R and *Bioconductor* for Genomic Analysis

Martin Morgan[1*]

[1]Roswell Park Comprehensive Cancer Center, Buffalo, New York

[*]Martin.Morgan@RoswellPark.org (mailto:Martin.Morgan@RoswellPark.org)

5 December 2019

Package

BiocIntro 0.0.10

Contents

1 Introduction

Description: This workshop will introduce you to the *Bioconductor* collection of R packages for statistical analysis and comprehension of high-throughput genomic data. The emphasis is on data exploration, using RNA-sequence gene expression experiments as a motivating example. How can I access common sequence data formats from R? How can I use information about gene models or gene annotations in my analysis? How do the properties of my data influence the statistical analyses I should perform? What common workflows can I perform with R and *Bioconductor*? How do I deal with very large data sets in R? These are the sorts of questions that will be tackled in this workshop.

Requirements: You will need to bring your own laptop. The workshop will use cloud-based resources, so your laptop will need a web browser and WiFi capabilities. Participants should have used *R* and *RStudio* for tasks such as those covered in introductory workshops earlier in the week. Some knowledge of the biology of gene expression and of concepts learned in a first course in statistics will be helpful.

Relevance: This workshop is relevant to anyone eager to explore genomic data in *R*. The workshop will help connect 'core' *R* concepts for working with data (e.g., data management via `data.frame()`, statistical modelling with `lm()` or `t.test()`, visualization using `plot()` or `ggplot()`) to the special challenges of working with large genomic data sets. It will be especially helpful to those who have or will have their own genomic data, and are interested in more fully understanding how to work with it in *R*.

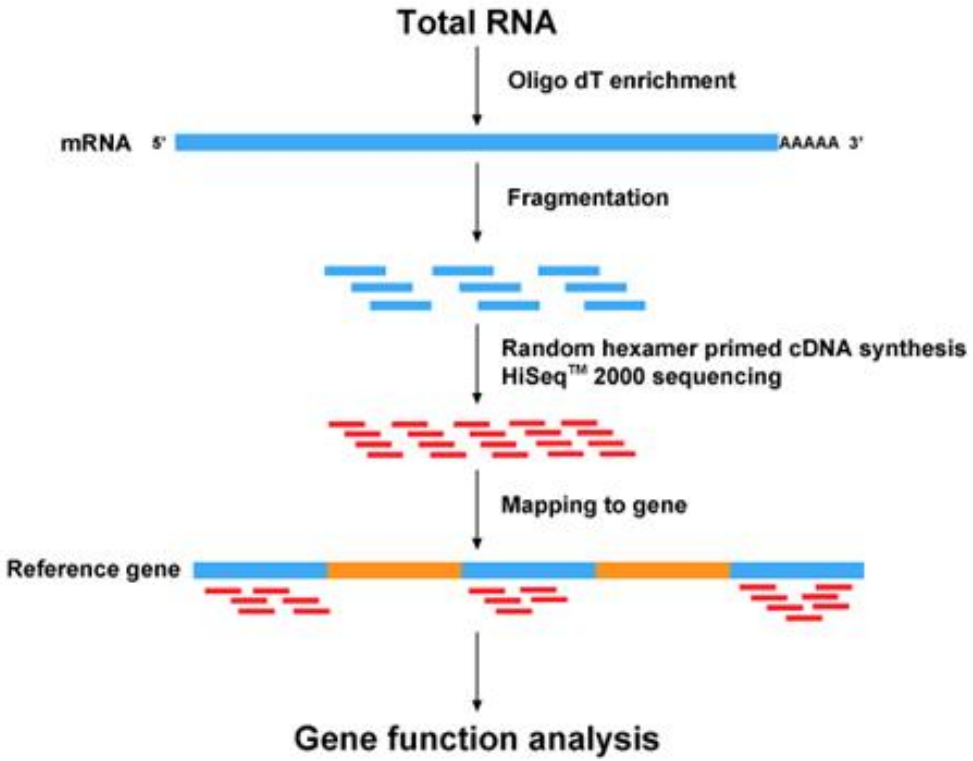
1.1 Our goal

RNA-seq

- Designed experiment, e.g., 8 samples from four cell lines exposed to two treatments (based on Himes et al., PMID: 24926665 (http://www.ncbi.nlm.nih.gov/pubmed/24926665); details in the airway (https://bioconductor.org/packages/airway) package vignette).

```
             cell   dex
SRR1039508  N61311 untrt
SRR1039509  N61311   trt
SRR1039512 N052611 untrt
SRR1039513 N052611   trt
SRR1039516 N080611 untrt
SRR1039517 N080611   trt
SRR1039520 N061011 untrt
SRR1039521 N061011   trt
```

- Library preparation: mRNA to stable double-stranded DNA
- DNA sequencing of 'short' mRNA-derived fragments
- Alignment to a reference genome or transcriptome



source: http://bio.lundberg.gu.se/courses/vt13/rnaseq.html (http://bio.lundberg.gu.se/courses/vt13/rnaseq.html)

- End result: a matrix of ‘counts’ – reads aligning to genes across samples.

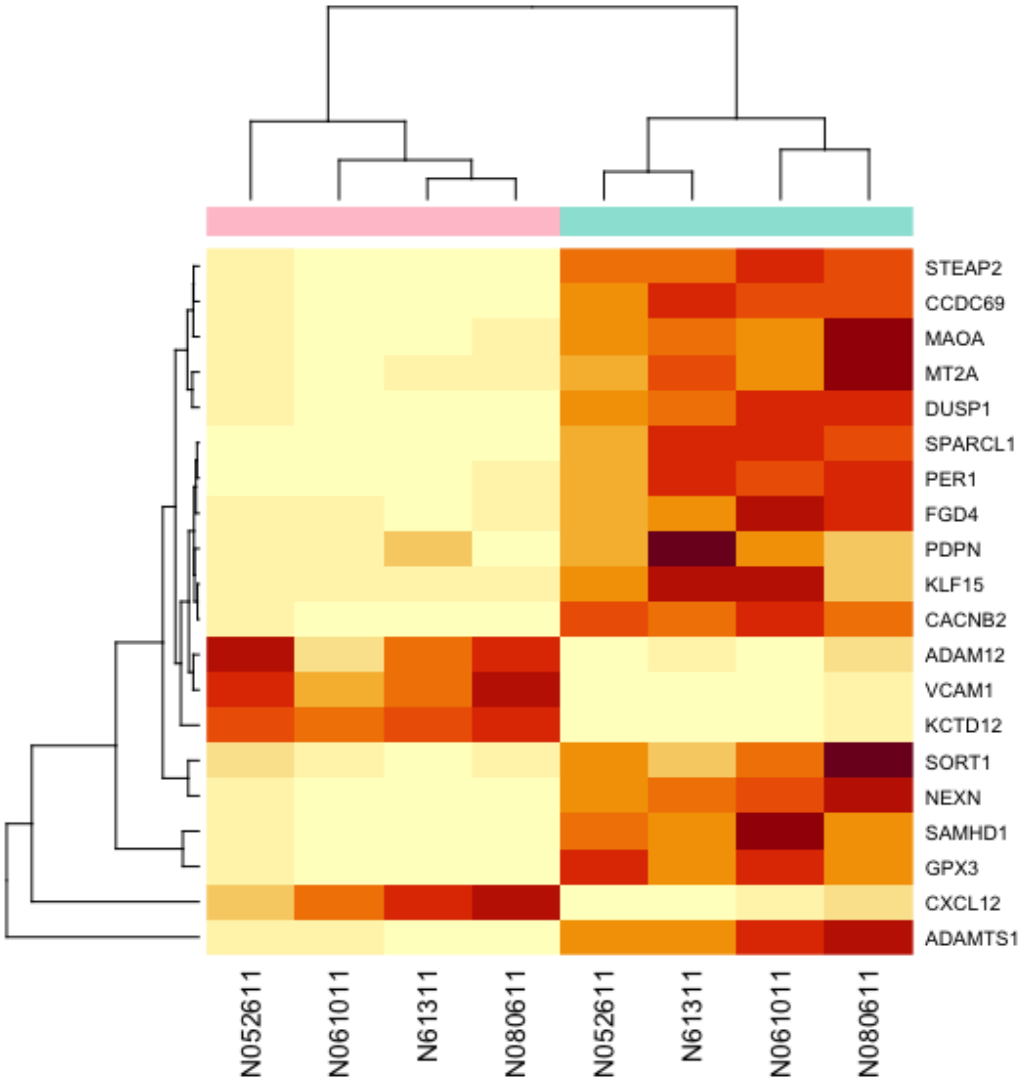
```
                SRR1039508 SRR1039509 SRR1039512 SRR1039513 SRR1039516
ENSG00000000003        679        448        873        408       1138
ENSG00000000005          0          0          0          0          0
ENSG00000000419        467        515        621        365        587
...                    ...        ...        ...        ...        ...
                SRR1039517 SRR1039520 SRR1039521
ENSG00000000003       1047        770        572
ENSG00000000005          0          0          0
ENSG00000000419        799        417        508
...                    ...        ...        ...
```

Research question

- Which gene counts are most different between dexamethasone `untrt` and `trt` experimental treatments?
- We’ll try to understand *how* to accomplish this, without going into statistical details.

Our goal

- Visualize 20 most differentially expressed genes as a heatmap.


STEAP2
CCDC69
MAOA
MT2A
DUSP1
SPARCL1
PER1
FGD4
PDPN
KLF15
CACNB2
ADAM12
VCAM1
KCTD12
SORT1
NEXN
SAMHD1
GPX3
CXCL12
ADAMTS1
N052611
N061011
N61311
N080611
N052611
N61311
N061011
N080611

2 Data gathering, input, representation, and cleaning

2.1 Base *R* data structures

Sample information

- Simple 'tab-separated value' text file, e.g., from Excel export.
- Input using base *R* command `read.table()`
- 'Atomic' vectors, e.g., `integer()`
- `factor()` and `NA`
- `data.frame()` : coordinated management
  - Column access with `$`
  - Subset with `[ , ]`

```
samples_file <-
    system.file(package="BiocIntro", "extdata", "samples.tsv")
samples <- read.table(samples_file)
samples
##                cell   dex avgLength
## SRR1039508  N61311 untrt       126
## SRR1039509  N61311   trt       126
## SRR1039512 N052611 untrt       126
## SRR1039513 N052611   trt        87
## SRR1039516 N080611 untrt       120
## SRR1039517 N080611   trt       126
## SRR1039520 N061011 untrt       101
## SRR1039521 N061011   trt        98
```

```
samples$dex <- relevel(samples$dex, "untrt")
```

Counts

- Another tsv file. Many rows, so use `head()` to view the first few.
- Row names: gene identifiers. Column names: sample identifiers.
- All columns are the same (numeric) type; represent as a `matrix()` rather than `data.frame()`

```
counts_file <-
    system.file(package="BiocIntro", "extdata", "counts.tsv")
counts <- read.table(counts_file)
dim(counts)
## [1] 63677     8

head(counts)
##                 SRR1039508 SRR1039509 SRR1039512 SRR1039513 SRR1039516
## ENSG00000000003        679        448        873        408       1138
## ENSG00000000005          0          0          0          0          0
## ENSG00000000419        467        515        621        365        587
## ENSG00000000457        260        211        263        164        245
## ENSG00000000460         60         55         40         35         78
## ENSG00000000938          0          0          2          0          1
##                 SRR1039517 SRR1039520 SRR1039521
## ENSG00000000003       1047        770        572
## ENSG00000000005          0          0          0
## ENSG00000000419        799        417        508
## ENSG00000000457        331        233        229
## ENSG00000000460         63         76         60
## ENSG00000000938          0          0          0

counts <- as.matrix(counts)
```

2.2 Genomic ranges (GRanges)

Row annotations.

- 'GTF' files contain information about gene models.
- The GTF file relevant to this experiment – same organism (*Homo sapiens*), genome (GRCh37) and gene model annotations (Ensembl release-75) as used in the alignment and counting step – is

```
url <- "ftp://ftp.ensembl.org/pub/release-75/gtf/homo_sapiens/Homo_sapiens.GRCh
37.75.gtf.gz"
```

- Use `BiocFileCache` to download the resource once to a location that persists across *R* sessions.

```
library(BiocFileCache)
```

- About *Bioconductor* packages
  - `BiocFileCache` is available from https://bioconductor.org (https://bioconductor.org)
  - Discover packages at https://bioconductor.org/packages (https://bioconductor.org/packages)
  - Learn about `BiocFileCache` from the BiocFileCache (https://bioconductor.org/packages/BiocFileCache) 'landing page'.
  - Explore the `BiocFileCache` package vignette (https://bioconductor.org/packages/release/bioc/vignettes/BiocFileCache/inst/doc/BiocFileCache.html) (access the vignette from within *R*: `browseVignettes("BiocFileCache")`).
  - Install the package with `BiocManager::install("BiocFileCache")`.
  - Find help on functions with, e.g., `?bfcrpath`.
  - Ask questions at https://support.bioconductor.org (https://support.bioconductor.org)

```
gtf_file <- bfcrpath(rnames = url)
## Using temporary cache /var/folders/yn/gmsh_22s2c55v816r6d51fx1tnyl61/T//Rtmp
LUzYG5/BiocFileCache
## adding rname 'ftp://ftp.ensembl.org/pub/release-75/gtf/homo_sapiens/Homo_sap
iens.GRCh37.75.gtf.gz'
```

- GTF files are plain text files and *could* be read using `read.table()` or similar, but contain structured information that we want to represent in *R*.
- Common sequence data formats
  - BED, GTF, bigWig: rtracklayer (https://bioconductor.org/packages/rtracklayer)
  - FASTA (DNA sequence): Biostrings (https://bioconductor.org/packages/Biostrings)
  - FASTQ (short reads & quality scores): ShortRead (https://bioconductor.org/packages/ShortRead)
  - BAM (aligned reads): Rsamtools (https://bioconductor.org/packages/Rsamtools), GenomicAlignments (https://bioconductor.org/packages/GenomicAlignments)
  - VCF (called variants): VariantAnnotation (https://bioconductor.org/packages/VariantAnnotation), VariantFiltering (https://bioconductor.org/packages/VariantFiltering). MAF: maftools (https://bioconductor.org/packages/maftools)
- Use the rtracklayer (https://bioconductor.org/packages/rtracklayer) package to import the file.

```
library(rtracklayer)
gtf <- import(gtf_file)
```

- A `GRanges` object
  - Range-specific information
  - Annotations on each range
  - Use functions to access core elements: `seqnames()` (e.g., chromosome), `start()` / `end()` / `width()`, `strand()`, etc.
  - Use `$` or `mcols()$` to access annotations on ranges.
  - *Bioconductor* conventions: 1-based, closed intervals (like Ensembl) rather than 0-based, 1/2 open intervals (like UCSC).
- Filter the information to gene-level annotations, keeping only some of the information about each genomic range. Use the `gene_id` column as `names()`.

```
rowidx <- gtf$type == "gene"
colidx <- c("gene_id", "gene_name", "gene_biotype")
genes <- gtf[rowidx, colidx]
names(genes) <- genes$gene_id
genes$gene_id <- NULL

genes
## GRanges object with 63677 ranges and 2 metadata columns:
##                    seqnames      ranges strand |   gene_name   gene_biotype
##                       <Rle>   <IRanges>  <Rle> | <character>    <character>
##   ENSG00000223972        1 11869-14412      + |     DDX11L1     pseudogene
##   ENSG00000227232        1 14363-29806      - |      WASH7P     pseudogene
##   ENSG00000243485        1 29554-31109      + |  MIR1302-10        lincRNA
##   ENSG00000237613        1 34554-36081      - |     FAM138A        lincRNA
##   ENSG00000268020        1 52473-54936      + |      OR4G4P     pseudogene
##               ...      ...         ...    ... .         ...            ...
##   ENSG00000198695       MT 14149-14673      - |      MT-ND6 protein_coding
##   ENSG00000210194       MT 14674-14742      - |       MT-TE        Mt_tRNA
##   ENSG00000198727       MT 14747-15887      + |      MT-CYB protein_coding
##   ENSG00000210195       MT 15888-15953      + |       MT-TT        Mt_tRNA
##   ENSG00000210196       MT 15956-16023      - |       MT-TP        Mt_tRNA
##   -------
##   seqinfo: 265 sequences from an unspecified genome; no seqlengths
```

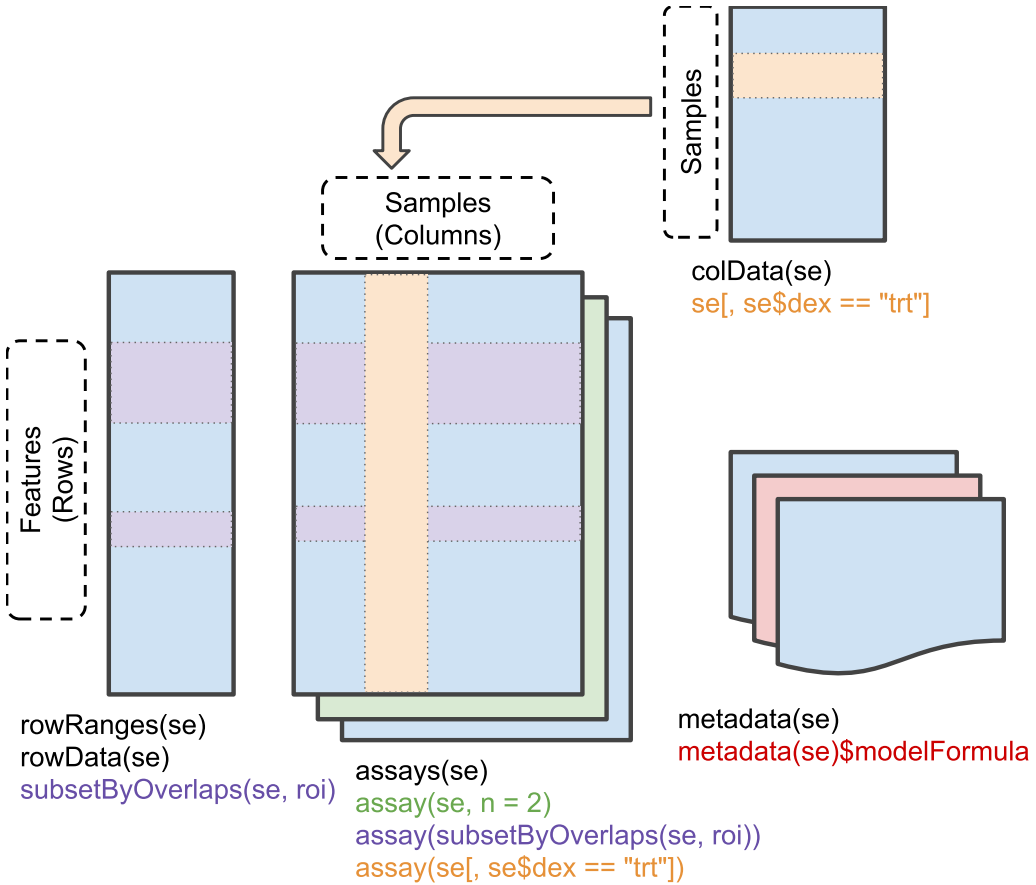
2.3 Coordinated management (`SummarizedExperiment`)

Three different data sets

- `counts` : results of the RNAseq workflow
- `samples` : sample and experimental design information
- `genes` : information about the genes that we've assayed.

Coordinate our manipulation

- Avoid 'bookkeeping' errors when, e.g., we subset one part of the data in a way different from another.
- Use the SummarizedExperiment (https://bioconductor.org/packages/SummarizedExperiment) package and data representation.
    - Two-dimensional structure, so subset with `[,]`
    - Use functions to access components: `assay()` , `rowData()` , `rowRanges()` , `colData()` , etc.




```
library(SummarizedExperiment)
```

- Make sure the order of the `samples` rows match the order of the samples in the columns of the `counts` matrix, and the order of the `genes` rows match the order of the rows of the `counts` matrix.
- Create a `SummarizedExperiment` to coordinate our data manipulation.

```
samples <- samples[colnames(counts),]
genes <- genes[rownames(counts),]
se <- SummarizedExperiment(
  assays = list(counts = counts),
  rowRanges = genes, colData = samples
)

se
## class: RangedSummarizedExperiment
## dim: 63677 8
## metadata(0):
## assays(1): counts
## rownames(63677): ENSG00000000003 ENSG00000000005 ... ENSG00000273492
##   ENSG00000273493
## rowData names(2): gene_name gene_biotype
## colnames(8): SRR1039508 SRR1039509 ... SRR1039520 SRR1039521
## colData names(3): cell dex avgLength
```

3 Analysis & visualization

3.1 Differential expression analysis

Gestalt

- Perform a `t.test()` for each row of the count matrix, asking whether the `trt` samples have on average counts that differ from the `untrt` samples.
- *Many* nuanced statistical issues

The DESeq2 (https://bioconductor.org/packages/DESeq2) pacakge

- Implements efficient, 'correct', robust algorithms for performing RNA-seq differential expression analysis of moderate-sized experiments.

```
library(DESeq2)
```

- Specify our experimental design, perform the analysis taking account of the nuanced statistical issues, and get a summary of the results. The details of this step are beyond the scope of this workshop.

```
dds <- DESeqDataSet(se, ~ cell + dex)
fit <- DESeq(dds)
## estimating size factors
## estimating dispersions
## gene-wise dispersion estimates
## mean-dispersion relationship
## final dispersion estimates
## fitting model and testing
destats <- results(fit)

destats
## log2 fold change (MLE): dex trt vs untrt
## Wald test p-value: dex trt vs untrt
## DataFrame with 63677 rows and 6 columns
##                         baseMean      log2FoldChange             lfcSE
##                        <numeric>           <numeric>         <numeric>
## ENSG00000000003  708.602169691234   -0.38125388742934 0.100654430181804
## ENSG00000000005                 0                  NA                NA
## ENSG00000000419  520.297900552084   0.206812715390398 0.112218674568195
## ENSG00000000457  237.163036796015  0.0379205923946151  0.14344471633862
## ENSG00000000460  57.9326331250967 -0.0881676962628265 0.287141995236272
## ...                           ...                 ...               ...
## ENSG00000273489 0.275899382507797    1.48372584344306  3.51394515550546
## ENSG00000273490                 0                  NA                NA
## ENSG00000273491                 0                  NA                NA
## ENSG00000273492 0.105978355992386  -0.463691271907546  3.52308373749196
## ENSG00000273493 0.106141666408122  -0.521381077922898  3.53139001322807
##                             stat               pvalue                padj
##                        <numeric>            <numeric>           <numeric>
## ENSG00000000003  -3.78775069056286 0.000152017272514002 0.00128292609656079
## ENSG00000000005                 NA                  NA                NA
## ENSG00000000419   1.84294384322566   0.0653372100662581   0.196469601297369
## ENSG00000000457   0.26435684326705    0.791504962999781    0.91141814384918
## ENSG00000000460 -0.307052600196215    0.75880333554496   0.895006448013164
## ...                           ...                 ...                 ...
## ENSG00000273489  0.422239328669782    0.672850337762336                  NA
## ENSG00000273490                 NA                  NA                  NA
## ENSG00000273491                 NA                  NA                  NA
## ENSG00000273492 -0.131615171950935    0.895288684444562                  NA
## ENSG00000273493 -0.147641884914972     0.88262539793309                  NA
```

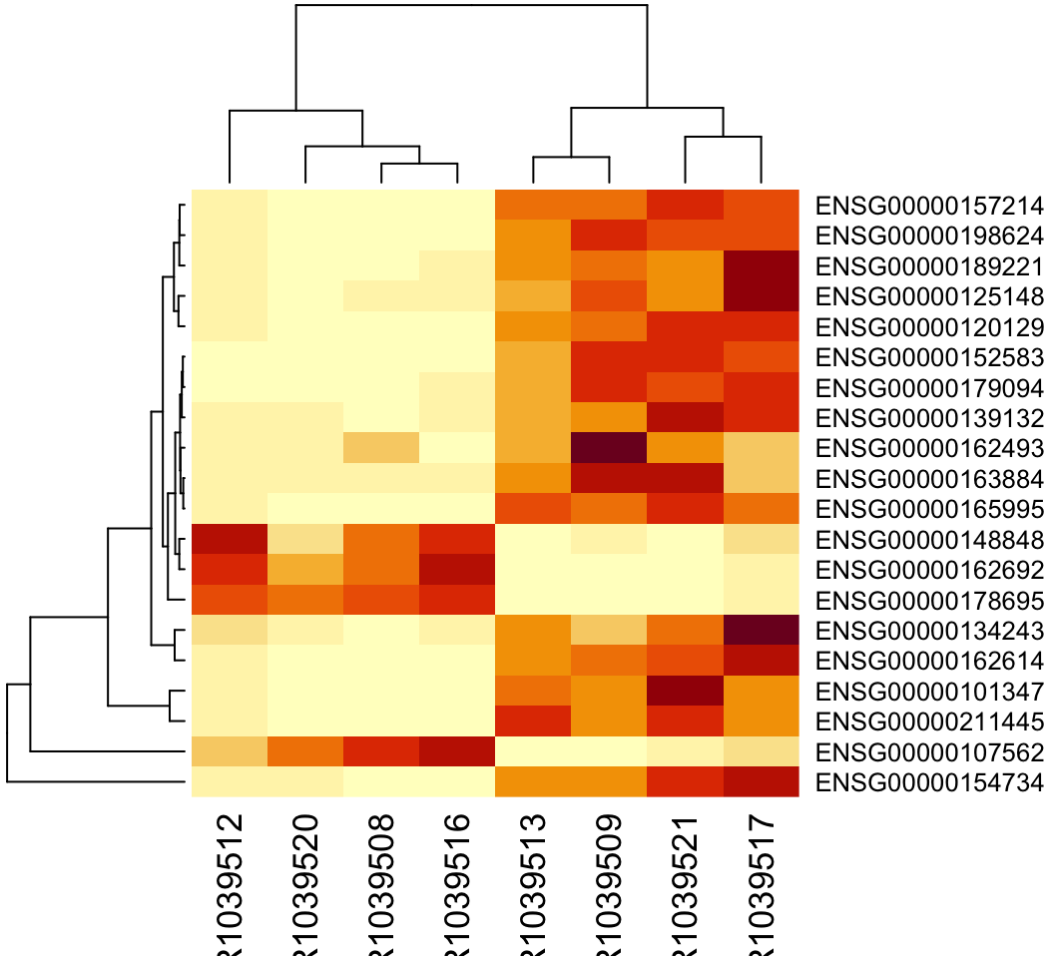
- Add the results to our `SummarizedExperiment`, so that we can manipulate these in a coordianted fashion too.

```
rowData(se) <- cbind(rowData(se), destats)
```

3.2 Heatmap

- Use `order()` and `head()` to identify the row indexes of the top 20 most differentially expressed (based on adjusted P-value) genes.
- Subset the our `SummarizedExperiment` to contain just these rows.
- Dispaly the `assay()` of our subset as a heatmap

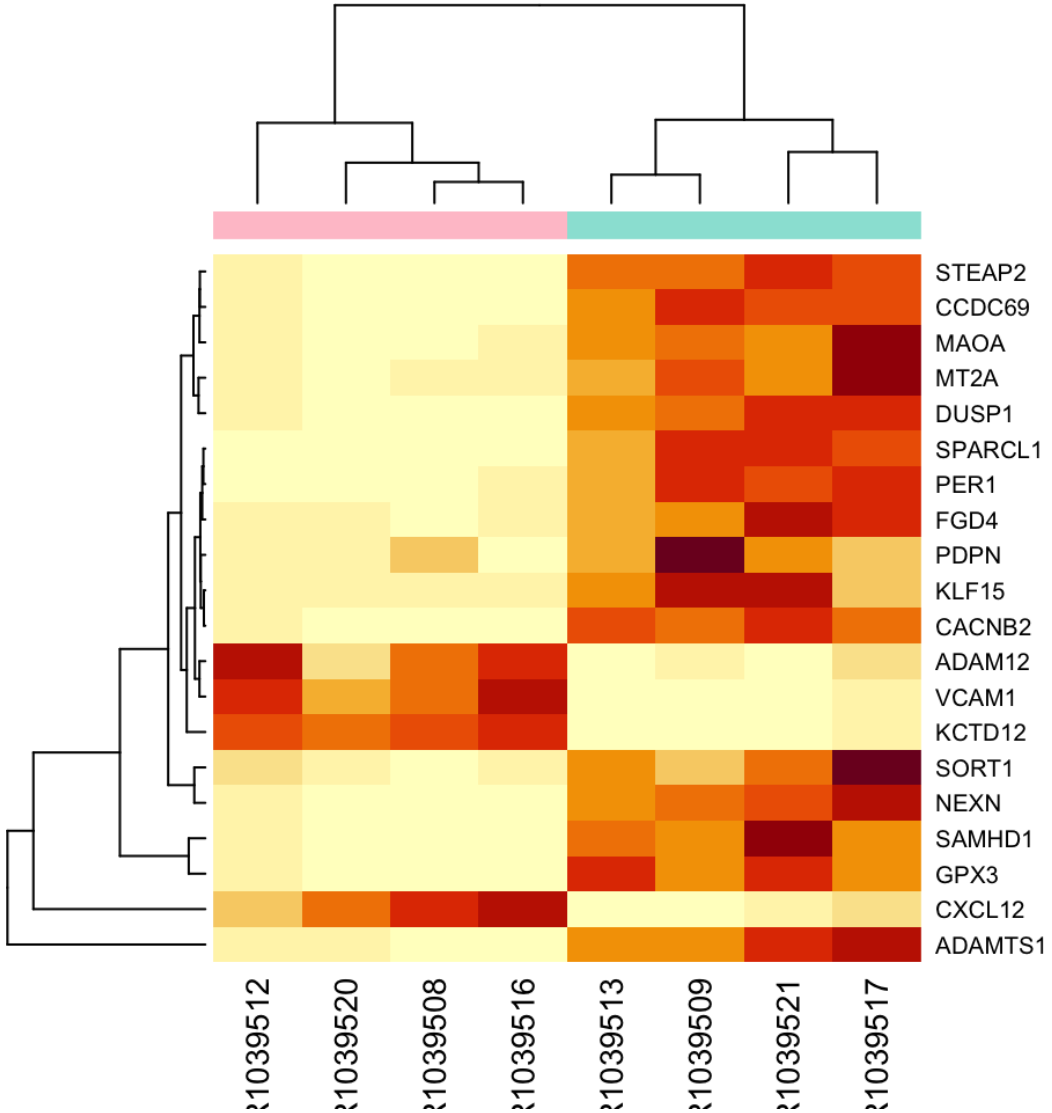
```
top20idx <- head( order(rowData(se)$padj), 20)
top20 <- se[top20idx,]
heatmap(assay(top20))
```



Update row labels and adding information about treatment group.

- Extract the top 20 matrix.
- Update the row names of the matrix with the corresponding gene names.
- Create a vector of colors, one for each sample, with the color determined by the dexamethasone treatment.

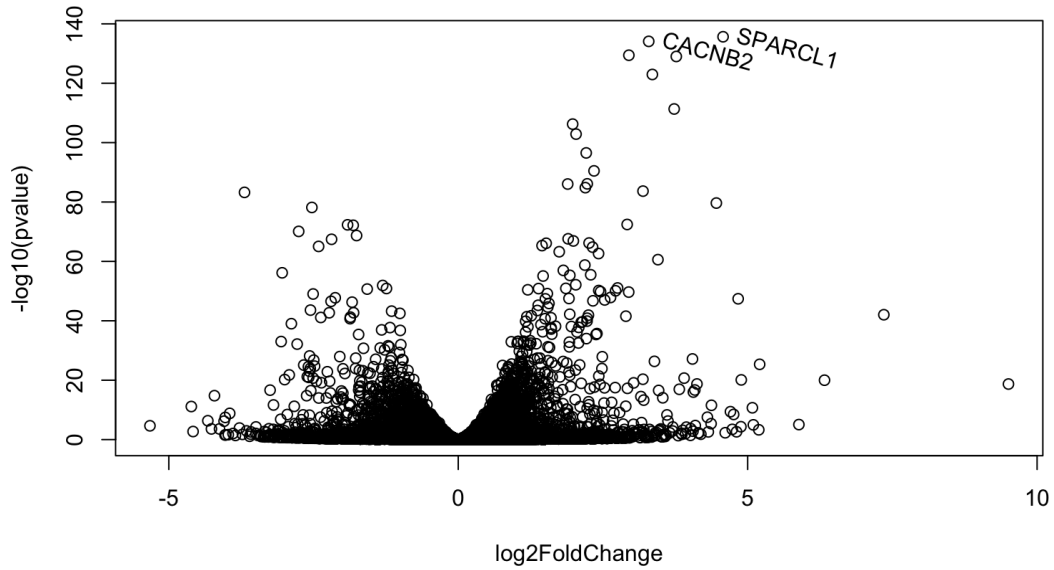
```
m <- assay(top20)
rownames(m) <- rowData(top20)$gene_name
trtcolor <- hcl.colors(2, "Pastel 1")[ colData(top20)$dex ]
heatmap(m, ColSideColors = trtcolor)
```



3.3 Volcano plot

- Plots 'statistical significance' on the Y-axis, 'biological significance' on the X-axis.
- Use `plot()` to create the points
- Use `text()` to add labels to the two most significant genes.

```
plot(-log10(pvalue) ~ log2FoldChange, rowData(se))
label <- with(
    rowData(se),
    ifelse(-log10(pvalue) > 130, gene_name, "")
)
text(
    -log10(pvalue) ~ log2FoldChange, rowData(se),
    label = label, pos = 4, srt=-15
)
```



3.4 Top table and tidy data

Goal

- Provide a concise summary of the 20 most differentially expressed genes.

dplyr (https://cran.r-project.org/package=dplyr) and ‘tidy’ data

- A convenient way to display and manipulate strictly tabular data.
  - ‘long form’ tables where each row represents an observation and each column an attribute measured on the observations.
- `tibble` : a `data.frame` with better display properties

- `%>%`, e.g., `mtcars %>% count(cyl)`: a pipe that takes a tibble (or data.frame) `tbl` on the left and uses it as an argument to a small number of functions like `count()` on the right.
- 'Tidy' functions usually return a `tibble()`, and hence can be chained together.

```
library(dplyr)
```

- Steps below:
    - `as_tibble()`: create a tibble from `rowData(se)`
    - `select()`: select specific columns.
    - `arrange()`: arrange all rows from smallest to largest `padj`
    - `head()`: filter to the first 20 rows

```
rowData(se) %>%
    as_tibble(rownames = "ensembl_id") %>%
    select(ensembl_id, gene_name, baseMean, log2FoldChange, padj) %>%
    arrange(padj) %>%
    head(n = 20)
```

```
## # A tibble: 20 x 5
##    ensembl_id      gene_name baseMean log2FoldChange      padj
##    <chr>           <chr>        <dbl>          <dbl>     <dbl>
##  1 ENSG00000152583 SPARCL1       997.           4.57 4.00e-132
##  2 ENSG00000165995 CACNB2        495.           3.29 7.06e-131
##  3 ENSG00000120129 DUSP1        3409.           2.95 2.20e-126
##  4 ENSG00000101347 SAMHD1      12703.           3.77 4.32e-126
##  5 ENSG00000189221 MAOA         2342.           3.35 3.96e-120
##  6 ENSG00000211445 GPX3        12286.           3.73 1.39e-108
##  7 ENSG00000157214 STEAP2       3009.           1.98 1.48e-103
##  8 ENSG00000162614 NEXN         5393.           2.04 2.98e-100
##  9 ENSG00000125148 MT2A         3656.           2.21 5.81e- 94
## 10 ENSG00000154734 ADAMTS1     30315.           2.35 5.87e- 88
## 11 ENSG00000139132 FGD4         1223.           2.23 1.24e- 83
## 12 ENSG00000162493 PDPN         1100.           1.89 1.32e- 83
## 13 ENSG00000134243 SORT1        5511.           2.20 2.01e- 82
## 14 ENSG00000179094 PER1          777.           3.19 2.73e- 81
## 15 ENSG00000162692 VCAM1         508.          -3.69 6.78e- 81
## 16 ENSG00000163884 KLF15         561.           4.46 2.51e- 77
## 17 ENSG00000178695 KCTD12       2650.          -2.53 7.07e- 76
## 18 ENSG00000198624 CCDC69       2057.           2.92 3.58e- 70
## 19 ENSG00000107562 CXCL12      25136.          -1.91 4.54e- 70
## 20 ENSG00000148848 ADAM12       1365.          -1.81 6.14e- 70
```

- Check out the plyranges (https://bioconductor.org/packages/plyranges) workshop!

4 Summary

4.1 What we've learned

Packages

- Discover at https://bioconductor.org/packages (https://bioconductor.org/packages)
- Install with `BiocManager::install("BiocFileCache")`

- Use with `library(BiocFileCache)`.
- Get help with `?bfcrpath`
- Get more help at https://support.bioconductor.org (https://support.bioconductor.org)
- Mature packages provide access to common sequence analysis data formats, e.g., BED, GTF, FASTQ, BAM, VCF.

Data structures

- Represent and coordinate 'complicated' data.
- Already prevalent in base *R*, e.g., `data.frame()`, `matrix()`.
- `GRanges` for representing genomic ranges
  - 'Accessor' functions `seqnames()`, `start()`, etc for core components
  - `$` or `mcols()$` for annotations
- `SummarizedExperiment` for coordinated manipulation of assay data with row and column annotation.
  - `[,]` to subset assay and annotaions in a coordinated fashion.
  - `assay()`, `rowRanges()`, `rowData()`, `colData()` to access components.

Analysis

- Mature packages like DESeq2 (https://bioconductor.org/packages/DESeq2) provide excellent vignettes, well-defined steps in analysis, integration with other workflow steps, and very robust support.
- Emerging areas are often represented by several packages implementing less complete or certain steps in analysis.

4.2 Next steps

Single-cell RNA-seq: an amazing resource: Orchestarting Single Cell Analysis (https://osca.bioconductor.org) with *Bioconductor*, including the scran (https://bioconductor.org/packages/scran) and scater (https://bioconductor.org/packages/scater) packages.

- Quality control
- Normalization
- Feature selection
- Dimensionality reduction

- Clustering
- Marker gene detection
- Cell type annotation (SingleR (https://bioconductor.org/packages/SingleR)) (this package has a great vignette!)
- Trajectory analysis (destiny (https://bioconductor.org/packages/destiny))
- Gene set enrichment
- Etc.!

Other prominent domains of analysis (check out biocViews (https://bioconductor.org/packages))

- Microarrays – epigenomic, classical expression, copy number
- Annotated variants
- Gene set enrichment
- Flow cytometry
- Proteomics

Participate!

- Get help on the support site (https://support.bioconductor.org).
- Join (https://bioc-community.herokuapp.com/) and use (https://community-bioc.slack.com) the slack community.
- Participate in other conferences (e.g., BioC2020 (https://bioc2020.bioconductor.org/) in Boston, July 29 - 31, 2020).

5 Acknowledgements

Research reported in this presentation was supported by the NCI and NHGRI of the National Institutes of Health under award numbers U24CA232979, U41HG004059, and U24CA180996. The content is solely the responsibility of the authors and does not necessarily represent the official views of the National Institutes of Health.

A portion of this work is supported by the Chan Zuckerberg Initiative DAF, an advised fund of Silicon Valley Community Foundation.

```
## R version 3.6.1 Patched (2019-12-01 r77489)
## Platform: x86_64-apple-darwin17.7.0 (64-bit)
## Running under: macOS High Sierra 10.13.6
##
## Matrix products: default
## BLAS:   /Users/ma38727/bin/R-3-6-branch/lib/libRblas.dylib
## LAPACK: /Users/ma38727/bin/R-3-6-branch/lib/libRlapack.dylib
##
## locale:
## [1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8
##
## attached base packages:
## [1] parallel  stats4    stats     graphics  grDevices utils     datasets
## [8] methods   base
##
## other attached packages:
##  [1] dplyr_0.8.3                 DESeq2_1.26.0
##  [3] SummarizedExperiment_1.16.0 DelayedArray_0.12.0
##  [5] BiocParallel_1.20.0         matrixStats_0.55.0
##  [7] Biobase_2.46.0              rtracklayer_1.46.0
##  [9] GenomicRanges_1.38.0        GenomeInfoDb_1.22.0
## [11] IRanges_2.20.1              S4Vectors_0.24.0
## [13] BiocGenerics_0.32.0         BiocFileCache_1.10.2
## [15] dbplyr_1.4.2                BiocStyle_2.14.0
##
## loaded via a namespace (and not attached):
##  [1] bitops_1.0-6             bit64_0.9-7              RColorBrewer_1.1-2
##  [4] httr_1.4.1               tools_3.6.1              backports_1.1.5
##  [7] utf8_1.1.4               R6_2.4.1                 rpart_4.1-15
## [10] Hmisc_4.3-0              DBI_1.0.0                lazyeval_0.2.2
## [13] colorspace_1.4-1         nnet_7.3-12              tidyselect_0.2.5
## [16] gridExtra_2.3            bit_1.1-14               curl_4.2
## [19] compiler_3.6.1           cli_1.1.0                htmlTable_1.13.2
## [22] bookdown_0.16            scales_1.1.0             checkmate_1.9.4
## [25] genefilter_1.68.0        rappdirs_0.3.1           stringr_1.4.0
## [28] digest_0.6.23            Rsamtools_2.2.1          foreign_0.8-72
## [31] rmarkdown_1.18           XVector_0.26.0           base64enc_0.1-3
## [34] pkgconfig_2.0.3          htmltools_0.4.0          htmlwidgets_1.5.1
## [37] rlang_0.4.2              rstudioapi_0.10          RSQLite_2.1.2
## [40] acepack_1.4.1            RCurl_1.95-4.12          magrittr_1.5
## [43] GenomeInfoDbData_1.2.2   Formula_1.2-3            Matrix_1.2-18
```

```
## [46] Rcpp_1.0.3              munsell_0.5.0           fansi_0.4.0
## [49] lifecycle_0.1.0         stringi_1.4.3           yaml_2.2.0
## [52] zlibbioc_1.32.0         grid_3.6.1              blob_1.2.0
## [55] crayon_1.3.4            lattice_0.20-38         Biostrings_2.54.0
## [58] splines_3.6.1           annotate_1.64.0         locfit_1.5-9.1
## [61] zeallot_0.1.0           knitr_1.26              pillar_1.4.2
## [64] geneplotter_1.64.0      codetools_0.2-16        XML_3.98-1.20
## [67] glue_1.3.1              evaluate_0.14           latticeExtra_0.6-28
## [70] data.table_1.12.6       BiocManager_1.30.10     vctrs_0.2.0
## [73] gtable_0.3.0            purrr_0.3.3             assertthat_0.2.1
## [76] ggplot2_3.2.1           xfun_0.11               xtable_1.8-4
## [79] survival_3.1-7          tibble_2.1.3            GenomicAlignments_1.2
2.1
## [82] AnnotationDbi_1.48.0    memoise_1.1.0           cluster_2.1.0
```