Bioconductor

BIOF 339

December 11th, 2018

Bioconductor

Bioconductor

Bioconductor provides tools for the analysis and comprehension of high-throughput genomic and biological data, using R.

- 1476 packages
- Covers the bioinformatic pipeline
- Software
- Annotation
- Experiments

Explore Bioconductor website

Installing Bioconductor packages

This is different from the usual `install.packages`. If you are using a version of R less than 3.5, the method is:

```r
source('http://bioconductor.org/biocLite.R')
biocLite(c('Biobase','limma','hgu95av2.db','Biostrings'))
```

Otherwise (for R version 3.5 and later):

```r
install.packages("BiocManager")
BiocManager::install(c('Biobase','limma','hgu95av2.db','Biostrings'))
```

DNA sequences

```r
library(Biostrings)
dna <- DNAStringSet(c("AACAT", "GGCGCCT"))
reverseComplement(dna)
```

```
#    A DNAStringSet instance of length 2
#      width seq
#   [1]     5 ATGTT
#   [2]     7 AGGCGCC
```

DNA sequences

```r
library(Biostrings)
data("phiX174Phage")
phiX174Phage
```

```
#    A DNAStringSet instance of length 6
#      width seq                                               names
#   [1]  5386 GAGTTTTATCGCTTCCATGAC...ATTGGCGTATCCAACCTGCA Genbank
#   [2]  5386 GAGTTTTATCGCTTCCATGAC...ATTGGCGTATCCAACCTGCA RF70s
#   [3]  5386 GAGTTTTATCGCTTCCATGAC...ATTGGCGTATCCAACCTGCA SS78
#   [4]  5386 GAGTTTTATCGCTTCCATGAC...ATTGGCGTATCCAACCTGCA Bull
#   [5]  5386 GAGTTTTATCGCTTCCATGAC...ATTGGCGTATCCAACCTGCA G97
#   [6]  5386 GAGTTTTATCGCTTCCATGAC...ATTGGCGTATCCAACCTGCA NEB03
```

DNA sequences

```
letterFrequency(phiX174Phage, 'GC', as.prob=TRUE)
```

```
#              G|C
#   [1,] 0.4476420
#   [2,] 0.4472707
#   [3,] 0.4472707
#   [4,] 0.4470850
#   [5,] 0.4472707
#   [6,] 0.4470850
```

Data in Bioconductor

The basic structure for expression data in a Bioconductor pipeline is the `ExpressionSet`

```
library(Biobase)
data("sample.ExpressionSet")
str(sample.ExpressionSet)
```

```
#  Formal class 'ExpressionSet' [package "Biobase"] with 7 slots
#    ..@ experimentData   :Formal class 'MIAME' [package "Biobase"] with 13 
#    .. .. ..@ name             : chr "Pierre Fermat"
#    .. .. ..@ lab              : chr "Francis Galton Lab"
#    .. .. ..@ contact          : chr "pfermat@lab.not.exist"
#    .. .. ..@ title            : chr "Smoking-Cancer Experiment"
#    .. .. ..@ abstract         : chr "An example object of expression set (
#    .. .. ..@ url              : chr "www.lab.not.exist"
#    .. .. ..@ pubMedIds        : chr ""
#    .. .. ..@ samples          : list()
#    .. .. ..@ hybridizations   : list()
#    .. .. ..@ normControls     : list()
```

Differences with usual R

Instead of storing data in named lists, ExpressionSet objects store data in **slots**, and we can see what the slots are with `slotNames`:

```
slotNames(sample.ExpressionSet)
```

```
#  [1] "experimentData"    "assayData"          "phenoData"
#  [4] "featureData"       "annotation"         "protocolData"
#  [7] ".__classVersion__"
```

Differences with usual R

You can access these slots using `@`, instead of the usual `$`:

```
sample.ExpressionSet@phenoData
```

```
#  An object of class 'AnnotatedDataFrame'
#    sampleNames: A B ... Z (26 total)
#    varLabels: sex type score
#    varMetadata: labelDescription
```

Differences with usual R

However, it's much easier to go with the built-in functions

```
pData(sample.ExpressionSet)
```

```
#       sex    type score
#   A Female Control  0.75
#   B   Male    Case  0.40
#   C   Male Control  0.73
#   D   Male    Case  0.42
#   E Female    Case  0.93
#   F   Male Control  0.22
#   G   Male    Case  0.96
#   H   Male    Case  0.79
#   I Female    Case  0.37
#   J   Male Control  0.63
#   K   Male    Case  0.26
#   L Female Control  0.36
#   M   Male    Case  0.41
#   N   Male    Case  0.80
```

Differences with usual R

```
head(exprs(sample.ExpressionSet))
```

```
#                         A         B        C        D        E       F
#  AFFX-MurIL2_at  192.7420  85.75330 176.7570 135.5750 64.49390 76.3569
#  AFFX-MurIL10_at  97.1370 126.19600  77.9216  93.3713 24.39860 85.5088
#  AFFX-MurIL4_at   45.8192   8.83135  33.0632  28.7072  5.94492 28.2925
#  AFFX-MurFAS_at   22.5445   3.60093  14.6883  12.3397 36.86630 11.2568
#  AFFX-BioB-5_at   96.7875  30.43800  46.1271  70.9319 56.17440 42.6756
#  AFFX-BioB-M_at   89.0730  25.84610  57.2033  69.9766 49.58220 26.1262
#                         G       H       I         J        K       L
#  AFFX-MurIL2_at  160.5050 65.9631 56.9039 135.60800 63.44320 78.2126
#  AFFX-MurIL10_at  98.9086 81.6932 97.8015  90.48380 70.57330 94.5418
#  AFFX-MurIL4_at   30.9694 14.7923 14.2399  34.48740 20.35210 14.1554
#  AFFX-MurFAS_at   23.0034 16.2134 12.0375   4.54978  8.51782 27.2852
#  AFFX-BioB-5_at   86.5156 30.7927 19.7183  46.35200 39.13260 41.7698
#  AFFX-BioB-M_at   75.0083 42.3352 41.1207  91.53070 39.91360 49.8397
#                        M       N       O       P        Q        R
#  AFFX-MurIL2_at  83.0943 89.3372 91.0615 95.9377 179.8450 152.4670
#  AFFX-MurIL10_at 75.3455 68.5827 87.4050 84.4581  87.6806 108.0320
```

Accessing Features (probes)

```r
affyIDs <- rownames(sample.ExpressionSet@featureData)
affyIDs[200:203]
```

```
#  [1] "31439_f_at" "31440_at"   "31441_at"   "31442_at"
```

The IDs for affy probes are singularly unhelpful if we wish to analyze our expression data with respect to genes or transcripts. To address this problem here (and in other data sets) we can turn to the “biomaRt” library.

BiomaRt

BiomaRt

“The biomaRt package, provides an interface to a growing collection of databases implementing the BioMart software suite.”

To use the biomaRt package, we will have to first select a BioMart database and a dataset from that database to query.

Selecting a Database

```r
# BiocManager::install('biomaRt')
library("biomaRt")
ensemblDatabase <- useMart("ensembl")
```

Selecting a Dataset

```
searchDatasets(mart = ensemblDatabase, pattern = "Human")
```

```
#                   dataset              description    version
#  54 hsapiens_gene_ensembl Human genes (GRCh38.p12) GRCh38.p12
```

```
ensemblHumanData <- useMart("ensembl",dataset="hsapiens_gene_ensembl")
```

Identifying Attributes

```
searchAttributes(mart = ensemblHumanData, pattern = "affy")
```

```
#                     name                    description         page
#   95          affy_hc_g110         AFFY HC G110 probe feature_page
#   96         affy_hg_focus        AFFY HG Focus probe feature_page
#   97         affy_hg_u133a        AFFY HG U133A probe feature_page
#   98       affy_hg_u133a_2      AFFY HG U133A 2 probe feature_page
#   99         affy_hg_u133b        AFFY HG U133B probe feature_page
#   100   affy_hg_u133_plus_2   AFFY HG U133 Plus 2 probe feature_page
#   101          affy_hg_u95a         AFFY HG U95A probe feature_page
#   102        affy_hg_u95av2       AFFY HG U95Av2 probe feature_page
#   103          affy_hg_u95b         AFFY HG U95B probe feature_page
#   104          affy_hg_u95c         AFFY HG U95C probe feature_page
#   105          affy_hg_u95d         AFFY HG U95D probe feature_page
#   106          affy_hg_u95e         AFFY HG U95E probe feature_page
#   107          affy_hta_2_0         AFFY HTA 2 0 probe feature_page
#   108   affy_huex_1_0_st_v2   AFFY HuEx 1 0 st v2 probe feature_page
#   109         affy_hugenefl       AFFY HuGeneFL probe feature_page
#   110 affy_hugene_1_0_st_v1 AFFY HuGene 1 0 st v1 probe feature_page
```

Identifying Attributes (Part 2)

```
searchAttributes(mart = ensemblHumanData, pattern = "hgnc")
```

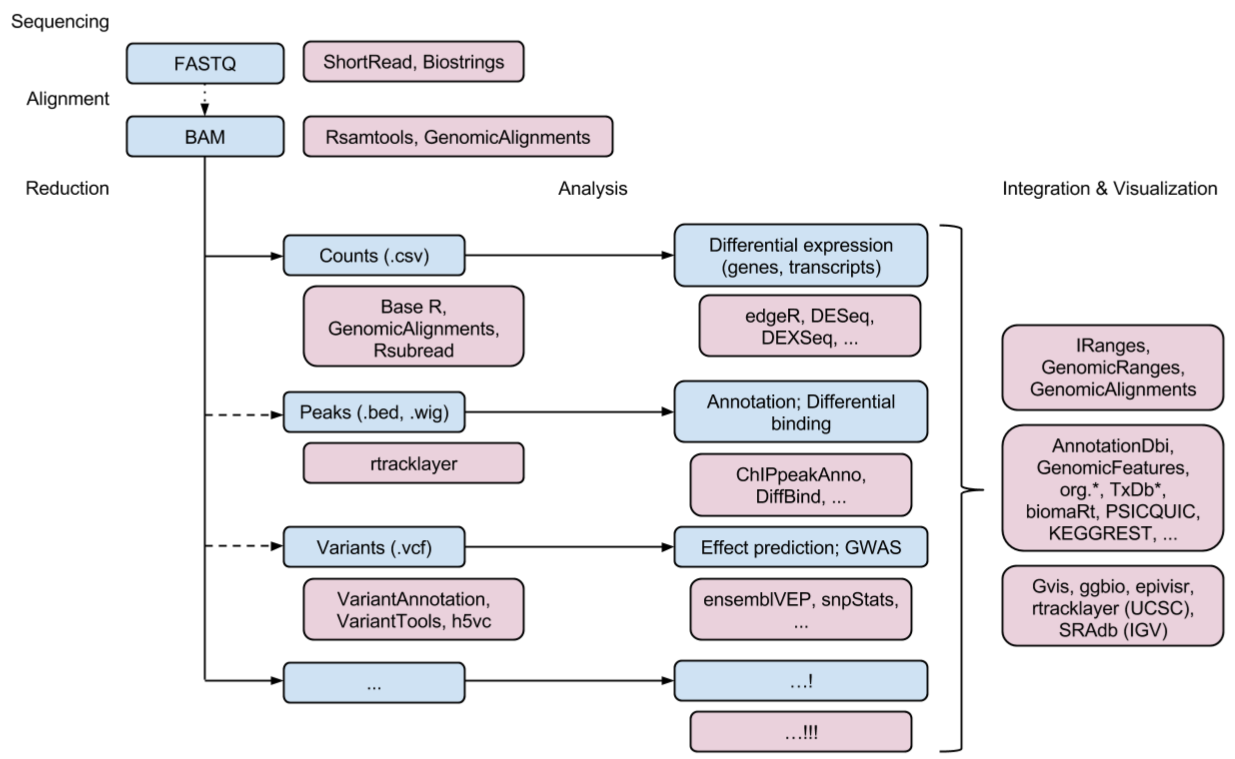
```
#                name                 description         page
#   58          hgnc_id                    HGNC ID feature_page
#   59      hgnc_symbol                HGNC symbol feature_page
#   60 hgnc_trans_name HGNC transcript name ID feature_page
```

Querying the Dataset

```
getBM(attributes = c('affy_hg_u95av2', 'hgnc_symbol'),
      filters = "affy_hg_u95av2",
      values = affyIDs[200:203],
      mart = ensemblHumanData)
```

```
#    affy_hg_u95av2 hgnc_symbol
#  1       31440_at       TCF7
#  2     31439_f_at       RHCE
#  3     31439_f_at        RHD
```

Bioconductor ecosystem


Sequencing
FASTQ
ShortRead, Biostrings
Alignment
BAM
Rsamtools, GenomicAlignments
Reduction
Analysis
Integration & Visualization
Counts (.csv)
Base R, GenomicAlignments, Rsubread
Differential expression (genes, transcripts)
edgeR, DESeq, DEXSeq, ...
Peaks (.bed, .wig)
rtracklayer
Annotation; Differential binding
ChIPpeakAnno, DiffBind, ...
Variants (.vcf)
VariantAnnotation, VariantTools, h5vc
Effect prediction; GWAS
ensemblVEP, snpStats, ...
...
...!
...!!!
IRanges, GenomicRanges, GenomicAlignments
AnnotationDbi, GenomicFeatures, org.*, TxDb*, biomaRt, PSICQUIC, KEGGREST, ...
Gvis, ggbio, epivisr, rtracklayer (UCSC), SRAdb (IGV)

Taken from <a href = “http://bioconductor.org/help/course-materials/2017/OSU/B1_Bioconductor_intro.html”, target="_blank">Morgan’s Bioconductor Tutorial

A RNA-Seq pipeline (link)

Goals

- Exploratory data analysis
- Differential expression analysis with DESeq2
- Visualization
- We will start after reads have been aligned to a reference genome and reads overlapping known genes have been counted

The experiment

- In the experiment, four primary human airway smooth muscle cell lines were treated with 1 micromolar dexamethasone for 18 hours.
- For each of the four cell lines, we have a treated and an untreated sample.

Start with prepared SummarizedExperiment

```r
# BiocManager::install('airway')
library(airway)
data(airway)
se <- airway
head(assay(se))
```

```
#                   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
```

Metadata for experiment

```
colData(se)
```

```
#  DataFrame with 8 rows and 9 columns
#             SampleName     cell      dex    albut        Run avgLength
#               <factor> <factor> <factor> <factor>   <factor> <integer>
#  SRR1039508 GSM1275862   N61311    untrt    untrt SRR1039508       126
#  SRR1039509 GSM1275863   N61311      trt    untrt SRR1039509       126
#  SRR1039512 GSM1275866  N052611    untrt    untrt SRR1039512       126
#  SRR1039513 GSM1275867  N052611      trt    untrt SRR1039513        87
#  SRR1039516 GSM1275870  N080611    untrt    untrt SRR1039516       120
#  SRR1039517 GSM1275871  N080611      trt    untrt SRR1039517       126
#  SRR1039520 GSM1275874  N061011    untrt    untrt SRR1039520       101
#  SRR1039521 GSM1275875  N061011      trt    untrt SRR1039521        98
#             Experiment    Sample    BioSample
#               <factor>  <factor>     <factor>
#  SRR1039508  SRX384345 SRS508568 SAMN02422669
#  SRR1039509  SRX384346 SRS508567 SAMN02422675
#  SRR1039512  SRX384349 SRS508571 SAMN02422678
#  SRR1039513  SRX384350 SRS508572 SAMN02422670
```

Genomic ranges over which counting occurred

```
rowRanges(se)
```

```
#  GRangesList object of length 64102:
#  $ENSG00000000003
#  GRanges object with 17 ranges and 2 metadata columns:
#         seqnames            ranges strand |   exon_id       exon_name
#            <Rle>         <IRanges>  <Rle> | <integer>     <character>
#     [1]        X 99883667-99884983      - |    667145 ENSE00001459322
#     [2]        X 99885756-99885863      - |    667146 ENSE00000868868
#     [3]        X 99887482-99887565      - |    667147 ENSE00000401072
#     [4]        X 99887538-99887565      - |    667148 ENSE00001849132
#     [5]        X 99888402-99888536      - |    667149 ENSE00003554016
#     ...      ...               ...    ... .       ...             ...
#    [13]        X 99890555-99890743      - |    667156 ENSE00003512331
#    [14]        X 99891188-99891686      - |    667158 ENSE00001886883
#    [15]        X 99891605-99891803      - |    667159 ENSE00001855382
```

Create a DESeqDataSet

```r
# BiocManager::install('DESeq2')
library("DESeq2")
dds <- DESeqDataSet(se, design = ~ cell + dex)
dds
```

```
#  class: DESeqDataSet
#  dim: 64102 8
#  metadata(2): '' version
#  assays(1): counts
#  rownames(64102): ENSG00000000003 ENSG00000000005 ... LRG_98 LRG_99
#  rowData names(0):
#  colnames(8): SRR1039508 SRR1039509 ... SRR1039520 SRR1039521
#  colData names(9): SampleName cell ... Sample BioSample
```

Fix factor label

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

Run differential expression pipeline

```
dds <- DESeq(dds)
```

```
#  estimating size factors

#  estimating dispersions

#  gene-wise dispersion estimates

#  mean-dispersion relationship

#  final dispersion estimates

#  fitting model and testing
```

Extracting results

```
(res <- results(dds))
```

```
#  log2 fold change (MLE): dex trt vs untrt
#  Wald test p-value: dex trt vs untrt
#  DataFrame with 64102 rows and 6 columns
#                          baseMean     log2FoldChange              lfcSE
#                         <numeric>          <numeric>          <numeric>
#  ENSG00000000003 708.602169691234  -0.381253887429316 0.100654430187038
#  ENSG00000000005                0                  NA                NA
#  ENSG00000000419 520.297900552084   0.206812715390385 0.112218674572541
#  ENSG00000000457 237.163036796015  0.0379205923945968 0.143444716340173
#  ENSG00000000460 57.9326331250967 -0.0881676962637897 0.287141995230742
#  ...                          ...                 ...               ...
#  LRG_94                         0                  NA                NA
#  LRG_96                         0                  NA                NA
#  LRG_97                         0                  NA                NA
#  LRG_98                         0                  NA                NA
#  LRG_99                         0                  NA                NA
#                              stat              pvalue              pad
```

Summarizing results

```
summary(res)
```

```
#
#  out of 33469 with nonzero total read count
#  adjusted p-value < 0.1
#  LFC > 0 (up)       : 2604, 7.8%
#  LFC < 0 (down)     : 2215, 6.6%
#  outliers [1]       : 0, 0%
#  low counts [2]     : 15441, 46%
#  (mean count < 5)
#  [1] see 'cooksCutoff' argument of ?results
#  [2] see 'independentFiltering' argument of ?results
```

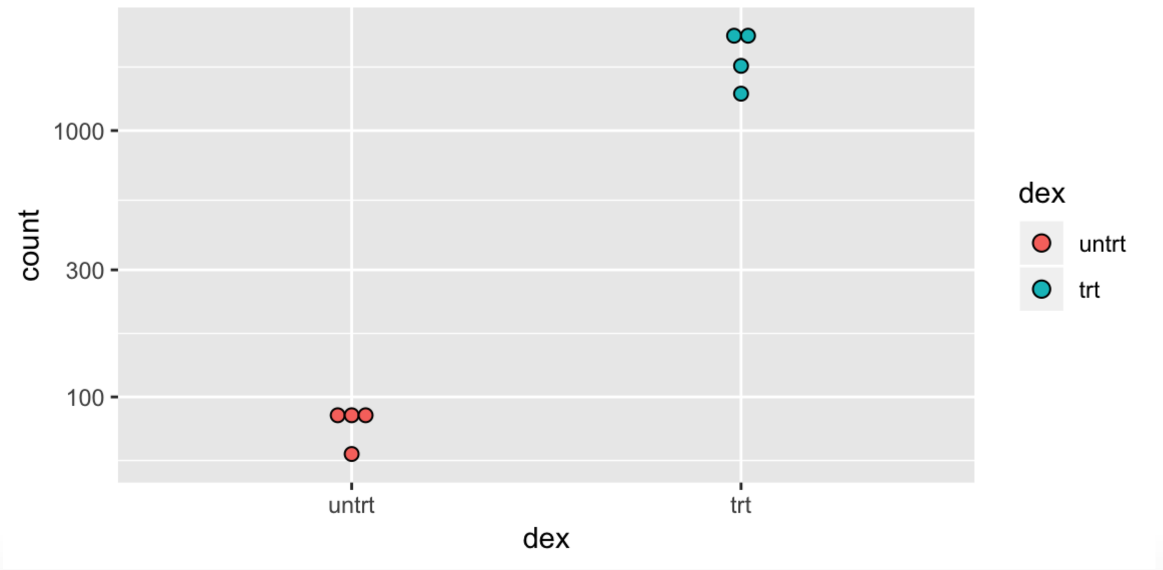
Summarizing results

```
library(tidyverse)
as.data.frame(res) %>%
  rownames_to_column(var = 'ID') %>%
  filter(padj < 0.1) %>%
  arrange(desc(abs(log2FoldChange))) %>% head()
```

```
#                ID   baseMean log2FoldChange     lfcSE      stat
#   1 ENSG00000179593  67.243048       9.505972 1.0545022  9.014654
#   2 ENSG00000109906 385.071029       7.352628 0.5363902 13.707610
#   3 ENSG00000250978  56.318194       6.327384 0.6777974  9.335214
#   4 ENSG00000132518   5.654654       5.885112 1.3240432  4.444803
#   5 ENSG00000128285   6.624741      -5.325905 1.2578165 -4.234247
#   6 ENSG00000127954 286.384119       5.207160 0.4930828 10.560419
#          pvalue         padj
#   1 1.974931e-19 1.253664e-17
#   2 9.141988e-43 2.257695e-40
#   3 1.007873e-20 7.210289e-19
#   4 8.797236e-06 1.000609e-04
```

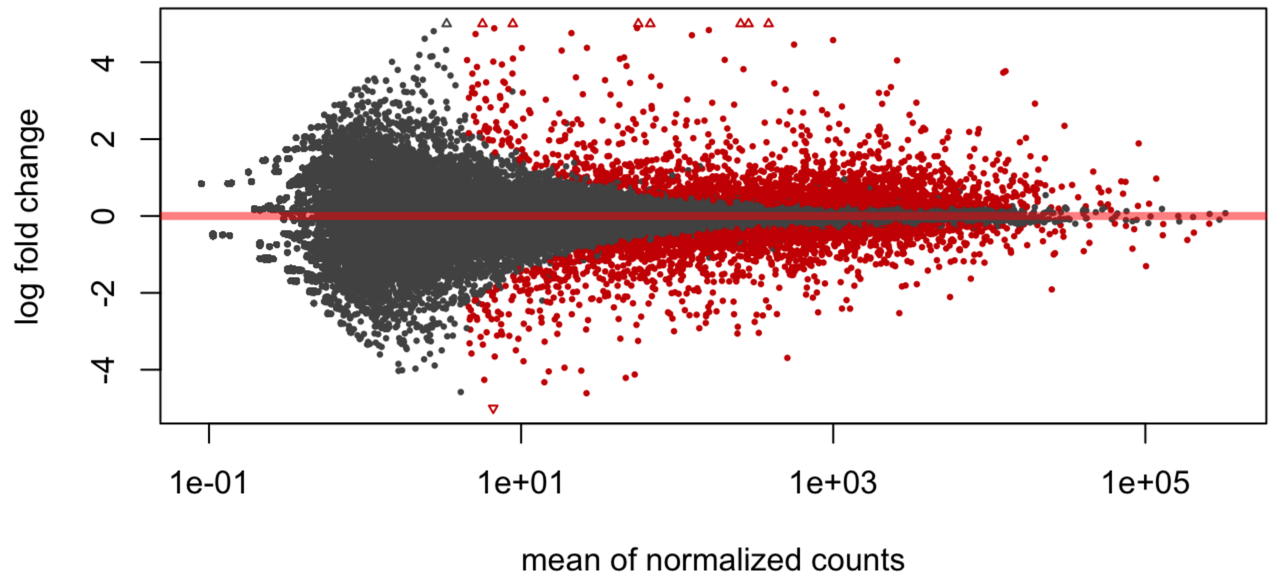
A visualization

```r
topGene <- rownames(res)[which.min(res$padj)]
dat <- plotCounts(dds, gene=topGene, intgroup=c("dex"), returnData=TRUE)
ggplot(dat, aes(x = dex, y = count, fill=dex))+
  geom_dotplot(binaxis='y', stackdir='center')+
  scale_y_log10()
```



Another visualization

```
plotMA(res, ylim=c(-5,5))
```



Making a heatmap

Heatmaps

There are several ways of doing heatmaps in R:

- http://sebastianraschka.com/Articles/heatmaps_in_r.html
- https://plot.ly/r/heatmaps/
- http://moderndata.plot.ly/interactive-heat-maps-for-r/
- http://www.siliconcreek.net/r/simple-heatmap-in-r-with-ggplot2
- https://rud.is/b/2016/02/14/making-faceted-heatmaps-with-ggplot2/

Some example data

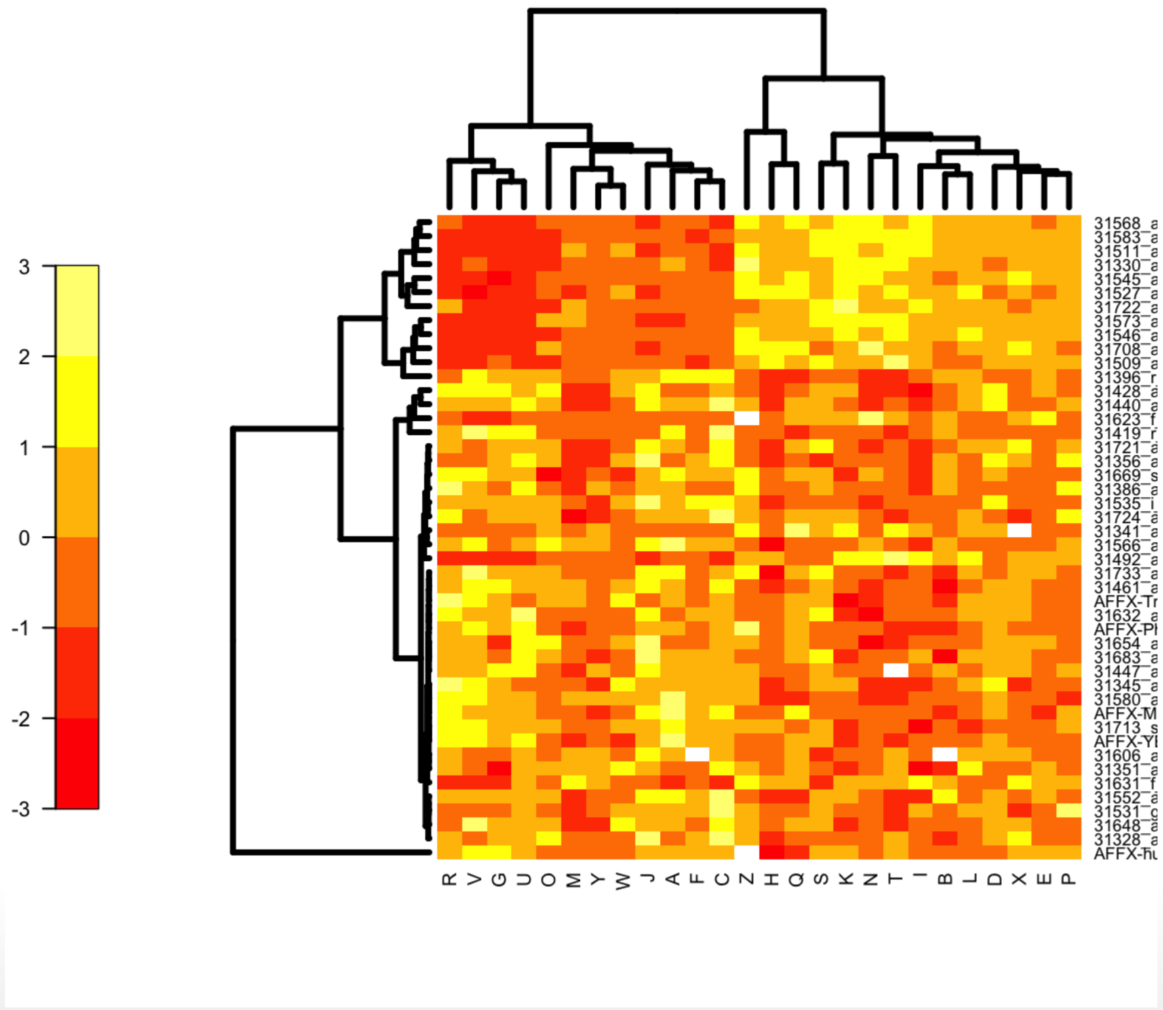
```r
library(Biobase)
data(sample.ExpressionSet)
exdat <-  sample.ExpressionSet
library(limma)
design1 <- model.matrix(~type, data=pData(exdat))
lm1 <- lmFit(exprs(exdat), design1)
lm1 <- eBayes(lm1) # compute linear model for each probeset
geneID <- rownames(topTable(lm1, coef=2, num=100, adjust='none',p.value=0.05
exdat2 <- exdat[geneID,] # Keep features with p-values < 0.05
exdat2
```

```
#  ExpressionSet (storageMode: lockedEnvironment)
#  assayData: 46 features, 26 samples
#    element names: exprs, se.exprs
#  protocolData: none
#  phenoData
#    sampleNames: A B ... Z (26 total)
#    varLabels: sex type score
#    varMetadata: labelDescription
```

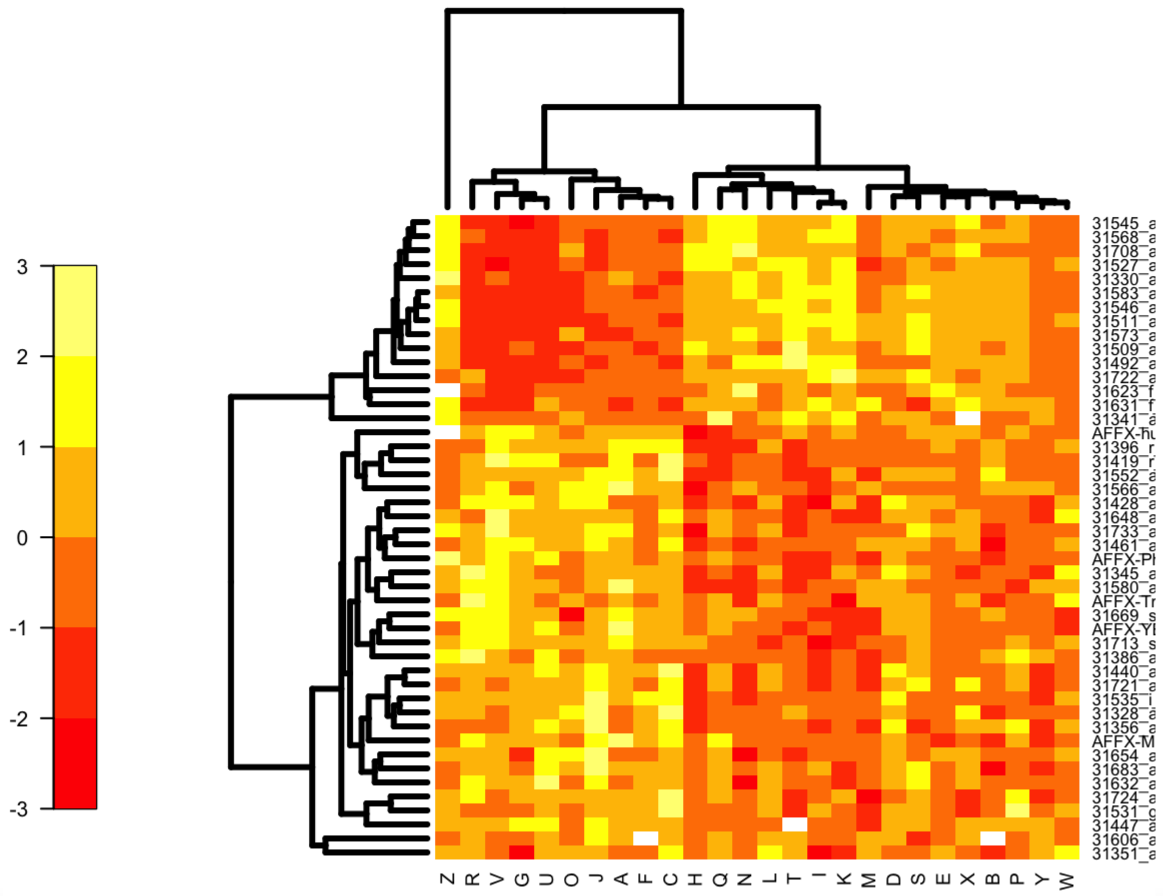
Heatmaps using `Heatplus`

```r
source('http://bioconductor.org/biocLite.R')
biocLite('Heatplus')
```

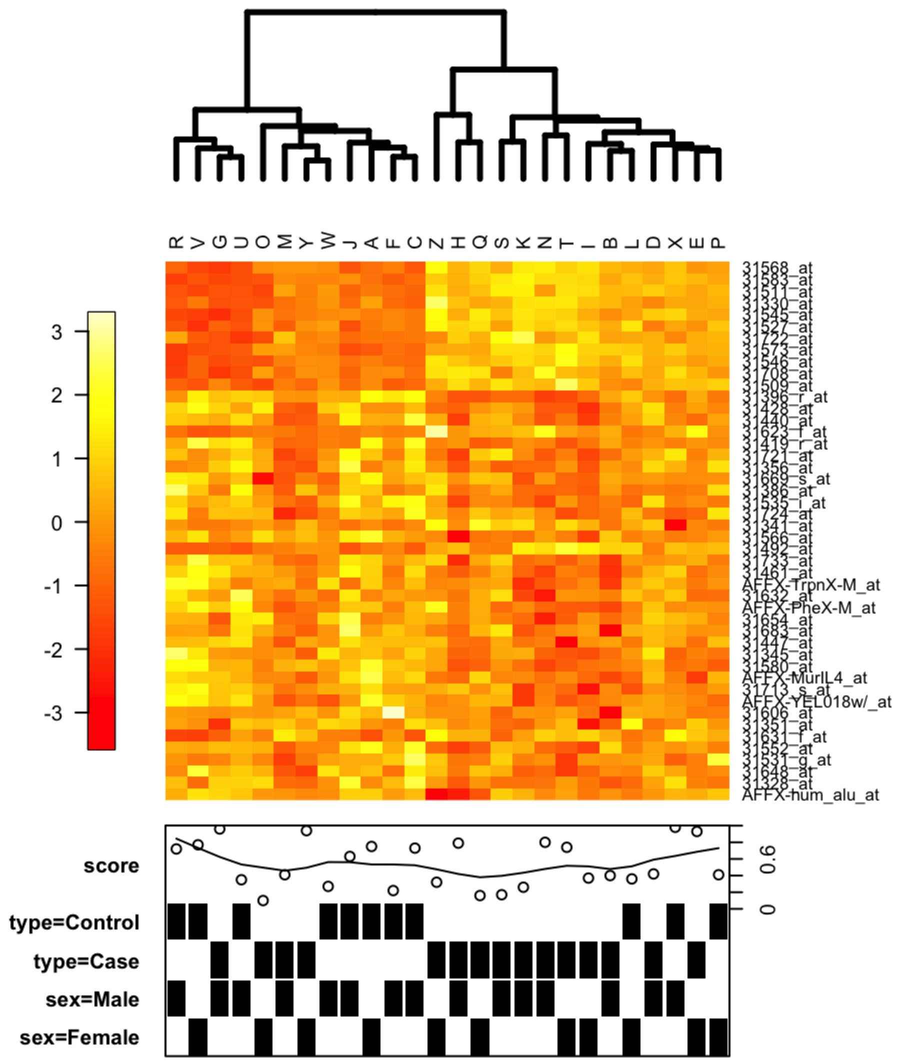
```r
# BiocManager::install('Heatplus')
library(Heatplus)
reg1 <- regHeatmap(exprs(exdat2), legend=2, col=heat.colors,
                   breaks=-3:3)
plot(reg1)
```


3
2
1
0
-1
-2
-3
R
V
G
U
O
M
Y
W
J
A
F
C
Z
H
Q
S
K
N
T
I
B
L
D
X
E
P
31568_a
31583_a
31511_a
31330_a
31545_a
31527_a
31722_a
31573_a
31546_a
31708_a
31509_a
31396_r
31428_a
31440_a
31623_f
31419_r
31721_a
31356_a
31669_s
31386_a
31535_i
31724_a
31341_a
31566_a
31492_a
31733_a
31461_a
AFFX-Tr
31632_a
AFFX-Pl
31654_a
31683_a
31447_a
31345_a
31580_a
AFFX-M
31713_s
AFFX-YI
31606_a
31351_a
31631_f
31552_a
31531_g
31648_a
31328_a
AFFX-hu

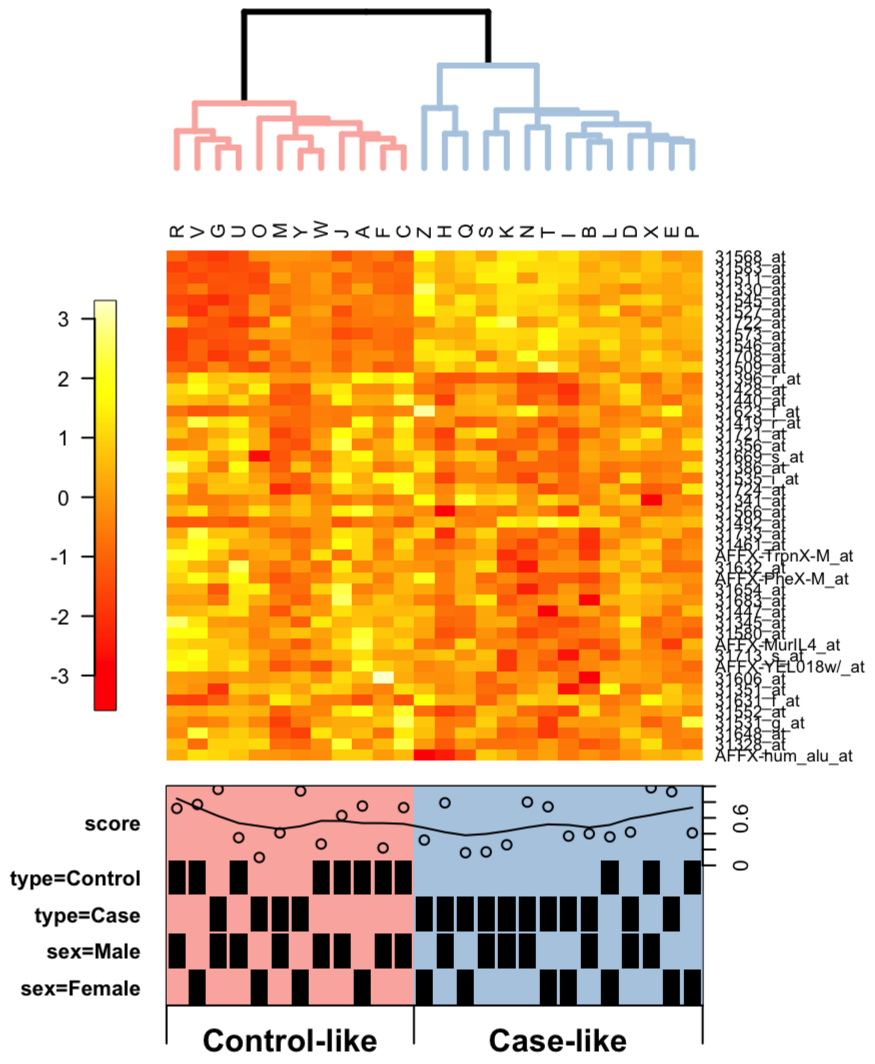
```r
corrdist <- function(x) as.dist(1-cor(t(x)))
hclust.avl <-  function(x) hclust(x, method='average')
reg2 <- regHeatmap(exprs(exdat2), legend=2, col=heat.colors,
                   breaks=-3:3,
                   dendrogram = list(clustfun=hclust.avl, distfun=corrdist))
plot(reg2)
```


3
2
1
0
-1
-2
-3
Z R V G U O J A F C H Q N L T I K M D S E X B P Y W
31545_a
31568_a
31708_a
31527_a
31330_a
31583_a
31546_a
31511_a
31573_a
31509_a
31492_a
31722_a
31623_f
31631_f
31341_a
AFFX-hu
31396_r
31419_r
31552_a
31566_a
31428_a
31648_a
31733_a
31461_a
AFFX-Pl
31345_a
31580_a
AFFX-Tr
31669_s
AFFX-Yl
31713_s
31386_a
31440_a
31721_a
31535_i
31328_a
31356_a
AFFX-M
31654_a
31683_a
31632_a
31724_a
31531_g
31447_a
31606_a
31351_a

```r
ann1 <- annHeatmap(exprs(exdat2), ann=pData(exdat2), col = heat.colors)
plot(ann1)
```

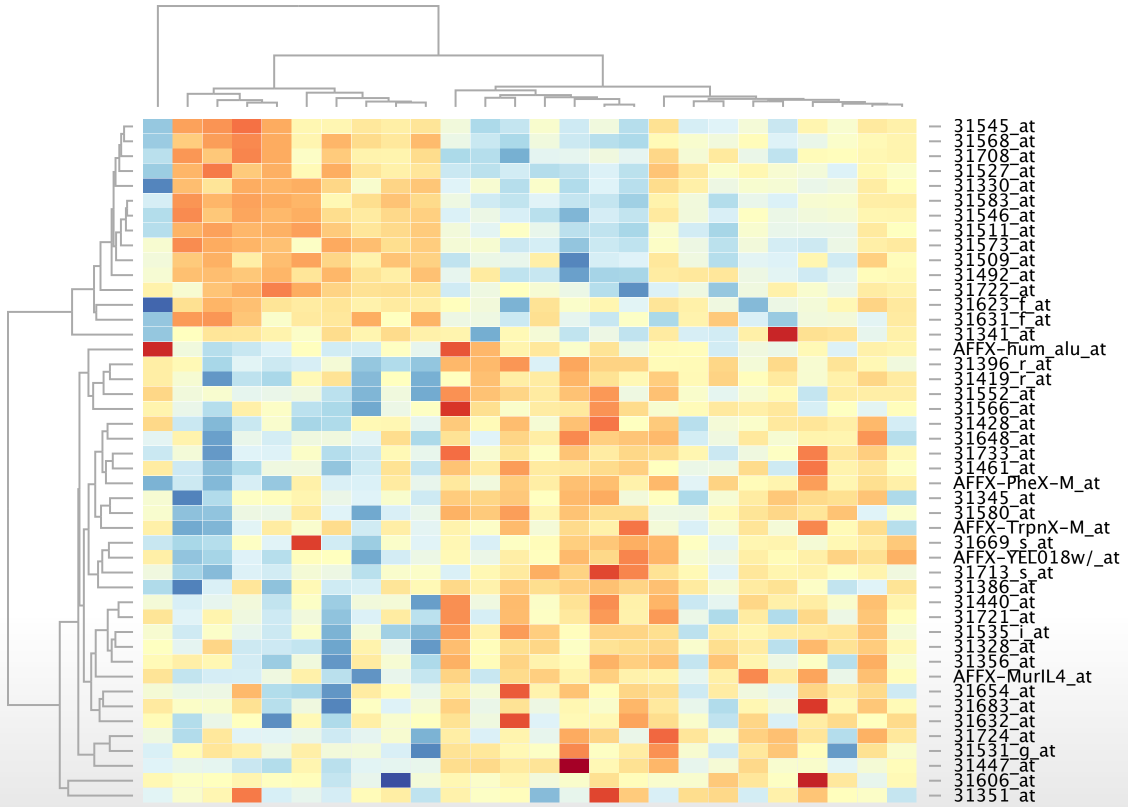

R V G U O M Y W J A F C Z H Q S K N T I B L D X E P
3
2
1
0
-1
-2
-3
31568_at
31583_at
31511_at
31330_at
31545_at
31527_at
31722_at
31573_at
31546_at
31708_at
31509_at
31396_r_at
31428_at
31440_at
31623_f_at
31419_r_at
31721_at
31356_at
31669_s_at
31386_at
31535_i_at
31724_at
31341_at
31566_at
31492_at
31733_at
31461_at
AFFX-TrpnX-M_at
31632_at
AFFX-PheX-M_at
31654_at
31683_at
31447_at
31345_at
31580_at
AFFX-MurIL4_at
31713_s_at
AFFX-YEL018w/_at
31606_at
31351_at
31631_f_at
31552_at
31531_g_at
31648_at
31328_at
AFFX-hum_alu_at
score
0.6
0
type=Control
type=Case
sex=Male
sex=Female

```r
ann2 <- annHeatmap(exprs(exdat2), ann=pData(exdat2), col = heat.colors,
                   cluster = list(cuth=7500,
                                  label=c('Control-like','Case-like')))
plot(ann2)
```


R V G U O M Y W J A F C Z H Q S K N T I B L D X E P
3
2
1
0
-1
-2
-3
31568_at
31583_at
31511_at
31330_at
31545_at
31527_at
31722_at
31573_at
31546_at
31708_at
31509_at
31396_r_at
31428_at
31440_at
31623_f_at
31419_r_at
31721_at
31356_at
31669_s_at
31386_at
31535_i_at
31724_at
31341_at
31566_at
31492_at
31733_at
31461_at
AFFX-TrpnX-M_at
31632_at
AFFX-PheX-M_at
31654_at
31683_at
31447_at
31345_at
31580_at
AFFX-MurIL4_at
31713_s_at
AFFX-YEL018w/_at
31606_at
31351_at
31631_f_at
31552_at
31531_g_at
31648_at
31328_at
AFFX-hum_alu_at
score
type=Control
type=Case
sex=Male
sex=Female
0.6
0
Control-like
Case-like

Put your mouse over each point :)

```r
# install.packages('d3heatmap')
library(d3heatmap)
d3heatmap(exprs(exdat2), distfun = corrdist,
          hclustfun = hclust.avl, scale='row')
```


31545_at
31568_at
31708_at
31527_at
31330_at
31583_at
31546_at
31511_at
31573_at
31509_at
31492_at
31722_at
31623_f_at
31631_f_at
31341_at
AFFX-hum_alu_at
31396_r_at
31419_r_at
31552_at
31566_at
31428_at
31648_at
31733_at
31461_at
AFFX-PheX-M_at
31345_at
31580_at
AFFX-TrpnX-M_at
31669_s_at
AFFX-YEL018w/_at
31713_s_at
31386_at
31440_at
31721_at
31535_i_at
31328_at
31356_at
AFFX-MurIL4_at
31654_at
31683_at
31632_at
31724_at
31531_g_at
31447_at
31606_at
31351_at