Comparison between RNA-seq and Immunohistochemistry data

Contents

In the SHIVA trial, estrogen, androgen and progesterone status (ER, AR and PR) were collected in order to associate the patients to a hormone therapy. The hormone receptors ER, AR and PR were collected using Immunohistochemistry (IHC). Immunohistochemestry values are available through TCGA for a limited number of samples but RNA-Seq could represent a valid proxy. In this section we want to explore the relationship between RNA-seq and Immunohistochemistry and, possibly, identify a threshold that we can use in the simulation to appropriately identify over-expressed samples. IHC categories for ER will be compared to *ESR1* expression values and IHC PR categories to the RNA values of *PGR* gene.

Comparative analysis

To run the comparison analysis we will need two datasets:

- Dataset with IHC values,
- Dataset with RNA-seq expression values

Both dataset need to have in common the same patients so that we can reconstruct the index.

The analysis will be run on two hormone receptors:

- ER
- PR

ER analysis

IHC dataset

As Input dataset we are choosing to use:

Clinical data downloaded from cBioportal for the dataset: **Breast Invasive Carcinoma (TCGA, Cell 2015)** - LINK

The dataset has been downloaded and stored as `brca_tcga_pub2015_clinical_data.tsv`.

```
suppressMessages(library(dplyr))
suppressMessages(library(ggplot2))
suppressMessages(library(plotly))
suppressMessages(library(readr))
suppressMessages(library(knitr))
suppressMessages(library(PrecisionTrialDesigner))

ihc <-readr::read_tsv("../external_resources/brca_tcga_pub2015_clinical_data.tsv")
ihcFilter <- ihc %>%
  dplyr::select(`Patient ID`
                , `ER Status By IHC`
                , `ER Status IHC Percent Positive`
                ) %>%
  dplyr::filter(!is.na(`ER Status By IHC`)) %>% # Remove the <NA>
  dplyr::filter(!is.na(`ER Status IHC Percent Positive`)) %>% # Remove the <NA>
  dplyr::rename(case_id=`Patient ID`, er_status=`ER Status By IHC`, ihc_value=`ER Status IHC Percent Po

# preview
kable(head(ihcFilter), caption="top 6 rows")
```

Table 1: top 6 rows

case_id	er_status	ihc_value
TCGA-A2-A3XV	Positive	<10%
TCGA-A2-A3Y0	Positive	90-99%
TCGA-LL-A50Y	Positive	90-99%
TCGA-LL-A5YP	Positive	<10%
TCGA-LL-A5YL	Positive	90-99%
TCGA-LL-A5YM	Positive	90-99%

RNA-seq dataset

The RNA-seq dataset was extracted using PTD function.!

```
panel_design <- data.frame(drug=""
    , gene_symbol="ESR1"
    , alteration="expression"
    , exact_alteration="up"
    ,   mutation_specification=""
    ,   group="")


panel <- newCancerPanel(panel_design)
```

```
## Checking panel construction...
## Calculating panel size...
## Connecting to ensembl biomart...
```

```
panel <- getAlterations(panel, tumor_type = "brca_tcga")
```

```
##
```

```
## Retrieving Expression data...

## getting Expression from this cancer study: brca_tcga
```

```
panel <- subsetAlterations(panel)
```

```
## Subsetting expression...
```

```
# Load data from SHIVA retrospective analaysis
#panel <- readRDS("../Temp/shiva_panel.rds")

# Fetch data
rnaseq <- panel@dataFull$expression$data %>%
  filter(tumor_type == "brca") %>%
  filter(gene_symbol == "ESR1") %>%
  select(case_id, expressionValue)

# Preview
kable(head(rnaseq), caption = "top 6 rows")
```

Table 2: top 6 rows

case_id	expressionValue
TCGA-3C-AAAU	-0.7191
TCGA-3C-AALI	-1.0102
TCGA-3C-AALJ	-0.3734
TCGA-3C-AALK	-0.8026
TCGA-4H-AAAK	-0.5421
TCGA-5L-AAT0	-0.4499

Inner Join datasets

Preview the results from the inner join.

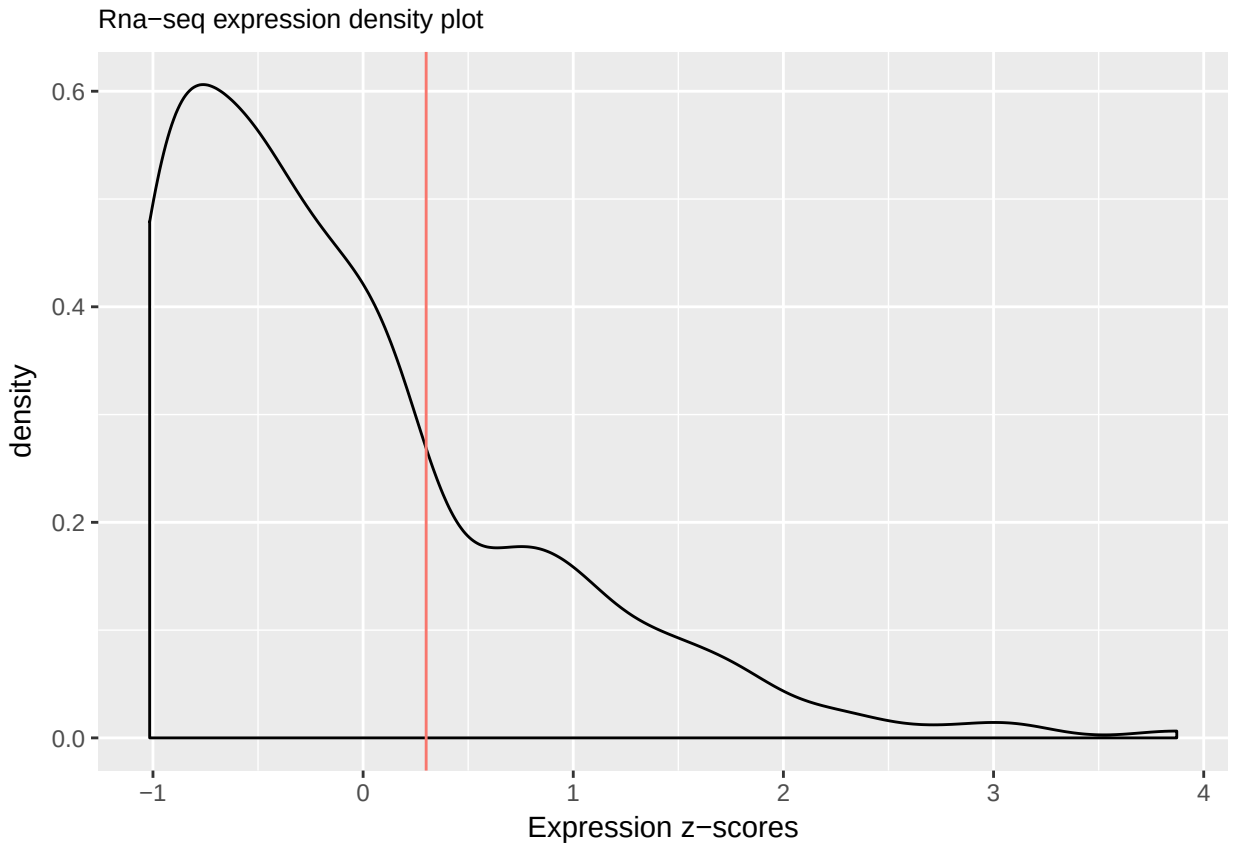
```
df <- dplyr::inner_join(rnaseq, ihcFilter, by="case_id")
# preview
kable(head(df
           # ADD BUTTONS TO THE TABLE
           , extensions = 'Buttons'
           , options = list(
                dom = 'lBfrtip'
               , buttons = c('copy', 'csv', 'excel')
               )
           , caption = "Comparison between Missing and Submitted regions (bp) in the panel"
           ))
```

case_id	expressionValue	er_status	ihc_value
TCGA-A1-A0SB	-0.9422	Positive	70-79%
TCGA-A1-A0SD	-0.1806	Positive	90-99%
TCGA-A1-A0SE	-0.6371	Positive	80-89%
TCGA-A1-A0SF	-0.3379	Positive	90-99%
TCGA-A1-A0SI	-0.6601	Positive	50-59%
TCGA-A1-A0SJ	-0.6137	Positive	70-79%

Comparison Analysis

Explore RNA-seq Z-score

```
# explore z-score value
p1 <- ggplot(df, aes(x=expressionValue)) +
        geom_density(kernel="gaussian") +
        geom_vline(aes(xintercept=0.3, color="red")) +
        labs(x="Expression z-scores", title="Rna-seq expression density plot") +
        theme(legend.position = "none", plot.title=element_text(size=10))
p1
```

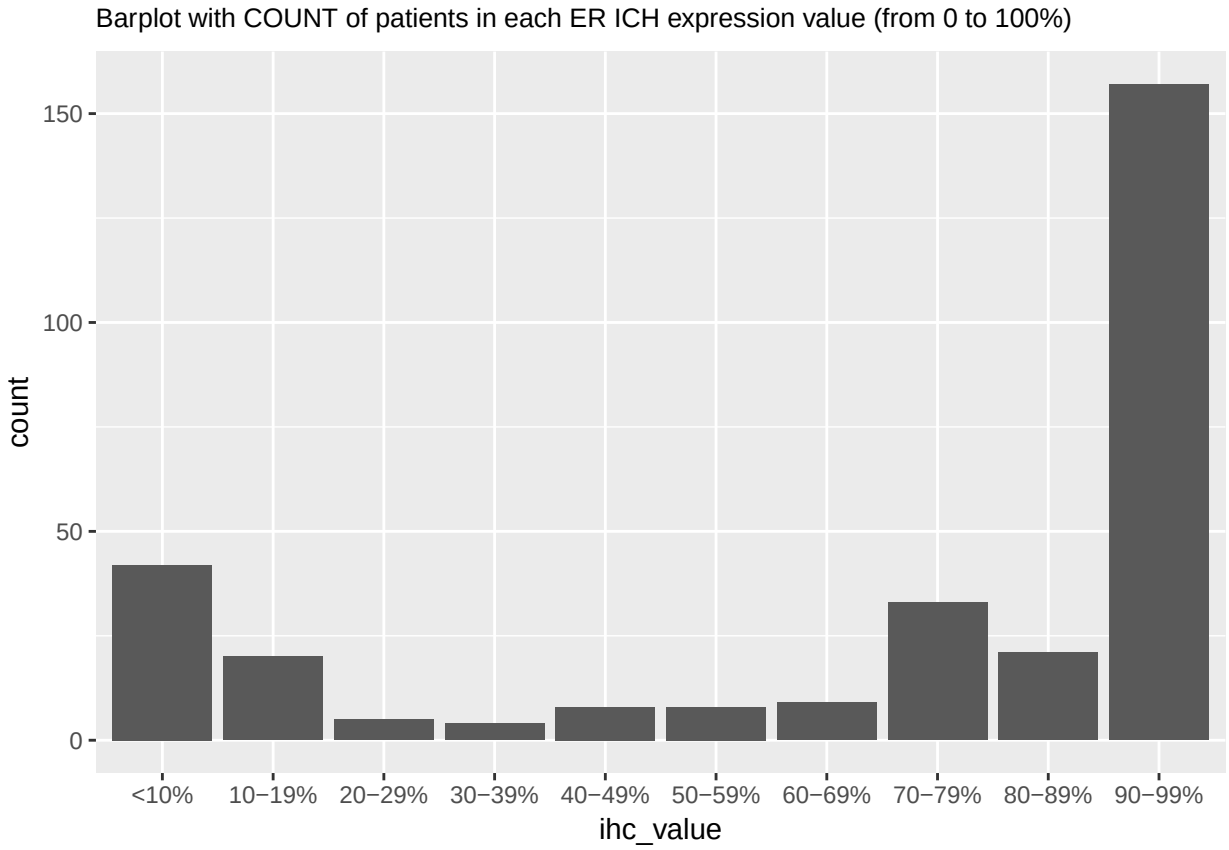



```
ggsave(filename="../Figures/fig_extra1.svg", plot=p1, device = "svg")
```

```
## Saving 6.5 x 4.5 in image
```

Explore ICH values

```
# barplot
p2 <- ggplot(data=df, aes(x=ihc_value)) +
        geom_bar(stat = "count", position = "stack") +
        labs(title="Barplot with COUNT of patients in each ER ICH expression value (from 0 to 100%)")+
        theme(legend.position = "none", plot.title=element_text(size=10))
p2
```




```
ggsave(filename="../Figures/fig_extra2.svg", plot=p2, device = "svg")
```

```
## Saving 6.5 x 4.5 in image
```

Compare

```
p3 <- ggplot(data=df, aes(x=ihc_value, y=expressionValue, group=1)) +
    geom_point(colour="red", size=1, shape=21, fill="white") +
    labs(title="Comparison between RNA-seq and IHC values for ER in Breast cancer") +
    xlab("IHC value") +
    ylab("RNA-seq z-score") +
    geom_smooth(method="lm") +
    geom_hline(yintercept =0.3) +
    theme(legend.position = "none", plot.title=element_text(size=10))

#ggplotly(p3,width = 650, height = 400, margin(t=1000))

ggsave(filename="../Figures/fig_extra3.svg", plot=p3, device = "svg")
```

```
## Saving 7 x 4.5 in image
```

Fit to a linear model

```
# Convert chategorical values to continue numerical value
# <10% = 1
# 10-19% = 2
# etc..
```

```r
df$ihc_value2 <- as.numeric(factor(df$ihc_value))
# Fit the data into a linea regressino model ache chek the coefficients
summary(lm(df$expressionValue ~ ihc_value2, df))
```

```
##
## Call:
## lm(formula = df$expressionValue ~ ihc_value2, data = df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.1513 -0.5161 -0.1986  0.3119  3.6855
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.91294    0.10600  -8.612 3.90e-16 ***
## ihc_value2   0.10993    0.01292   8.510 7.99e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7789 on 305 degrees of freedom
## Multiple R-squared:  0.1919, Adjusted R-squared:  0.1892
## F-statistic: 72.42 on 1 and 305 DF,  p-value: 7.987e-16
```

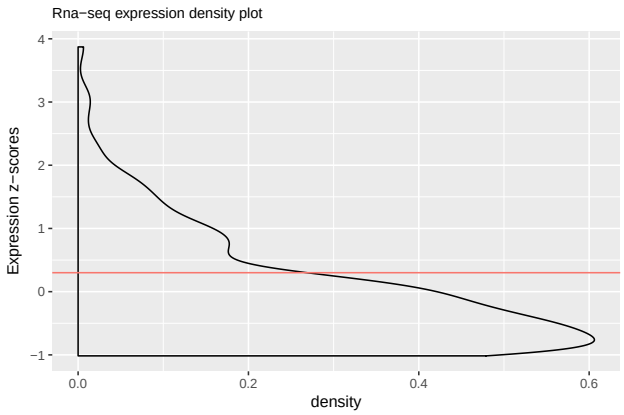
There is a significant linear relationship between the predictor and the outcome. Although the R^2 value is very low (R^2 indicates the percentage of total variation explained by the linear relationship with the predictors).

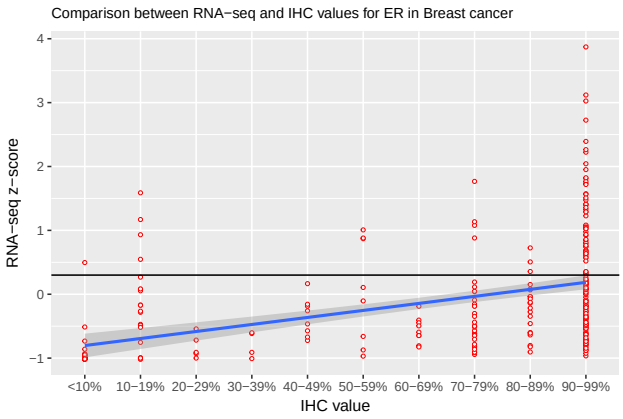
- Pearson correlation: **0.4380385**

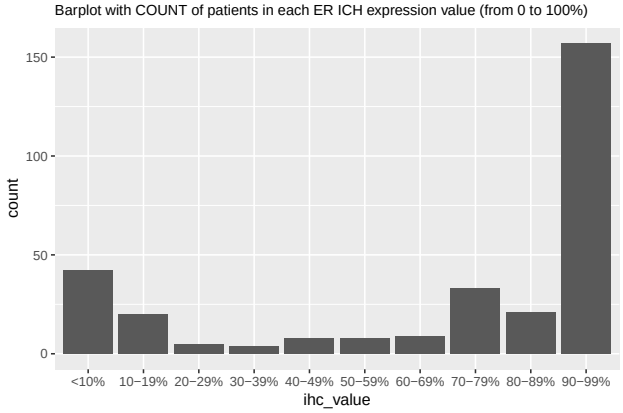
Put all the plots together

```
##
## Attaching package: 'gridExtra'
```

```
## The following object is masked from 'package:dplyr':
##
##     combine
```







PR analysis

IHC dataset

```
ihcPRFilter <- ihc %>%
  dplyr::select(`Patient ID`
                , `PR status by ihc`
                , `PR status ihc percent positive`
                ) %>%
  dplyr::filter(!is.na(`PR status by ihc`)) %>% # Remove the <NA>
  dplyr::filter(!is.na(`PR status ihc percent positive`)) %>% # Remove the <NA>
  dplyr::rename(case_id=`Patient ID`, pr_status=`PR status by ihc`, ihc_value=`PR status ihc percent pos
```

RNA-seq dataset

The RNA-seq dataset was extracted using PTD function.!

```
panel_design <- data.frame(drug=""
    , gene_symbol="PGR"
    , alteration="expression"
    , exact_alteration="up"
    ,   mutation_specification=""
    ,   group="")
```

```
panel <- newCancerPanel(panel_design)
```

```
## Checking panel construction...
## Calculating panel size...
## Connecting to ensembl biomart...
```

```
panel <- getAlterations(panel, tumor_type = "brca_tcga")
```

```
##
## Retrieving Expression data...
## getting Expression from this cancer study: brca_tcga
```

```
panel <- subsetAlterations(panel)
```

```
## Subsetting expression...
```

```
# Load data from SHIVA retrospective analaysis
# Fetch data
rnaseq_PR <- panel@dataFull$expression$data %>%
  filter(tumor_type == "brca") %>%
  filter(gene_symbol == "PGR") %>%
  select(case_id, expressionValue)
```

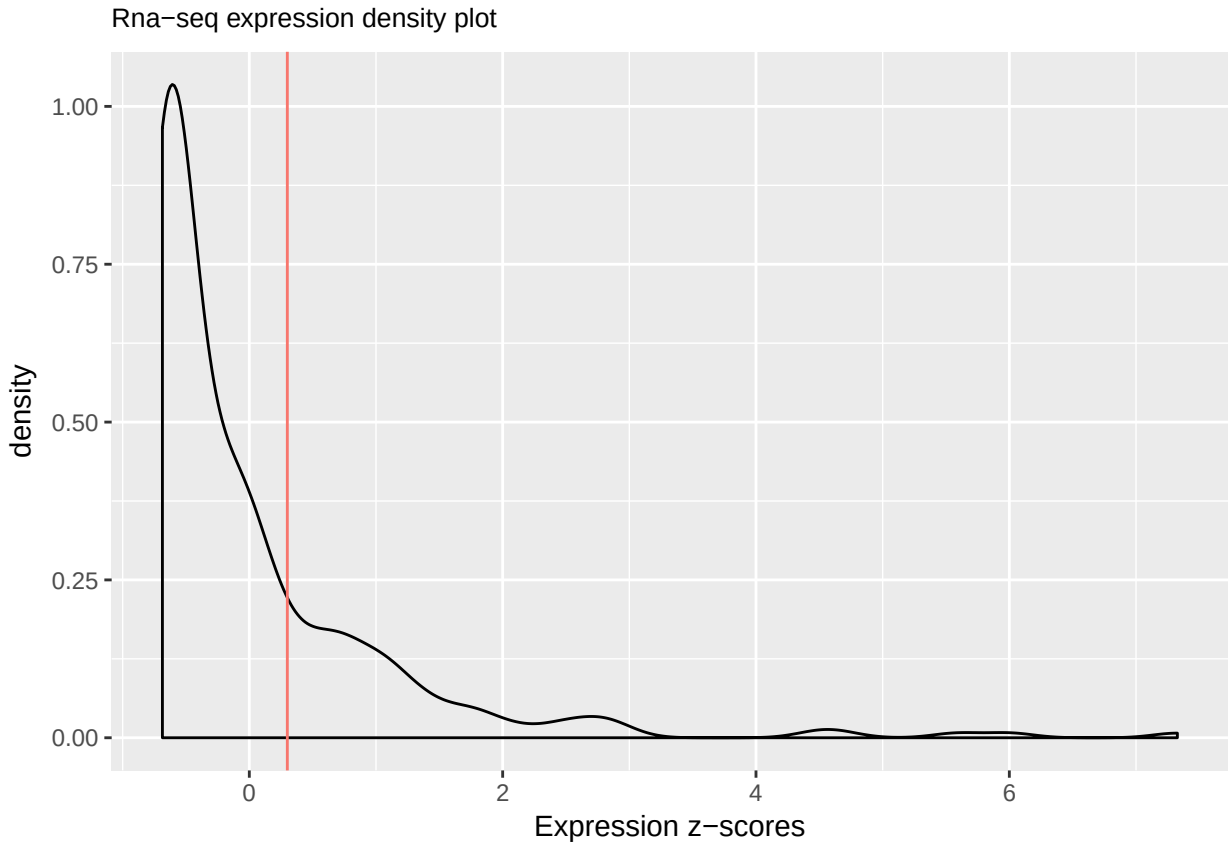
Inner join datasets

```
# join
dfPR <- dplyr::inner_join(rnaseq_PR, ihcPRFilter, by="case_id")
```

Comparison analysis

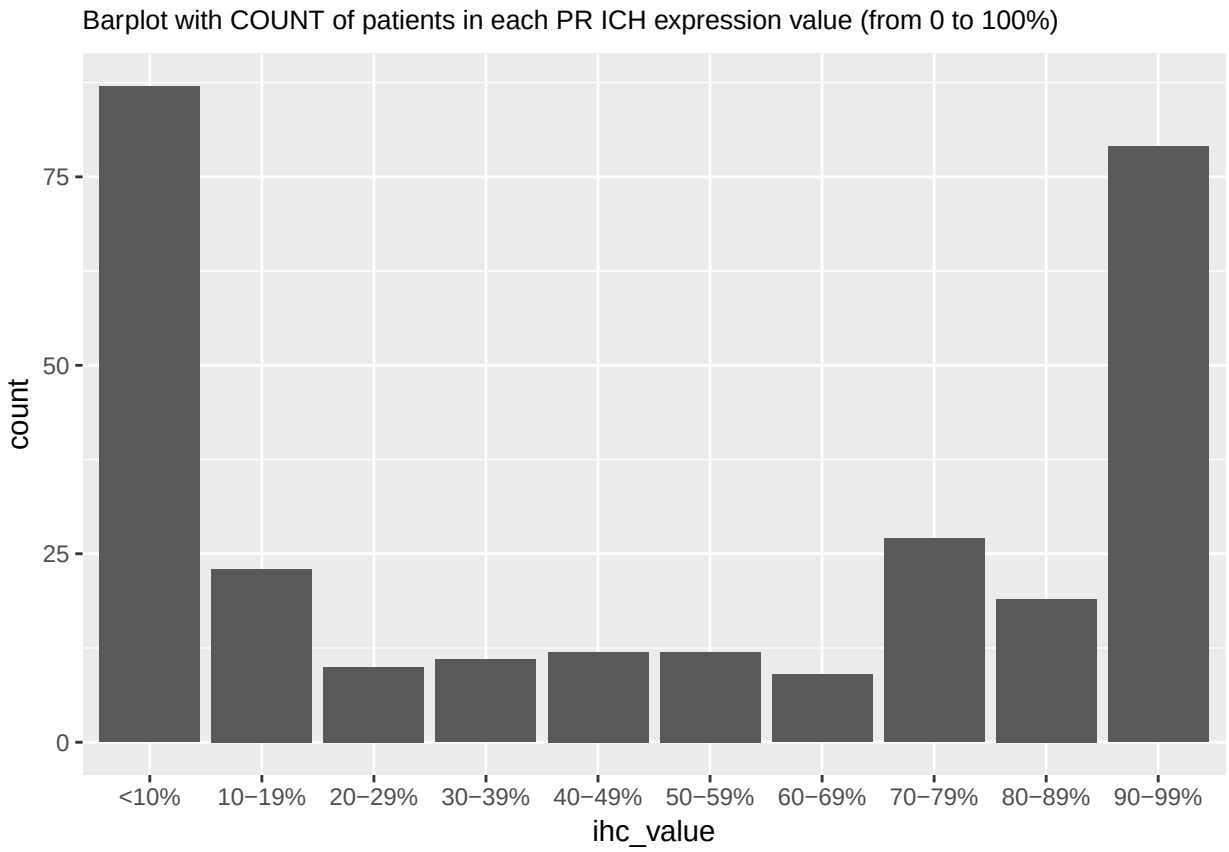
Explore RNA-seq Z-score

```
p1 <- ggplot(dfPR, aes(x=expressionValue)) +
        geom_density(kernel="gaussian") +
        geom_vline(aes(xintercept=0.3, color="red")) +
        labs(x="Expression z-scores", title="Rna-seq expression density plot") +
        theme(legend.position = "none", plot.title=element_text(size=10))
p1
```



Explore ICH values

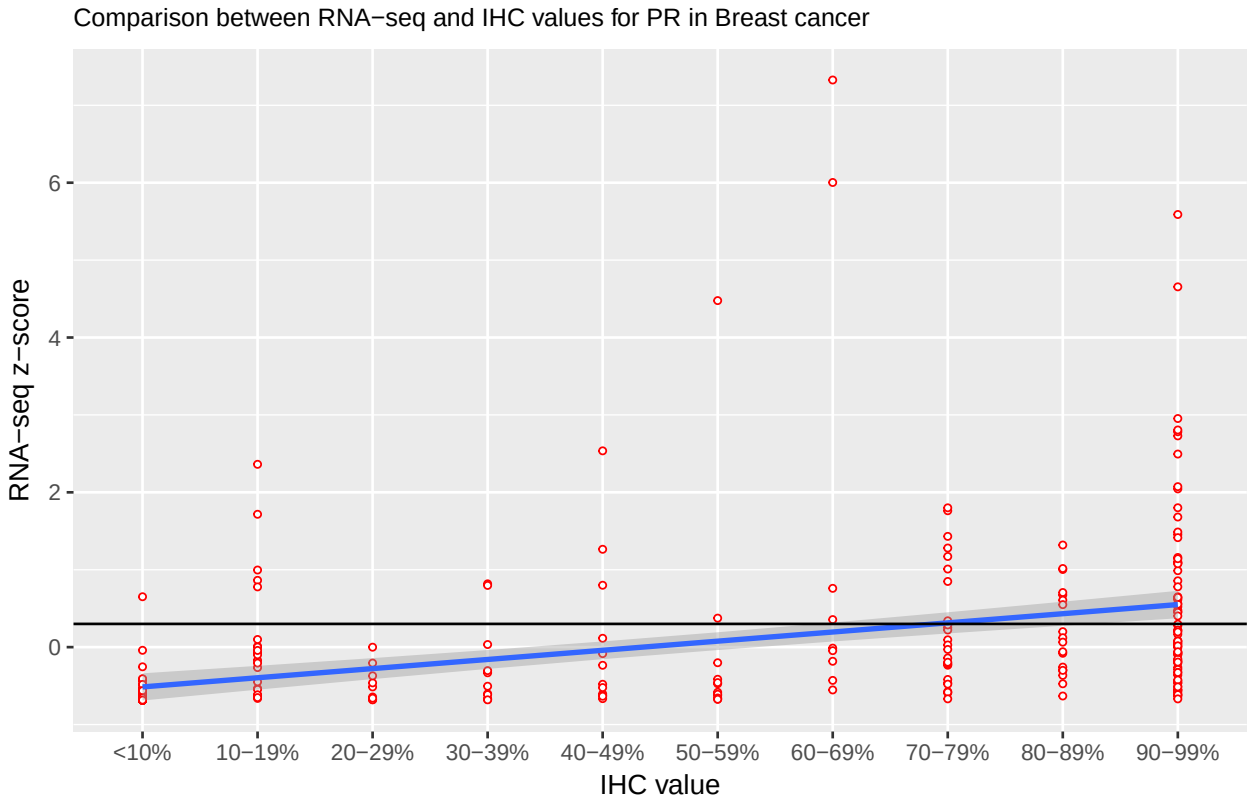
```
p2 <- ggplot(data=dfPR, aes(x=ihc_value)) +
        geom_bar(stat = "count", position = "stack") +
        labs(title="Barplot with COUNT of patients in each PR ICH expression value (from 0 to 100%)")+
        theme(legend.position = "none", plot.title=element_text(size=10))
p2
```



Compare

```
p3 <- ggplot(data=dfPR, aes(x=ihc_value, y=expressionValue, group=1)) +
    geom_point(colour="red", size=1, shape=21, fill="white") +
    labs(title="Comparison between RNA-seq and IHC values for PR in Breast cancer") +
    xlab("IHC value") +
    ylab("RNA-seq z-score") +
    geom_smooth(method="lm") +
    geom_hline(yintercept =0.3) +
    theme(legend.position = "none", plot.title=element_text(size=10))

#ggplotly(p3,width = 650, height = 400, margin(t=1000))
p3
```

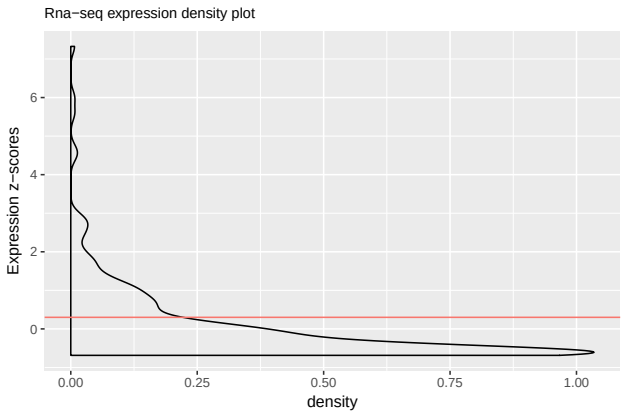


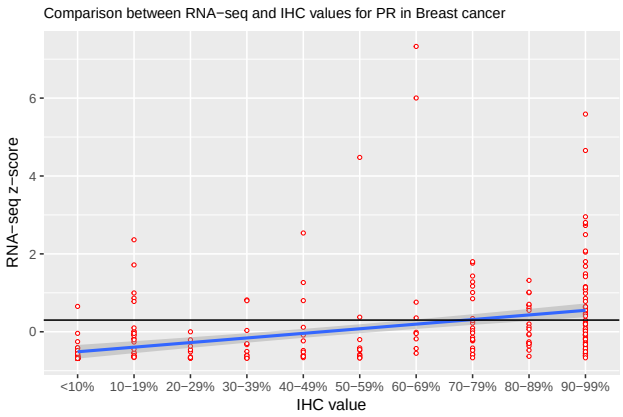
Fit to a linear model

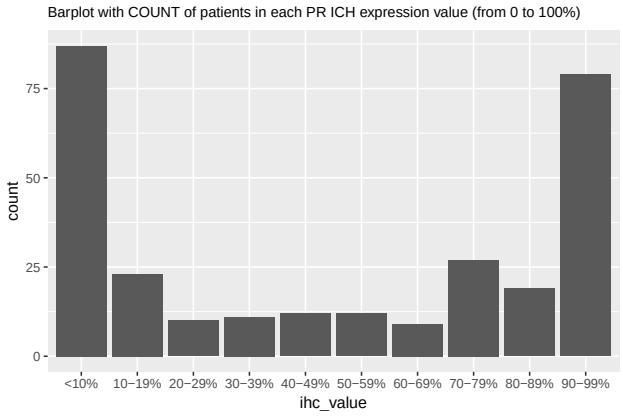
```
# Convert chategorical values to continue numerical value
# <10% = 1
# 10-19% = 2
# etc..
dfPR$ihc_value2 <- as.numeric(factor(dfPR$ihc_value))
# Fit the data into a linea regressino model ache chek the coefficients
summary(lm(expressionValue ~ ihc_value2, dfPR))
```

```
##
## Call:
## lm(formula = expressionValue ~ ihc_value2, data = dfPR)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.2185 -0.4923 -0.1667  0.0834  7.1311
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.63187    0.10126  -6.240 1.57e-09 ***
## ihc_value2   0.11819    0.01522   7.763 1.47e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9817 on 287 degrees of freedom
## Multiple R-squared:  0.1736, Adjusted R-squared:  0.1707
## F-statistic: 60.27 on 1 and 287 DF,  p-value: 1.466e-13
```

Put all the plots together







Session Info

```
sessionInfo()
```

```
## R version 3.3.3 (2017-03-06)
## Platform: x86_64-apple-darwin13.4.0 (64-bit)
## Running under: macOS  10.13.1
##
## 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] grid      stats     graphics  grDevices utils     datasets  methods
## [8] base
##
## other attached packages:
## [1] gridExtra_2.3                gdtools_0.1.6
## [3] bindrcpp_0.2                 PrecisionTrialDesigner_0.99.0
## [5] knitr_1.17                   readr_1.1.1
## [7] plotly_4.7.1                 ggplot2_2.2.1.9000
## [9] dplyr_0.7.4
##
## loaded via a namespace (and not attached):
```

```
##  [1] Biobase_2.34.0                httr_1.3.1
##  [3] tidyr_0.7.2                   bit64_0.9-7
##  [5] jsonlite_1.5                  viridisLite_0.2.0
##  [7] AnnotationHub_2.6.5           shiny_1.0.5
##  [9] assertthat_0.2.0              LowMACAAnnotation_0.99.3
## [11] interactiveDisplayBase_1.12.0 highr_0.6
## [13] stats4_3.3.3                  blob_1.1.0
## [15] Rsamtools_1.26.2              yaml_2.1.14
## [17] ggrepel_0.7.0                 lattice_0.20-35
## [19] RSQLite_2.0                   backports_1.1.1
## [21] glue_1.2.0                    digest_0.6.12
## [23] GenomicRanges_1.26.4          RColorBrewer_1.1-2
## [25] XVector_0.14.1                cgdsr_1.2.6
## [27] colorspace_1.3-2              Matrix_1.2-12
## [29] htmltools_0.3.6               httpuv_1.3.5
## [31] R.oo_1.21.0                   plyr_1.8.4
## [33] XML_3.98-1.9                  pkgconfig_2.0.1
## [35] biomaRt_2.30.0                zlibbioc_1.20.0
## [37] purrr_0.2.4                   xtable_1.8-2
## [39] scales_0.5.0                  svglite_1.2.1
## [41] brglm_0.6.1                   BiocParallel_1.8.2
## [43] tibble_1.3.4                  IRanges_2.8.2
## [45] DT_0.2                        SummarizedExperiment_1.4.0
## [47] BiocGenerics_0.20.0           lazyeval_0.2.1
## [49] magrittr_1.5                  mime_0.5
## [51] memoise_1.1.0                 evaluate_0.10.1
## [53] R.methodsS3_1.7.1             BiocInstaller_1.24.0
## [55] tools_3.3.3                   data.table_1.10.4-3
## [57] hms_0.3                       stringr_1.2.0
## [59] googleVis_0.6.2               S4Vectors_0.12.2
## [61] munsell_0.4.3                 Biostrings_2.42.1
## [63] AnnotationDbi_1.36.2          GenomeInfoDb_1.10.3
## [65] profileModel_0.5-9            rlang_0.1.4
## [67] RCurl_1.95-4.8                htmlwidgets_0.9
## [69] labeling_0.3                  bitops_1.0-6
## [71] shinyBS_0.61                  rmarkdown_1.7
## [73] gtable_0.2.0                  codetools_0.2-15
## [75] DBI_0.7                       reshape2_1.4.2
## [77] R6_2.2.2                      GenomicAlignments_1.10.1
## [79] rtracklayer_1.34.2            bit_1.1-12
## [81] bindr_0.1                     rprojroot_1.2
## [83] stringi_1.1.5                 parallel_3.3.3
## [85] Rcpp_0.12.13
```