促进探索性数据可视化：ggpubr在TCGA基因组数据中的应用

Jerry Wang

2017年9月18日

1. 数据准备

TCGA 数据

下面的R代码将会安装RTCGA核心包以及clinical和mRNA基因表达数据包。

```
source("https://bioconductor.org/biocLite.R")
# Install the main RTCGA package
biocLite("RTCGA")
# Install the clinical and mRNA gene expression data packages
biocLite("RTCGA.clinical")
biocLite("RTCGA.mRNA")
```

查看对于每种癌症可提供的数据类型，用如下代码：

```
library(RTCGA)
infoTCGA()
```

```
##                   Cohort  BCR Clinical   CN LowP Methylation mRNA mRNASeq
## ACC-counts           ACC   92       92   90    0          80    0      79
## BLCA-counts         BLCA  412      412  410  112         412    0     408
## BRCA-counts         BRCA 1098     1097 1089   19        1097  526    1093
## CESC-counts         CESC  307      307  295   50         307    0     304
## CHOL-counts         CHOL   51       45   36    0          36    0      36
## COAD-counts         COAD  460      458  451   69         457  153     457
## COADREAD-counts COADREAD  631      629  616  104         622  222     623
## DLBC-counts         DLBC   58       48   48    0          48    0      48
## ESCA-counts         ESCA  185      185  184   51         185    0     184
## FPPP-counts         FPPP   38       38    0    0           0    0       0
## GBM-counts           GBM  613      595  577    0         420  540     160
## GBMLGG-counts     GBMLGG 1129     1110 1090   52         936  567     676
## HNSC-counts         HNSC  528      528  522  108         528    0     520
## KICH-counts         KICH  113      113   66    0          66    0      66
## KIPAN-counts       KIPAN  973      941  883    0         892   88     889
## KIRC-counts         KIRC  537      537  528    0         535   72     533
## KIRP-counts         KIRP  323      291  289    0         291   16     290
## LAML-counts         LAML  200      200  197    0         194    0     179
## LGG-counts           LGG  516      515  513   52         516   27     516
## LIHC-counts         LIHC  377      377  370    0         377    0     371
## LUAD-counts         LUAD  585      522  516  120         578   32     515
## LUSC-counts         LUSC  504      504  501    0         503  154     501
## MESO-counts         MESO   87       87   87    0          87    0      87
## OV-counts             OV  602      591  586    0         594  574     304
## PAAD-counts         PAAD  185      185  184    0         184    0     178
## PCPG-counts         PCPG  179      179  175    0         179    0     179
## PRAD-counts         PRAD  499      499  492  115         498    0     497
## READ-counts         READ  171      171  165   35         165   69     166
## SARC-counts         SARC  261      261  257    0         261    0     259
## SKCM-counts         SKCM  470      470  469  118         470    0     469
## STAD-counts         STAD  443      443  442  107         443    0     415
## STES-counts         STES  628      628  626  158         628    0     599
## TGCT-counts         TGCT  150      134  150    0         150    0     150
## THCA-counts         THCA  503      503  499   98         503    0     501
## THYM-counts         THYM  124      124  123    0         124    0     120
## UCEC-counts         UCEC  560      548  540  106         547   54     545
## UCS-counts           UCS   57       57   56    0          57    0      57
## UVM-counts           UVM   80       80   80   51          80    0      80
##                 miR miRSeq RPPA MAF rawMAF
## ACC-counts        0     80   46  90      0
## BLCA-counts       0    409  344 130    395
## BRCA-counts       0   1078  887 977      0
## CESC-counts       0    307  173 194      0
## CHOL-counts       0     36   30  35      0
## COAD-counts       0    406  360 154    367
## COADREAD-counts   0    549  491 223    489
## DLBC-counts       0     47   33  48      0
## ESCA-counts       0    184  126 185      0
## FPPP-counts       0     23    0   0      0
## GBM-counts      565      0  238 290    290
## GBMLGG-counts   565    512  668 576    806
## HNSC-counts       0    523  212 279    510
## KICH-counts       0     66   63  66     66
## KIPAN-counts      0    873  756 644    799
## KIRC-counts       0    516  478 417    451
## KIRP-counts       0    291  215 161    282
## LAML-counts       0    188    0 197      0
## LGG-counts        0    512  430 286    516
## LIHC-counts       0    372   63 198    373
## LUAD-counts       0    513  365 230    542
## LUSC-counts       0    478  328 178      0
## MESO-counts       0     87   63   0      0
## OV-counts       570    453  426 316    469
## PAAD-counts       0    178  123 150    184
## PCPG-counts       0    179   80 179      0
## PRAD-counts       0    494  352 332    498
## READ-counts       0    143  131  69    122
## SARC-counts       0    259  223 247      0
## SKCM-counts       0    448  353 343    366
## STAD-counts       0    436  357 289    395
## STES-counts       0    620  483 474    395
## TGCT-counts       0    150  118 149      0
## THCA-counts       0    502  222 402    496
## THYM-counts       0    124   90 123      0
## UCEC-counts       0    538  440 248      0
## UCS-counts        0     56   48  57      0
## UVM-counts        0     80   12  80      0
```

基因表达数据

R函数 `expressionsTCGA()` (`RTCGA` 包内)可以轻松提取一种或者多种癌症中你感兴趣基因的表达值。

在接下来的代码中将会提取三个如下不同数据集中感兴趣的五个基因 GATA3, PTEN, XBP1, ESR1 和 MUC1 的 mRNA 表达值：

- Breast invasive carcinoma (BRCA),
- Ovarian serous cystadenocarcinoma (OV) and
- Lung squamous cell carcinoma (LUSC)

```
library(RTCGA)
library(RTCGA.mRNA)
expr <- expressionsTCGA(BRCA.mRNA, OV.mRNA, LUSC.mRNA,
                        extract.cols = c("GATA3", "PTEN", "XBP1","ESR1", "MUC1"))
expr
```

```
## # A tibble: 1,305 x 7
##             bcr_patient_barcode   dataset     GATA3       PTEN      XBP1
##                           <chr>     <chr>     <dbl>      <dbl>     <dbl>
##  1 TCGA-A1-A0SD-01A-11R-A115-07 BRCA.mRNA  2.870500  1.3613571  2.983333
##  2 TCGA-A1-A0SE-01A-11R-A084-07 BRCA.mRNA  2.166250  0.4283571  2.550833
##  3 TCGA-A1-A0SH-01A-11R-A084-07 BRCA.mRNA  1.323500  1.3056429  3.020417
##  4 TCGA-A1-A0SJ-01A-11R-A084-07 BRCA.mRNA  1.841625  0.8096429  3.131333
##  5 TCGA-A1-A0SK-01A-12R-A084-07 BRCA.mRNA -6.025250  0.2508571 -1.451750
##  6 TCGA-A1-A0SM-01A-11R-A084-07 BRCA.mRNA  1.804500  1.3107857  4.041083
##  7 TCGA-A1-A0SO-01A-22R-A084-07 BRCA.mRNA -4.879250 -0.2369286 -0.724750
##  8 TCGA-A1-A0SP-01A-11R-A084-07 BRCA.mRNA -3.143250 -1.2432143 -1.193083
##  9 TCGA-A2-A04N-01A-11R-A115-07 BRCA.mRNA  2.034000  1.2074286  2.278833
## 10 TCGA-A2-A04P-01A-31R-A034-07 BRCA.mRNA -0.293125  0.2883571 -1.605083
## # ... with 1,295 more rows, and 2 more variables: ESR1 <dbl>, MUC1 <dbl>
```

显示每个数据集中样本的数量可以使用如下命令：

```
nb_samples <- table(expr$dataset)
nb_samples
```

```
##
## BRCA.mRNA LUSC.mRNA   OV.mRNA
##       590       154       561
```

我们可以减缓数据集的名字，即删除标签中的“mRNA”。完成这个过程可以使用R的基本函数 `gsub()` 。同时简化病人的barcode列。

```
expr$dataset <- gsub(pattern = ".mRNA", replacement = "",  expr$dataset)
expr$bcr_patient_barcode <- paste0(expr$dataset, c(1:590, 1:561, 1:154))
expr
```

```
## # A tibble: 1,305 x 7
##    bcr_patient_barcode dataset     GATA3       PTEN      XBP1        ESR1
##                  <chr>   <chr>     <dbl>      <dbl>     <dbl>       <dbl>
##  1               BRCA1    BRCA  2.870500  1.3613571  2.983333  3.0842500
##  2               BRCA2    BRCA  2.166250  0.4283571  2.550833  2.3860000
##  3               BRCA3    BRCA  1.323500  1.3056429  3.020417  0.7912500
##  4               BRCA4    BRCA  1.841625  0.8096429  3.131333  2.4954167
##  5               BRCA5    BRCA -6.025250  0.2508571 -1.451750 -4.8606667
##  6               BRCA6    BRCA  1.804500  1.3107857  4.041083  2.7970000
##  7               BRCA7    BRCA -4.879250 -0.2369286 -0.724750 -4.4860833
##  8               BRCA8    BRCA -3.143250 -1.2432143 -1.193083 -1.6274167
##  9               BRCA9    BRCA  2.034000  1.2074286  2.278833  4.1155833
## 10              BRCA10    BRCA -0.293125  0.2883571 -1.605083  0.4731667
## # ... with 1,295 more rows, and 1 more variables: MUC1 <dbl>
```

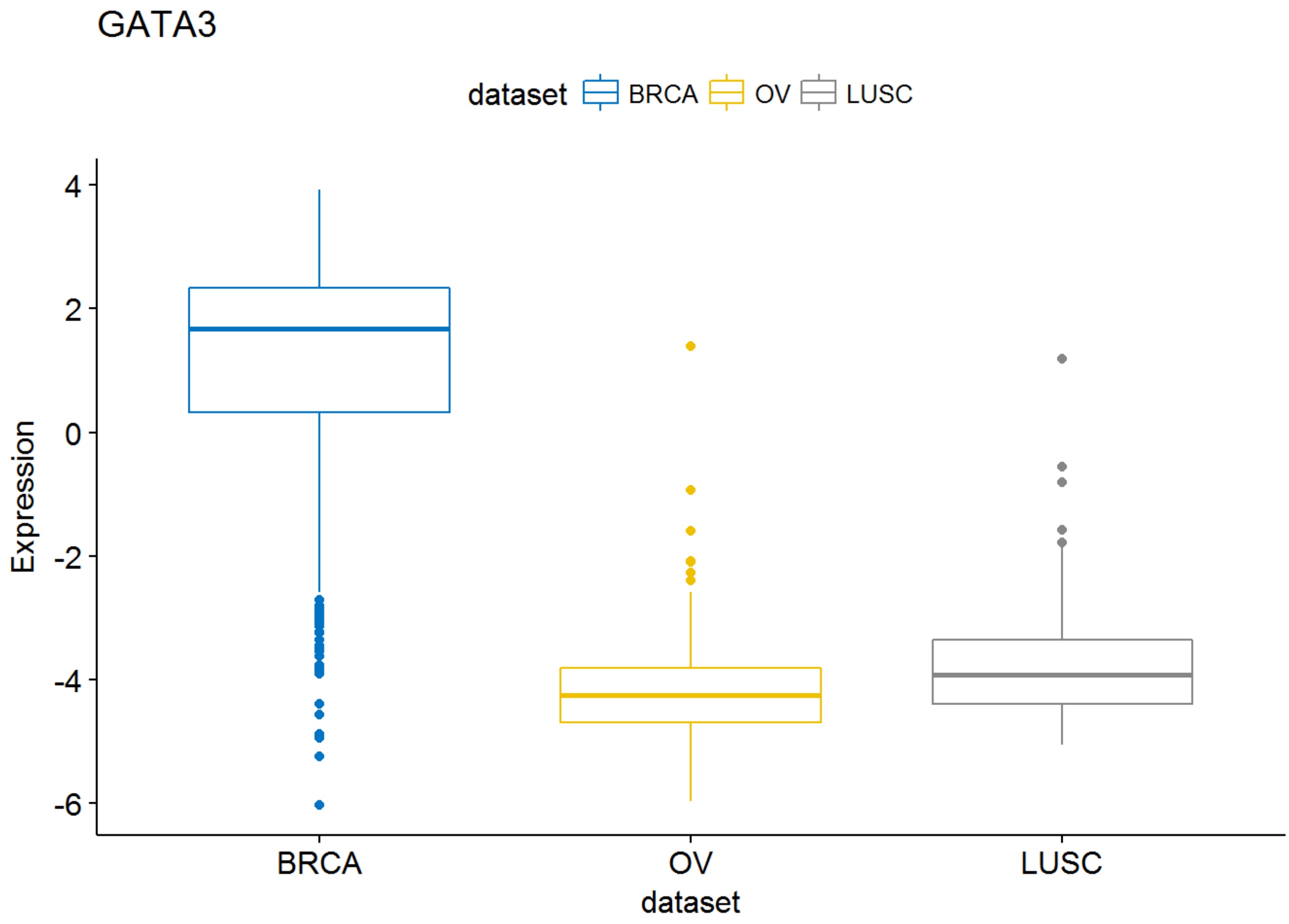
如果你执行安装RTCGA包有问题可以直接从本地文件中读取所需数据：

```
expr <- read.delim("expr_tcga.txt", stringsAsFactors = FALSE)
```

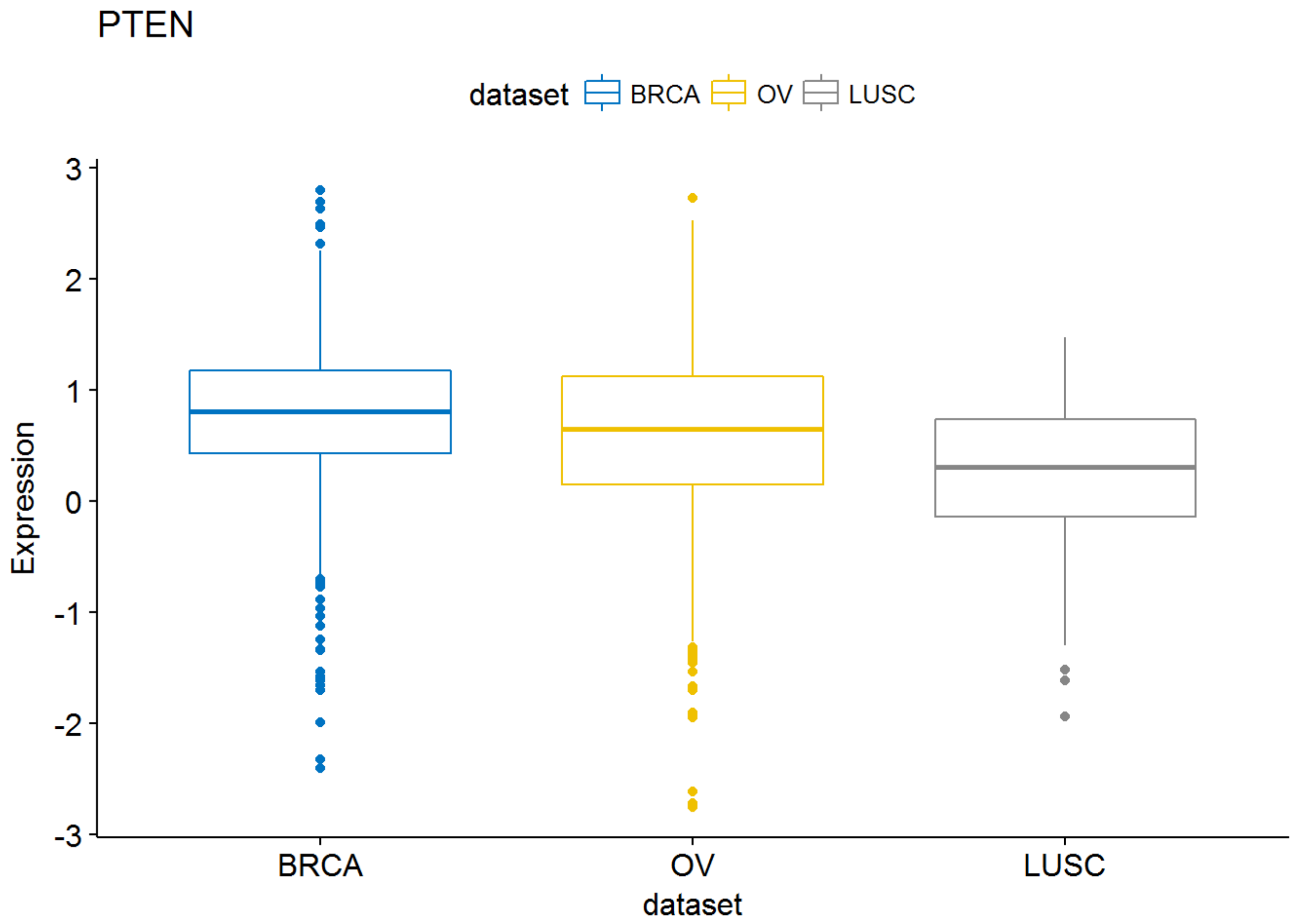
2. 箱线图

创建基因表达谱的箱线图，不同分组（数据集或者说癌症类型）使用不同的颜色：

```
library(ggpubr)
# GATA3
ggboxplot(expr, x = "dataset", y = "GATA3",
          title = "GATA3", ylab = "Expression",
          color = "dataset", palette = "jco")
```



```
# PTEN
ggboxplot(expr, x = "dataset", y = "PTEN",
          title = "PTEN", ylab = "Expression",
          color = "dataset", palette = "jco")
```

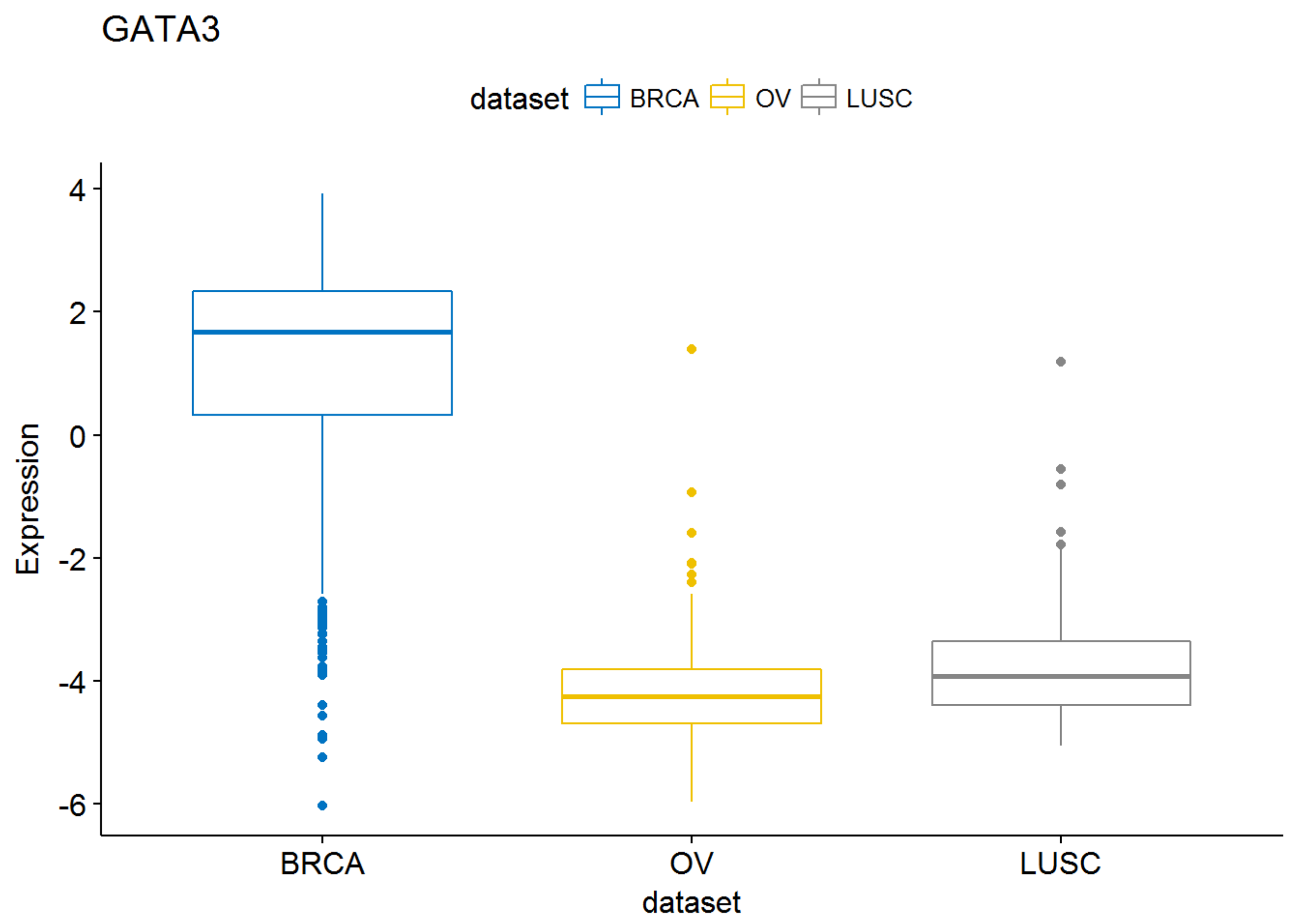


注意： 参数 **palette** 用来改变颜色调色板。允许使用的颜色调色板包括：

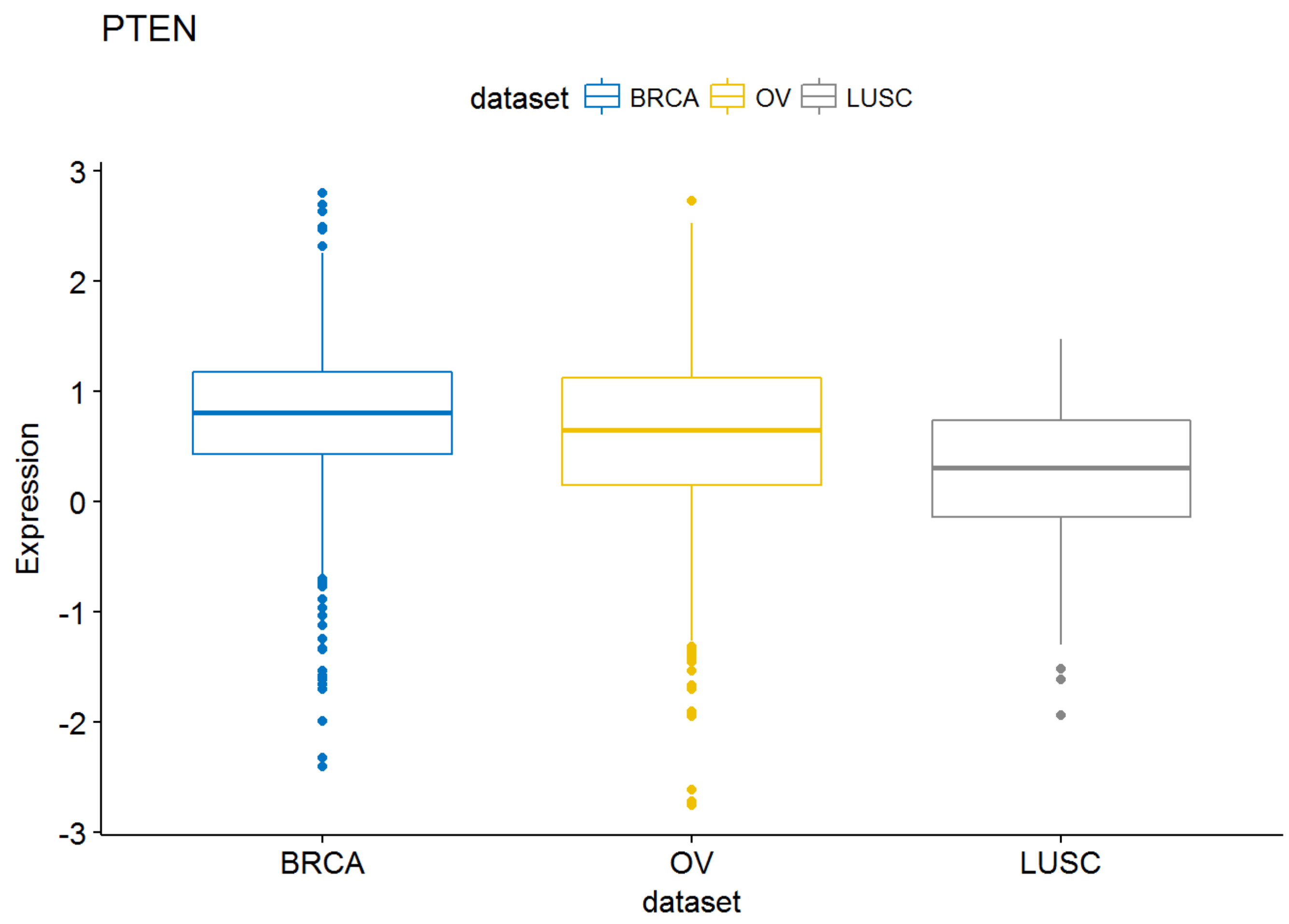
- “grey”使用灰度调色板。
- brewer palettes，例如： “RdBu”, “Blues” 等等。可以使用函数 `RColorBrewer::display.brewer.all()` 来查看可用调色板。
- 自定义调色板： c(“blue”, “red”) 或 c(“#00AFBB”, “#E7B800”)。
- 来自 `ggsci` (https://cran.r-project.org/web/packages/ggsci/vignettes/ggsci.html)包的科学类杂志调色板，例如： “npg”, “aaas”, “lancet”, “jco”, “ucscgb”, “uchicago”, “simpsons” 和 “rickandmorty”。

如果不想为每个基因都重复相同的R代码，可以创建一个plot list。

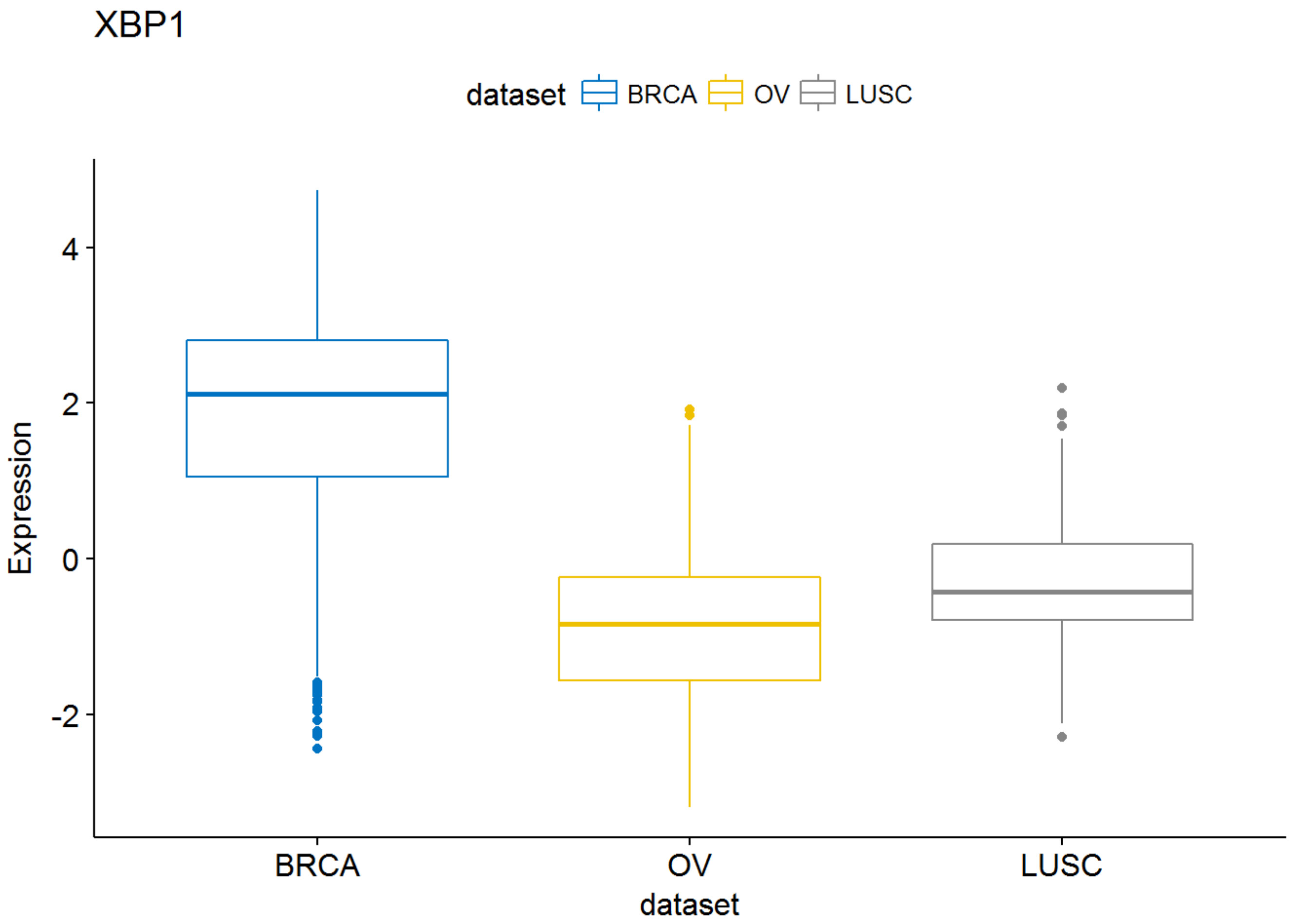
```
# Create a  list of plots
p <- ggboxplot(expr, x = "dataset",
               y = c("GATA3", "PTEN", "XBP1"),
               title = c("GATA3", "PTEN", "XBP1"),
               ylab = "Expression",
               color = "dataset", palette = "jco")
# View GATA3
p$GATA3
```



```
# View PTEN
p$PTEN
```



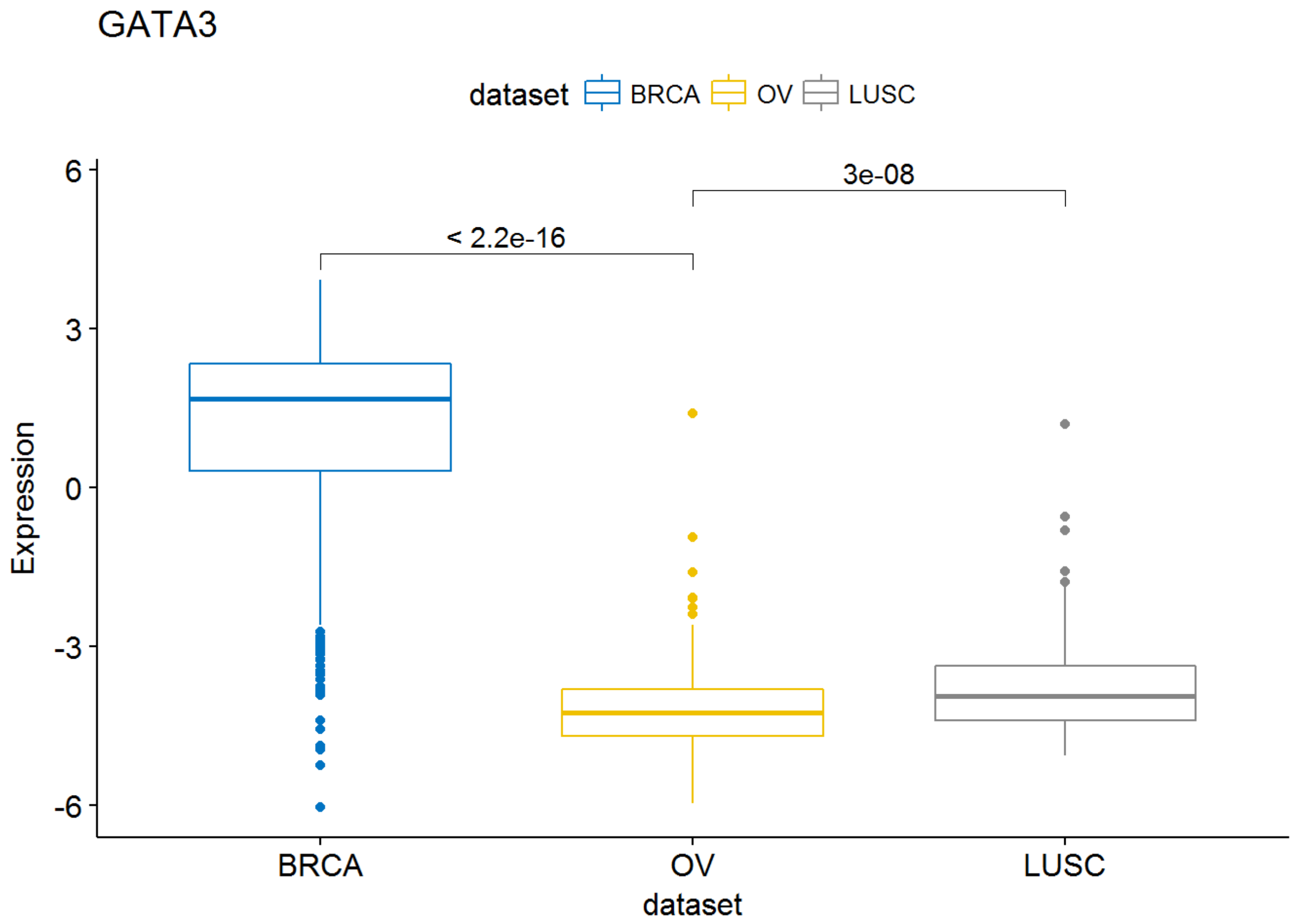
```
# View XBP1
p$XBP1
```



> 当参数y包含多个变量(这里是多个基因名字)，那么参数title, xlab 和 ylab 可以是与 y 等长的字符串向量。也可以是单个字符串，若如此该字符串将应用于所有的图片。

给箱线图添加p-values和显著性水平：

```
my_comparisons <- list(c("BRCA", "OV"), c("OV", "LUSC"))
ggboxplot(expr, x = "dataset", y = "GATA3",
          title = "GATA3", ylab = "Expression",
          color = "dataset", palette = "jco")+
  stat_compare_means(comparisons = my_comparisons)
```



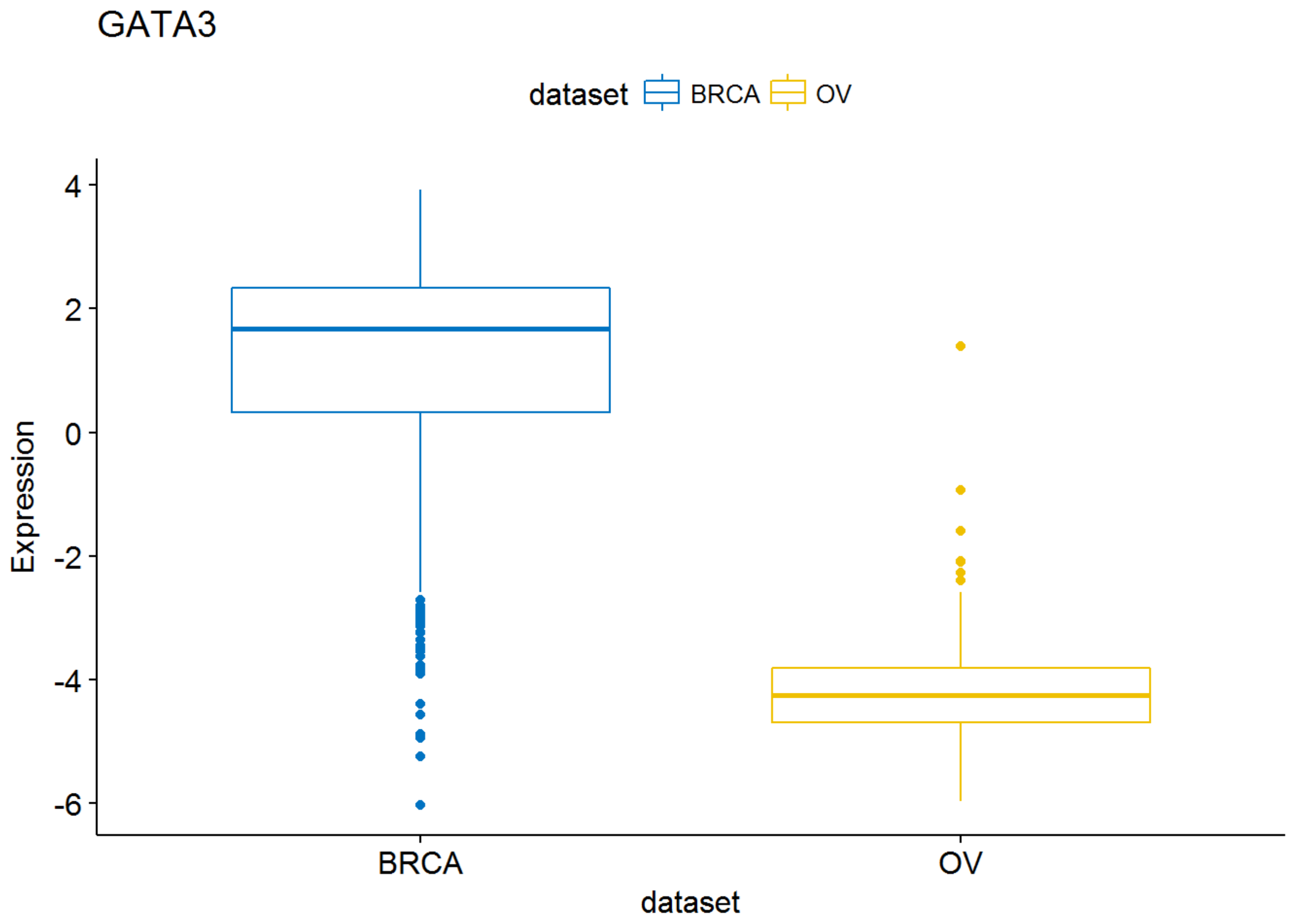
对于每一个基因，你可以比较不同的组间的差异：

```
compare_means(c(GATA3, PTEN, XBP1) ~ dataset, data = expr)
```

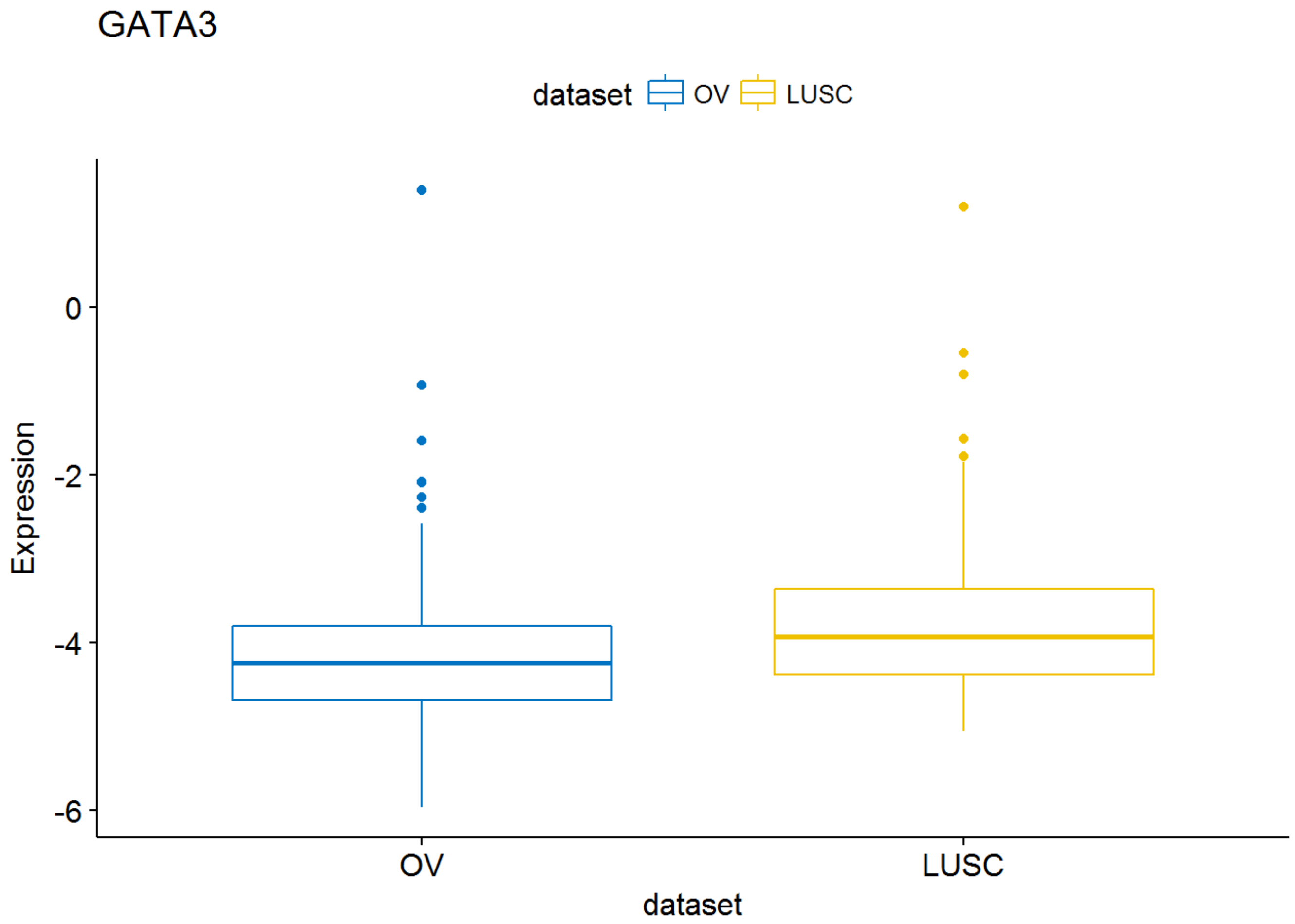
```
## # A tibble: 9 x 8
##      .y.  group1 group2             p         p.adj  p.format p.signif
##   <fctr>  <chr>  <chr>          <dbl>         <dbl>    <chr>    <chr>
## 1  GATA3   BRCA     OV 1.111768e-177 3.335304e-177  < 2e-16     ****
## 2  GATA3   BRCA   LUSC  6.684016e-73  1.336803e-72  < 2e-16     ****
## 3  GATA3     OV   LUSC  2.965702e-08  2.965702e-08  3.0e-08     ****
## 4   PTEN   BRCA     OV  6.791940e-05  6.791940e-05  6.8e-05     ****
## 5   PTEN   BRCA   LUSC  1.042830e-16  3.128489e-16  < 2e-16     ****
## 6   PTEN     OV   LUSC  1.280576e-07  2.561153e-07  1.3e-07     ****
## 7   XBP1   BRCA     OV 2.551228e-123 7.653685e-123  < 2e-16     ****
## 8   XBP1   BRCA   LUSC  1.950162e-42  3.900324e-42  < 2e-16     ****
## 9   XBP1     OV   LUSC  4.239570e-11  4.239570e-11  4.2e-11     ****
## # ... with 1 more variables: method <chr>
```

如果你想选择或者删除某些组（这里是癌症类型）来作图展示，那么可以使用参数 select 或者 remove：

```
# Select BRCA and OV cancer types
ggboxplot(expr, x = "dataset", y = "GATA3",
          title = "GATA3", ylab = "Expression",
          color = "dataset", palette = "jco",
          select = c("BRCA", "OV"))
```

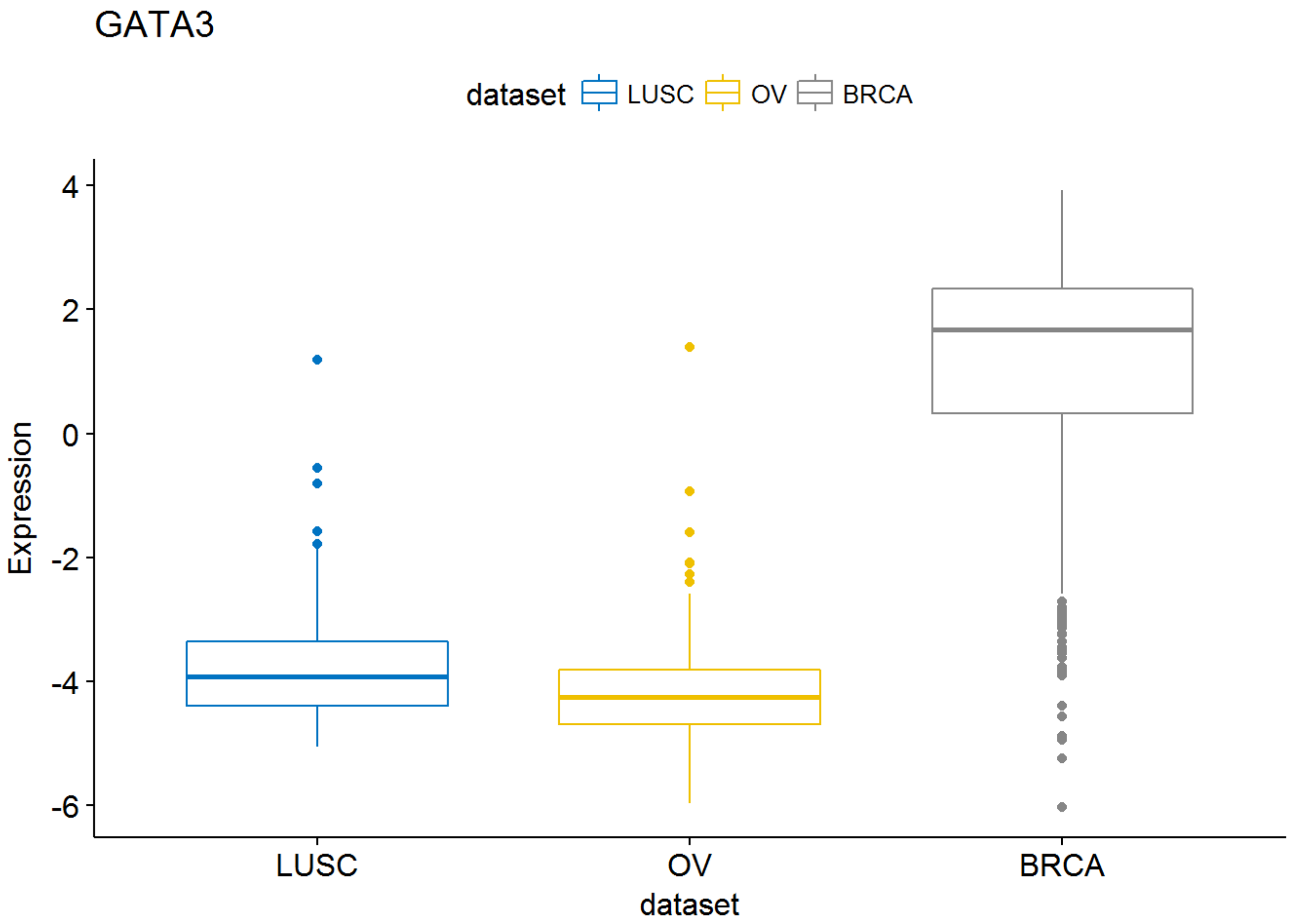


```
# or remove BRCA
ggboxplot(expr, x = "dataset", y = "GATA3",
          title = "GATA3", ylab = "Expression",
          color = "dataset", palette = "jco",
          remove = "BRCA")
```



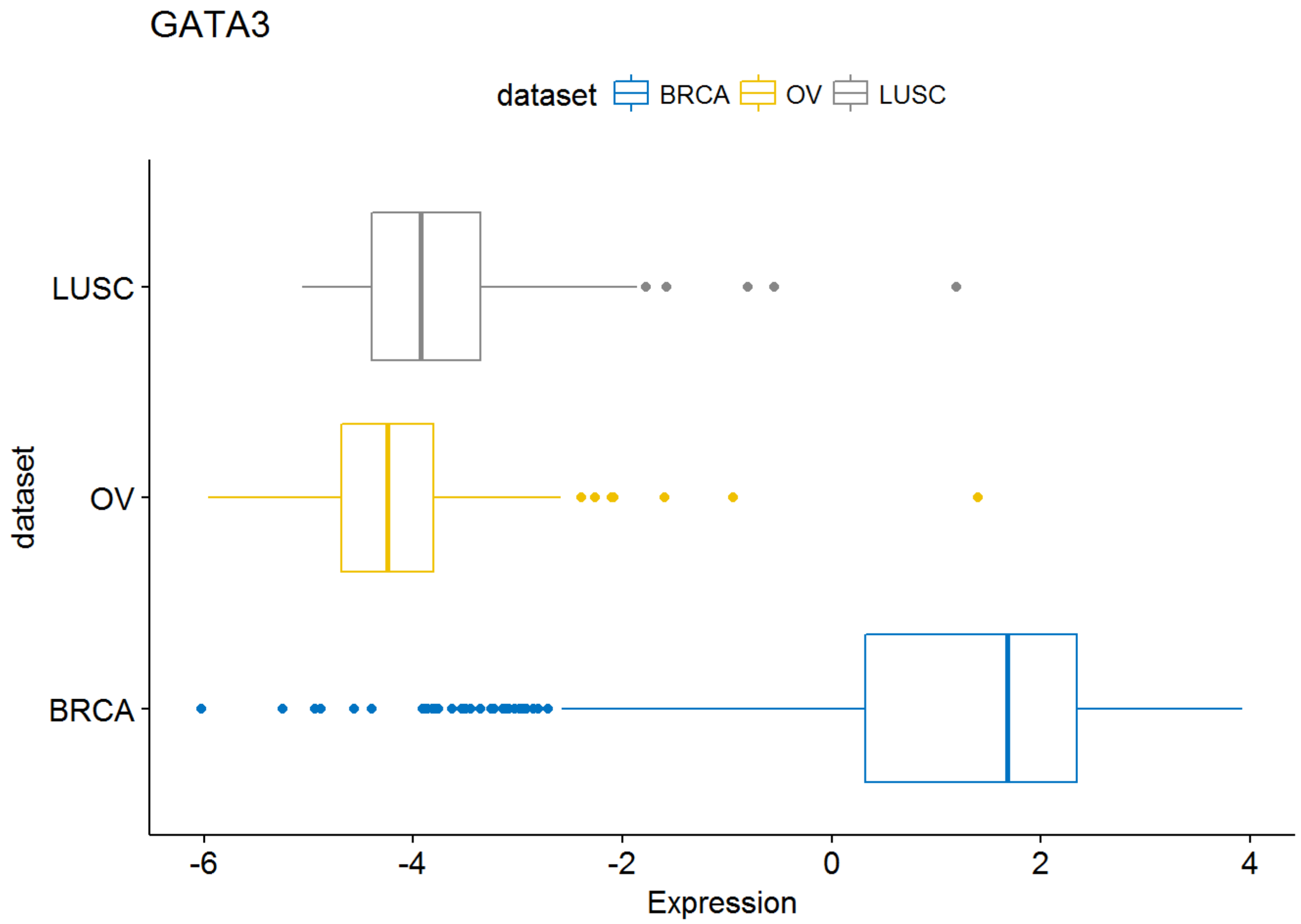
改变分组在x轴上的位置可以使用参数 `order`：

```
# Order data sets
ggboxplot(expr, x = "dataset", y = "GATA3",
          title = "GATA3", ylab = "Expression",
          color = "dataset", palette = "jco",
          order = c("LUSC", "OV", "BRCA"))
```



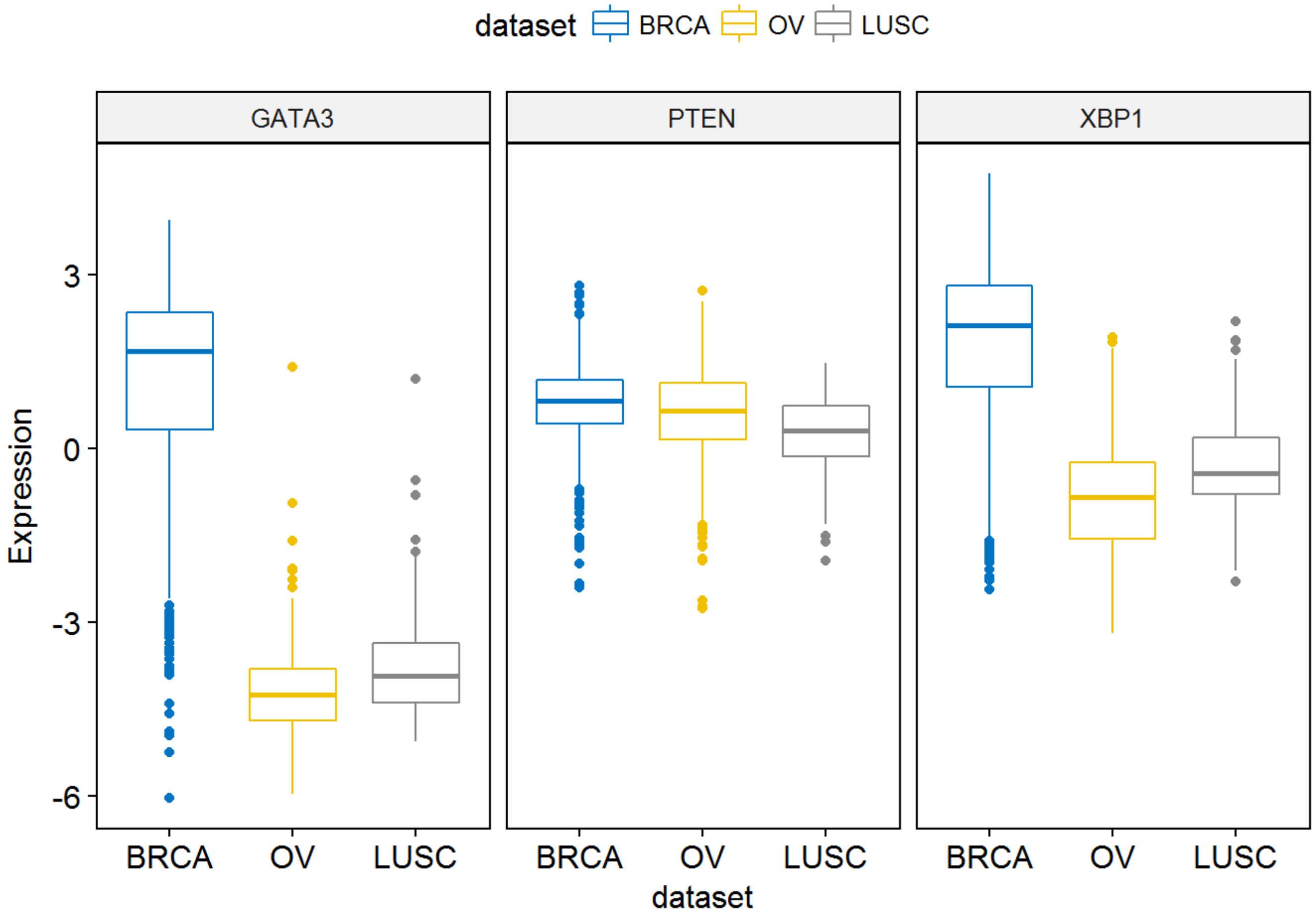
可以使用参数 rotate = TRUE 实现水平箱线图：

```
ggboxplot(expr, x = "dataset", y = "GATA3",
          title = "GATA3", ylab = "Expression",
          color = "dataset", palette = "jco",
          rotate = TRUE)
```



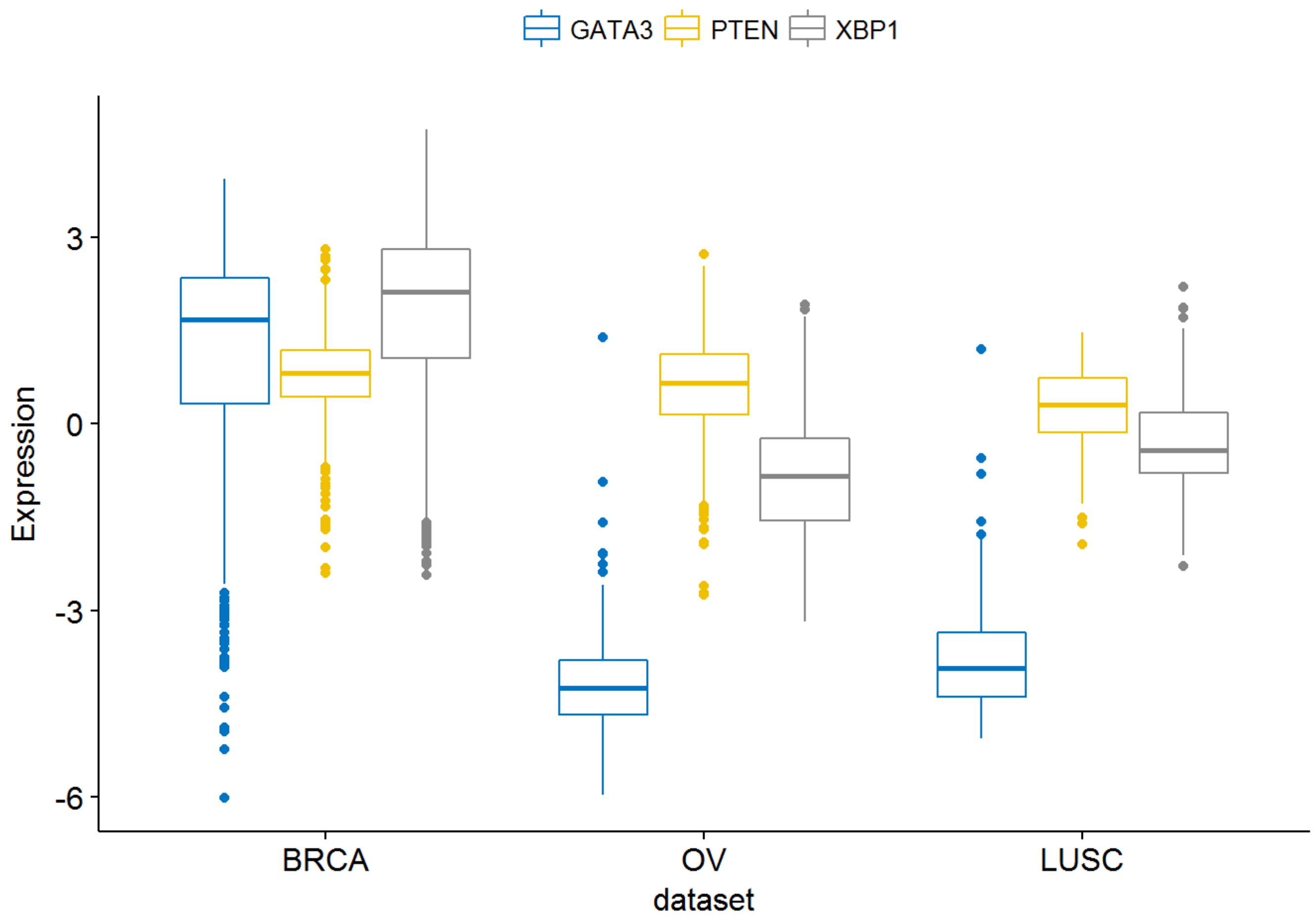
将三个基因的表达图放到一个作图区域可以使用参数 `combine = TRUE`：

```
ggboxplot(expr, x = "dataset",
          y = c("GATA3", "PTEN", "XBP1"),
          combine = TRUE,
          ylab = "Expression",
          color = "dataset", palette = "jco")
```



还可以将三种图合并为一张图展示只要使用 merge = TRUE 或者 merge = “asis”：

```
ggboxplot(expr, x = "dataset",
          y = c("GATA3", "PTEN", "XBP1"),
          merge = TRUE,
          ylab = "Expression",
          palette = "jco")
```

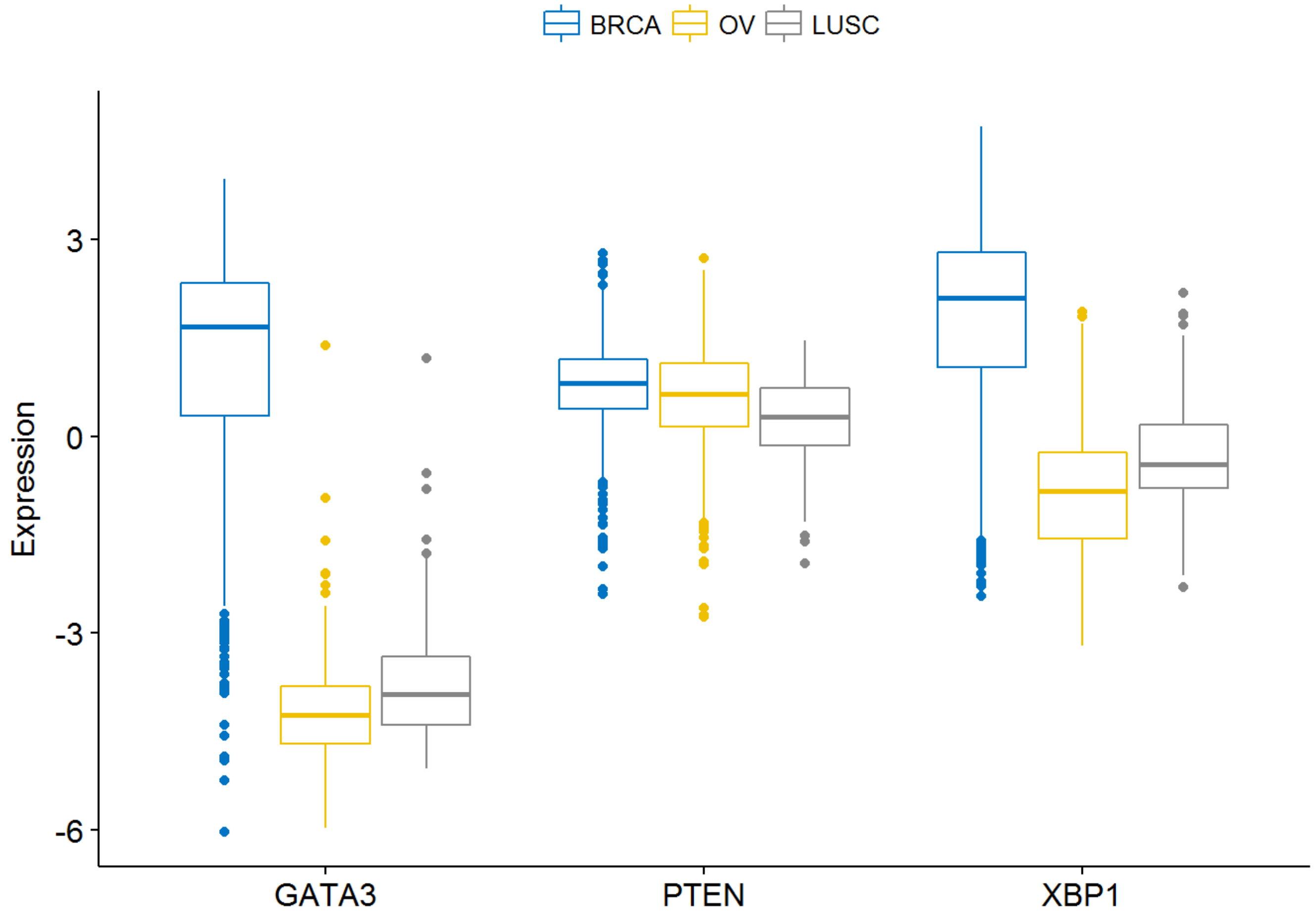


上面的图很容易去比较每种癌症不同基因的表达水平。

但是，你可能想把基因放到x轴方便去比较同一个基因在不同癌症中的表达水平。

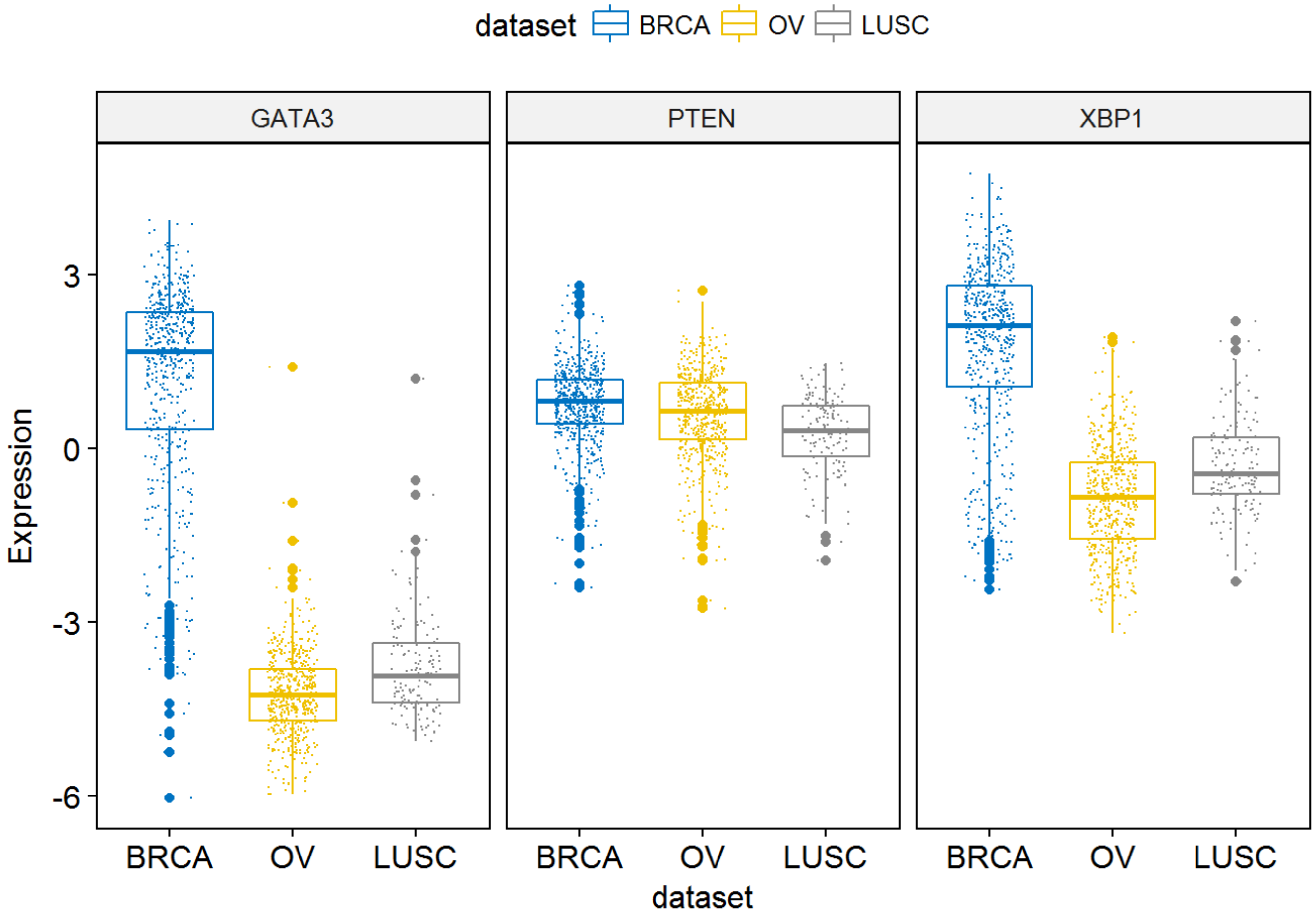
在这种条件下，y变量（这里是基因）变成x轴标签，而x变量（这里是癌种）变成分组变量。如此做的话需要使用参数 merge = “flip” 。

```
ggboxplot(expr, x = "dataset",
          y = c("GATA3", "PTEN", "XBP1"),
          merge = "flip",
          ylab = "Expression",
          palette = "jco")
```



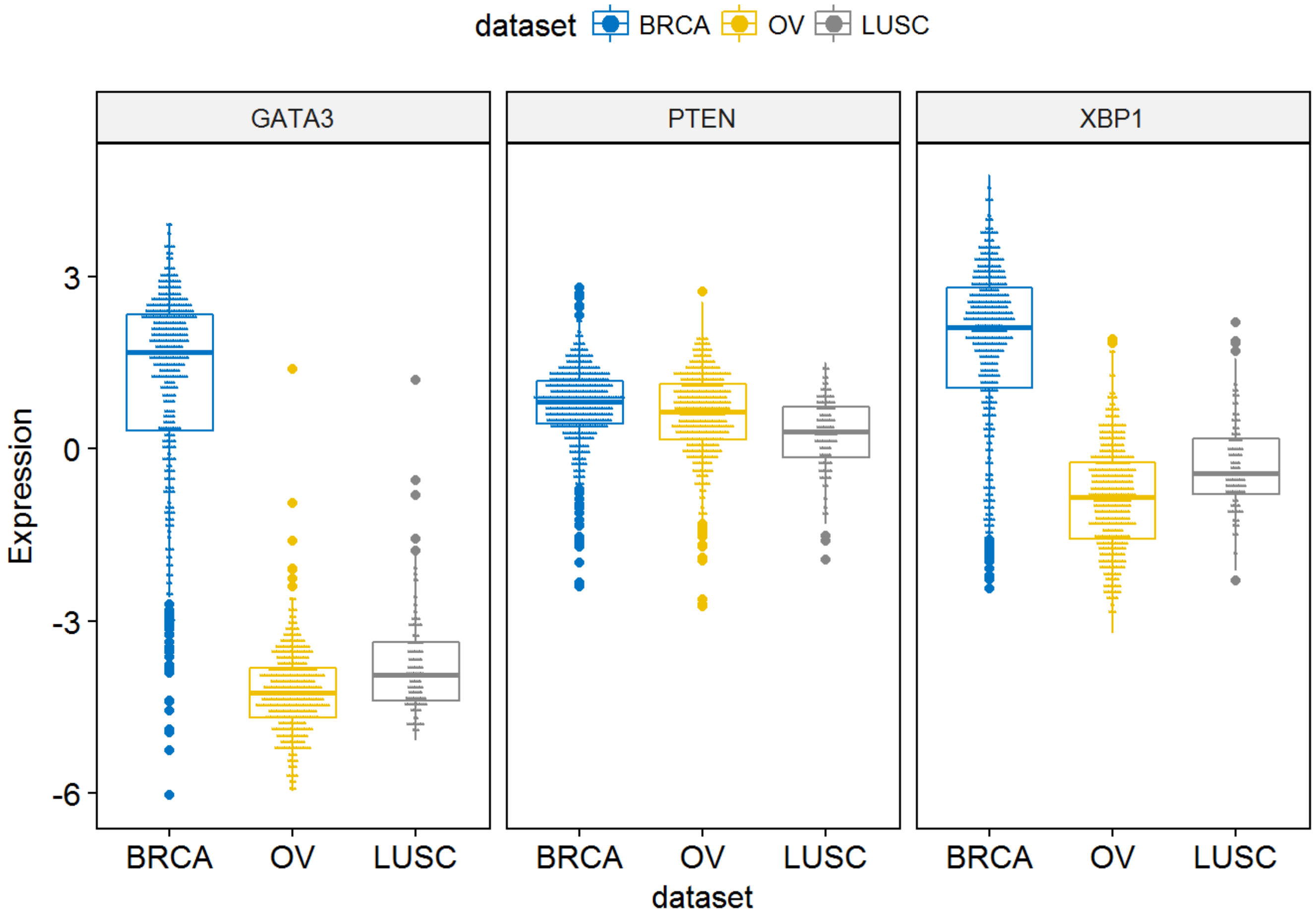
如果要在箱线图上添加打散的点。每一个点就是一个独立的观测值。可以添加 add = “jitter” 。自定义添加元素的特性，指定参数 add.params 。

```
ggboxplot(expr, x = "dataset",
          y = c("GATA3", "PTEN", "XBP1"),
          combine = TRUE,
          color = "dataset", palette = "jco",
          ylab = "Expression",
          add = "jitter",                              # Add jittered points
          add.params = list(size = 0.1, jitter = 0.2)  # Point size and the amount of jittering
          )
```



你还可以在箱线图上添加 dotplot，并调整。

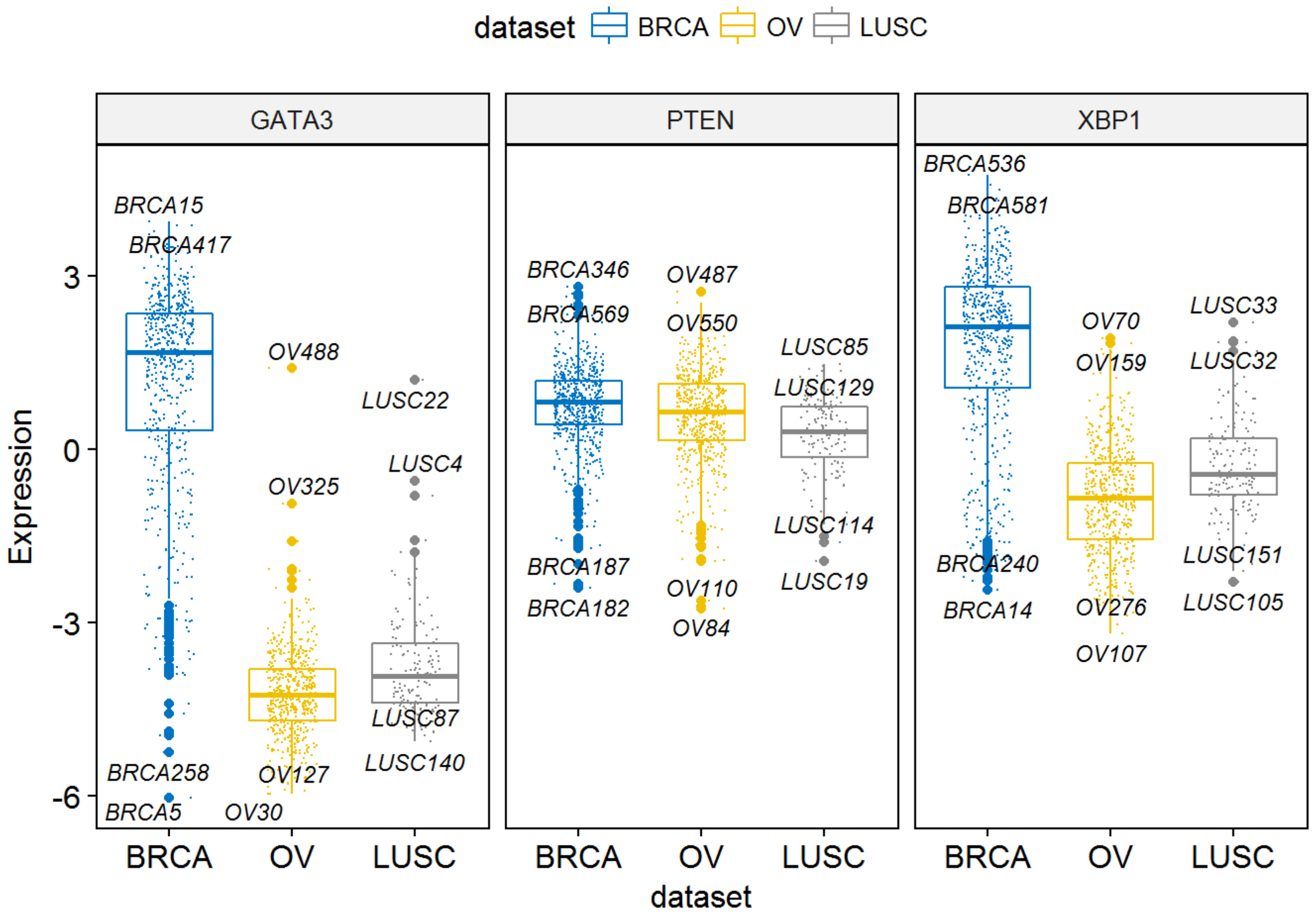
```
ggboxplot(expr, x = "dataset",
          y = c("GATA3", "PTEN", "XBP1"),
          combine = TRUE,
          color = "dataset", palette = "jco",
          ylab = "Expression",
          add = "dotplot",                              # Add dotplot
          add.params = list(binwidth = 0.1, dotsize = 0.3)
          )
```



你可能想在箱线图上将前n最高或最低值的样品的名称显示出来。在这种情况下，您可以使用以下参数：

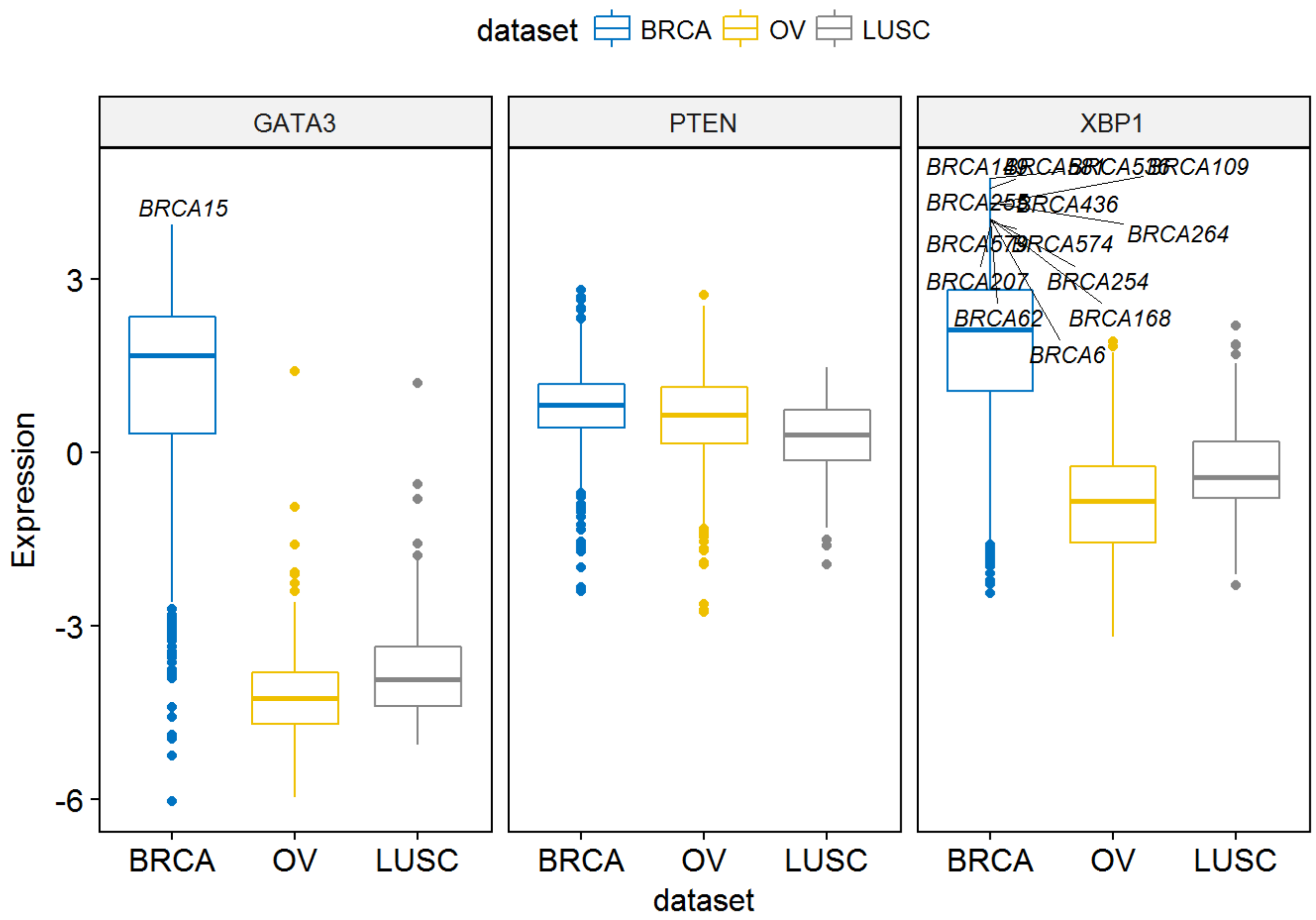
- `label`：包含点标签的列的名字
- `label.select`：可以有两种格式：
    - 一个字符串向量指定需要显示的标签名字
    - 一个list包含一个或者以下多个组分的组合:
        - `top.up` 和 `top.down`：用来显示 top up/down 的点。例如：label.select = list(top.up = 10, top.down = 4)。
        - `criteria`：利用x和y变量值用来过滤满足条件的点。例如：label.select = list(criteria = "`y` > 3.9 & `y` < 5 & `x` %in% c('BRCA', 'OV')")。

```
ggboxplot(expr, x = "dataset",
          y = c("GATA3", "PTEN", "XBP1"),
          combine = TRUE,
          color = "dataset", palette = "jco",
          ylab = "Expression",
          add = "jitter",                              # Add jittered points
          add.params = list(size = 0.1, jitter = 0.2), # Point size and the amount of jitterin
g
          label = "bcr_patient_barcode",               # column containing point labels
          label.select = list(top.up = 2, top.down = 2),# Select some labels to display
          font.label = list(size = 9, face = "italic"), # label font
          repel = TRUE                                 # Avoid label text overplotting
          )
```



一个复杂的标签显示规则的例子如下:

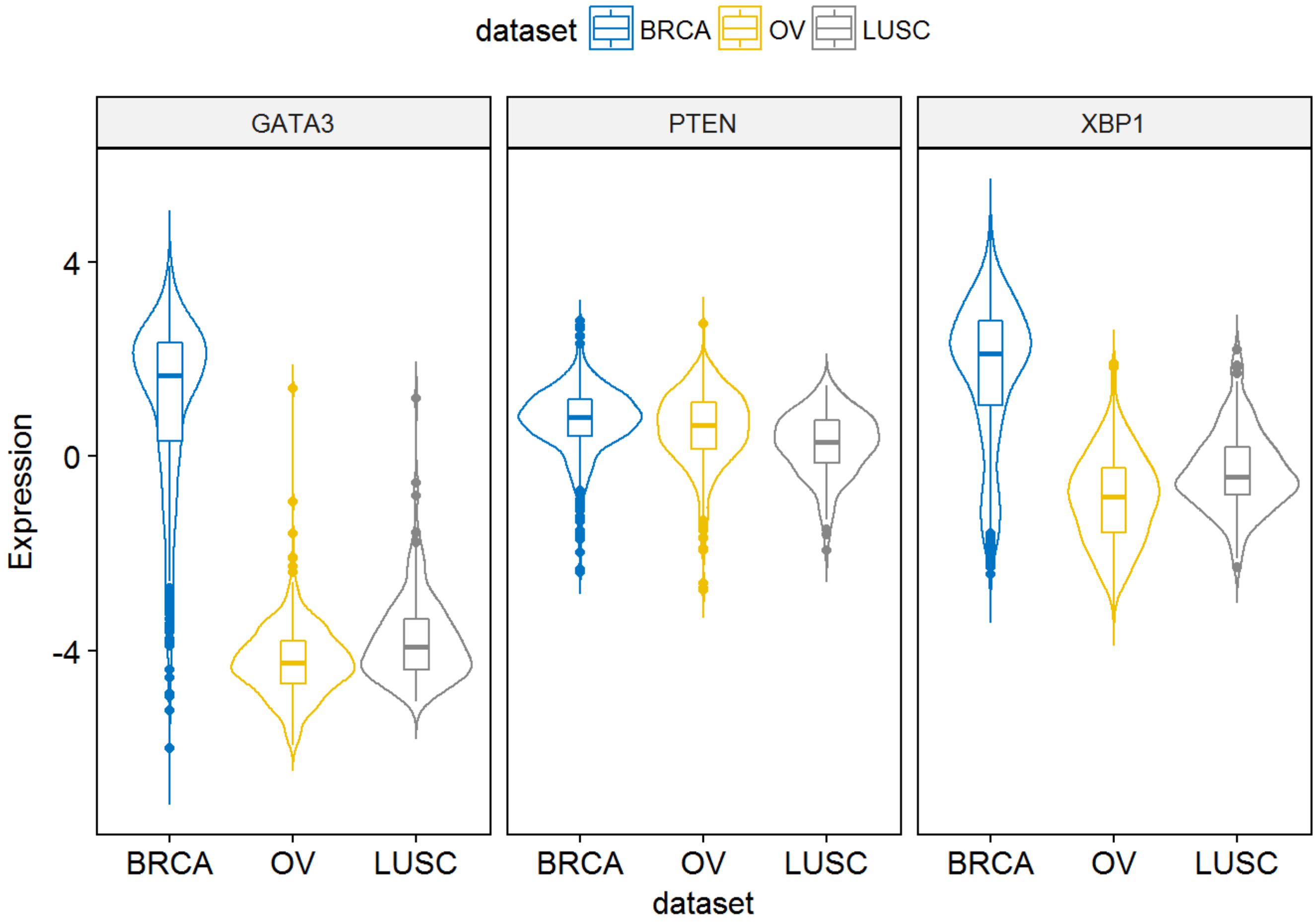
```
label.select.criteria <- list(criteria = "`y` > 3.9 & `x` %in% c('BRCA', 'OV')")
ggboxplot(expr, x = "dataset",
          y = c("GATA3", "PTEN", "XBP1"),
          combine = TRUE,
          color = "dataset", palette = "jco",
          ylab = "Expression",
          label = "bcr_patient_barcode",             # column containing point labels
          label.select = label.select.criteria,       # Select some labels to display
          font.label = list(size = 9, face = "italic"), # label font
          repel = TRUE                                # Avoid label text overplotting
          )
```



3. 小提琴图

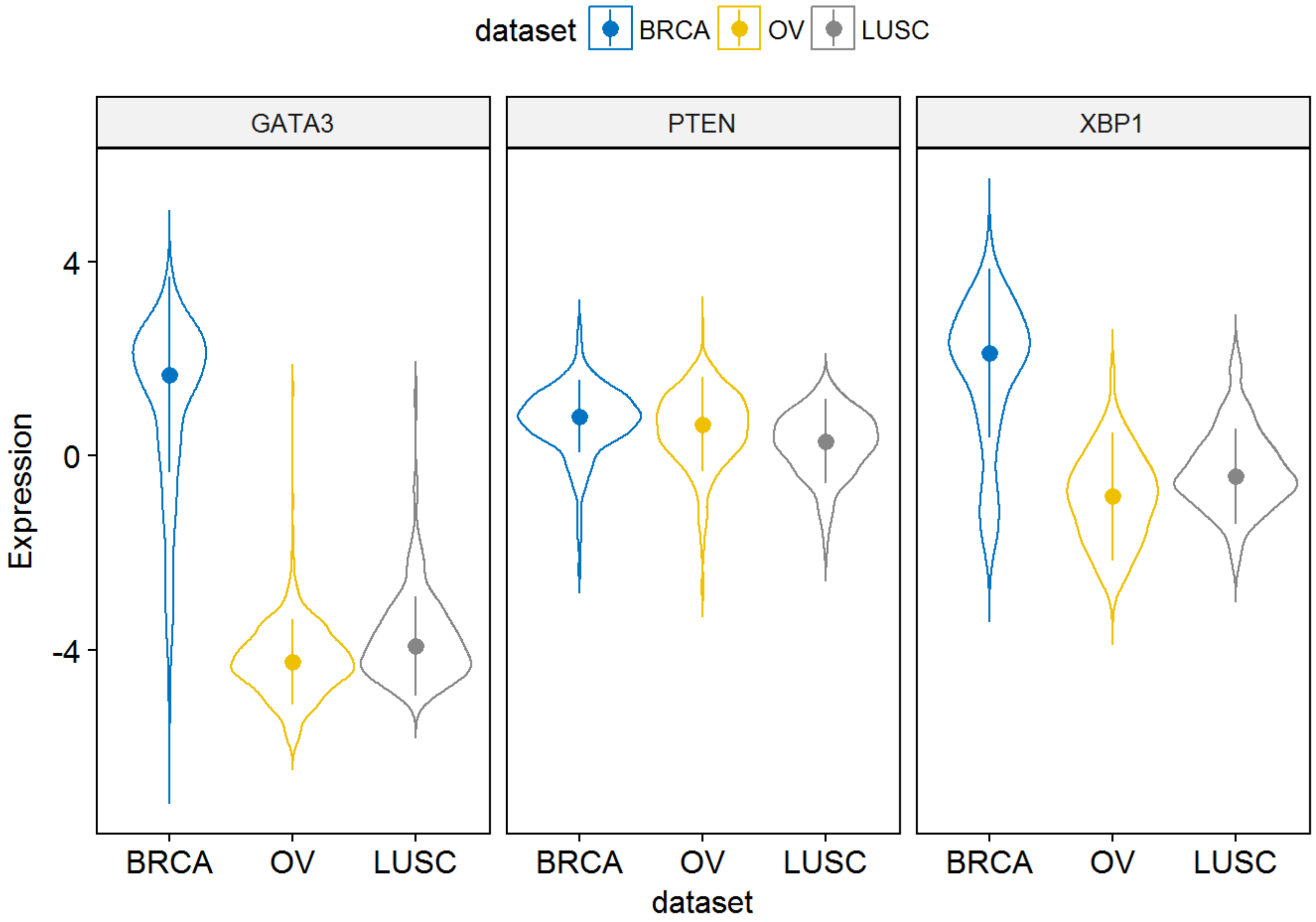
接下来的R代码将作出小提琴图并且其内部是箱线图:

```
ggviolin(expr, x = "dataset",
          y = c("GATA3", "PTEN", "XBP1"),
          combine = TRUE,
          color = "dataset", palette = "jco",
          ylab = "Expression",
          add = "boxplot")
```



其内部不仅可以添加箱线图，也可以添加中位数+四分点范围（median + interquantile range）：

```
ggviolin(expr, x = "dataset",
          y = c("GATA3", "PTEN", "XBP1"),
          combine = TRUE,
          color = "dataset", palette = "jco",
          ylab = "Expression",
          add = "median_iqr")
```

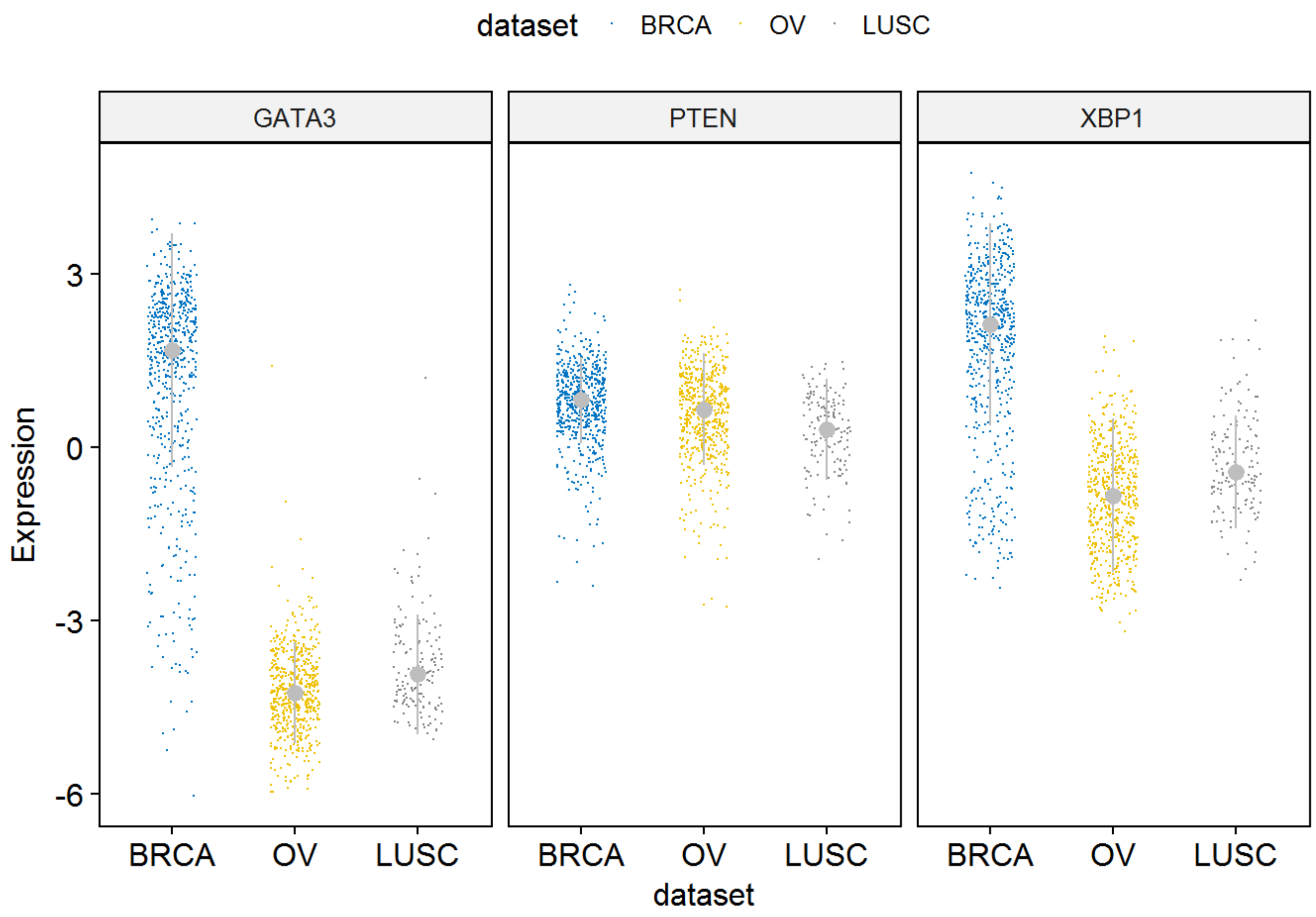


当使用函数 ggviolin() 时，add 参数合适的值可以是：“mean”, “mean_se”, “mean_sd”, “mean_ci”, “mean_range”, “median”, “median_iqr”, “median_mad”, “median_range”.

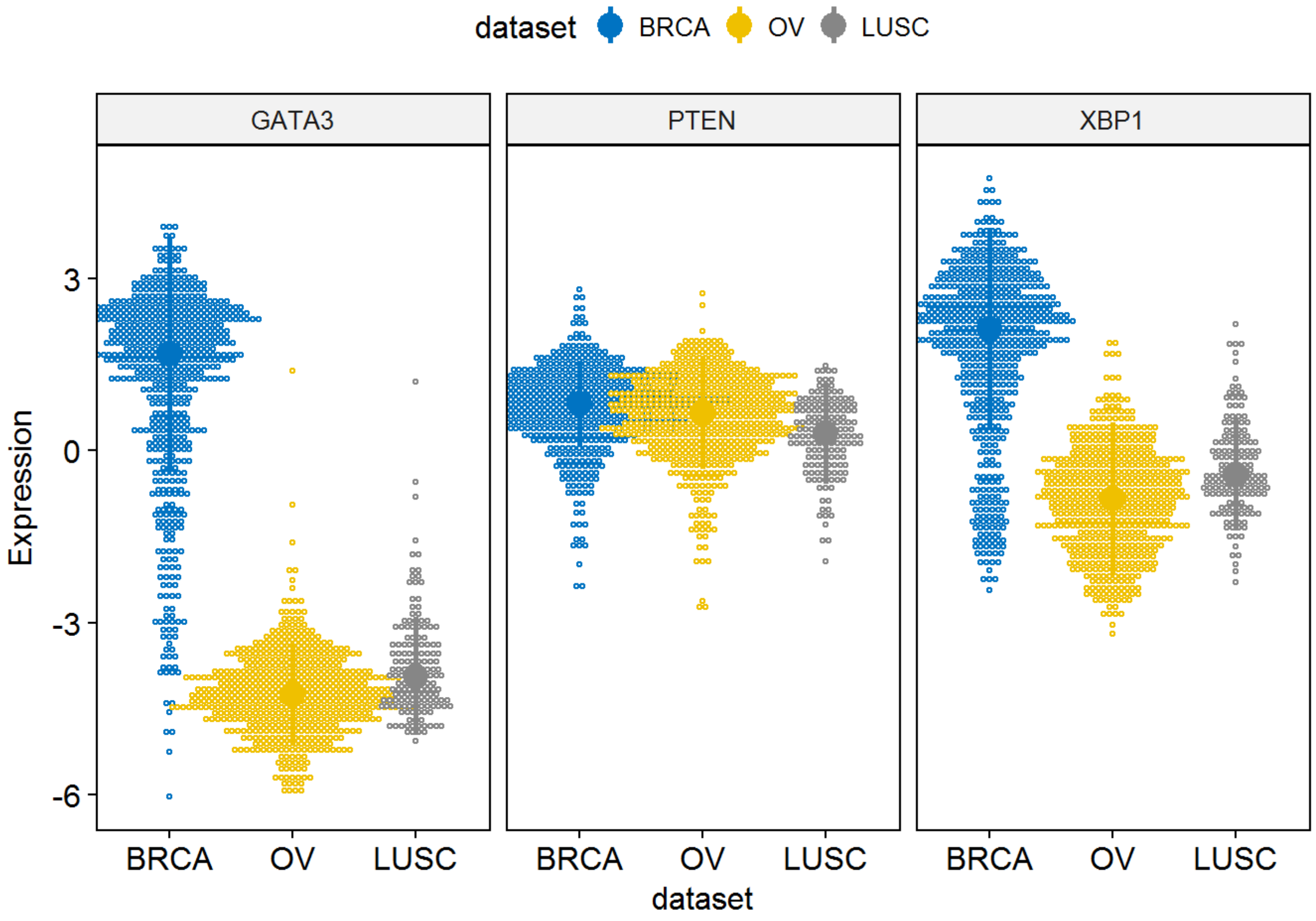
也可以添加 “jitter” 点 与 “dotplot” 到小提琴图的内部。

4. 带状图与点图（Stripcharts and dot plots）

```
ggstripchart(expr, x = "dataset",
             y = c("GATA3", "PTEN", "XBP1"),
             combine = TRUE,
             color = "dataset", palette = "jco",
             size = 0.1, jitter = 0.2,
             ylab = "Expression",
             add = "median_iqr",
             add.params = list(color = "gray"))
```



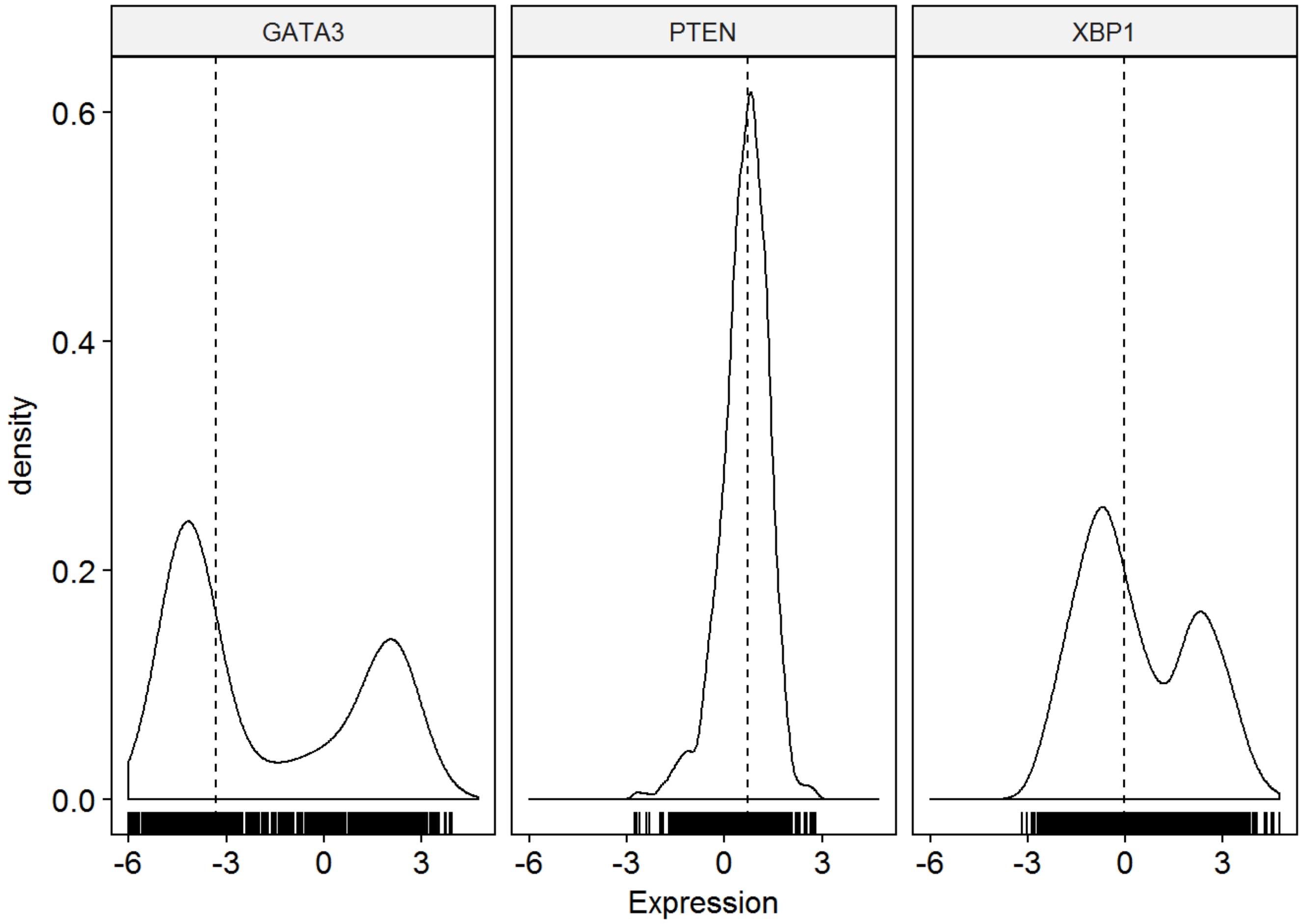
```
ggdotplot(expr, x = "dataset",
          y = c("GATA3", "PTEN", "XBP1"),
          combine = TRUE,
          color = "dataset", palette = "jco",
          fill = "white",
          binwidth = 0.1,
          ylab = "Expression",
          add = "median_iqr",
          add.params = list(size = 0.9))
```



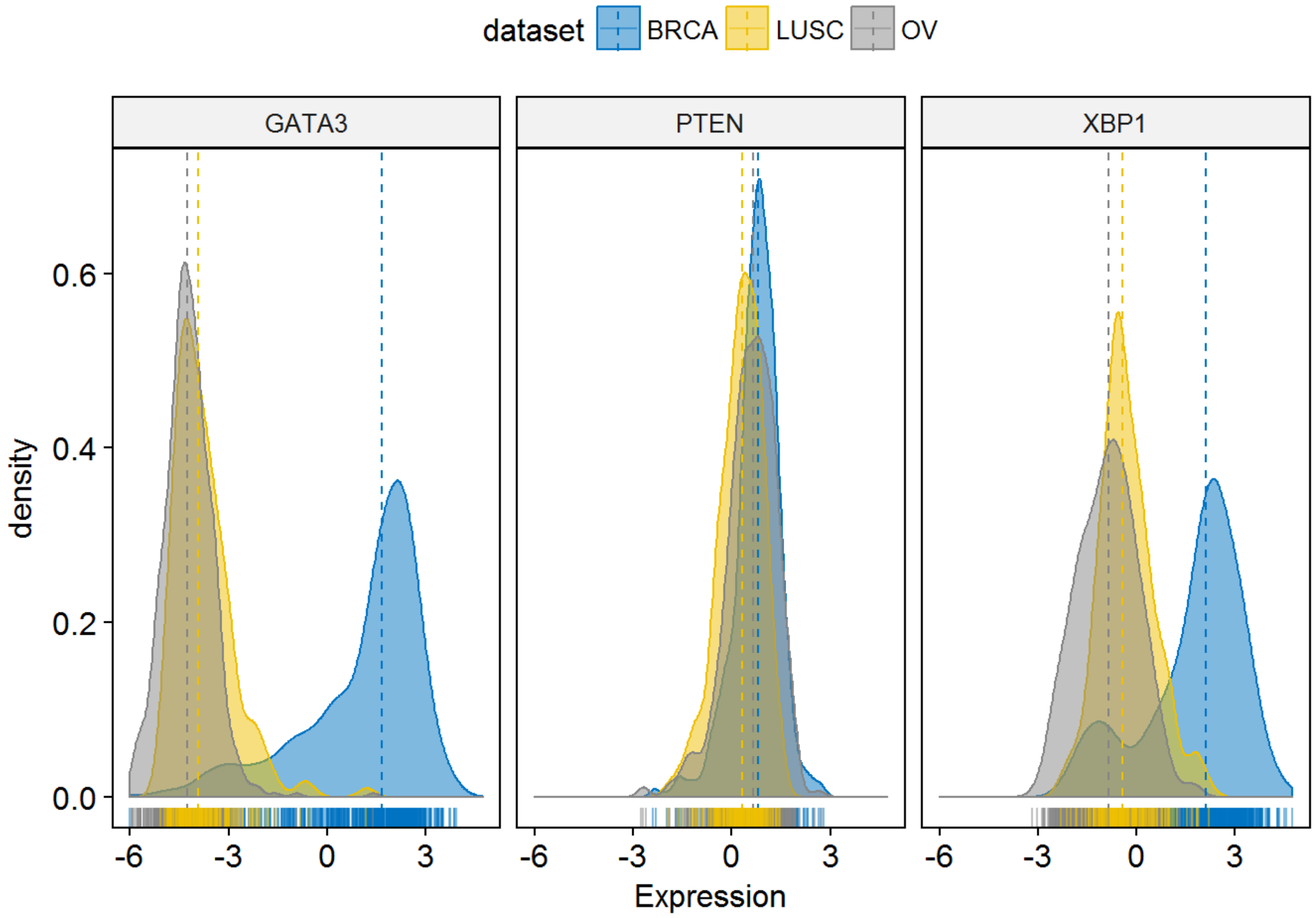
5. 密度图

用密度图显示数据分布可以使用 ggdensity() 函数。

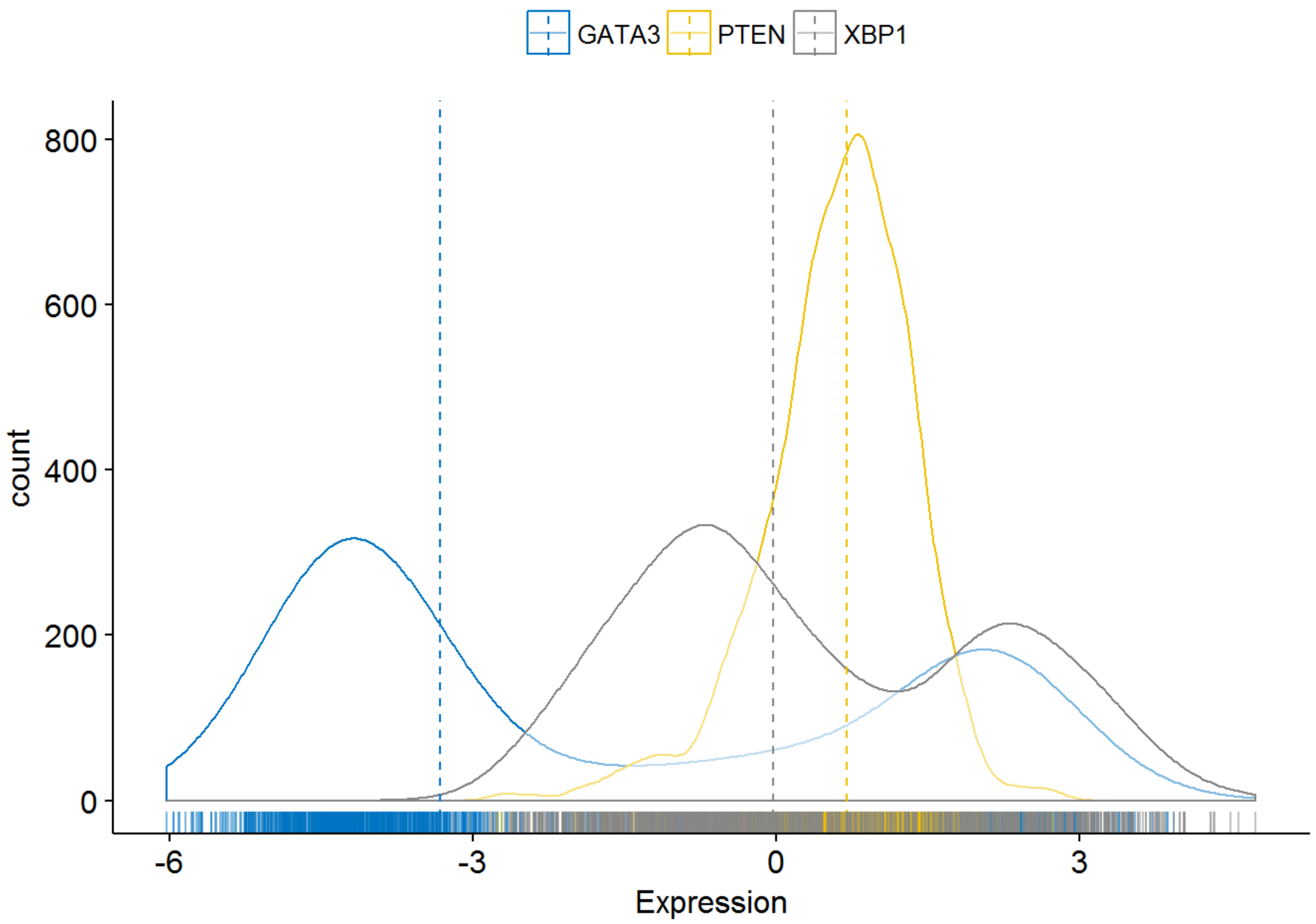
```
ggdensity(expr,
       x = c("GATA3", "PTEN",  "XBP1"),
       y = "..density..",
       combine = TRUE,                  # Combine the 3 plots
       xlab = "Expression",
       add = "median",                  # Add median line.
       rug = TRUE                       # Add marginal rug
)
```



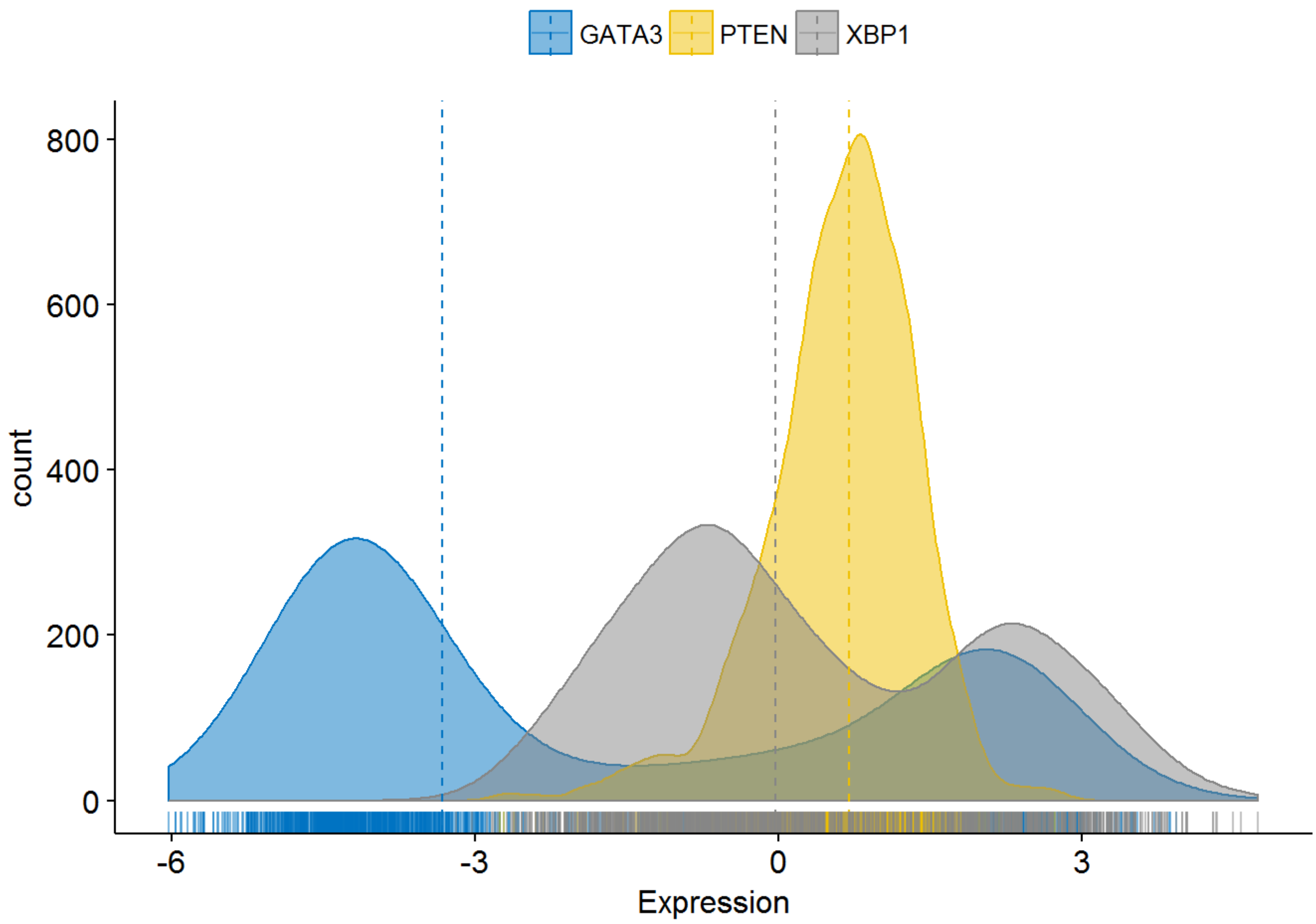
```
# Change color and fill by dataset
ggdensity(expr,
       x = c("GATA3", "PTEN",  "XBP1"),
       y = "..density..",
       combine = TRUE,                  # Combine the 3 plots
       xlab = "Expression",
       add = "median",                  # Add median line.
       rug = TRUE,                      # Add marginal rug
       color = "dataset",
       fill = "dataset",
       palette = "jco"
)
```



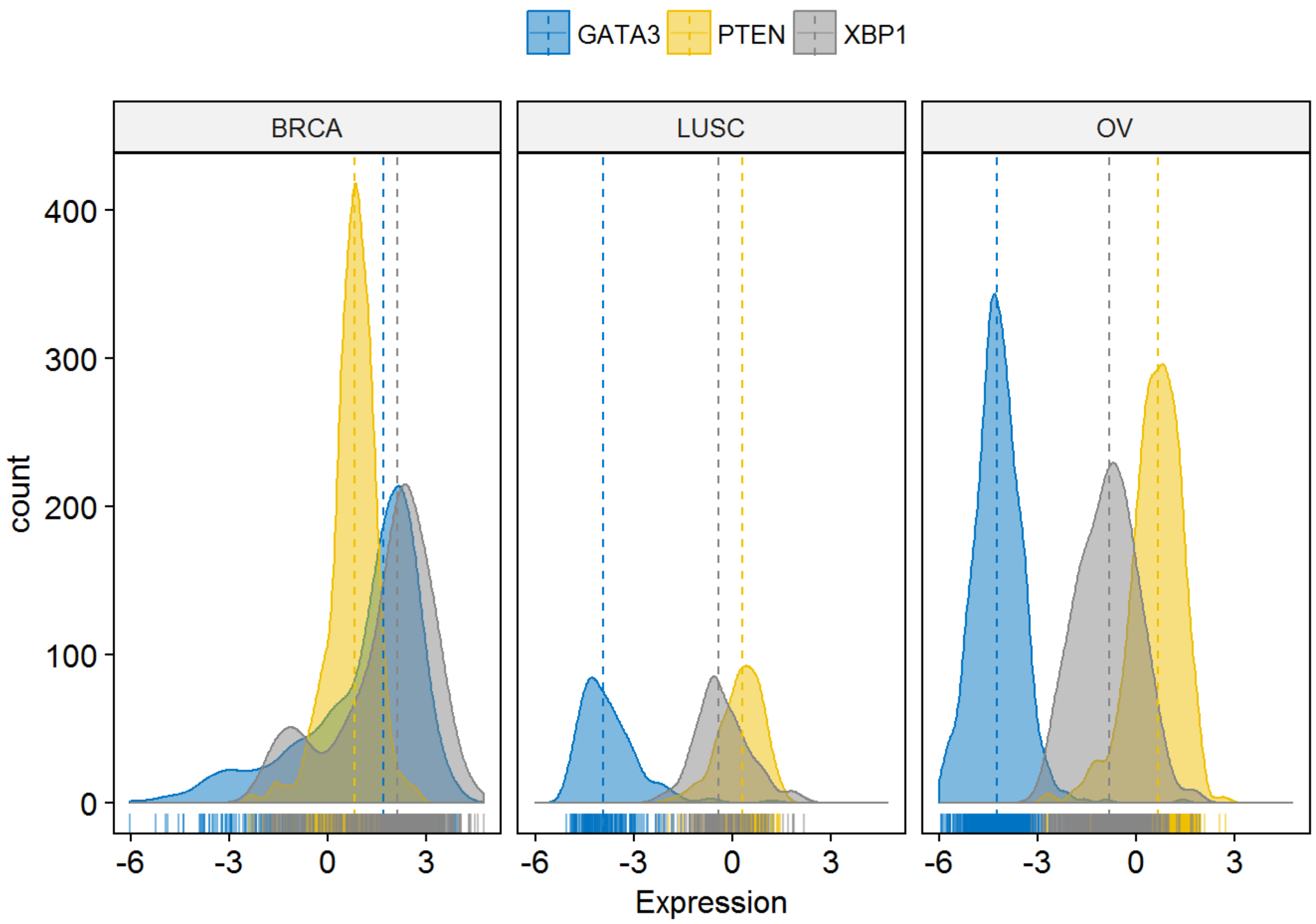
```
# Merge the 3 plots
# and use y = "..count.." instead of "..density.."
ggdensity(expr,
       x = c("GATA3", "PTEN",  "XBP1"),
       y = "..count..",
       merge = TRUE,                    # Merge the 3 plots
       xlab = "Expression",
       add = "median",                  # Add median line.
       rug = TRUE ,                     # Add marginal rug
       palette = "jco"                  # Change color palette
)
```



```
# color and fill by x variables
ggdensity(expr,
       x = c("GATA3", "PTEN",  "XBP1"),
       y = "..count..",
       color = ".x.", fill = ".x.",    # color and fill by x variables
       merge = TRUE,                   # Merge the 3 plots
       xlab = "Expression",
       add = "median",                 # Add median line.
       rug = TRUE ,                    # Add marginal rug
       palette = "jco"                 # Change color palette
)
```

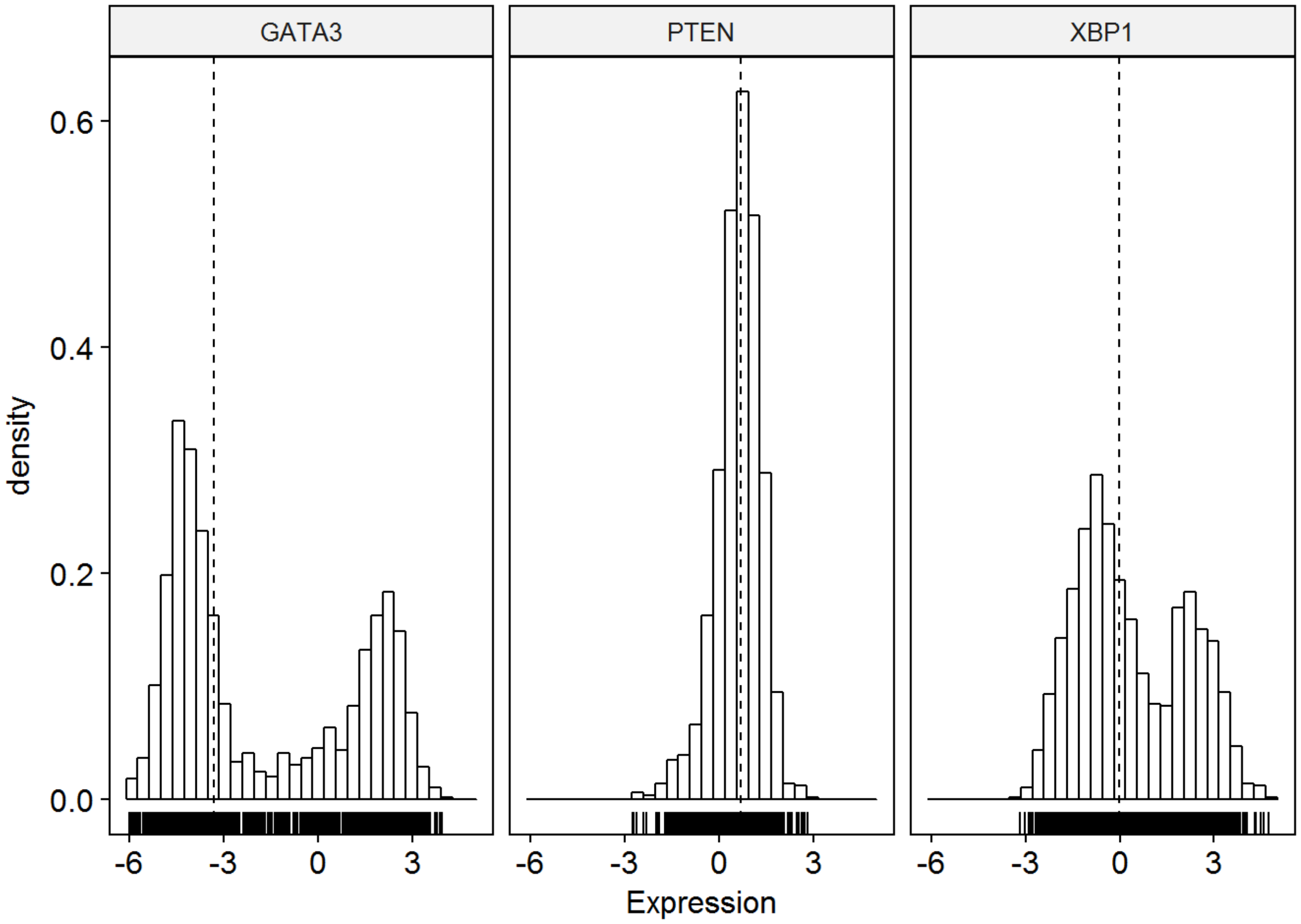


```
# Facet by "dataset"
ggdensity(expr,
       x = c("GATA3", "PTEN",  "XBP1"),
       y = "..count..",
       color = ".x.", fill = ".x.",
       facet.by = "dataset",            # Split by "dataset" into multi-panel
       merge = TRUE,                    # Merge the 3 plots
       xlab = "Expression",
       add = "median",                  # Add median line.
       rug = TRUE ,                     # Add marginal rug
       palette = "jco"                  # Change color palette
)
```

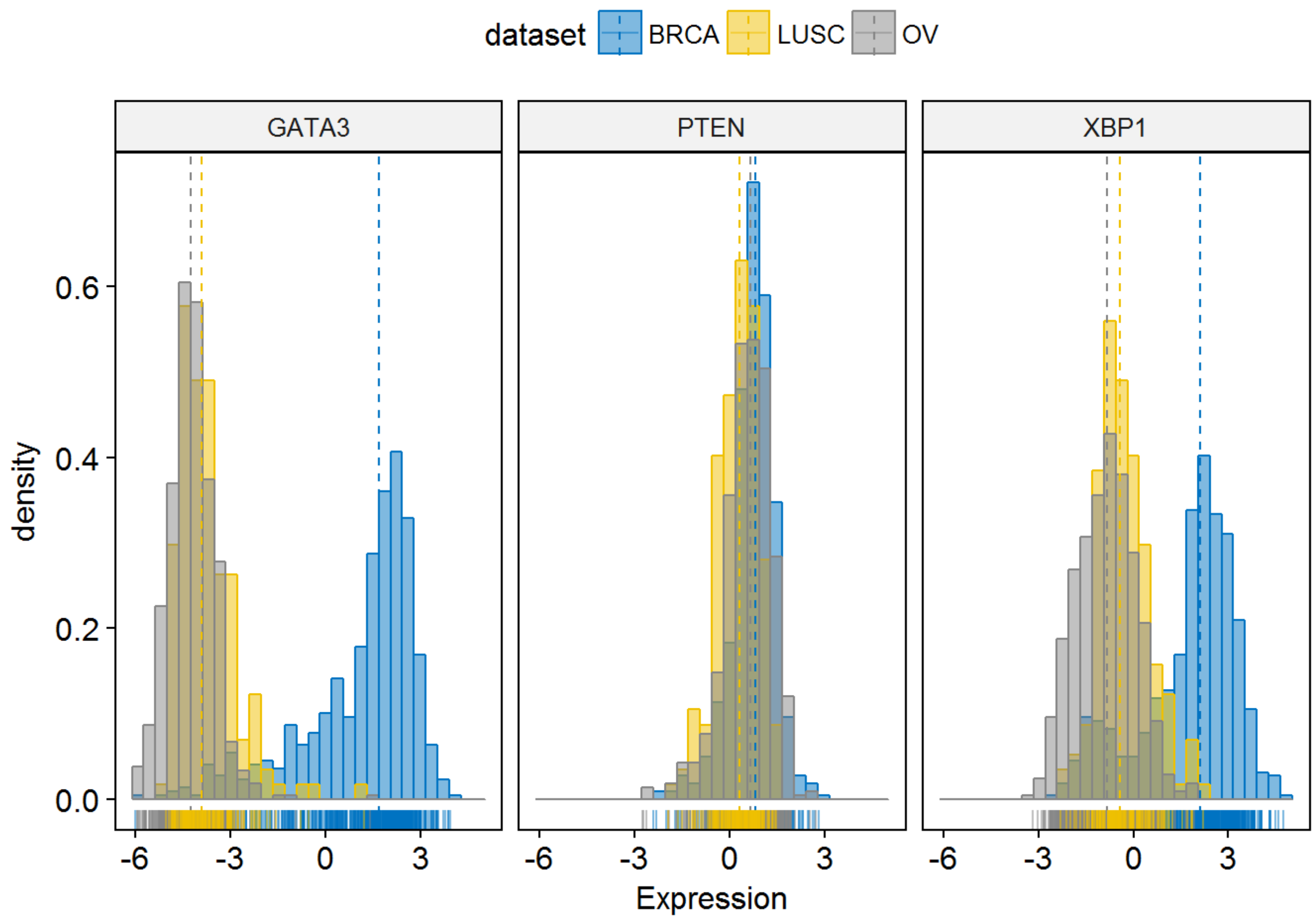


6. 直方图

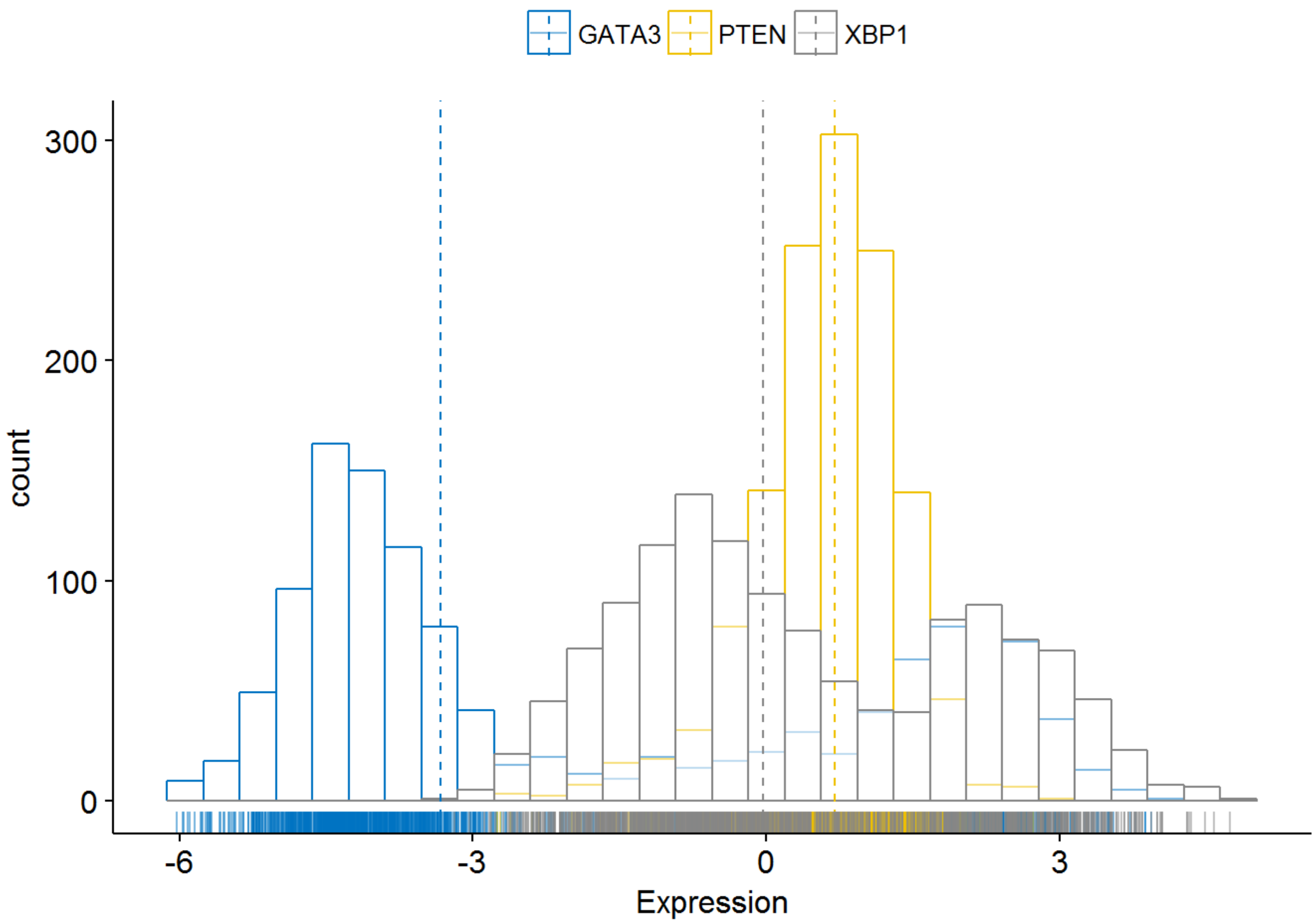
```
# Basic histogram plot
gghistogram(expr,
       x = c("GATA3", "PTEN",  "XBP1"),
       y = "..density..",
       combine = TRUE,                  # Combine the 3 plots
       xlab = "Expression",
       add = "median",                  # Add median line.
       rug = TRUE                       # Add marginal rug
)
```



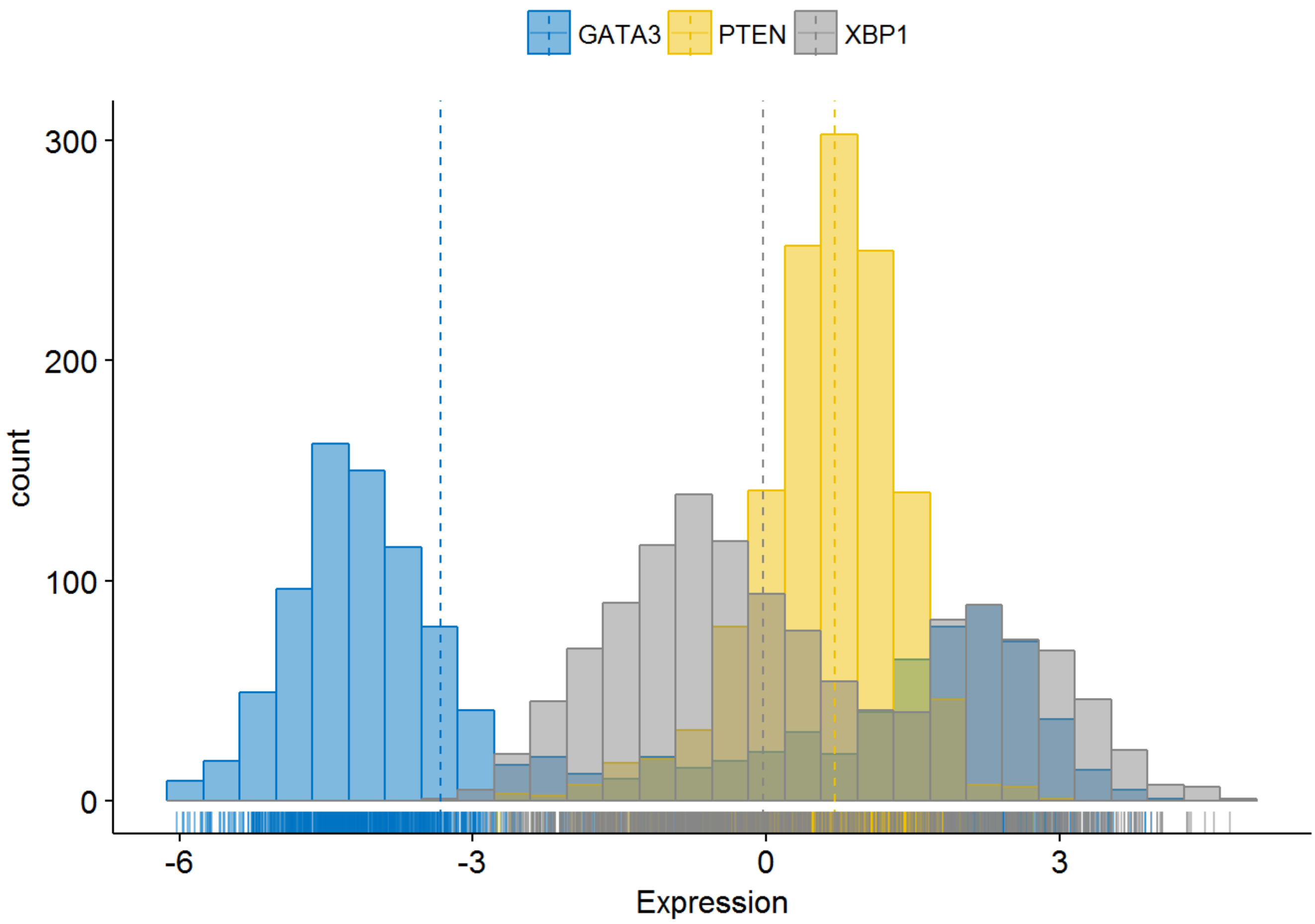
```
# Change color and fill by dataset
gghistogram(expr,
       x = c("GATA3", "PTEN",  "XBP1"),
       y = "..density..",
       combine = TRUE,                  # Combine the 3 plots
       xlab = "Expression",
       add = "median",                  # Add median line.
       rug = TRUE,                      # Add marginal rug
       color = "dataset",
       fill = "dataset",
       palette = "jco"
)
```



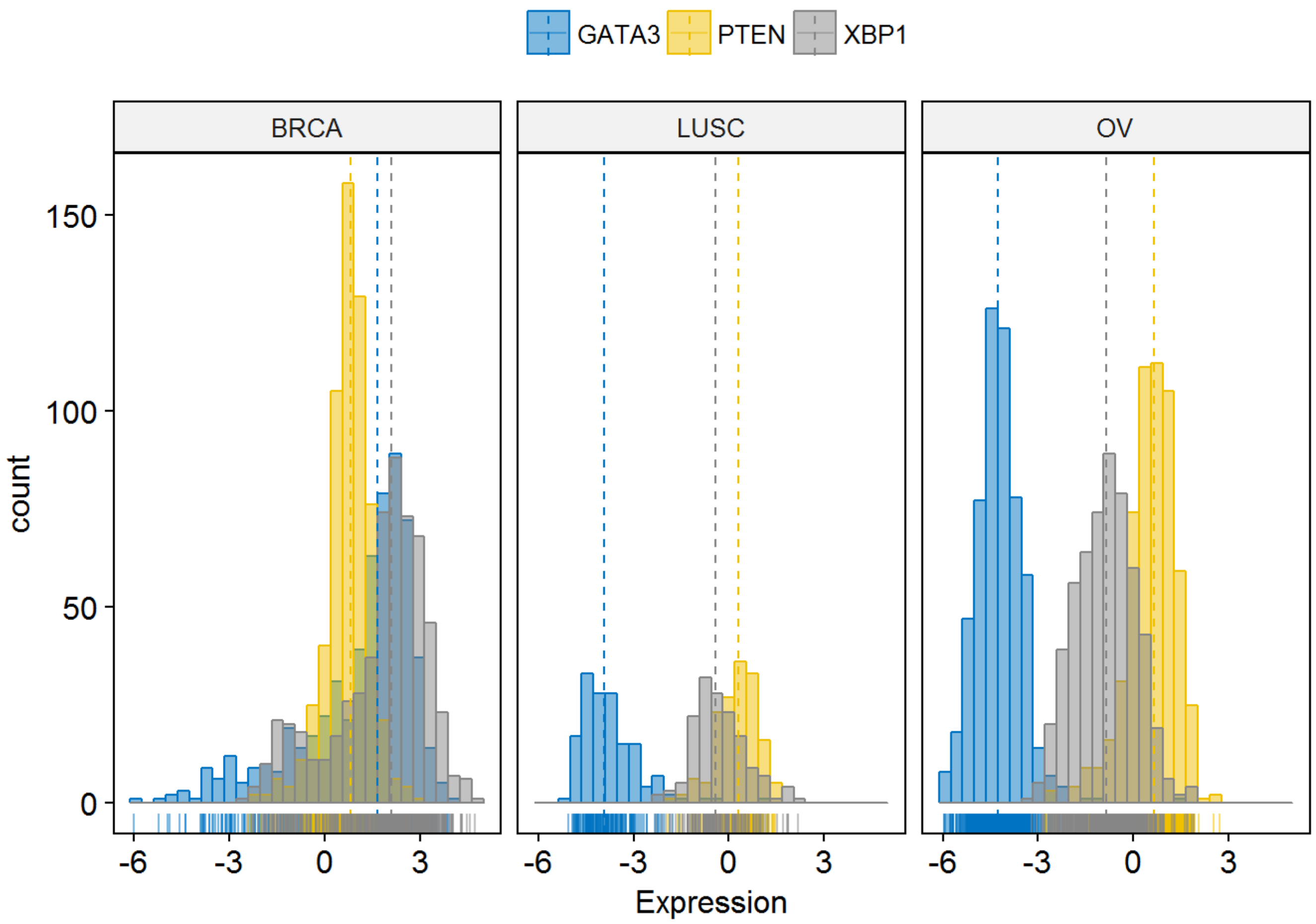
```
# Merge the 3 plots
# and use y = ″..count..″ instead of ″..density..″
gghistogram(expr,
       x = c(″GATA3″, ″PTEN″,  ″XBP1″),
       y = ″..count..″,
       merge = TRUE,                    # Merge the 3 plots
       xlab = ″Expression″,
       add = ″median″,                  # Add median line.
       rug = TRUE ,                     # Add marginal rug
       palette = ″jco″                  # Change color palette
)
```



```
# color and fill by x variables
gghistogram(expr,
       x = c(″GATA3″, ″PTEN″,  ″XBP1″),
       y = ″..count..″,
       color = ″.x.″, fill = ″.x.″,     # color and fill by x variables
       merge = TRUE,                    # Merge the 3 plots
       xlab = ″Expression″,
       add = ″median″,                  # Add median line.
       rug = TRUE ,                     # Add marginal rug
       palette = ″jco″                  # Change color palette
)
```

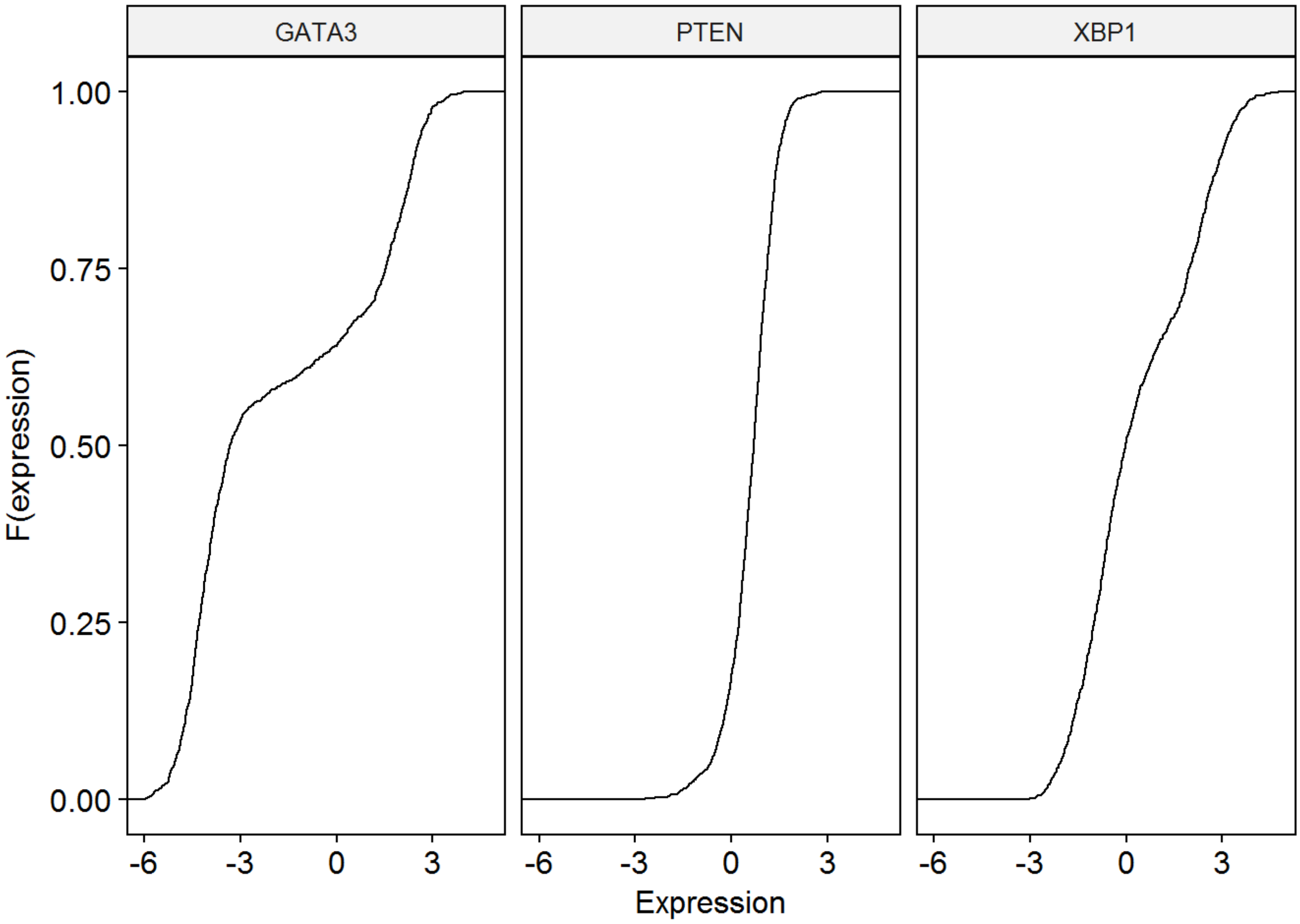


```
# Facet by ″dataset″
gghistogram(expr,
        x = c(″GATA3″, ″PTEN″,  ″XBP1″),
        y = ″..count..″,
        color = ″.x.″, fill = ″.x.″,
        facet.by = ″dataset″,           # Split by ″dataset″ into multi-panel
        merge = TRUE,                   # Merge the 3 plots
        xlab = ″Expression″,
        add = ″median″,                 # Add median line.
        rug = TRUE ,                    # Add marginal rug
        palette = ″jco″                 # Change color palette
)
```

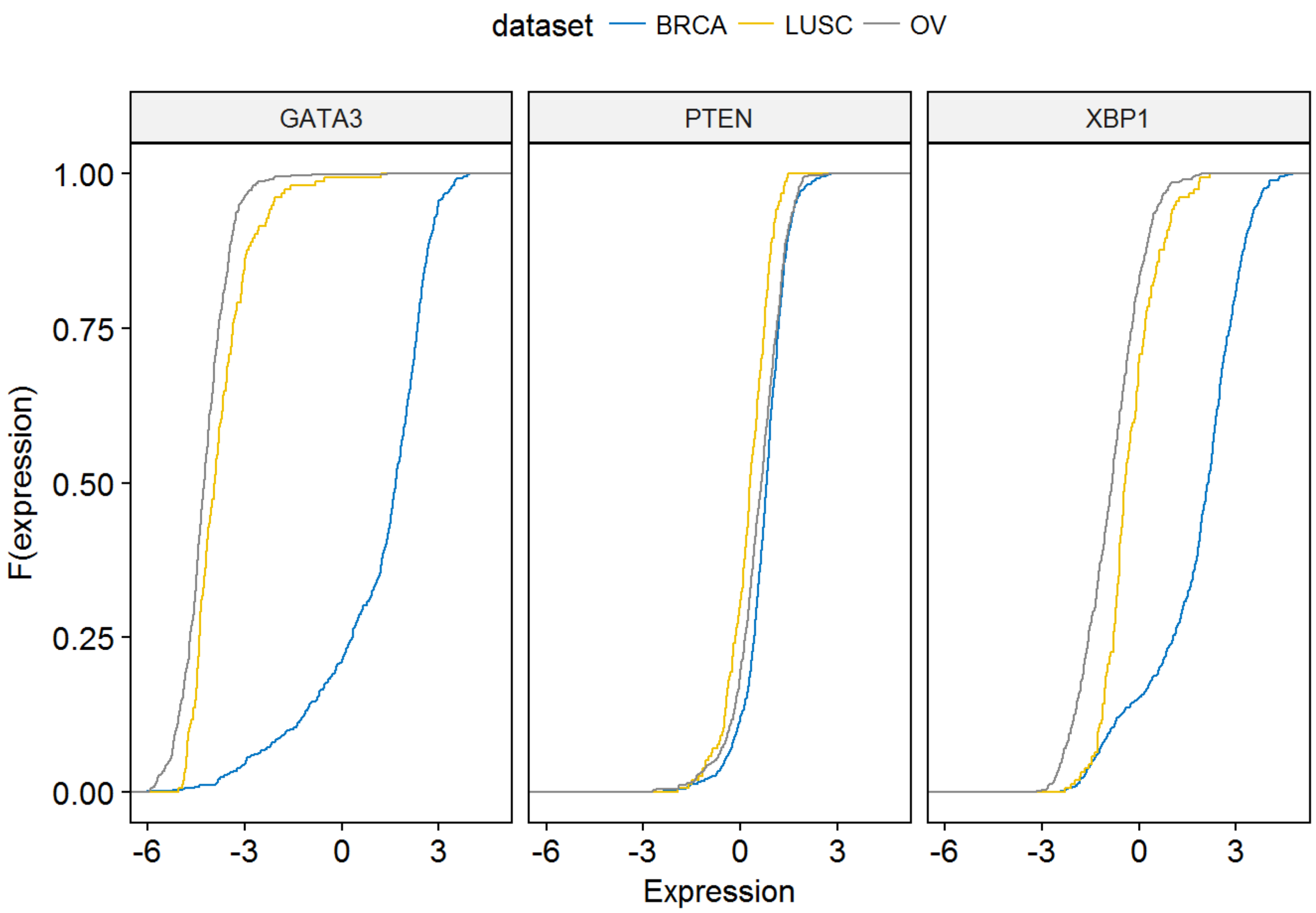


7.经验累积密度函数（Empirical cumulative density function）

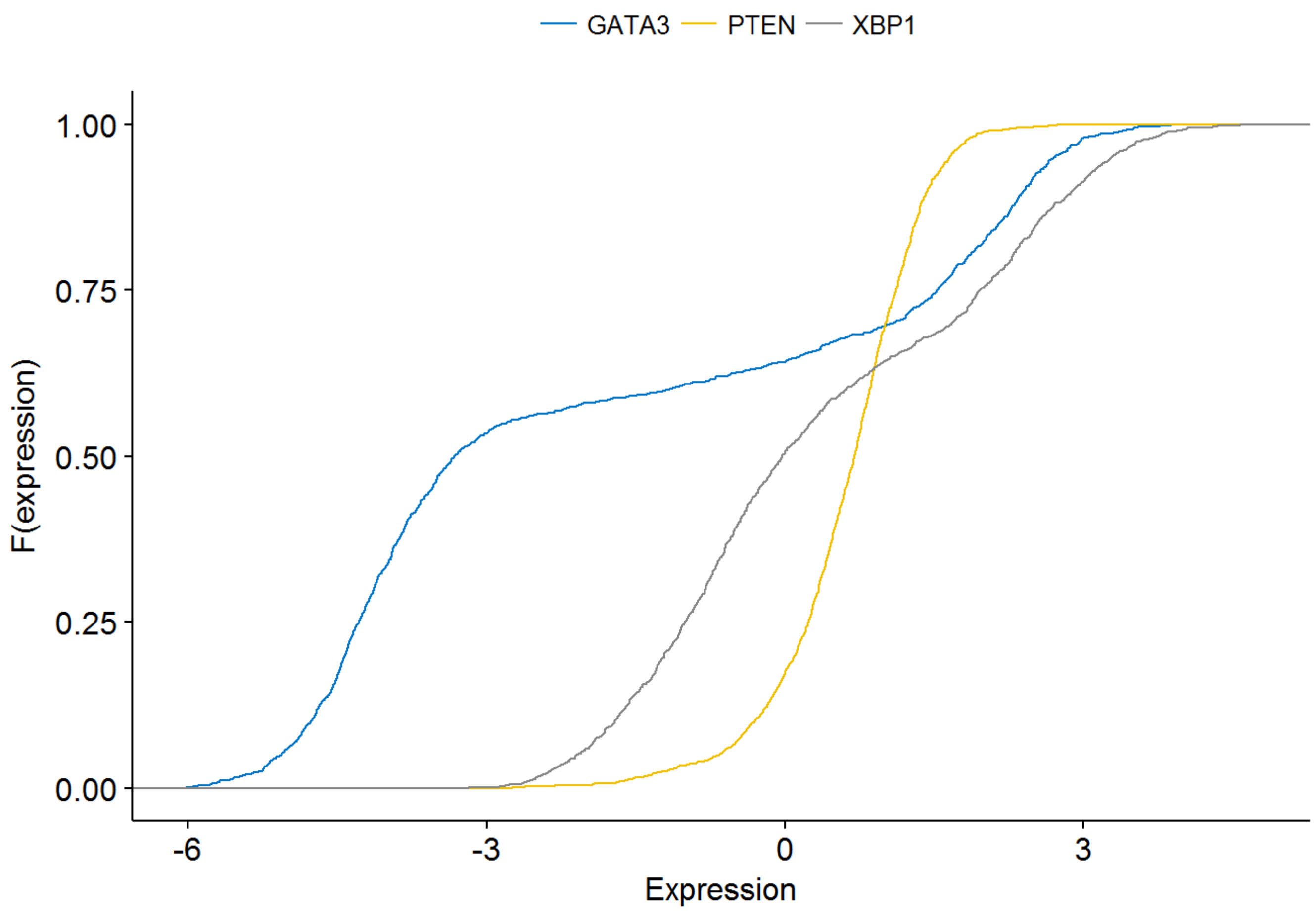
```
# Basic ECDF plot
ggecdf(expr,
       x = c("GATA3", "PTEN",  "XBP1"),
       combine = TRUE,
       xlab = "Expression", ylab = "F(expression)"
)
```



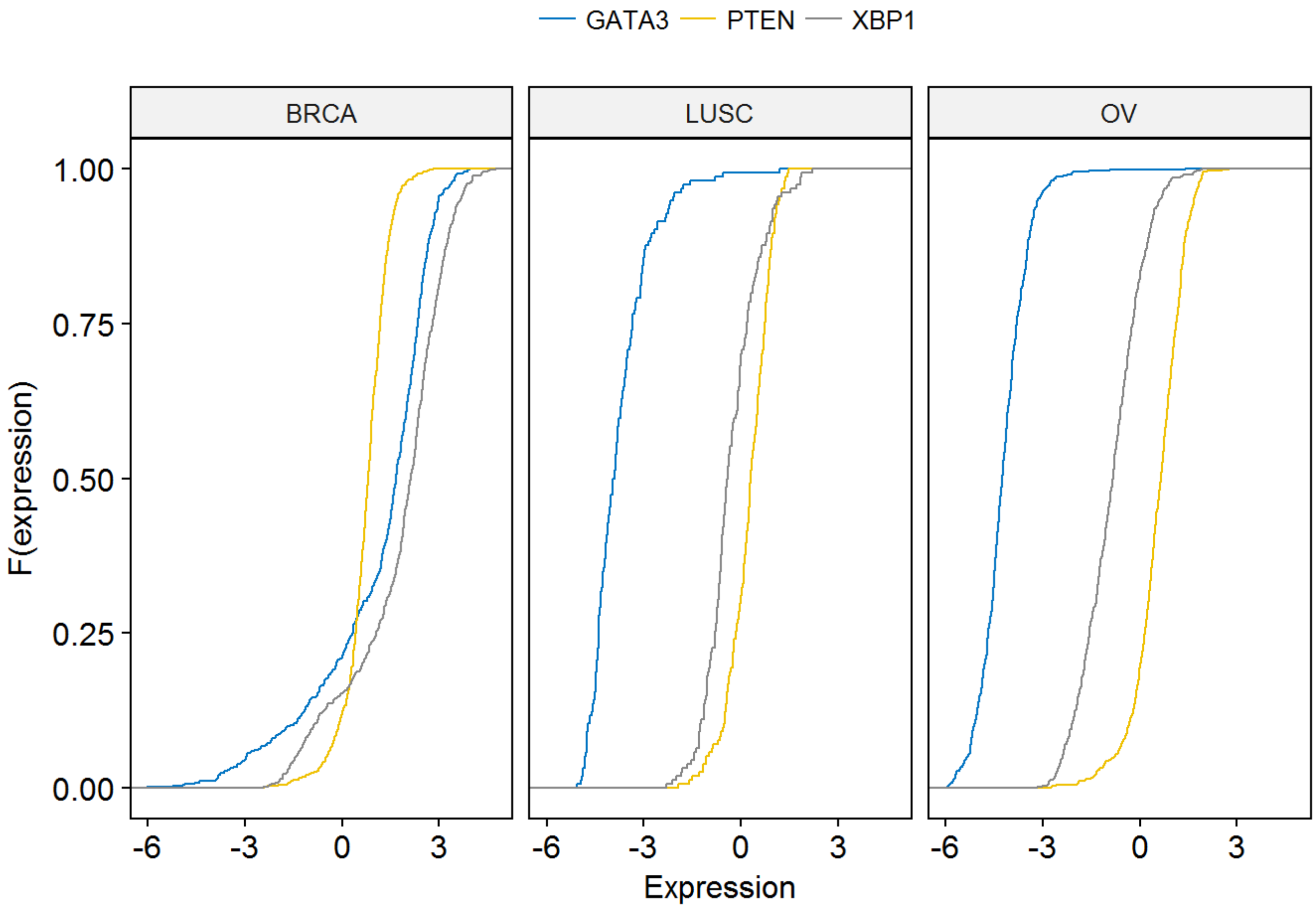
```
# Change color  by dataset
ggecdf(expr,
       x = c("GATA3", "PTEN",  "XBP1"),
       combine = TRUE,
       xlab = "Expression", ylab = "F(expression)",
       color = "dataset", palette = "jco"
)
```



```
# Merge the 3 plots and color by x variables
ggecdf(expr,
       x = c(″GATA3″, ″PTEN″,  ″XBP1″),
       merge = TRUE,
       xlab = ″Expression″, ylab = ″F(expression)″,
       color = ″.x.″, palette = ″jco″
)
```

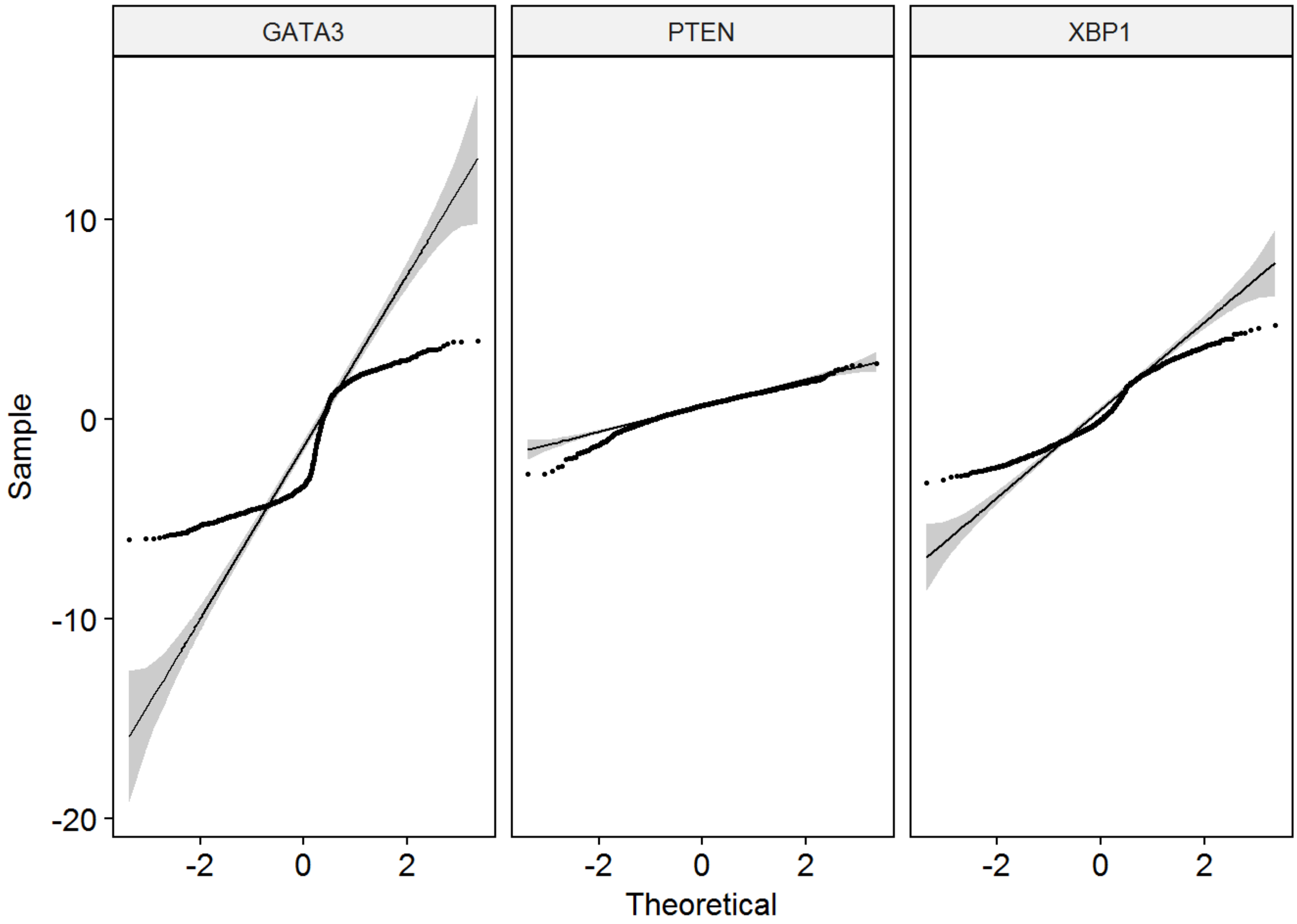


```
# Merge the 3 plots and color by x variables
# facet by "dataset" into multi-panel
ggecdf(expr,
       x = c("GATA3", "PTEN",  "XBP1"),
       merge = TRUE,
       xlab = "Expression", ylab = "F(expression)",
       color = ".x.", palette = "jco",
       facet.by = "dataset"
)
```

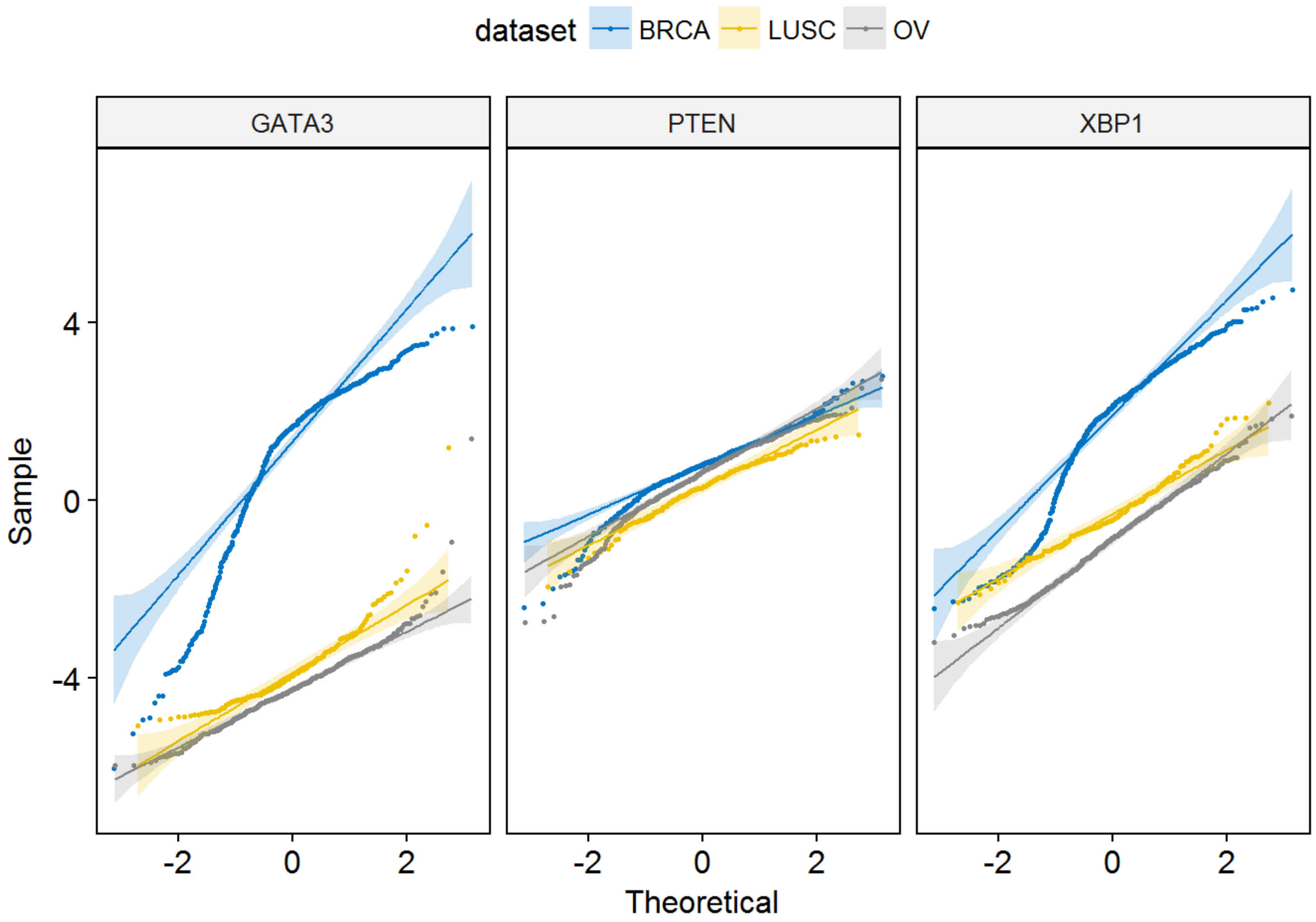


8. QQ图（Quantile - Quantile plot）

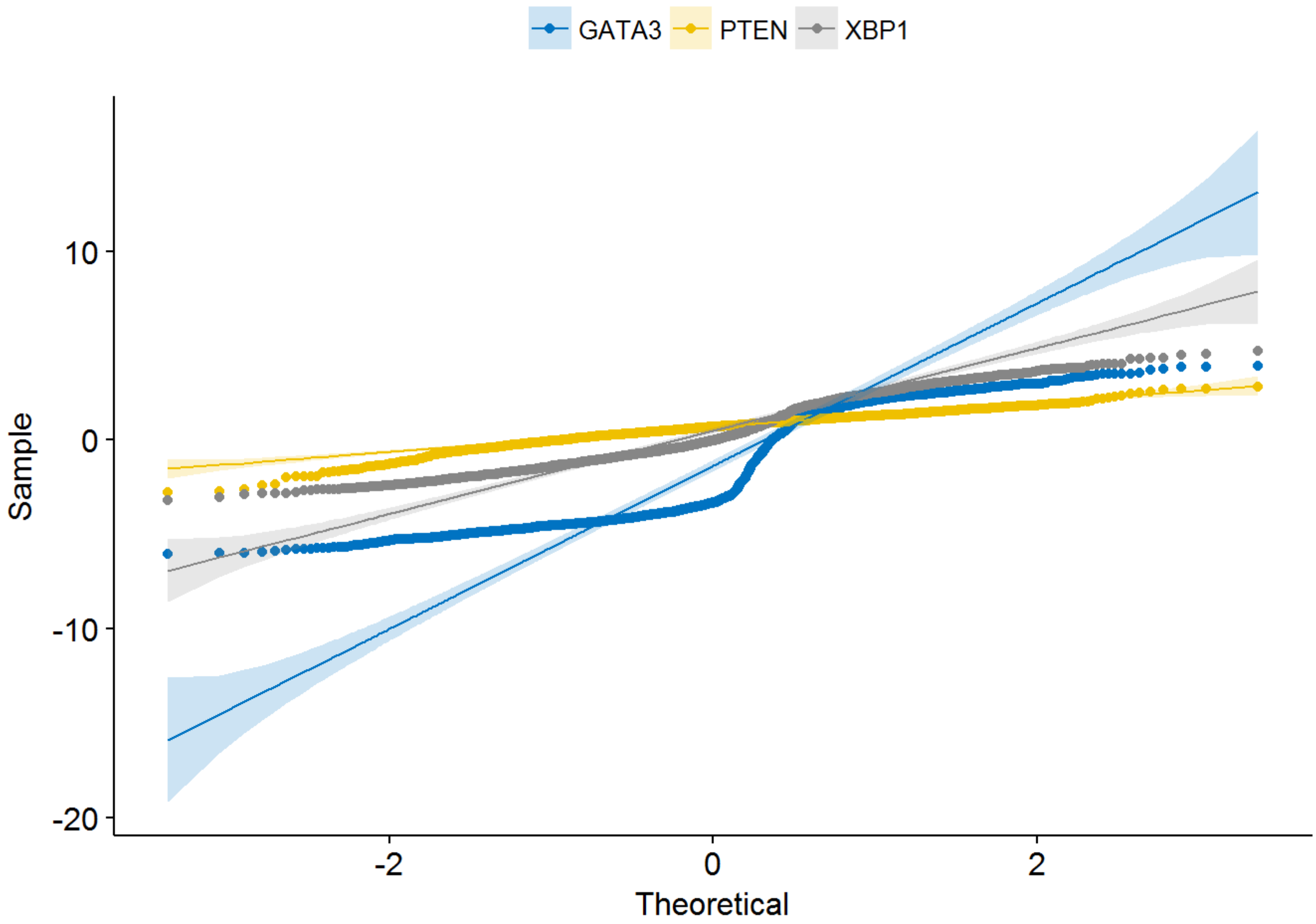
```
# Basic ECDF plot
ggqqplot(expr,
       x = c(″GATA3″, ″PTEN″,  ″XBP1″),
       combine = TRUE, size = 0.5
)
```



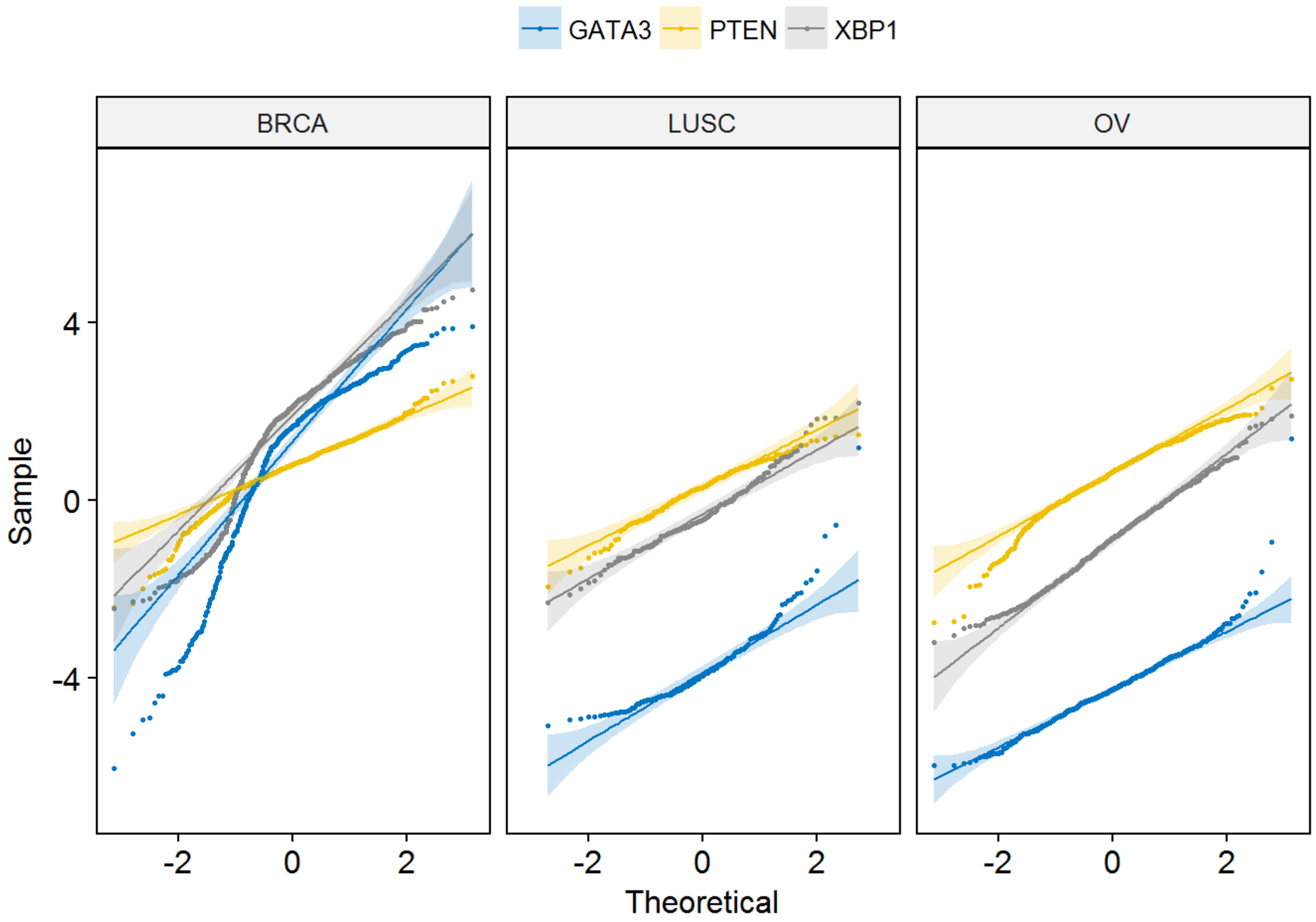
```
# Change color  by dataset
ggqqplot(expr,
       x = c("GATA3", "PTEN",  "XBP1"),
       combine = TRUE, color = "dataset", palette = "jco",
       size = 0.5
)
```



```
# Merge the 3 plots and color by x variables
ggqqplot(expr,
       x = c("GATA3", "PTEN",  "XBP1"),
       merge = TRUE,
       color = ".x.", palette = "jco"
)
```



```
# Merge the 3 plots and color by x variables
# facet by "dataset" into multi-panel
ggqqplot(expr,
       x = c("GATA3", "PTEN",  "XBP1"),
       merge = TRUE, size = 0.5,
       color = ".x.", palette = "jco",
       facet.by = "dataset"
)
```



参考：Facilitating Exploratory Data Visualization: Application to TCGA Genomic Data (http://www.sthda.com/english/articles/24-ggpubr-publication-ready-plots/77-facilitating-exploratory-data-visualization-application-to-tcga-genomic-data/)