Practical - intermediate

Aurelien Ginolhac

2nd June 2016

Project - set-up

- Create a new project in a meaningful folder name on your computer (such as `R_workshop/day1-intermediate`) using the project manager utility on the upper-right part of the rstudio window.
- Check if you have all those libraries installed

```r
library("tidyr")
library("dplyr", warn.conflicts = FALSE)
library("ggplot2")
library("broom")
suppressPackageStartupMessages(library("GEOquery")) # bioconductor is verbose
theme_set(theme_bw(14)) # if you wish to get this theme by default
```

Aim

Working with GEO datasets could be an hassle and you are going to experience it. Extensive manipulation of tables (`data.frame` and `matrix`) is required and provides a nice exercise. Here, we will investigate the relationship between the expression of *ENTPD5* and mir-182 as it was described by the authors. Even if the data are normalised and should be ready to use, quite an extensive amount of work is still required to reproduce the claimed result.

Retrieve GEO study

The GEO dataset of interest is GSE35834

- load the study using the `getGEO` function

```r
gse35834 <- getGEO("GSE35834", GSEMatrix = TRUE)
```

```
## ftp://ftp.ncbi.nlm.nih.gov/geo/series/GSE35nnn/GSE35834/matrix/

## Found 2 file(s)

## GSE35834-GPL15236_series_matrix.txt.gz

## File stored at:

## /var/folders/7x/14cplkhj0fn34yltb3c0j9bczm4jkt/T//RtmpW0ogVQ/GPL15236.soft

## Warning in read.table(file = file, header = header, sep = sep, quote =
## quote, : not all columns named in 'colClasses' exist

## GSE35834-GPL8786_series_matrix.txt.gz

## File stored at:

## /var/folders/7x/14cplkhj0fn34yltb3c0j9bczm4jkt/T//RtmpW0ogVQ/GPL8786.soft

## Warning in read.table(file = file, header = header, sep = sep, quote =
## quote, : not all columns named in 'colClasses' exist
```

```r
show(gse35834)
```

```
## $`GSE35834-GPL15236_series_matrix.txt.gz`
## ExpressionSet (storageMode: lockedEnvironment)
## assayData: 22486 features, 80 samples
##   element names: exprs
## protocolData: none
## phenoData
##   sampleNames: GSM875933 GSM875934 ... GSM876012 (80 total)
##   varLabels: title geo_accession ... data_row_count (39 total)
##   varMetadata: labelDescription
## featureData
##   featureNames: 10000_at 10001_at ... 9_at (22486 total)
##   fvarLabels: ID ENTREZ_GENE_ID Description SPOT_ID
##   fvarMetadata: Column Description labelDescription
## experimentData: use 'experimentData(object)'
## Annotation: GPL15236
##
## $`GSE35834-GPL8786_series_matrix.txt.gz`
## ExpressionSet (storageMode: lockedEnvironment)
## assayData: 7815 features, 78 samples
##   element names: exprs
## protocolData: none
## phenoData
##   sampleNames: GSM875855 GSM875856 ... GSM875932 (78 total)
##   varLabels: title geo_accession ... data_row_count (39 total)
##   varMetadata: labelDescription
## featureData
##   featureNames: 14q-0_st 14qI-1_st ... zma-miR408_st (7815 total)
##   fvarLabels: ID miRNA_ID_LIST ... SEQUENCE (11 total)
##   fvarMetadata: Column Description labelDescription
## experimentData: use 'experimentData(object)'
## Annotation: GPL8786
```

- what kind of object is `gse35834`?

Solution

As shown in the Environment tab, it is a list composed by two elements. Each list is also a list with a special class 'ExpressionSet'.

- Two platforms were used in this study, which ones?

Solution

according to the GEO webpage: - GPL15236 ([HuEx-1_0-st] Affymetrix Human Exon 1.0 ST Array) http://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GPL15236 - GPL8786 ([miRNA-1_0] Affymetrix miRNA Array) http://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GPL8786

- How can you assign the mRNA or mir data to each element of `gse35834`?

Solution

The function `show()` displays 1. GPL15236 2. GPL8786 Thus, gse35834[[1]] is mRNA (22486 probes) gse35834[[2]] is mir (7815 probes)

Explore the mRNA expression meta-data

Informations about samples are accessible using `phenoData()` and can directly be retrieved as a `data.frame` with `pData()`.

for example, the following command will return the mRNA meta-data as a `data.frame`

```
pData(gse35834[[1]])
```

- Extract as a `tbl_df` named `rna_meta` the mRNA meta-data and
  - rename `geo_accession` to `sample`
  - select `source_name_ch1` and all columns that start with “charact”

Solution

```
rna_meta <- pData(gse35834[[1]]) %>%
  tbl_df() %>%
  select(sample = geo_accession,
         source_name_ch1,
         starts_with("charact"))
```

Explore the mir expression meta-data

- Extract as a `tbl_df` named `mir_meta` the mRNA meta-data and
  - rename `geo_accession` to `sample`
  - select `source_name_ch1` and all columns that start with “charact”

Solution

```
mir_meta <- pData(gse35834[[2]]) %>%
  tbl_df() %>%
  select(sample = geo_accession,
         source_name_ch1,
         starts_with("charact"))
```

Join meta-data

- Explore the two data frames with `View(rna_meta)` and `View(mir_meta)`. Are the samples `GSM*` identical?

Solution

No, they aren't. This is really annoying as the expression data contain only GSM ids.

Then, we would like to somehow join both informations.
Knowing that both data frames have different “sample” columns, merge them to get the correspondence between RNA `GSM*` and mir `GSM*`. Save the result as `rna_mir`.

Note

When 2 data.frames are joined by specific columns and the remaining columns have have identical names, a '.x' or '.y' suffix is appended for the first and second data frames respectively

Solution

```
inner_join(rna_meta, mir_meta,
           by = c("characteristics_ch1.1", "characteristics_ch1",
                  "source_name_ch1", "characteristics_ch1.2",
                  "characteristics_ch1.3", "characteristics_ch1.4",
                  "characteristics_ch1.5", "characteristics_ch1.6",
                  "characteristics_ch1.7", "characteristics_ch1.8")) -> rna_mir
```

```
## Warning in inner_join_impl(x, y, by$x, by$y): joining factors with
## different levels, coercing to character vector

## Warning in inner_join_impl(x, y, by$x, by$y): joining factors with
## different levels, coercing to character vector

## Warning in inner_join_impl(x, y, by$x, by$y): joining factors with
## different levels, coercing to character vector
```

Get RNA expression data for the ENTPD5 gene

Expression data can be accessed via `exprs()` which returns a matrix.

Warning

If you do not pipe the command to `head`, `R` would print **ALL** rows (or until it reaches `max.print`).

```
exprs(gse35834[[1]]) %>% head()
```

rows are probes and columns are sample ids in the form `GSM*`.

Probe ids are not meaningful, but `fData()` provides features.

```
fData(gse35834[[1]]) %>% head()
```

Again, we need to merge both informations to assign the expression data to the gene of interest.

1. Find the common values that could help us joining.

Solution

the probe ids are the common values

2. A `matrix` contains only numerical values. But, the `rownames` contain the necessary info.
   Transform the `matrix` into a `data.frame`. Then, convert the `rownames` to a column using `tibble::rownames_to_column(var = "ID")`.
   Save as `rna_expression`

Solution

```
exprs(gse35834[[1]]) %>%
  as.data.frame() %>%
  tibble::rownames_to_column(var = "ID") -> rna_expression
```

3. merge expression data to platform annotation (`fData(gse35834[[1]])`). Save as `rna_expression`. `R` is always working on temporary objects, you won't erase the object you are working on.

Note

Warnings about *factors being coerced to characters* can be ignored. Factors shouldn't be in the first place (default of `readr` functions)

Solution

```
rna_expression %>%
 inner_join(fData(gse35834[[1]])) -> rna_expression
```

```
## Joining by: "ID"

## Warning in inner_join_impl(x, y, by$x, by$y): joining character vector and
## factor, coercing into character vector
```

4. Find the Entrez gene id for *ENTPD5*. Usually, the gene symbol is given in the annotation, but each GEO submission is a new discovery.

Solution

957, for Homo sapiens

5. Filter `rna_expression` for the gene of interest and tidy the samples:
   A column `sample` for all `GSM*` and a column `rna_expression` containing the expression values. Save the result as `rna_expression_melt`. At this point you should get a `data.frame` of 80 values.

Solution

```
rna_expression %>%
  filter(ENTREZ_GENE_ID == 957) %>%
  gather(sample, rna_expression, starts_with("GSM")) -> rna_expression_melt
```

6. Add the meta-data and discard the columns `ID`, `SPOT_ID` and `sample.x`. Save the result as `rna_expression_melt`.

Solution

```
rna_expression_melt %>%
  inner_join(rna_mir, by = c("sample" = "sample.x")) %>%
  select(-ID, -SPOT_ID, -sample.y) -> rna_expression_melt
```

```
## Warning in inner_join_impl(x, y, by$x, by$y): joining character vector and
## factor, coercing into character vector
```

Get mir expression data for miR-182

1. Repeat the previous step but using `exprs(gse35834[[2]])` for the `mir_expression`. This time, the mir names are nicely provided by `fData(gse35834[[2]])` in the column `miRNA_ID_LIST`

Solution

```r
exprs(gse35834[[2]]) %>%
  as.data.frame() %>%
  tibble::rownames_to_column(var = "ID") %>%
  # match expression data to platform annotation
  inner_join(fData(gse35834[[2]])) %>%
  gather(sample, mir_expression, starts_with("GSM")) %>% # melt patients
  filter(miRNA_ID_LIST == "hsa-mir-182") -> mir_expression_melt
```

```
## Joining by: "ID"
```

```
## Warning in inner_join_impl(x, y, by$x, by$y): joining character vector and
## factor, coercing into character vector
```

2. How many rows do you obtain? How many are expected?

Solution

78 samples for the mir experiment, so expect 78, obtain twice this number

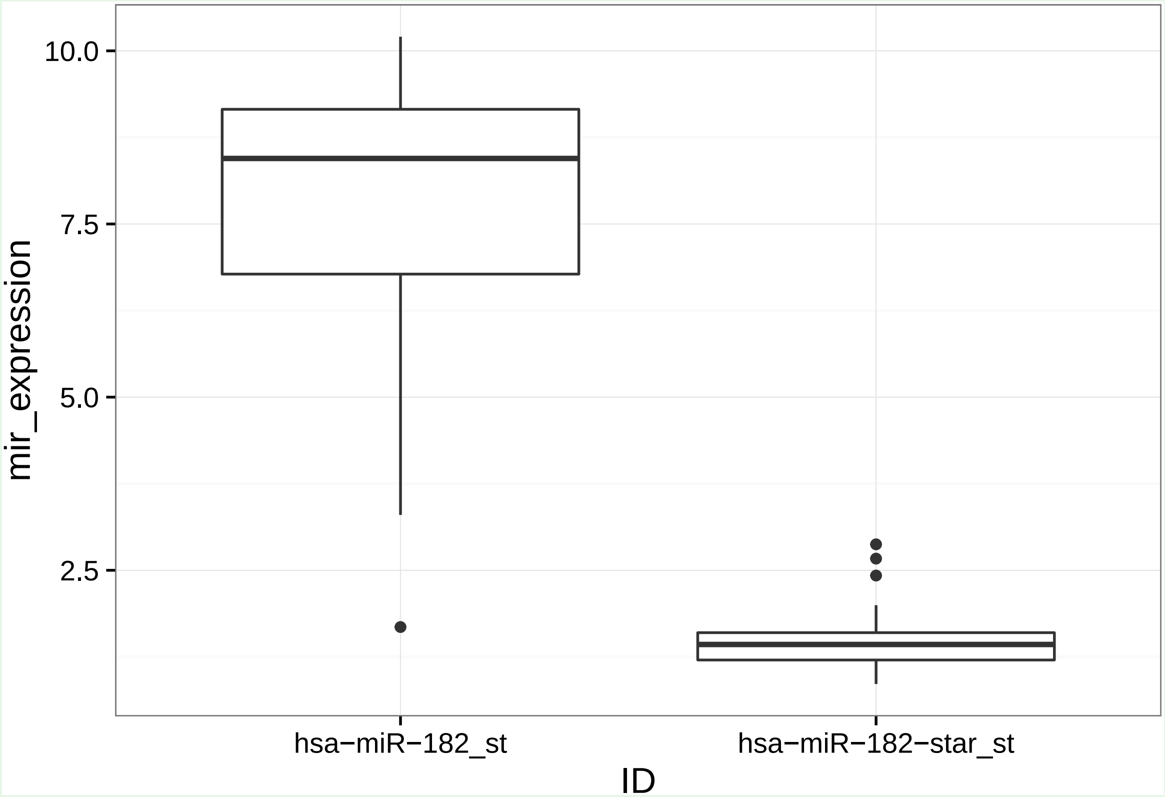
3. Find out what happened, and plot the boxplot distribution of `expression` by `ID`

Solution

The mir array contains probes for both strands of mir: - mature mir - immature mir, named "*", star.

Solution

```
mir_expression_melt %>%
  ggplot(aes(x = ID, y = mir_expression))+
  geom_boxplot()
```



Solution

The immature mir, named star is indeed merely expressed

4. Filter out the irrelevant IDs using `grepl` in the `filter` function.

Hint

adding `!` to a condition means `NOT`. Example `filter(iris, !grepl("a", Species))`: remove all Species that contain an "a".

Solution

```
mir_expression_melt %>%
  filter(!grepl("star", ID)) -> mir_expression_melt
```

5. Add the meta-data, count the number of rows. Discard the column `sample.x` after joining.

Solution

```
mir_expression_melt %>%
  inner_join(rna_mir, by = c("sample" = "sample.y")) %>%
  select(-sample.x) -> mir_expression_melt
```

```
## Warning in inner_join_impl(x, y, by$x, by$y): joining character vector and
## factor, coercing into character vector
```

Solution

77 rows, we lost GSM875854, which is not present in the meta-data nor the GSE description. Let it down

join both expression

Join `rna_expression_melt` and `mir_expression_melt` by their common columns EXCEPT `sample`. Save the result as `expression`.

Solution

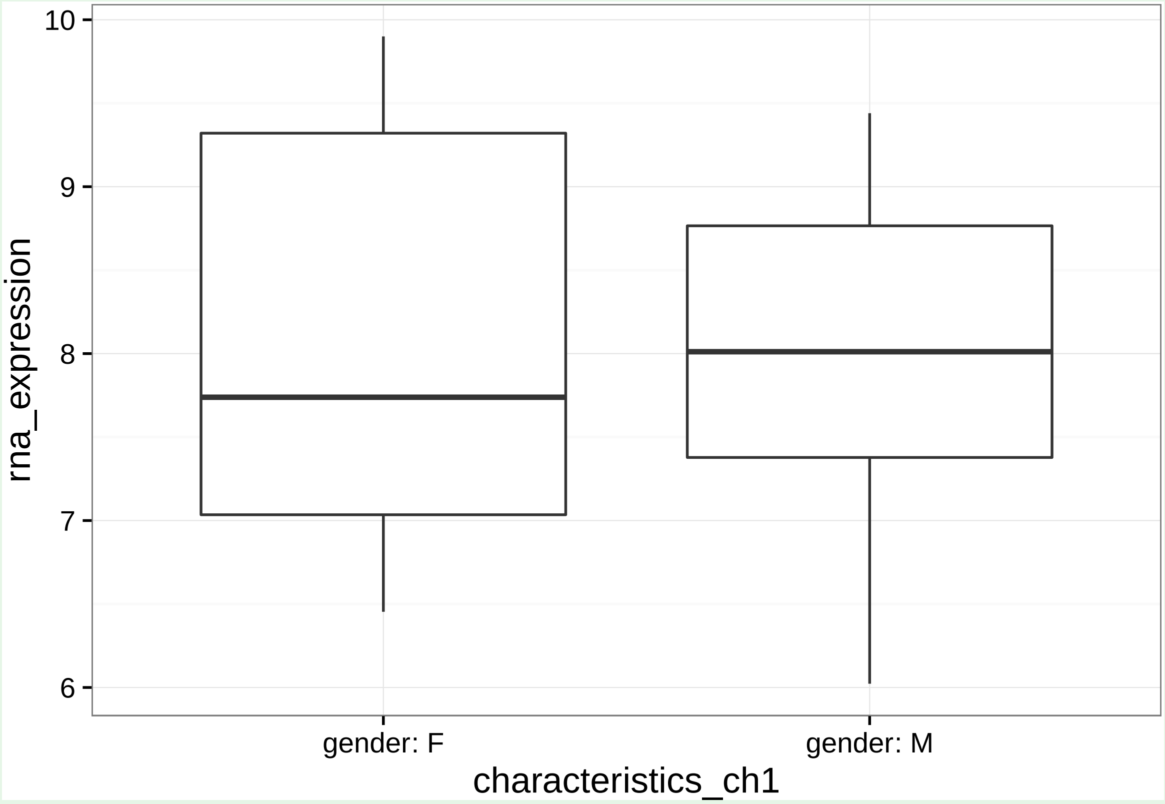
```
expression <- inner_join(rna_expression_melt, mir_expression_melt,
                         by = c("source_name_ch1", "characteristics_ch1", "characteristics_ch1.1",
```

Examine gene expression according to meta data

1. Plot the gene expression distribution by Gender. Is there any obvious difference?

Solution

```
expression %>%
  ggplot(aes(y = rna_expression, x = characteristics_ch1))+
  geom_boxplot()
```



Solution

no relation to gender

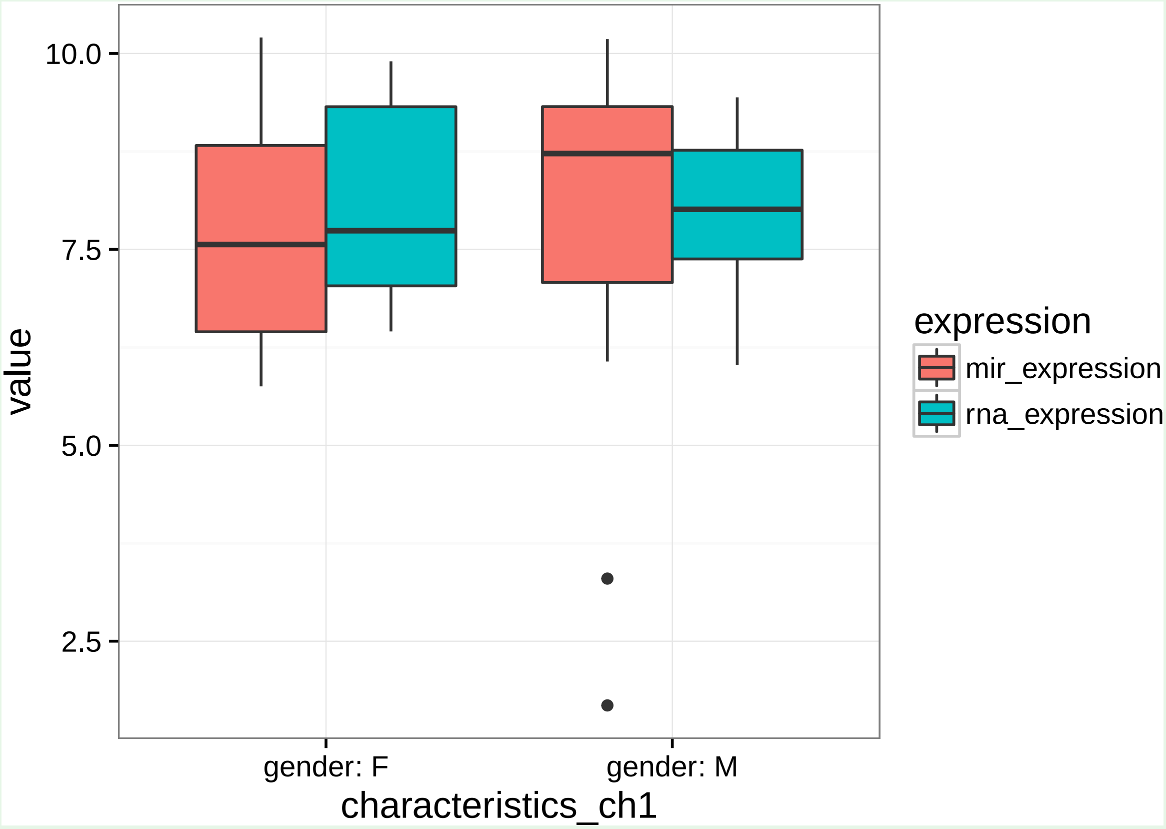
2. Plot gene AND mir expression distribution by Gender. Is there any obvious difference?

Hint

You will need to tidy by gathering rna and mir expression

Solution

```
expression %>%
  gather(expression, value, ends_with("expression")) %>%
  ggplot(aes(y = value, x = characteristics_ch1, fill = expression))+
  geom_boxplot()
```



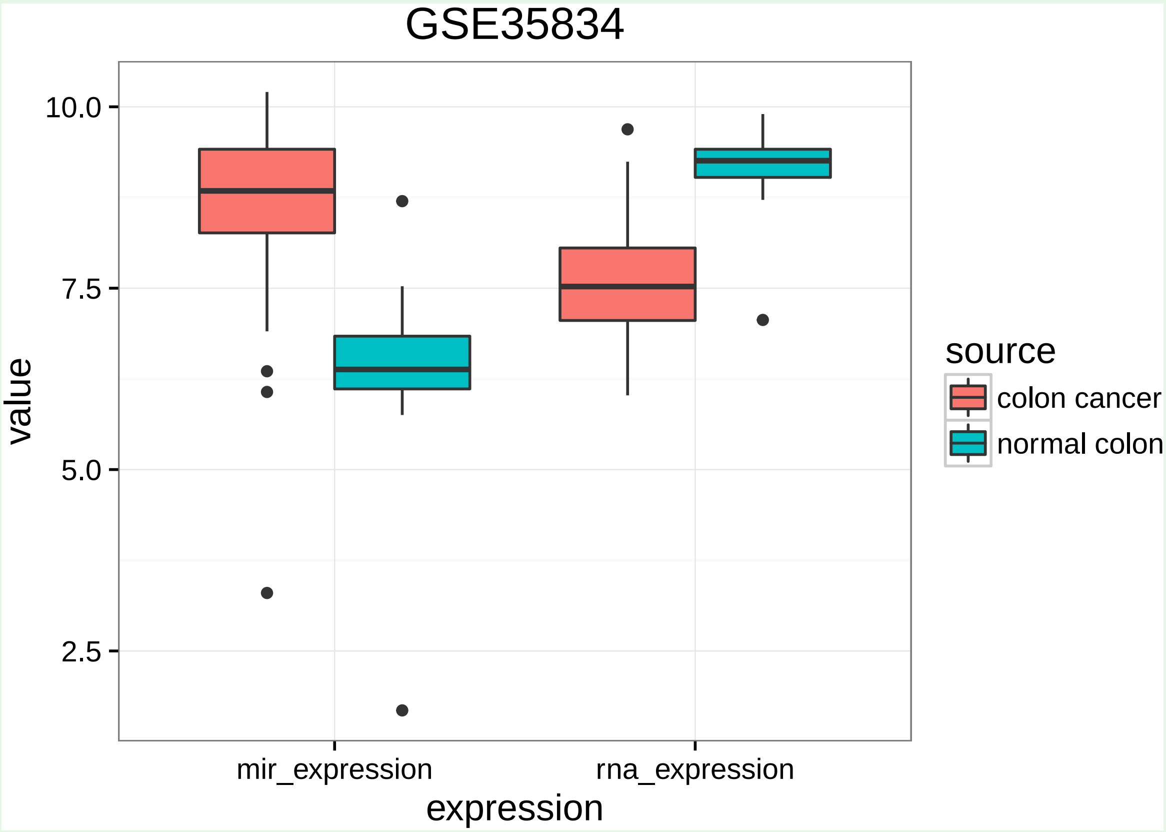
Solution

no relation to gender for both expressions

3. Plot gene AND mir expression distributions by source (control / cancer). To make it easier, a quick hack is `separate(expression, source_name_ch1, c("source", "rest"), sep = 12)` to get `source` as control / cancer. Is there any difference?

Solution

```
expression %>%
  gather(expression, value, ends_with("expression")) %>%
  separate(source_name_ch1, c("source", "rest"), sep = 12) %>%
  ggplot(aes(y = value, fill = source, x = expression))+
  geom_boxplot()+ ggtitle("GSE35834")
```



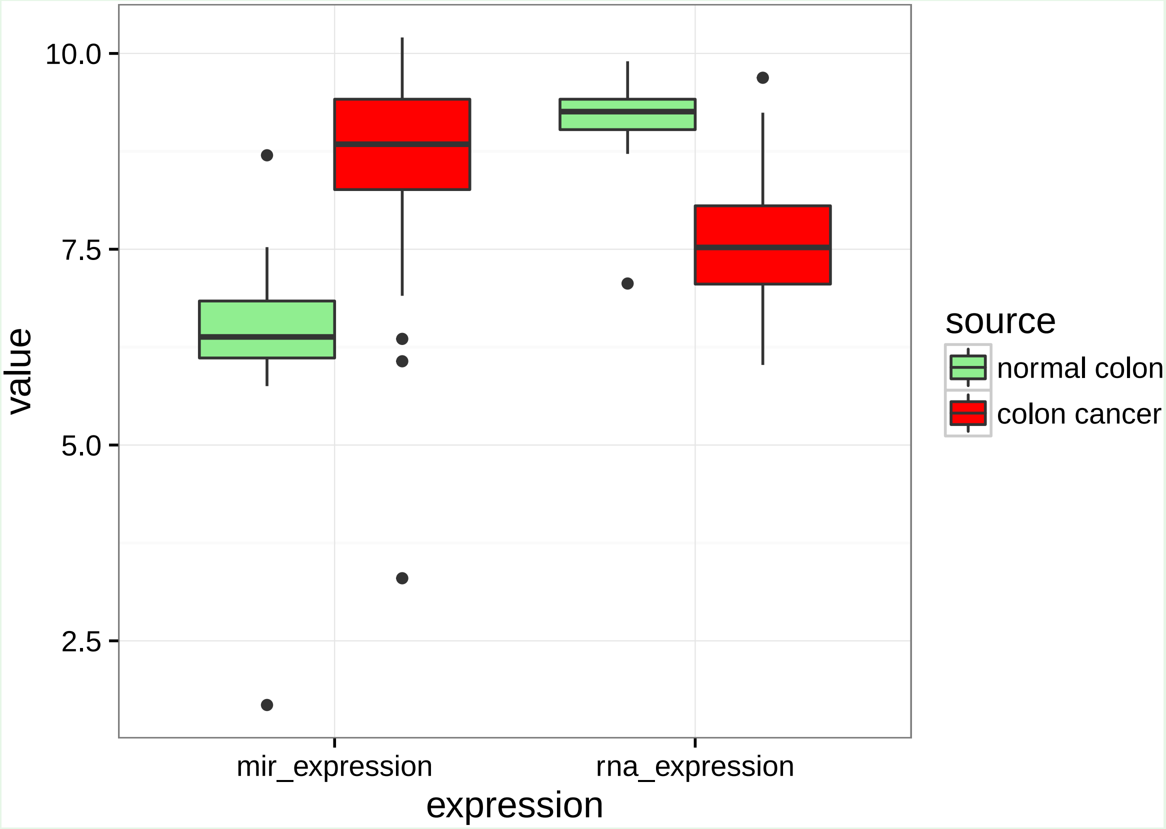
Solution

Like it is stated in the summary of the study, the expression of mir-182 seems indeed higher in cancer while the ENTPD5 expression seems lower.

4. Replot 3. but reordering the levels so normal colon comes first. Display *normal* in "lightgreen" and *cancer* in "red" using `scale_fill_manual()`

Solution

```
expression %>%
  gather(expression, value, ends_with("expression")) %>%
  separate(source_name_ch1, c("source", "rest"), sep = 12)  %>%
  mutate(source = factor(source, levels = c("normal colon", "colon cancer"))) %>%
  ggplot(aes(y = value, fill = source, x = expression))+
  geom_boxplot()+
  scale_fill_manual(values = c("lightgreen", "red"))
```

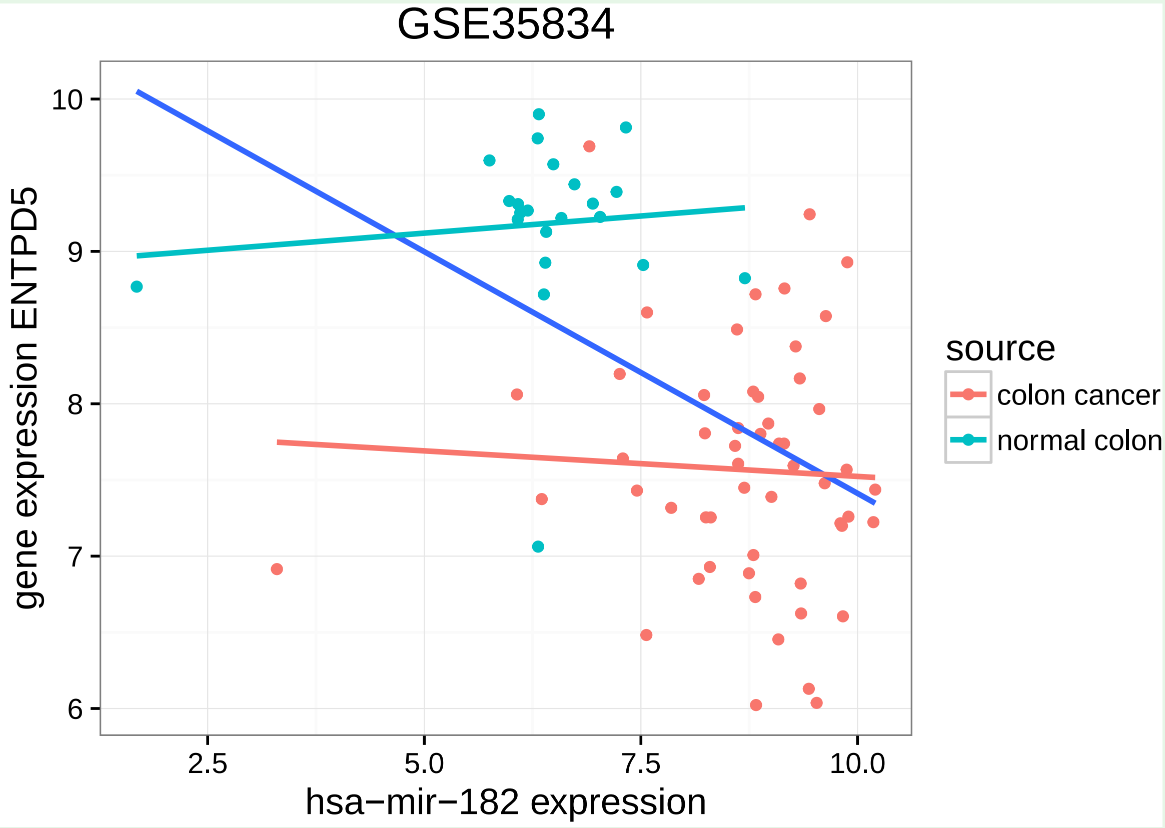


plot relation ENTPD5 ~ mir-182 as scatter-plot for all patients

- add a linear trend using `geom_smooth()` for all data + per source

Solution

```
expression %>%
  separate(source_name_ch1, c("source", "rest"), sep = 12) %>%
  ggplot(aes(x = mir_expression, y = rna_expression))+
  geom_point(aes(colour = source))+
  geom_smooth(method = "lm", se = FALSE)+
  geom_smooth(aes(colour = source), method = "lm", se = FALSE)+
  labs(y = "gene expression ENTPD5",
       x = "hsa-mir-182 expression")+
  ggtitle("GSE35834")
```



- does it support the claim of the study?

Solution

the two dot clouds between normal and cancer origin do split by - high mir expression / low gene expression - mild mir expression / high gene expression but the trend is not so clear

Supplementary exercise - linear regression

- get the estimate from the linear trend. linear models are outputted by `lm()` as lists. Since `data.frame` are much easier to work with, David Robinson developed `broom`. We will present the use of `broom` during the advanced lecture, but you can have an insight here and test `broom::tidy()` coupled with `dplyr::do()`.

```
library("broom")
expression %>%
  separate(source_name_ch1, c("source", "rest"), sep = 12) %>%
  group_by(source) %>%
  do(tidy(lm(rna_expression ~ mir_expression, data = .))) %>%
  filter(term != "(Intercept)")
```

```
## Source: local data frame [2 x 6]
## Groups: source [2]
##
##         source           term    estimate  std.error  statistic   p.value
##          (chr)          (chr)       (dbl)      (dbl)      (dbl)     (dbl)
## 1 colon cancer mir_expression -0.03354124 0.09217281 -0.3638951 0.7174117
## 2 normal colon mir_expression  0.04496954 0.10042656  0.4477853 0.6588934
```

The estimate of the intercept is not meaningful thus it is filtered out. One can easily see that the slope is not significant when data are slipped by source.

- Perform the linear regression and tidy the results for all data, is it significant?

Solution

```r
expression %>%
  do(tidy(lm(rna_expression ~ mir_expression, data = .))) %>%
  filter(term != "(Intercept)")
```

```
##             term   estimate  std.error statistic      p.value
## 1 mir_expression -0.3172285 0.06592259 -4.812137 7.545623e-06
```

Solution

with a pvalue of 7.54e-6, the negative is highly significant

- replace `tidy` by `glance` to extract the r^2. Is this value satisfactory?

Solution

```r
expression %>%
  do(glance(lm(rna_expression ~ mir_expression, data = .)))
```

```
##   r.squared adj.r.squared     sigma statistic      p.value df   logLik
## 1 0.2359153     0.2257275 0.9136575  23.15666 7.545623e-06  2 -101.292
##       AIC      BIC deviance df.residual
## 1 208.584 215.6154 62.60775          75
```

Solution

with a r^2 of 0.236, i.e only 23.6% of the variance explained, a linear fit sounds bad due to outliers

Perform a linear model for the expression of *ENTPD5* and ALL mirs

- Count how many `hsa-mir`, which are not star, are present on the array GPL8786

Solution

```r
fData(gse35834[[2]]) %>%
  filter(grepl("^hsa", ID)) %>%
  filter(!grepl("star", ID)) %>%
  nrow()
```

```
## [1] 677
```

- Retrieve the expression values for the 677 human mir like you did before. Same procedure, except that you don't filter for mir-182. Save as `all_mir_rna_expression`

Solution

```
exprs(gse35834[[2]]) %>%
  as.data.frame() %>%
  tibble::rownames_to_column(var = "ID") %>%
  filter(grepl("^hsa", ID)) %>%
  # match expression data to platform annotation
  gather(sample, mir_expression, starts_with("GSM")) %>%
  filter(!grepl("star", ID)) %>%
  inner_join(fData(gse35834[[2]])) %>%
  inner_join(rna_mir, by = c("sample" = "sample.y")) %>%
  select(-sample.x) %>%
  inner_join(rna_expression_melt,
             by = c("source_name_ch1", "characteristics_ch1",
                    "characteristics_ch1.1", "characteristics_ch1.2",
                    "characteristics_ch1.3",  "characteristics_ch1.4",
                    "characteristics_ch1.5", "characteristics_ch1.6",
                    "characteristics_ch1.7", "characteristics_ch1.8")) -> all_mir_rna_expression
```

```
## Joining by: "ID"

## Warning in inner_join_impl(x, y, by$x, by$y): joining character vector and
## factor, coercing into character vector

## Warning in inner_join_impl(x, y, by$x, by$y): joining character vector and
## factor, coercing into character vector
```

- Perform the 677 linear models, tidy the results and arrange by the `adj.r.squared`

Solution

```
all_mir_rna_expression %>%
  group_by(ID) %>%
   do(glance(lm(rna_expression ~ mir_expression, data = .))) %>%
  ungroup() %>%
  arrange(desc(adj.r.squared))
```

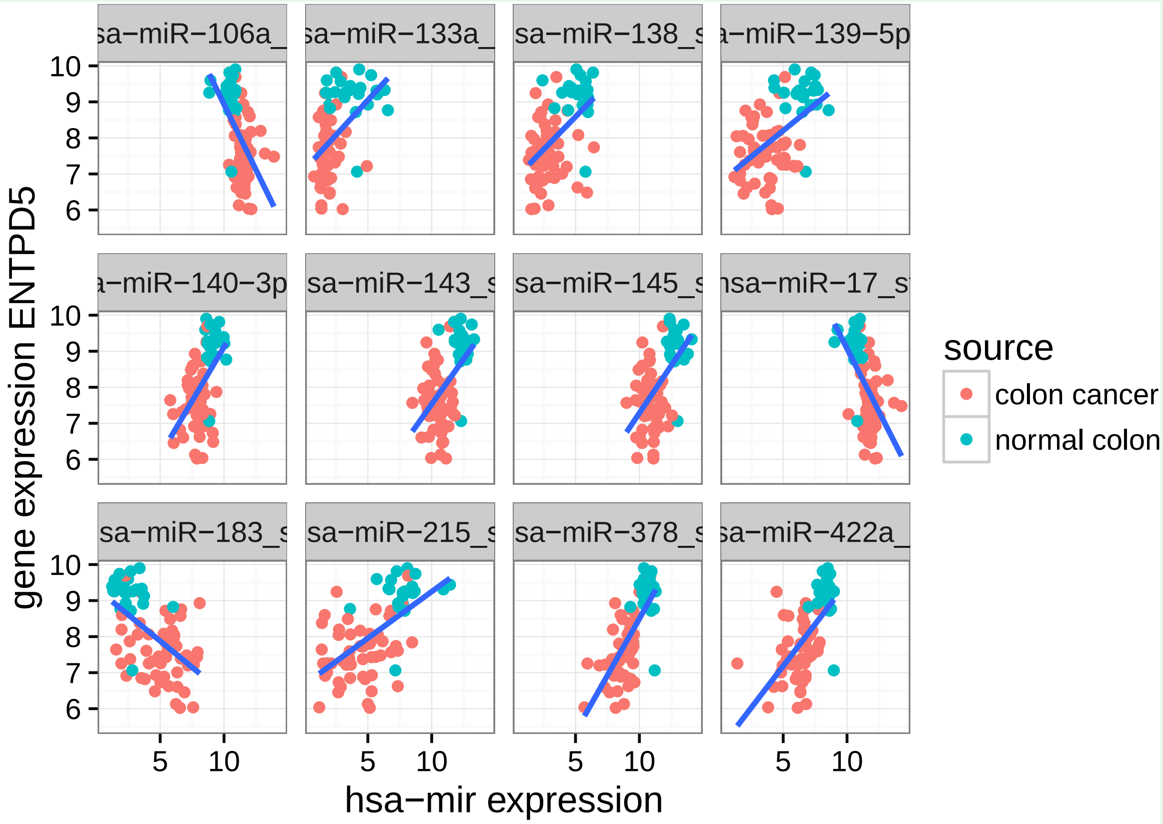
```
## Source: local data frame [677 x 12]
##
##                    ID r.squared adj.r.squared     sigma statistic
##                 <chr>     <dbl>         <dbl>     <dbl>     <dbl>
## 1     hsa-miR-378_st 0.5337446     0.5275279 0.7137146  85.85605
## 2    hsa-miR-422a_st 0.3880594     0.3799002 0.8176497  47.56092
## 3     hsa-miR-215_st 0.3649550     0.3564878 0.8329423  43.10187
## 4     hsa-miR-145_st 0.3239984     0.3149851 0.8593825  35.94649
## 5     hsa-miR-183_st 0.3066374     0.2973925 0.8703479  33.16851
## 6      hsa-miR-17_st 0.2964648     0.2870843 0.8767093  31.60447
## 7    hsa-miR-106a_st 0.2875369     0.2780374 0.8822545  30.26861
## 8  hsa-miR-140-3p_st 0.2812168     0.2716330 0.8861589  29.34301
## 9     hsa-miR-138_st 0.2693901     0.2596486 0.8934195  27.65396
## 10 hsa-miR-139-5p_st 0.2689871     0.2592402 0.8936659  27.59736
## ..                ...       ...           ...       ...       ...
## Variables not shown: p.value <dbl>, df <int>, logLik <dbl>, AIC <dbl>, BIC
##   <dbl>, deviance <dbl>, df.residual <int>.
```

- Get the top 12 mir and plot the scatter plot

Solution

```
top12_mir <- all_mir_rna_expression %>%
  group_by(ID) %>%
   do(glance(lm(rna_expression ~ mir_expression, data = .))) %>%
  ungroup() %>%
  arrange(desc(adj.r.squared)) %>%
  head(12) %>%
  .$ID

all_mir_rna_expression %>%
  filter(ID %in% top12_mir) %>%
  separate(source_name_ch1, c("source", "rest"), sep = 12) %>%
  ggplot(aes(x = mir_expression, y = rna_expression))+
  geom_point(aes(colour = source))+
  geom_smooth(method = "lm", se = FALSE)+
  facet_wrap(~ ID, ncol = 4)+
  labs(y = "gene expression ENTPD5",
       x = "hsa-mir expression")
```


sa−miR−106a_
sa−miR−133a_
sa−miR−138_s
a−miR−139−5p
a−miR−140−3p
sa−miR−143_s
sa−miR−145_s
hsa−miR−17_s
sa−miR−183_s
sa−miR−215_s
sa−miR−378_s
sa−miR−422a_
gene expression ENTPD5
hsa−mir expression
source
● colon cancer
● normal colon