `bladderbatch` dataset from Bioconductor is a collection of gene expression data on bladder cancers from 5 different batches.

```
if (!requireNamespace("BiocManager", quietly = TRUE))
    install.packages("BiocManager")

BiocManager::install("bladderbatch")

library(bladderbatch)
data(bladderdata)

# Get the expression data
edata = exprs(bladderEset)
# Get the pheno data
pheno = pData(bladderEset)
```

Create a reduced dataset containing only batches 1-3. Save the subsetted expression data as `expr` and save the subsetted sample data as `pdata`:

```
ind = which(pheno$batch %in% 1:3)

# subset expression data
expr = edata[ ,ind]

# subset pheno data and redefine factor levels
pdata = data.frame(batch = factor(pheno$batch[ind]),
                   cancer = factor(pheno$cancer[ind]))
```

Question 1

1/1 point (graded)

Make a table of cancer status by batch.

Which of the following are true?
Check ALL correct answers.

☐ All of the cancer samples are in the same batch.

☐ All of the normal samples are in the same batch.

☑ One batch contains only cancer samples.

☑ One batch contains only normal samples.

☐ No batches contain a mix of cancer and normal samples.

✓

Explanation

```
table(pdata$batch, pdata$cancer)
```

```
##
##      Cancer Normal
##    1     11      0
##    2     14      4
##    3      0      4
```

Submit

You have used 2 of 5 attempts

ℹ Answers are displayed within the problem

Question 2

1/1 point (graded)

Compare gene expression in the normal samples from batches 2 and 3. Use this code to extract the relevant subset of the data:

```
index = which(pdata$cancer == "Normal")
expr_norm = edata[ ,index]
batch_norm = factor(pdata$batch[index])
```

Use `rowttests()` from the **genefilter** package to compare expression across the two batches and extract p-values. Then use the `qvalue()` function from the **qvalue** package to obtain q-values for each gene.

What proportion of genes have an FDR less than 0.05 when comparing normal samples across batches?

0.4955796

✓ **Answer:** 0.4955796

0.4955796

Explanation

```
library(genefilter)
library(qvalue)

pval = rowttests(expr_norm, batch_norm)$p.value

qval = qvalue(pval)$qvalue
mean(qval < 0.05)
```

```
## [1] 0.4955796
```

Under the null hypothesis, there should be no significant gene expression differences between normal samples. However, nearly 50% of the genes appear differentially expressed across batches. Batch appears to be a confounding variable.

Submit

You have used 1 of 5 attempts

Answers are displayed within the problem

Question 3

1/1 point (graded)

Use `rowttests()` from the **genefilter** library to find which genes in `expr` appear to be differentially expressed between cancer and normal samples. Do not include batch effects. Then use the `qvalue()` function from the **qvalue** package to obtain q-values for each gene.

What proportion of genes appear differentially expressed between cancer and normal samples at an q-value cutoff of 0.05?

0.645835

✓ **Answer:** 0.6458735

0.645835

Explanation

```
library(genefilter)

pval = rowttests(expr, pdata$cancer)$p.value

qval = qvalue(pval)$qvalue
mean(qval < 0.05)
```

```
## [1] 0.6458735
```

The data suggest over 60% of the genes are differentially expressed. Even for a strong phenotype like cancer, this seems excessive.

Submit

You have used 1 of 5 attempts

Answers are displayed within the problem

Question 4

1/1 point (graded)

The `pdata` sample information associated with this experiment includes a variable `batch`. It is not immediately clear what these batches represent, whether they include all the major sources of experimental variability, and whether they will be useful for improving interpreation of the data.

Define a model matrix `X` that includes both cancer status and batch as variables.

Which of these commands correctly defines `X`?

○ `X = cbind(pdata$cancer, pdata$batch)`

○ `X = model.matrix(~cancer, batch)`

● `X = model.matrix(~pdata$cancer + pdata$batch)`

○ `X = model.matrix(~cancer + batch)`

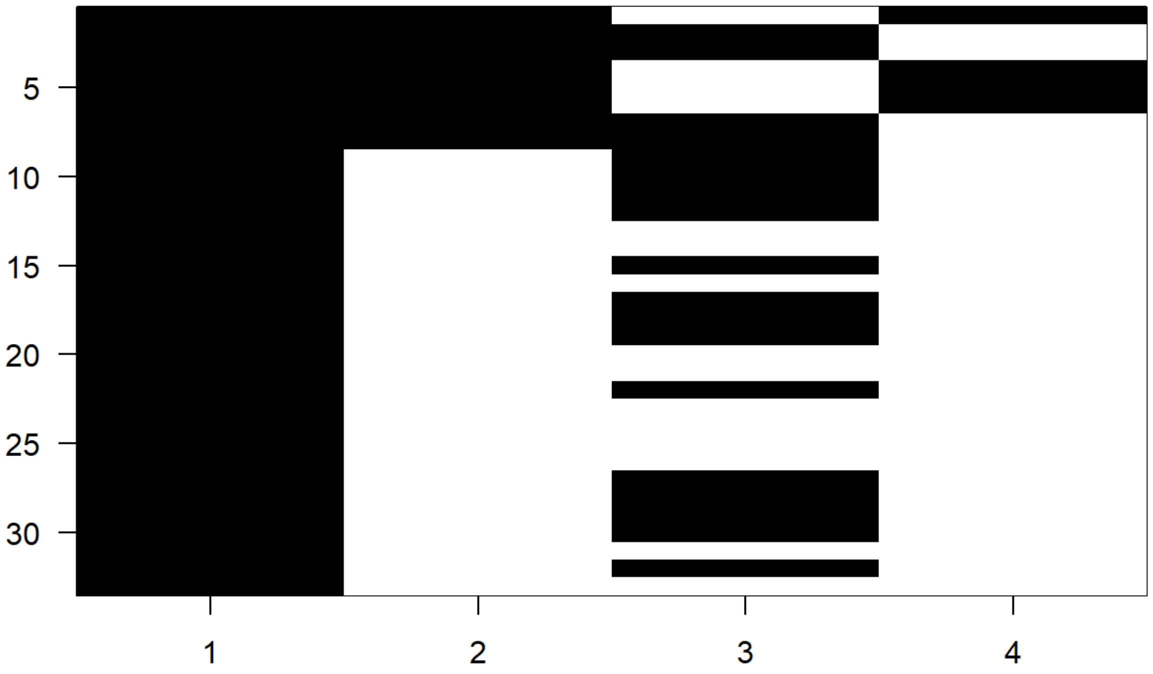
○ `X = model.matrix(~pdata$cancer, pdata$batch)`

○ `X = cbind(cancer, batch)`

✓

Explanation

```
X = model.matrix(~pdata$cancer + pdata$batch)
rafalib::imagemat(X)
```



Submit

You have used 1 of 2 attempts

Answers are displayed within the problem

Question 5

3/3 points (graded)

Now use the model matrix `X` defined above to fit a regression model using `lm()` for each gene. Note that you can obtain p-values for estimated parameters using `summary()`. Here is an example for the first gene:

```
i = 1
y = expr[i,]
fit = lm(y~X-1)
summary(fit)$coef
```

```
##                          Estimate Std. Error   t value
## X(Intercept)            9.8416126  0.1183420 83.162485 4.65
## Xpdata$cancerNormal -1.3616028  0.2225243 -6.118896 1.15
## Xpdata$batch2          0.3327226  0.1581411  2.103960 4.41
## Xpdata$batch3          1.4309186  0.3194294  4.479608 1.07
```

Find the p-value (`Pr(>|t|)`) for the expression difference between cancer and normal samples for each gene. You can do this by modifying the example code above and using `sapply()`. Then use the `qvalue()` function from the **qvalue** package to obtain q-values for each gene.

A. What proportion of genes appear to be differentially expressed between cancer and normal samples at a q-value cutoff of 0.05 when including batch in the model matrix?

0.7076246

✓ **Answer:** 0.7076246

0.7076246

B. What proportion of genes appear to be differentially expressed between batch 1 and batch 2?

0.2418884

✓ **Answer:** 0.2418884

0.2418884

C. What proportion of genes appear to be differentially expressed between batch 1 and batch 3?

0.1446394

✓ **Answer:** 0.1446394

0.1446394

Explanation

Part A:

```
pvals_cancer = sapply(1:nrow(expr),function(i){
    y = expr[i,]
    fit = lm(y~X-1)
    summary(fit)$coef[2,4]
})

qvals_cancer = qvalue(pvals_cancer)$qvalue
mean(qvals_cancer < 0.05)
```

```
## [1] 0.7076246
```

Part B:

```
pvals_1v2 = sapply(1:nrow(expr),function(i){
    y = expr[i,]
    fit = lm(y~X-1)
    summary(fit)$coef[3,4]
})

qvals_1v2 = qvalue(pvals_1v2)$qvalue
mean(qvals_1v2 < 0.05)
```

```
## [1] 0.2418884
```

Part C:

```
pvals_1v3 = sapply(1:nrow(expr),function(i){
    y = expr[i,]
    fit = lm(y~X-1)
    summary(fit)$coef[4,4]
})

qvals_1v3 = qvalue(pvals_1v3)$qvalue
mean(qvals_1v3 < 0.05)
```

```
## [1] 0.1446394
```

Submit

You have used 1 of 5 attempts

Answers are displayed within the problem

Question 6

1/1 point (graded)

Subtract the average expression of each gene from `expr` and save these results as `y`:

```
y = expr - rowMeans(expr)
```

Use the `svd()` function to obtain the principal components (PCs) for our detrended gene expression data `y`.

How many principal components (PCs) explain more than 5% each of the variability?

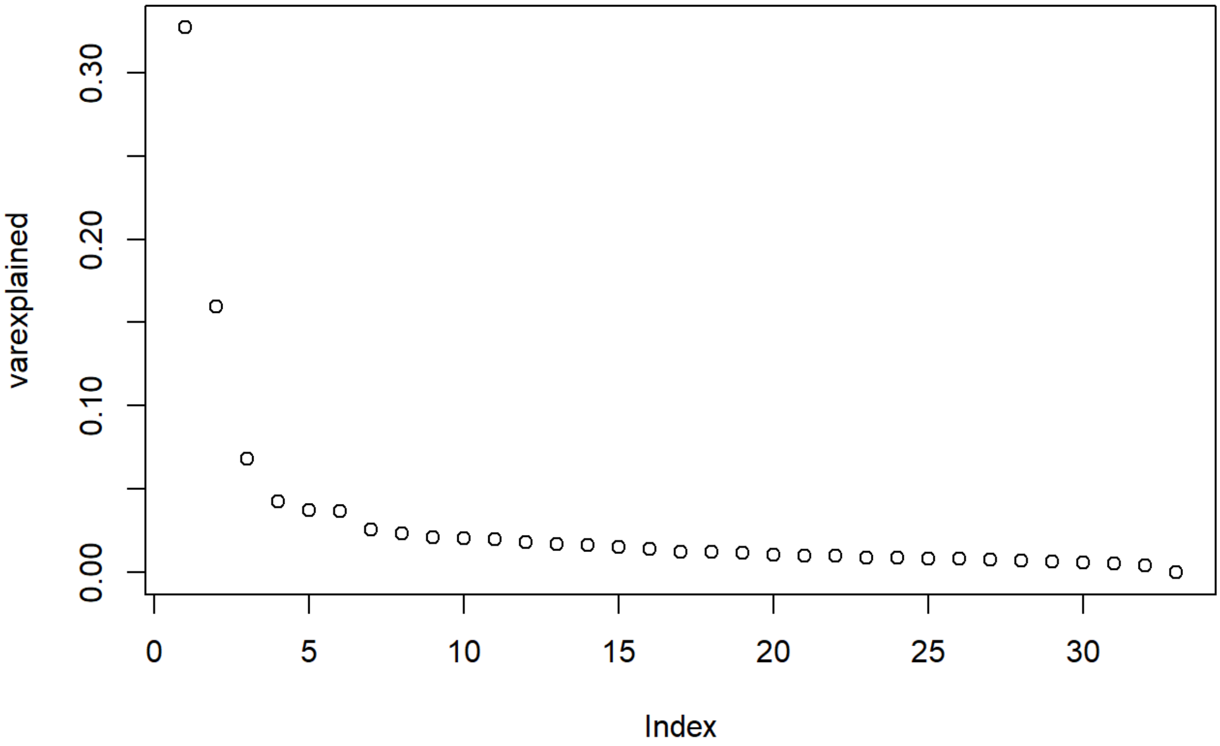
3

✓ **Answer:** 3

3

Explanation

```
s = svd(y)
varexplained = s$d^2/ sum(s$d^2)
plot(varexplained)
```



```
sum(varexplained > 0.05)
```

```
## [1] 3
```

Submit You have used 1 of 5 attempts

ℹ Answers are displayed within the problem

Question 7

1/1 point (graded)

Plot the first 2 principal components on the x and y axis respectively. Try coloring the points by either cancer status or batch number.

Which of the following are true?

Check all correct answers.

☑ Normal samples tend to have lower values of PC1 compared to cancer samples.

□ The samples with the lowest values of PC1 are in batch 3

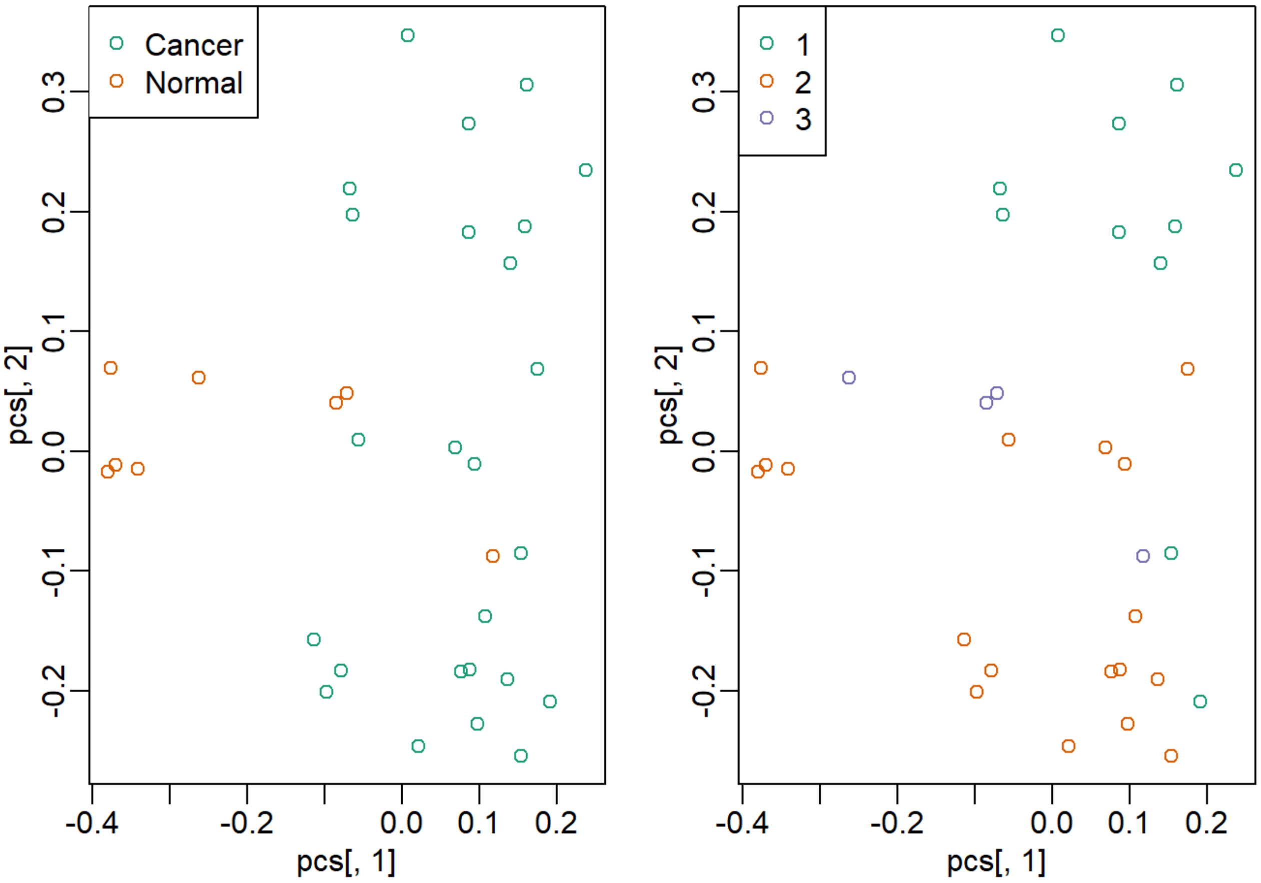
☑ Samples with high values of PC1 and high values of PC2 tend to be in batch 1

✓

Explanation

```
pcs = s$v[,1:2]

library(rafalib)
mypar(1,2)
plot(pcs[,1], pcs[,2], col=pdata$cancer)
legend("topleft", legend=levels(pdata$cancer), pch=1, col=1
plot(pcs[,1], pcs[,2], col=pdata$batch)
legend("topleft", legend=levels(pdata$batch), pch=1, col=1:
```



Submit You have used 1 of 3 attempts

Answers are displayed within the problem

Question 8

1/1 point (graded)

What is the absolute value of the correlation coefficient between the first principal component and cancer status?

0.7184334 ✓ **Answer:** 0.7184334

0.7184334

Explanation

```
abs(cor(pcs[,1], pdata$cancer=="Cancer"))
```

```
## [1] 0.7184334
```

Submit You have used 1 of 5 attempts

Answers are displayed within the problem

Question 9

1/1 point (graded)

Load the **sva** library and use it to infer the surrogate variables in `expr` other than cancer status.

Define `mod` as a model matrix including cancer status as a variable. Do not include `batch` as a variable - we will infer the batch effects with this approach. Then, use `sva()` to estimate the surrogate variables and store the output as `sv`.

How many significant surrogate variables affect the data?

6 ✓ **Answer:** 6

6

Explanation

```
library(sva)

## Loading required package: mgcv

## Loading required package: nlme

##
## Attaching package: 'nlme'

## The following object is masked from 'package:dplyr':
##
##     collapse

## This is mgcv 1.8-31. For overview type 'help("mgcv-package")'.

## Loading required package: BiocParallel
```

```
mod = model.matrix(~cancer, data=pdata)

sv = sva(expr, mod)
```

```
## Number of significant surrogate variables is:  6
## Iteration (out of 5 ):1  2  3  4  5
```

```
sv$n.sv
```

```
## [1] 6
```

Submit

You have used 1 of 5 attempts

ℹ Answers are displayed within the problem

Question 10

1/1 point (graded)

Define `mod0` as a null model matrix:

```
mod0 = model.matrix(~1, data=pdata)
```

The `f.pvalue()` function from **sva** quickly calculates p-values for each gene (row) given a design matrix `mod` with the variable of interest and a null matrix `mod0` that contains all variables except the variable of interest:

```
fpvals = f.pvalue(expr, mod, mod0)
```

Note that the q-values from this function are the same as the results from using `rowttests()` in question 3:

```
fqvals = qvalue(fpvals)$qvalue
mean(fqvals < 0.05)
```

```
## [1] 0.6458735
```

Now, alter the alternative and null model matrices to adjust for the surrogate variables:

```
modSv = cbind(mod,sv$sv)
mod0Sv = cbind(mod0,sv$sv)
```

Use `f.pvalue()` to calculate p-values for each gene given these new model matrices.

After adjusting for surrogate variables, what proportion of genes have a q-value below 0.05?

0.3314186

✓ **Answer:** 0.3314186

0.3314186

Explanation

```
fpValuesSv = f.pvalue(expr,modSv,mod0Sv)
fqSv = qvalue(fpValuesSv)$qvalue
mean(fqSv < 0.05)
```

```
## [1] 0.3314186
```

This is much lower than the original percentage of significant genes, suggesting that some batch effects have been removed.

Submit

You have used 1 of 5 attempts

ℹ Answers are displayed within the problem