Gene Expression Analysis with Microarrays

Leila Taher

WS 2022/2023

Contents

1 Affymetrix GeneChips

DNA microarrays are slides that are printed with thousands of tiny spots in defined positions, with each spot containing a known DNA sequence. These DNA sequences are known as *probes*. Typically, the probes represent genes. To determine which genes in a sample are being transcribed into messenger RNA (mRNA) (*expressed*), the following steps are undertaken:

(i) mRNA is isolated from the sample and converted into complementary DNA (cDNA). This cDNA is known as the *target*.

(ii) The target is labeled with a fluorescent dye.

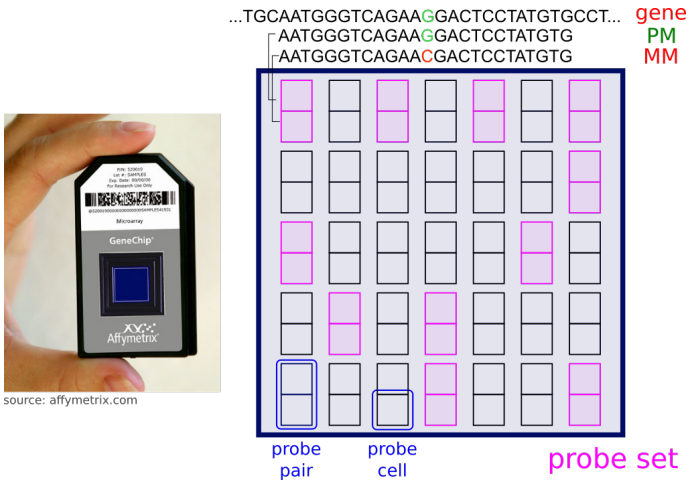
(iii) The fluorophore-labeled target is loaded onto the microarray and allowed to bind (*hybridize*) to the probes.

(iv) After washing and drying, microarray images are acquired on a laser scanner and fluorescent intensities are extracted using dedicated software to detect and quantify probe-target hybridization.

> The core principle behind microarrays is *hybridization*, the property of complementary nucleic acid sequences to specifically pair with each other by forming hydrogen bonds between complementary nucleotide base pairs.

During a microarray experiment, mRNA is typically isolated from two or more samples groups to determine the relative level of expression of each gene between the groups.

Affymetrix GeneChips allow whole genome gene expression analysis for a wide variety of experimental organisms (`https://pubmed.ncbi.nlm.nih.gov/16938544/`).



On an Affymetrix GeneChip, each gene is represented multiple times, by 11 to 20 different probes pairs scattered across the microarray. This collection of probes is called a *probe set*. Specifically, each probe set consists of:

- 11 to 20 different **perfect match (PM)** probes. These probes are 25 nucleotides long and each of them is perfectly complementary to a given mRNA sequence.
  - PM probes are designed for **specific hybridization**.
- 11 to 20 (exactly the same number as for PM probes) **mismatch (MM)** probes. These probes are 25 nucleotides long, but are not perfectly complementary. Instead, the $13th$ nucleotide has been changed to its complement.
  - The control MM probe intensities should reflect most of the background non-specific binding.
- A PM probe and its corresponding MM probe are referred to as a *probe pair*.

2 Importing data

The intensities measured for each microarray are normally stored in a `.CEL` file. This file contains an intensity value for each physical location (*cel*) of the microarray. The information needed to map each physical location to a probe set is described in a `.CDF` file.

2.1 The Gene Expression Omnibus

The Gene Expression Omnibus (GEO, `www.ncbi.nlm.nih.gov/geo/`) is a public repository that archives and freely distributes microarray, next-generation sequencing, and other forms of high-throughput genomic data submitted by the scientific community. We will work a with a microarray dataset of HUVEC and ocular vascular endothelial cells generated by Andrew C. Browning et al. (`https://pubmed.ncbi.nlm.nih.gov/22028475/`) (Table 1).

GEO accession number	Tissue
GSM524662	iris
GSM524663	retina
GSM524664	retina
GSM524665	iris
GSM524666	retina
GSM524667	iris

Table 1: GEO series GSE20986: gene expression profiling of HUVEC and ocular vascular endothelial cells.

1. What was the aim of the study by Browning *et al*?

   Open your favorite Web Browser, go to `https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE20986`, and download the `.CEL` files for the GEO accession numbers in the table. Note that the entire dataset consists of 12 microarrays, but we will only be using six.

2. Create and go into a folder "`GSE20986`".

3. Download the data set either by going to the appropriate GEO web-page or use following script to download via ftp:

```
#!/bin/bash

SERIES="GSE20986"

wget https://ftp.ncbi.nlm.nih.gov/geo/series/GSE20nnn/"$SERIES"/suppl/"$SERIES"
    _RAW.tar

tar -xf "$SERIES"_RAW.tar

rm $SERIES_RAW.tar

META_DATA="phenodata.txt"

cat "$META_DATA"

while read line;do
  if [[ $line =~ ^GSM ]];then
    SAMPLE=$(echo $line | sed "s/CEL.*/CEL/" )
    echo $SAMPLE
    gunzip $SAMPLE.gz
  fi
done < $META_DATA

rm *gz
```

4. If you have downloaded the files manually, you will need to generate the meta-file, describing the experimental factors, yourself. Using your favorite plain text editor (e.g., Notepad), type the following information:

```
FileName        Target
GSM524662.CEL   iris
GSM524663.CEL   retina
GSM524664.CEL   retina
GSM524665.CEL   iris
GSM524666.CEL   retina
GSM524667.CEL   iris
```

   - Use tab character to separate the fields.
   - Save it as a plain text file called "`phenodata.txt`".

5. The .CEL files can be imported into R using the "simpleaffy" (https://www.bioconductor.org/packages//2.7/bioc/html/simpleaffy.html) R/Bioconductor package. hgu133a2.db contains annotation data of the chip.

   Installation:

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

BiocManager::install(version = "3.12")
BiocManager::install("simpleaffy")
```

   This package also provides access to a variety of metrics for assessing the quality of RNA samples and of the intermediate stages of sample preparation and hybridization.

6. Use the function read.affy() to read the .CEL and "phenodata.txt" files into an object named "celfiles". "celfiles" is an "AffyBatch" object. You can verify it using the class() function.

7. Obtain a summary of the celfiles object by typing its name.

   - Based on the information obtained, can you find the product page on the website of the microarray manufacturer? What organism is this product for? Which genes are represented on the microarray?
   - How many samples have you read?
   - How many *genes* are represented on each microarray? Are these really genes?

   In addition to the intensities, the AffyBatch object contains information about the microarray experiments:

```
phenoData(celfiles)
pData(celfiles)
varLabels(celfiles)
```

   - What are these three functions doing? What kind of information do they provide?

3 Quality control

As with other high-throughput technologies, quality control (QC) is a fundamental aspect of the analysis process. Data quality can be checked using various diagnostic plots.
Overview:

(3.1) visual inspection

(3.2) probes & probe sets

  (3.2.1) probes

  (3.2.2) probesets

(3.3) intensity variability across samples

  (3.3.1) average background

  (3.3.2) scale factors

  (3.3.3) MA plots

(3.4) hybridization quality

(3.4.1) 3' to 5' ratios

(3.4.2) spike-in probe sets

(3.5) expression variability across samples

(3.5.1) RLE plots

(3.5.2) NUSE plots

3.1 Visual Inspection

The `image()` function creates an image of colors or shades of gray that represent the intensities for each spot on a microarray.
This image can be used to explore any spatial effects across the microarray. For some examples of images showing different artifacts, have a look at: `http://plmimagegallery.bmbolstad.com/`.

8. Examine all the microarrays. Do you see any obvious problems?

3.2 Probes & Probe sets

3.2.1 Probes

In principle, the MM probes provide a measure of *cross-hybridization* to the corresponding PM probes. Cross-hybridization is the tendency for a nucleic acid sequence to bind to other nucleic acid sequences that are almost, not perfectly complementary. Hence, subtracting the signal intensities of MM probes from those of PM probes should allow canceling of the effect of cross-hybridization. The intensities of the PM (MM) probes can be accessed with the `pm()` (`mm()`) function.

9. How many PM probes are present on each microarray?

10. How many MM probes do you expect on each microarray? Now check. How many MM probes are there on each microarray?

Typically, probe intensities are log-transformed. This is done for several reasons: it stabilizes the variance; it compresses the range of the data; and it makes the data more normally distributed.

11. Generate histograms of the intensities of the PM and MM probes for each microarray. Do you see any difference in the distributions of the PM and MM probes? Is this expected? Why?

3.2.2 Probe sets

You can access the probe set identifiers through the `geneNames()` function. Note that these are not gene names!

12. How many probe sets are on each microarray?

    Given one or several probe set identifiers, you can extract the corresponding intensities using the `probeset()` function.

13. How many probe pairs are there for "1552277_a_at"? The probe set "1552277_a_at" represents the *MSANTD3* gene (`https://www.genecards.org/cgi-bin/carddisp.pl?gene=MSANTD3`). Little is known about this gene, except that it is associated with swine influenza infections.

    Probe set are `S4` objects consisting of three slots:

    - "id": the probe set identifier.
    - "pm": a matrix with the PM probe intensities.
      - Columns represent samples; rows represent probes.
    - "mm": a matrix with the MM probe intensities.
      - Columns represent samples; rows represent probes.

14. How many probe pairs are there for "1007_s_at"? Extract the PM and MM intensities for "1007_s_at".

15. Use the `barplot.ProbeSet()` function to visualize the intensities for the PM and MM probes of probe set "1552277_a_at".

16. How are the intensities for the PM probes compared to those of the MM probes? What does this suggest for the probe set (and the corresponding gene)?

    - Examine the intensities of the probes for probe set "1007_s_at". What can we conclude about the expression of this probe set (and the corresponding gene)?

3.3 Comparison Between Microarrays

3.3.1 Average Background

The *average background* of a given microarray is computed as the $2nd$ percentile of the PM and MM probe intensities in a given region of the microarray. This summary statistic is intended to measure optical background. Average background values ranging between 20 and 100 are typical for a good quality microarray.

17. What could lead to a higher average background?

3.3.2 Scale factors

The `Bioconductor` packages usually contain shortcuts to visualize complicated data types using common functions. For example, we can use the `hist()` function to visualize the distribution of the data:

```
hist(celfiles, which="both", col=1:6, lty=1, lwd=1)
```

18. Similarly, you can create a boxplot using the `boxplot()` function. Try it.

19. Are the intensities on the different microarrays comparable?

Assuming that all other factors remain constant, the overall intensity observed for a given microarray reflects how much labeled target was hybridized to the probes. The median intensity of all PM and MM probes on a microarray are used to compute a summary statistic known as *scale factor*. Since the scale factor provide a measure of the overall intensities observed for a microarray, large variations in the scale factors of analyzed microarrays might indicate issues with RNA extraction, labeling, scanning or even microarray manufacture. Within an experiment, arrays are expected to have scale factors within 3-fold of each other; arrays whose scale factors are outside this range are considered to have poor quality.

3.3.3 MA plots

A MA plot is a scatter plot designed to detect differences between the labels in two-color microarrays. In two-color microarrays, two biological samples are labeled with different fluorescent dyes, usually Cyanine 3 (Cy3) and Cyanine 5 (Cy5), and then hybridized to the same array. M and A are defined as follows:

$$M = \log_2(R/G) = \log_2(R) - \log_2(G)$$

and

$$A = \frac{1}{2}\log_2(RG) = \frac{1}{2}(\log_2(R) + \log_2(G))$$

where R is the Cy5 intensity and G is the Cy3 intensity. M is plotted against A.
Affymetrix GeneChips use single-color labeling. Given a microarray,

- M is the difference between the (logarithm of the) intensity of a probe on the microarray and the median (logarithm of the) intensity of the probe across all analyzed microarrays.

- *A* is the average of the (logarithm of the) intensity of a probe on the microarray and the median (logarithm of the) intensity of that probe across all analyzed microarrays.

A MA plot identifies intensity-related biases.

20. Produce a MA plot for each microarray using the `MAplot()` function.

    - The red line is a locally estimated scatterplot smoothing (loess) regression curve based on the data. Ideally, the cloud of data points in the MA-plot should be centered around M=0 (blue line). What is the assumption?

3.4 Hybridization

3.4.1 3' to 5' ratios

Affymetrix's standard labeling protocols rely on a reverse-transcription step. Reverse transcription proceeds from the 3' end of a sequence. Therefore, RNA degradation or problems during labeling, often lead to under representation at the 5' end of the sequence.
The mRNA transcripts of *GAPDH* (`https://www.genecards.org/cgi-bin/carddisp.pl?gene=GAPDH`) and *ACTB* (`https://www.genecards.org/cgi-bin/carddisp.pl?gene=ACTB`) are relatively long and abundant in most cell types. Most Affymetrix microarrays contain separate probe sets targeting the 5', mid and 3' regions of these transcripts. By comparing the intensitiy values of the 3' probe set to either the mid or 5' probe set, it is possible to obtain a measure of the quality of the RNA hybridised to the microarray.

- High ratios indicate RNA degradation or problems during labeling. *GAPDH* is shorter than *ACTB*.
  - The ratio for *GAPDH* should be at or around 1 (and always less than 1.25).
  - The ratio for *ACTB* should be below 3 to be considered acceptable.
- All analyzed microarrays should have ratios in a similar range.

Assess the average background, scale factors, and 3' to 5' ratios of the microarrays using the `qc()` function. The results can be visualized using the `plot()` function.
In the resulting figure:

- Dotted horizontal lines separate the figure into rows, one for each microarray.
- The fraction of probe sets that are called *present* on each microarray (top) and the average background intensities (bottom) are written along the left hand side of the figure. In order to determine whether a given probe set is present, its PM and MM probe intensities are compared using one-sided Wilcoxon's signed rank test.
  - If any microarray has a substantially different value compared to the others, the numbers will be shown in red, otherwise they will be blue.
  - `GAPDH` ratios are plotted as circles. Values below 1 are displayed in blue; values above 1, in red.
  - *ACTB* ratios are plotted as triangles. Values below three (recommended) are given in blue, those above three in red.
- The central blue stripe represents the range where scale factors are within 3-fold of the mean across the scale factors of all analyzed microarrays:
  - The scale factors are plotted as horizontal lines departing from the central vertical line of the figure. A line to the left corresponds to a down-scaling, to the right, to an up-scaling. If any scale factors fall outside this 3-fold region, they are all colored red, otherwise they are blue.
- Dotted vertical lines provide a scale from −3 to 3.

21. Why should the fraction of probe sets that are called present on each microarray be similar among the different microarrays? How do you interpret your results?

3.4.2 Spike-in probe sets

In addition to the probe sets for the endogenous *GAPDH* and *ACTB*, Affymetrix GeneChips contain probe sets for *BioB*, *BioC*, *BioD* and *CreX*. *BioB*, *BioC*, *BioD* and *CreX* are *Bacillus subtiliis* genes and no human mRNA transcript is supposed to bind them. The mRNA transcripts of *BioB*, *BioC*, *BioD* and *CreX* are added (*spiked in*) into the samples during their preparation.

- *BioB* is added at a concentration of 1.5 pM, corresponding to approximately three transcripts per cell, the lower limit of detection for the system.
- *BioC*, *BioD* and *CreX* are spiked-in at increasing concentrations.

22. Use the `spikeInProbes()` function to summarize the intensities of the spike-in probe sets, and the `matplot()` function to visualize them.
    - How do you interpret your results?

3.5 Expression Variability

3.5.1 RLE Plots

For a given microarray, the RLE (Relative Log Expression) value of a probe set is the ratio between the *expression value* of the probe set and the median *expression value* of the probe set across all microarrays analyzed. The ratios are then log-transformed. The expression value of a probe set is estimated using robust regression, as a function of probe and microarray effects. Basically, the aim is to down-weight outlier probes. This is implemented in the "`affyPLM`" (`https://www.bioconductor.org/packages/release/bioc/html/affyPLM.html`) `R/Bioconductor` package.

23. Use the "`fitPLM()`" function to estimate the probe set expression values.
24. Visualize the distribution of the RLE values of the probe sets of each microarray using the `Mbox()` function.
25. Assuming that the expression levels of most genes (and, hence, their probe sets) will remain constant during an experiment, what should be the median of each distribution in an ideal experiment? What about the spread of the distribution? How should it look like in an ideal experiment?
    - Do any of your microarrays deviate from the expectation?

3.5.2 NUSE plots

The Normalized Unscaled Standard Error (NUSE) plot visualizes the distribution of the standard errors of the model. On the NUSE plot, the standard errors are standardized so that that for each probe set, the median standard error across all analyzed microarrays is equal to 1.

- As a rule of thumb, a median equal to or greater than 1.05 indicates quality problems, which are often also detected by other QC analyses.

26. Produce a NUSE plot and interpret your results.
    - Do any of your microarrays deviate from the expectation?

4 From Intensities to Expression Values

Many background correction and normalization algorithms have been developed to correct for non-biological differences in microarray signals (Table 2).

Method	Background Adjustment	Normalization	MM Correct	Probe set Summarization	Reference
MAS5	regional adjustment	scaling by constant	subtract idealized MM	Tukey biweight average	Affymetrix Documentation: MAS5
gcRMA	by GC content	quantile normalization	/	robust fit of linear model	Wu et al. (2004), JASA, 99, 909-917.
RMA	array background	quantile normalization	/	robust fit of linear model	Irizarry et al. (2003), Nuc Acids Res, 31, e15.

Table 2: Popular background correction and normalization methods for Affymetrix GeneChip microarrays.

- These algorithms employ different methods for background correction, normalization, and probe set summarization.

4.1 Robust Multi-array Average (RMA)

The Robust Multi-array Average (RMA) algorithm summarizes each probe set as a robust multi-array average of log-transformed, background-adjusted, normalized PM intensities. The resulting value is known as the *expression value* of the probe set.

- RMA ignores MM probes entirely: although the MM probes provide information, implementing this without adding noise is challenging.

(i) **Background correction**
Let F_{ijk} be the foreground intensity for the j^{th} PM probe in the k^{th} probe set on the i^{th} microarray and B_{ijk} be the corresponding background intensity, then the background-adjusted intensity of the PM probe is

$$PM_{ijk} = F_{ijk} + B_{ijk}$$

where

$$F_{ijk} \sim e^{\lambda_{ijk}}$$

and

$$B_{ijk} \sim N(\mu_i, \sigma^2_{ijk}).$$

The estimates for μ_{ijk}, σ_{ijk} and λ_{ijk} are computed for each microarray using the distribution of PM probe intensities by fitting the model

$$\mathrm{E}[F_{ijk}|PM_{ijk}] = PM_{ijk} - \mu_i - \lambda_{ijk}\,\sigma^2_{ijk} > 0.$$

(ii) **Normalization between arrays**

Between-array normalization is performed using quantile normalization (`https://en.wikipedia.org/wiki/Quantile_normalization`).

(iii) **Summarization of the probe sets**

The probe sets are summarized as the robust average of their log-transformed, background-adjusted, normalized PM probe intensities using the median polish algorithm (`https://en.wikipedia.org/wiki/Median_polish`). The median polish algorithm is a type of robust 2-way ANOVA, where one factor is the microarray and the other is the probe set.

- The algorithm is robust to outliers, so that single probes with large values are down-weighted.
- Because both the quantile normalization and the median polish algorithms use the data from all the microarrays, using just a subset of the microarrays or removing a single bad microarray affects the normalization step for all microarrays.

The resulting value is *expression value* of the probe set.

Apply the RMA algorithm as implemented in the (`rma()`) function:

```
eset <- rma(celfiles)
```

27. What object is returned by the `rma()` function? Which slots does the object contain ? Ask for some help about this class of objects.

    The `exprs()` function returns the expression values of the probe sets as a matrix:

    ```
    rma.exprs <- exprs(eset)
    ```

    Following background correction, normalization, and probe set summarization, diagnostic plots should be used to assess the quality of the data.

28. Visualize the distribution of the expression values using the `boxplot()` function.

29. What is the difference between these boxplots and the ones in 18?

30. Do the background correction and normalization procedures correct all the differences between the analyzed microarrays?

Depending on the QC assessment,ayou may decide to discard some of the data before proceeding with the differential expression analysis.

31. Would you discard any of the microarrays based on your previous results?

5 Differential Expression Analysis

The goal of differential expression analysis is to identify genes whose expression differs under different conditions. We assume a linear model

$$\mathbf{Y} = \mathbf{X}\beta + \epsilon$$

where $\mathbf{Y}$ is a vector containing the expression values of the probe sets, $\mathbf{X}$ is the *design matrix*, β is a vector of regression coefficients and ϵ is a normally distributed error term.
The choice of design matrix is a critical step in linear modeling since it encodes which coefficients will be fit in the model, as well as the inter-relationship between the samples. Here we will compare between two groups of samples: iris versus retina. We encode this experimental design in R with two steps:

(i) We start a formula with the tilde symbol ($\sim$).

(ii) Then we write the name of the variable that separates the samples into two groups.

32. In our case, the variable is called “Target”:

    ```
    X<-model.matrix(~ celfiles$Target)
    ```

    This design matrix has two columns because there are two parameters in this conceptual design: the expression level in each of the two groups. Column 1 represents the expression of the iris group and column 2 represents the difference in expression level from the retina group to the iris group.

    - Testing that the two groups have the same expression level is done by testing whether the second parameter (equal to the difference in expression between the two groups) is equal to zero.

    A different parametrization is

```
X<--model.matrix(~ celfiles$Target-1)
colnames(X) <- c("iris", "retina")
```

In this design matrix, the two parameters corresponding to the two columns represent the expression levels in the two groups.

- Testing that the two groups have the same expression level is done by asking whether these two parameters are the same.

33. We will use the `limma` R/Bioconductor (`https://www.bioconductor.org/packages/release/bioc/html/limma.html`) package to fit a linear model to each probe set.

```
fit1 <- lmFit(eset, X)
```

34. Significance testing can be done for each parameter, to which `limma` refers as "contrasts". We are interested in an association between probe set expression and tissue of origin:

```
contrast.matrix <- makeContrasts(irisvsretina="iris-retina", levels=X)
```

35. Given a linear model and a set of contrasts, we can now compute the estimated coefficients and standard errors for that particular set of contrasts:

```
fit2 <- contrasts.fit(fit1, contrast.matrix)
```

36. Finally, we compute the moderated t-statistic, empirically shrinking the residual variance of each probe set:

```
fit3 <- eBayes(fit2)
```

and use the function `topTable` to extract the results. The column "`adj.P.Val`" contains false discovery rate (FDR) adjusted P-values:

```
topTable(fit3, coef=1, n=5, adjust.method="fdr")
```

37. At a significance level of 0.05, how many probe sets are differentially expressed?

5.1 Volcano plots

A volcano plot is a scatter plot that is used to identify changes between two conditions. It plots the significance ($-\log_{10}$ P-value) on the y-axis and the fold-change on the x axes.

38. Generate a volcano plot.

6 version information R and associated packages

```
> sessionInfo()
R version 4.0.5 (2021-03-31)
Platform: x86_64-pc-linux-gnu (64-bit)
Running under: Ubuntu 20.04.2 LTS

Matrix products: default
BLAS:   /usr/lib/x86_64-linux-gnu/openblas-pthread/libblas.so.3
```

```
LAPACK: /usr/lib/x86_64-linux-gnu/openblas-pthread/liblapack.so.3

locale:
 [1] LC_CTYPE=en_US.UTF-8       LC_NUMERIC=C
 [3] LC_TIME=en_US.UTF-8        LC_COLLATE=en_US.UTF-8
 [5] LC_MONETARY=en_US.UTF-8    LC_MESSAGES=en_US.UTF-8
 [7] LC_PAPER=en_US.UTF-8       LC_NAME=C
 [9] LC_ADDRESS=C               LC_TELEPHONE=C
[11] LC_MEASUREMENT=en_US.UTF-8 LC_IDENTIFICATION=C

attached base packages:
[1] parallel  stats     graphics  grDevices utils     datasets  methods
[8] base

other attached packages:
 [1] BiocManager_1.30.16   simpleaffy_2.64.0     genefilter_1.70.0
 [4] ggrepel_0.9.1         gridExtra_2.3         ggsci_2.9
 [7] ggplot2_3.3.5         affyPLM_1.64.0        preprocessCore_1.50.0
[10] gcrma_2.60.0          affy_1.66.0           Biobase_2.50.0
[13] BiocGenerics_0.36.0   dplyr_1.0.7

loaded via a namespace (and not attached):
 [1] Rcpp_1.0.7            lattice_0.20-45       Biostrings_2.56.0
 [4] assertthat_0.2.1      utf8_1.2.2            R6_2.5.1
 [7] stats4_4.0.5          RSQLite_2.2.8         pillar_1.6.3
[10] zlibbioc_1.36.0       rlang_0.4.12          annotate_1.66.0
[13] blob_1.2.2            S4Vectors_0.28.1      Matrix_1.3-4
[16] splines_4.0.5         RCurl_1.98-1.5        bit_4.0.4
[19] munsell_0.5.0         compiler_4.0.5        pkgconfig_2.0.3
[22] tidyselect_1.1.1      tibble_3.1.6          IRanges_2.24.1
[25] XML_3.99-0.8          fansi_0.5.0           crayon_1.4.2
[28] withr_2.4.2           bitops_1.0-7          grid_4.0.5
[31] xtable_1.8-4          gtable_0.3.0          lifecycle_1.0.1
[34] DBI_1.1.1             magrittr_2.0.1        scales_1.1.1
[37] cachem_1.0.6          XVector_0.30.0        affyio_1.58.0
[40] ellipsis_0.3.2        generics_0.1.0        vctrs_0.3.8
[43] tools_4.0.5           bit64_4.0.5           glue_1.4.2
[46] purrr_0.3.4           fastmap_1.1.0         survival_3.2-13
[49] AnnotationDbi_1.50.3  colorspace_2.0-2      memoise_2.0.0
```