

LLA_MO_GRP78_.R

Guadalupe Rivera Torruco

2021-07-16

GRP78 expresion in childhood B-LLA

by Guadalupe Rivera Torruco

This analysis correspond to multiparametric cytometry bone marrow samples of childhood LLA de novo diagnosed. The stained samples were runned in Cytoflex LX cytometer and pre-analysed in FlowJo v10.2. The cells analysed correspond to compensated Live CD45 cells.

```
options(warn=-1)
```

```
packages <- c("knitr",
             "cowplot",
             "ggplot2",
             "readxl",
             "lattice",
             "uwot",
             "rstudioapi",
             "BiocManager",
             "knitr")

for (i in packages){
  if(!is.element(i, .packages(all.available = TRUE))){
    install.packages(i)
  }
  library(i, character.only = TRUE)
}

## Loading required package: Matrix

## Bioconductor version '3.12' is out-of-date; the current release version '3.15'
##   is available with R version '4.2'; see https://bioconductor.org/install
on.bioc <- TRUE

# Use fig.width = 7 for html and fig.width = 6 for pdf
fig.width <- ifelse(on.bioc, 8, 6)
knitr:::opts_chunk$set(cache = 2, cache.path = "cache/",
                       warning = FALSE, message = FALSE, error = FALSE,
                       fig.path = "figure/", fig.width = fig.width)
options(width=75)

packages_bioconductor <- c("CATALYST",
                           "diffcyt",
                           "flowCore",
                           "scater",
                           "flowViz",
```

```

    "BiocStyle")

if (!requireNamespace("BiocManager", quietly = TRUE))
  install.packages("BiocManager")
BiocManager::install(packages_bioconductor, force = TRUE, ask = FALSE)

## package 'CATALYST' successfully unpacked and MD5 sums checked
## package 'diffcyt' successfully unpacked and MD5 sums checked
## package 'flowCore' successfully unpacked and MD5 sums checked
## package 'scater' successfully unpacked and MD5 sums checked
## package 'flowViz' successfully unpacked and MD5 sums checked
## package 'BiocStyle' successfully unpacked and MD5 sums checked
##
## The downloaded binary packages are in
##   C:\Users\lupit\AppData\Local\Temp\RtmpkBWG1D\downloaded_packages
##
## There are binary versions available but the source versions are
##   later:
##       binary source needs_compilation
## cli      3.2.0  3.3.0          TRUE
## dplyr    1.0.8  1.0.9          TRUE
## igraph   1.3.0  1.3.1          TRUE
## ps       1.6.0  1.7.0          TRUE

#Bind the lists of packages and multi-load them
#all_packages <- c(packages, packages_bioconductor)

multi_library <- function(packages){
  for(i in packages){
    library(i, character.only = TRUE)
  }
}

multi_library(packages_bioconductor)

path <- "C:/Users/lupit/OneDrive/Analisis14feb2021" #path to files
setwd(path)

md <- "LLA_MO_GRP78_metadata.xlsx"
md <- read_excel(md)
head(data.frame(md))

##           file_name      sample_id condition      LLA_id
## 1      LLA_MO_89_19_SR.fcs 89_19_SR Standard risk Std_risk_1
## 2 LLA_MO_MICHELLE_AGT_SR.fcs MICHELLE_AGT_SR Standard risk Std_risk_2
## 3      LLA_MO_NAOMI_DAA_SR.fcs NAOMID_AA_SR Standard risk Std_risk_3
## 4      LLA_MO_ESTEBAN_SR.fcs ESTEBAN_SR Standard risk Std_risk_4
## 5      LLA_MO_JOSE_M_SR.fcs JOSE_M_SR Standard risk Std_risk_5
## 6      LLA_MO_MELANI_HR.fcs MELANI_HR     High risk High_risk_1
##       date
## 1 2019-08-23
## 2 2019-06-22
## 3 2019-06-22
## 4 2021-01-29
## 5 2021-08-02

```

```

## 6 2019-09-13

#Read a flowset - multiple FCS files (when in the working directory)
#fs <- read.flowSet(path = 'C:/User/bla/bla/fullFCS', pattern = ".fcs") <- when outside directory

flowset_LLA <- read.flowSet(path ="./Exports2",
                             pattern = '.fcs')

#Compensation matrix
Spill <- flowset_LLA@frames[["LLA_MO_MELANI_HR.fcs"]]\$description\$`$SPILLOVER`


flowset_LLA.comp <- compensate(flowset_LLA, Spill)

```

We must create a csv or xlsx with the channel colnames in the FCS files and the antigen of our markers. i.e. channel FL1-A correspond to FITC channel, the antibody to recognize GRP78 is marked with FITC.

```

panel <- "LLA_MO_GRP78_panel_v3.xlsx"
panel <- read_excel(panel)
head(data.frame(panel))

```

```

##   fcs_colname antigen marker_class
## 1     FL1-A    GRP78      type
## 2     FL3-A    CD19      type
## 3     FL5-A    CD10      type
## 4     FL11-A   CD38      type
## 5     FL9-A    CD34      type
## 6     FL12-A   CXCR4     type

```

We need to check if every fcs_colname is present in our flowset, if the panel.xlsx is correct

```

all(panel$fcs_colname %in% colnames(flowset_LLA))

## [1] TRUE

md$condition <- factor(md$condition, levels = c("Standard risk", "High risk"))
md$sample_id <- factor(md$sample_id, levels =
                         md$sample_id[order(md$condition)])

sce <- prepData(flowset_LLA, panel, md,
                 md_cols = list(file = "file_name", id = "sample_id",
                               factors = c("condition", "LLA_id")))

```

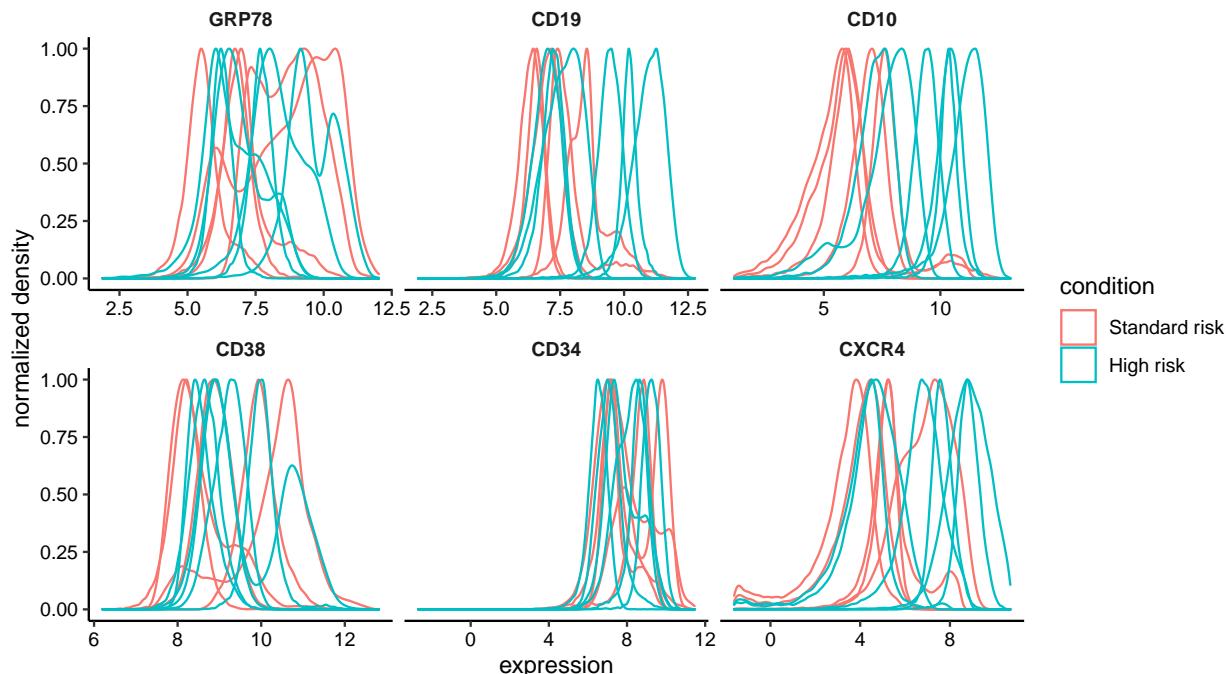
After creating sce experiment we can see some diagnostic plots. If we want to save it as pdf we can use:
`pdf(file = "desired_name.pdf")` `code_to_plot dev.off()`

Expression plot help us to see homogeneity between groups and see differences between them.

```

plotExprs(sce, color_by = "condition", assay = "exprs")

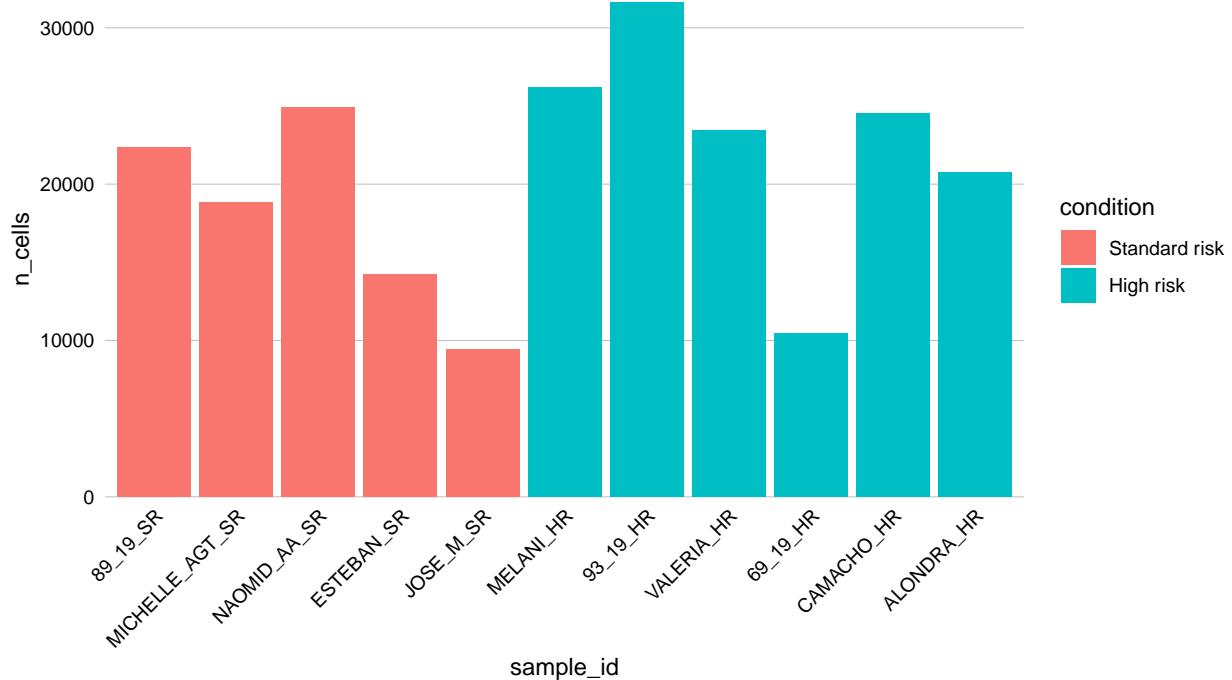
```



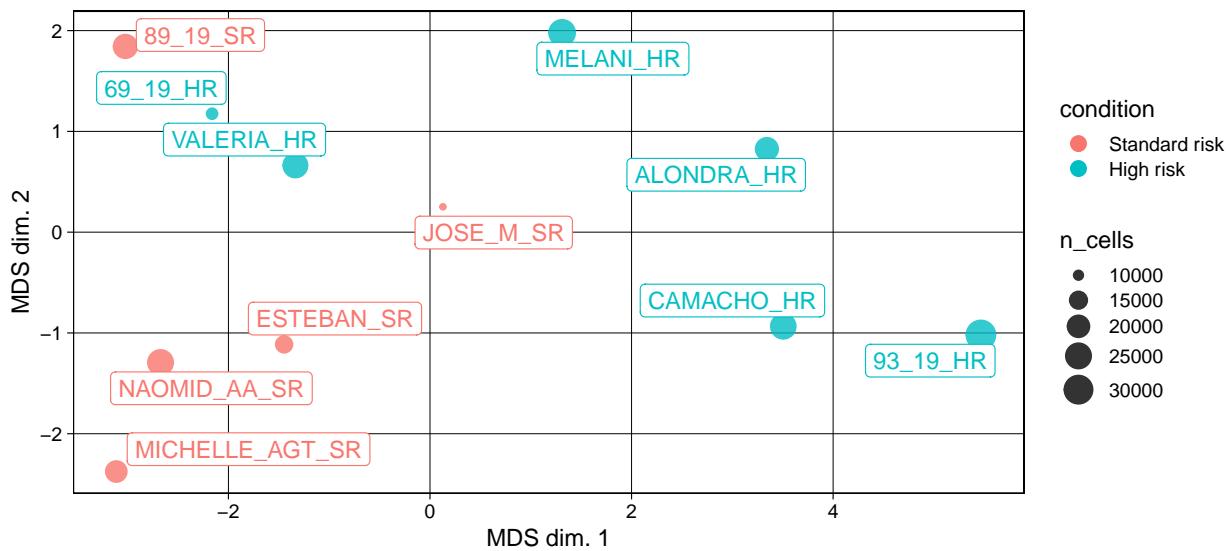
```
n_cells(sce)

##
##      89_19_SR MICHELLE_AGT_SR      NAOMID_AA_SR      ESTEBAN_SR
##      22343          18865          24895          14236
##      JOSE_M_SR      MELANI_HR      93_19_HR      VALERIA_HR
##      9452          26212          31656          23417
##      69_19_HR      CAMACHO_HR      ALONDRA_HR
##      10471         24495          20745

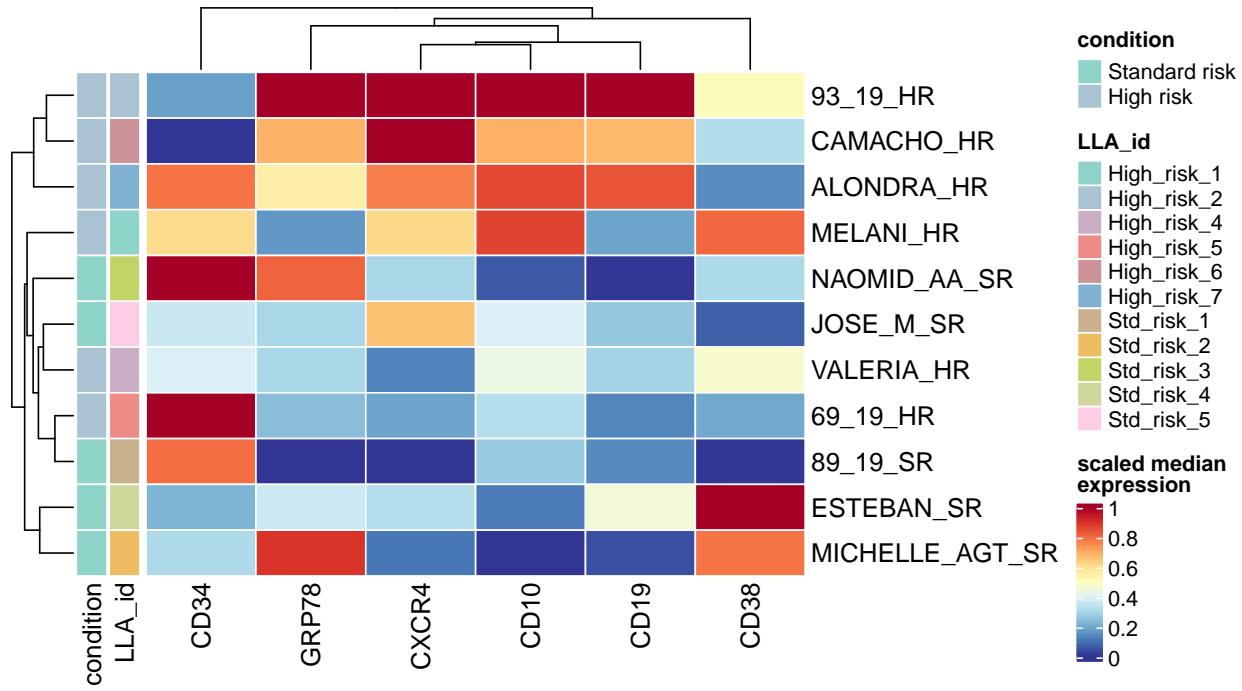
plotCounts(sce, group_by = "sample_id", color_by = "condition")
```



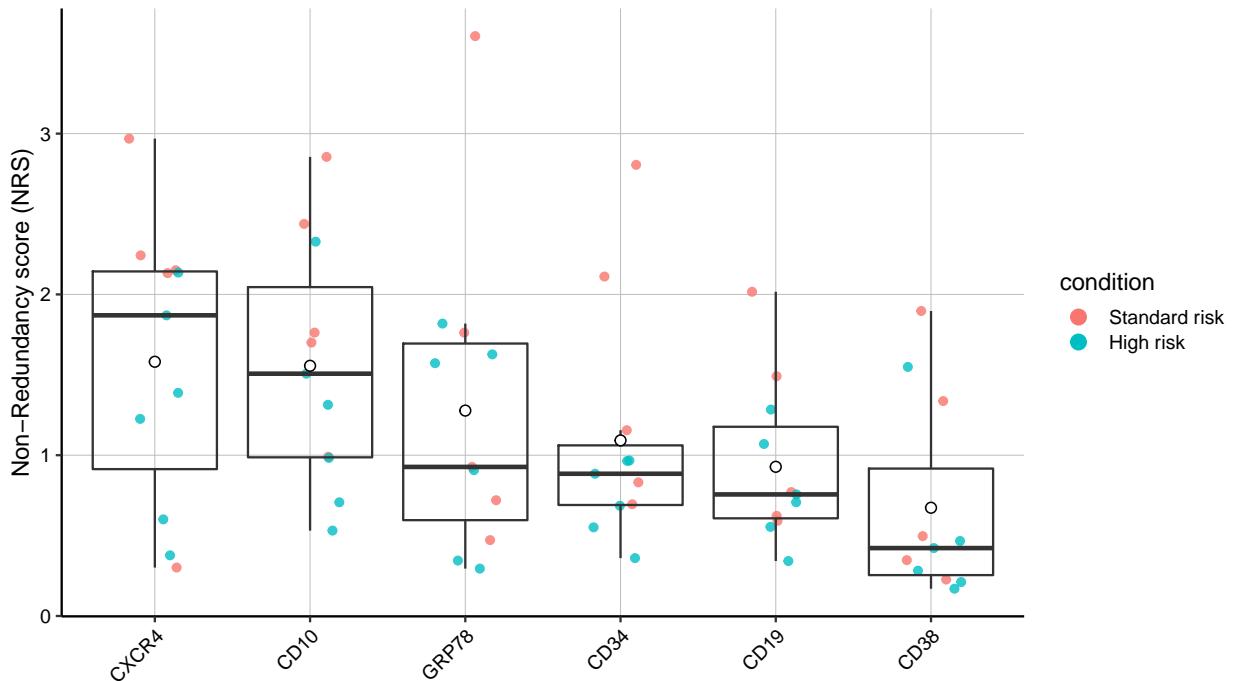
```
pbMDS(sce, color_by = "condition", label_by = "sample_id")
```



```
plotExprHeatmap(sce, scale = "last")
```



```
#pdf(file = "plotNRS.pdf")
plotNRS(sce, features = "type", color_by = "condition") #ERROR
```



```
#dev.off()
```

Following the exploratory analysis we will start clustering with FlowSOM and ConsensusClustering tools. We might need to set a seed, we can try different numbers:

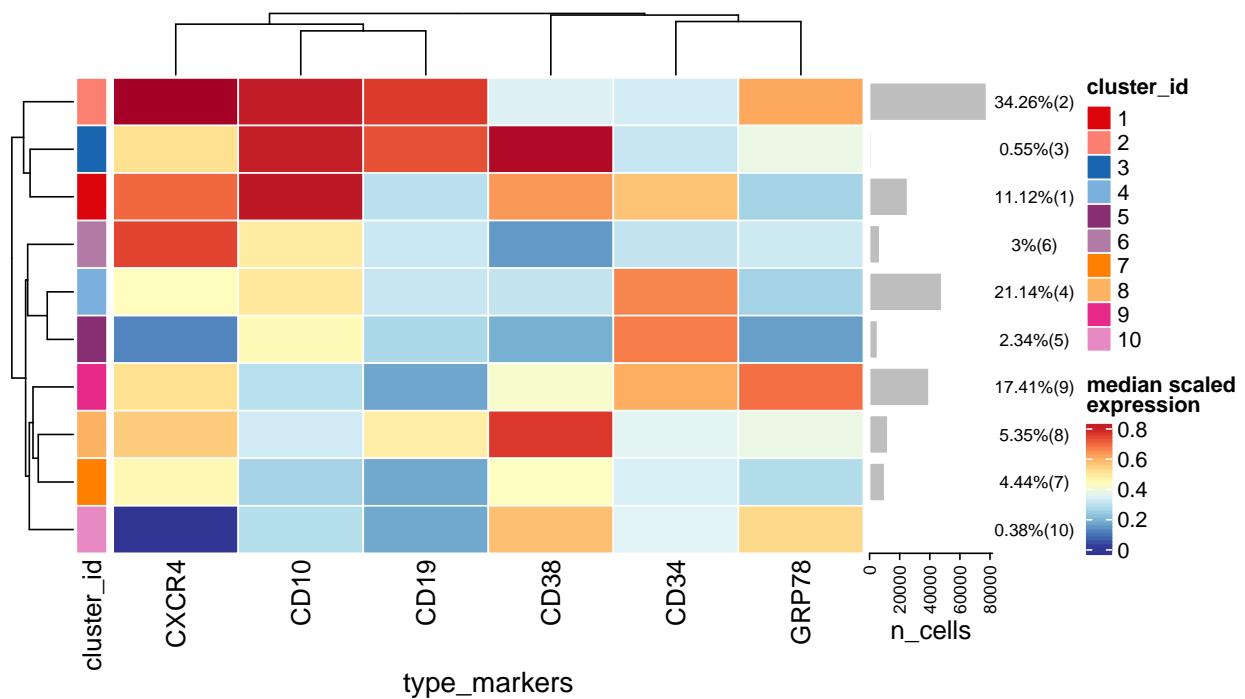
example: set.seed(1234)

set.seed(3332)

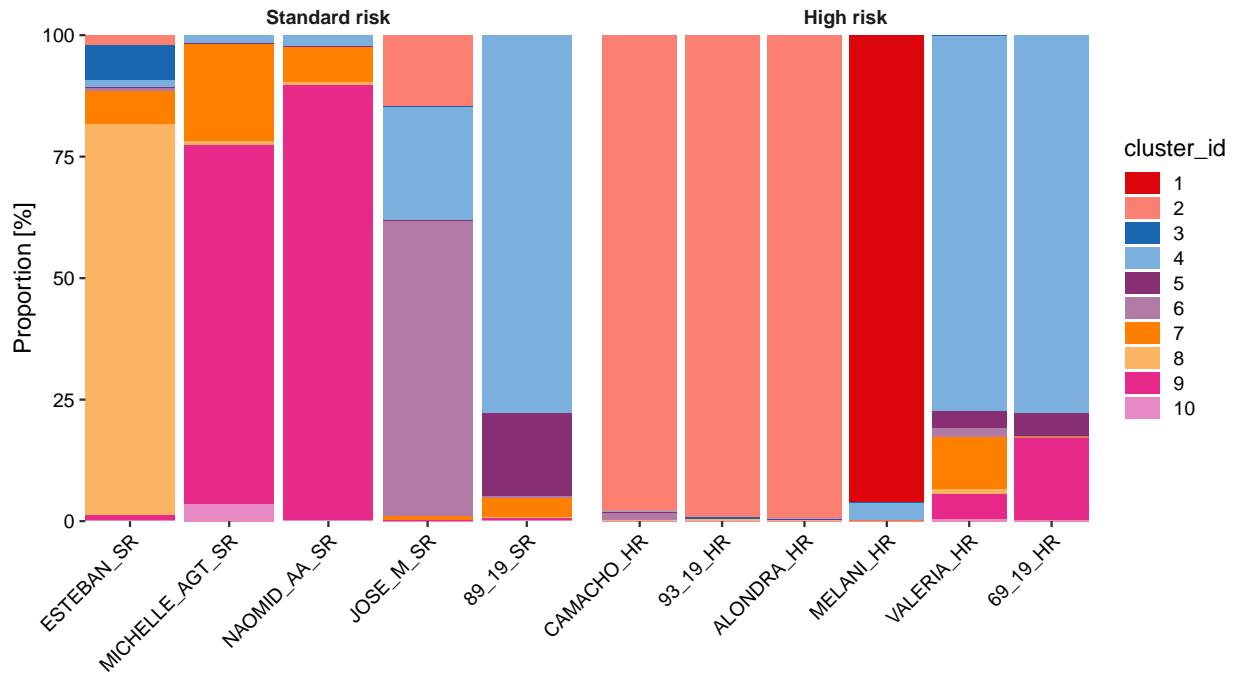
when not provided with a seed, overwrite our seed using the current system time: set.seed(as.numeric(Sys.time()))

```
set.seed(6666)
sce <- cluster(sce, features = "type",
                xdim = 10, ydim = 10, maxK = 10, seed = 6666)

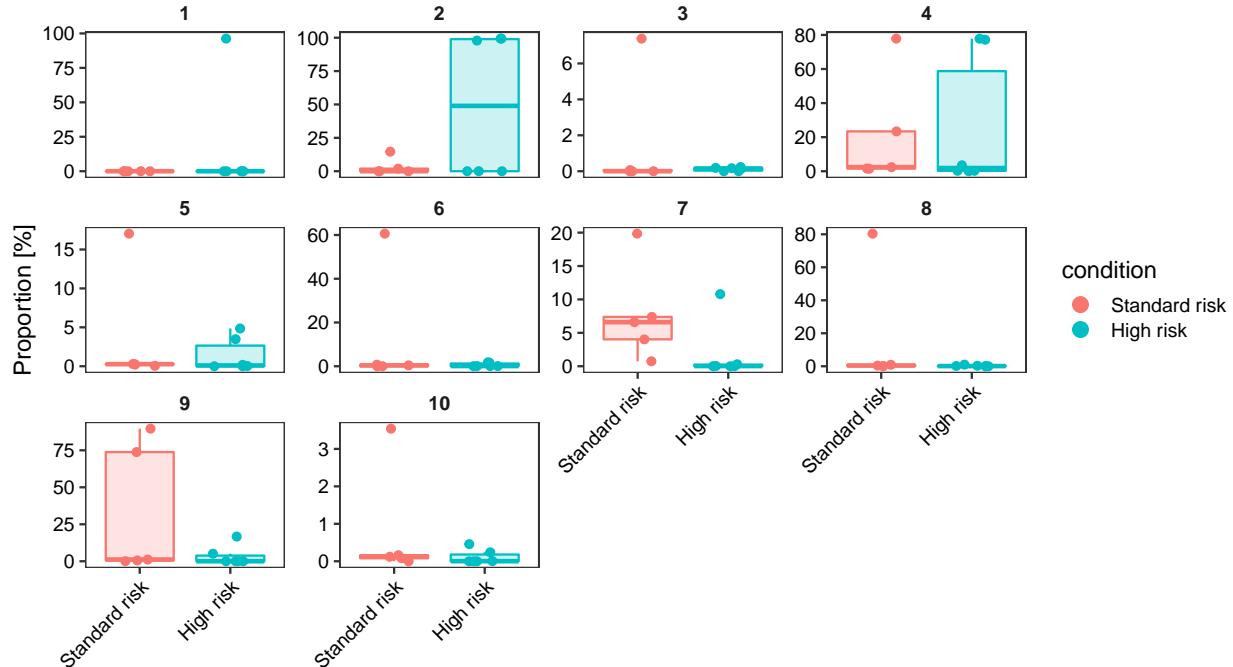
plotExprHeatmap(sce, features = "type",
                 by = "cluster_id", k = "meta10",
                 bars = TRUE, perc = TRUE)
```



```
plotAbundances(sce, k = "meta10", by = "sample_id")
```



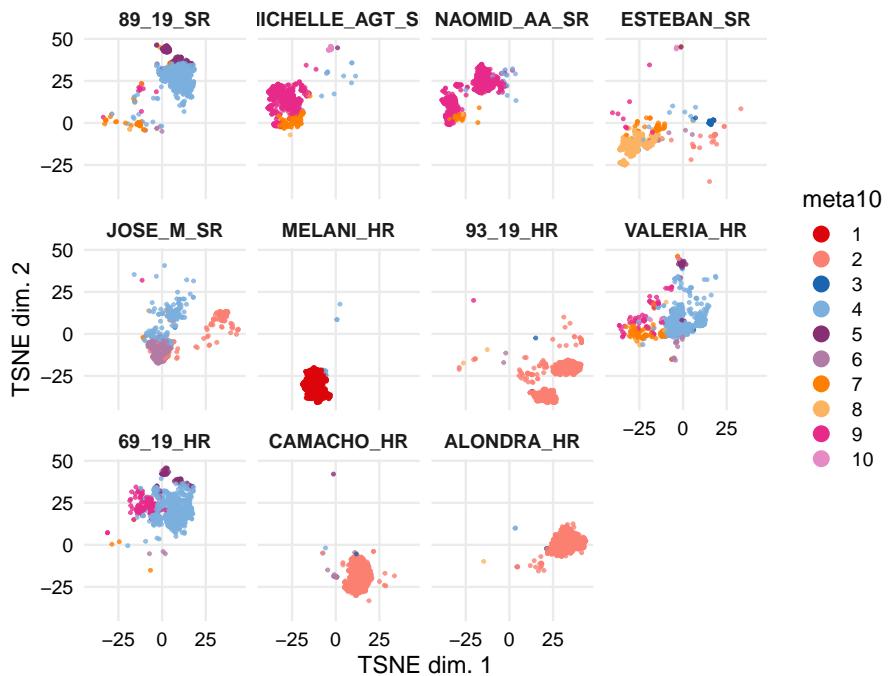
```
plotAbundances(sce, k = "meta10", by = "cluster_id")
```



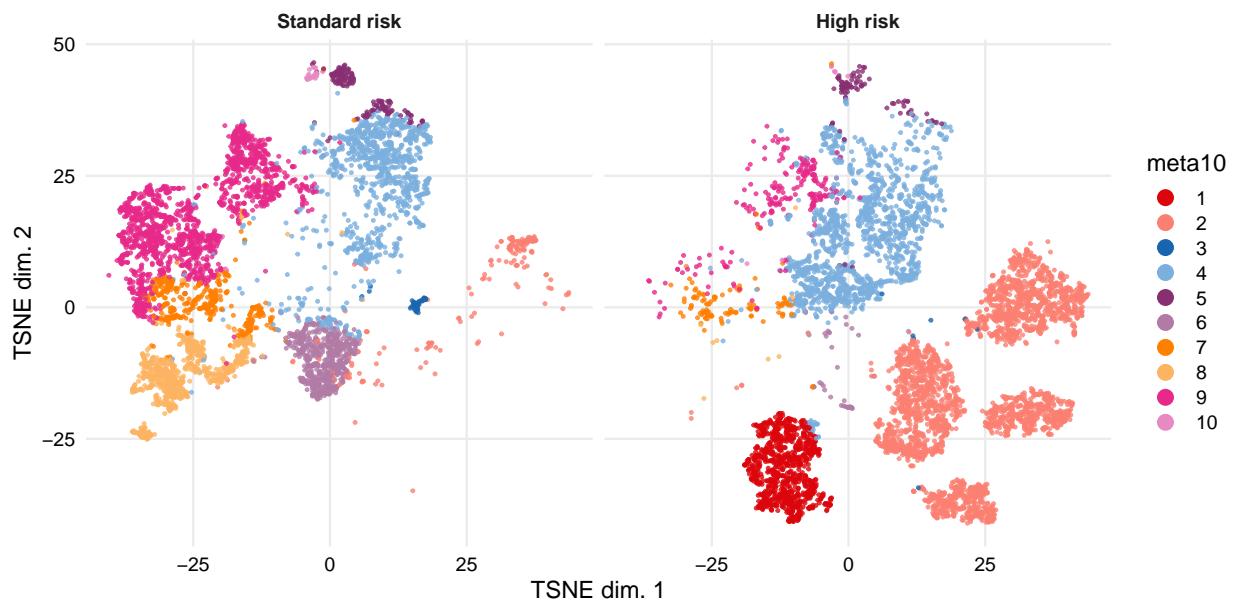
t-SNE and UMAP based on the arcsinh-transformed expression of the 6 lineage markers in the cells from the dataset. From each of the 6 samples, 1000 cells were randomly selected. Cells are colored according to the 10 cell subpopulations obtained with FlowSOM after the metaclustering step with ConsensusClusterPlus.

```
set.seed(6666)
sce <- runDR(sce, "TSNE", cells = 1000, features = "type")
```

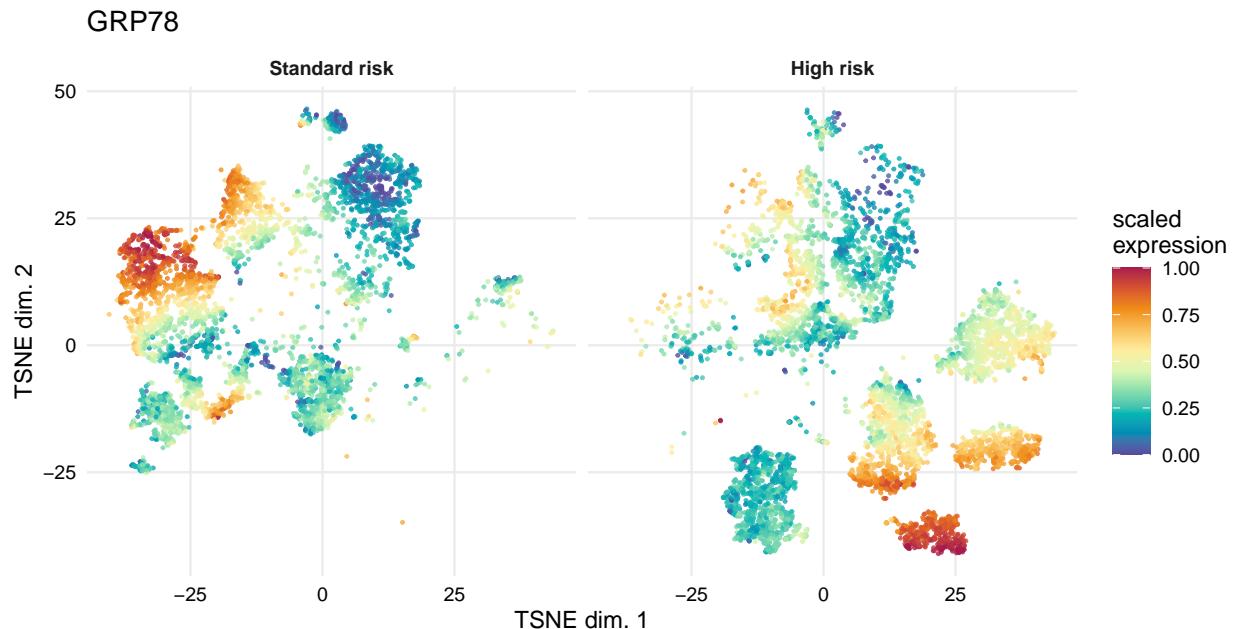
```
plotDR(sce, "TSNE",
       color_by = "meta10",
       facet_by = "sample_id")
```



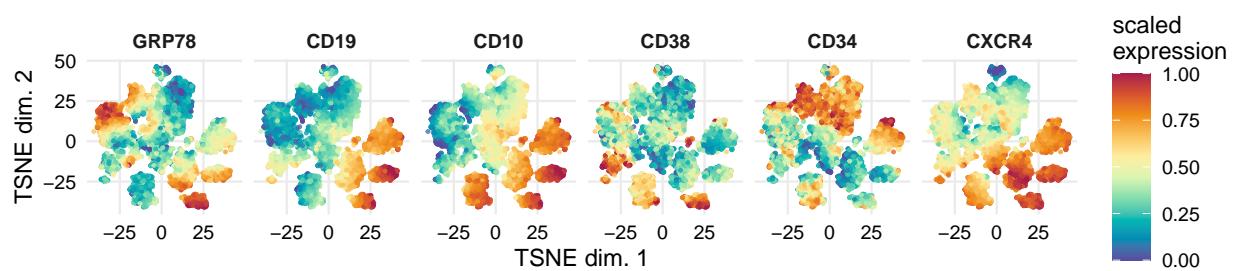
```
plotDR(sce, "TSNE",
       color_by = "meta10",
       facet_by = "condition")
```



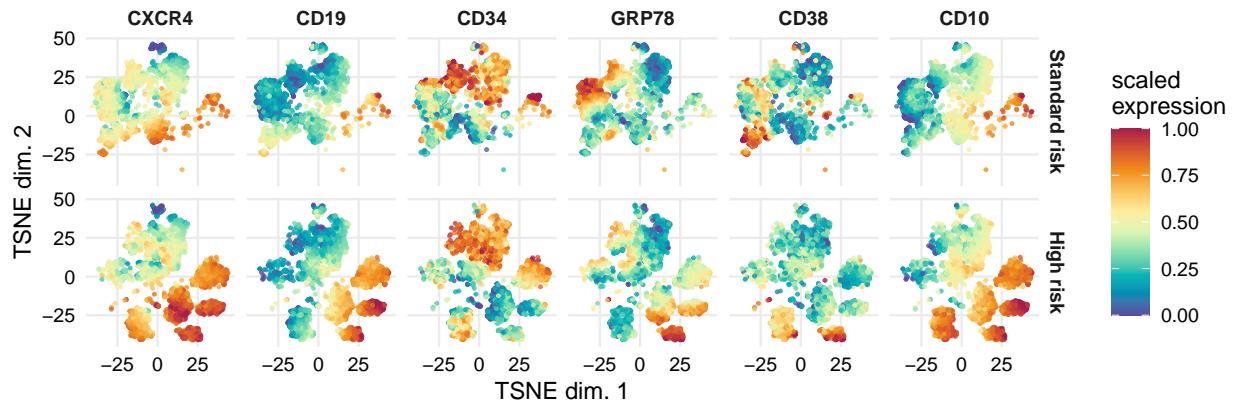
```
plotDR(sce, "TSNE", color_by = "GRP78",
       facet_by = "condition",
       ncol = 4,
       a_pal = rev(hcl.colors(10, "Spectral")))
```



```
plotDR(sce, "TSNE", color_by = rownames(sce), ncol = 6,
       a_pal = rev(hcl.colors(10, "Spectral")))
```

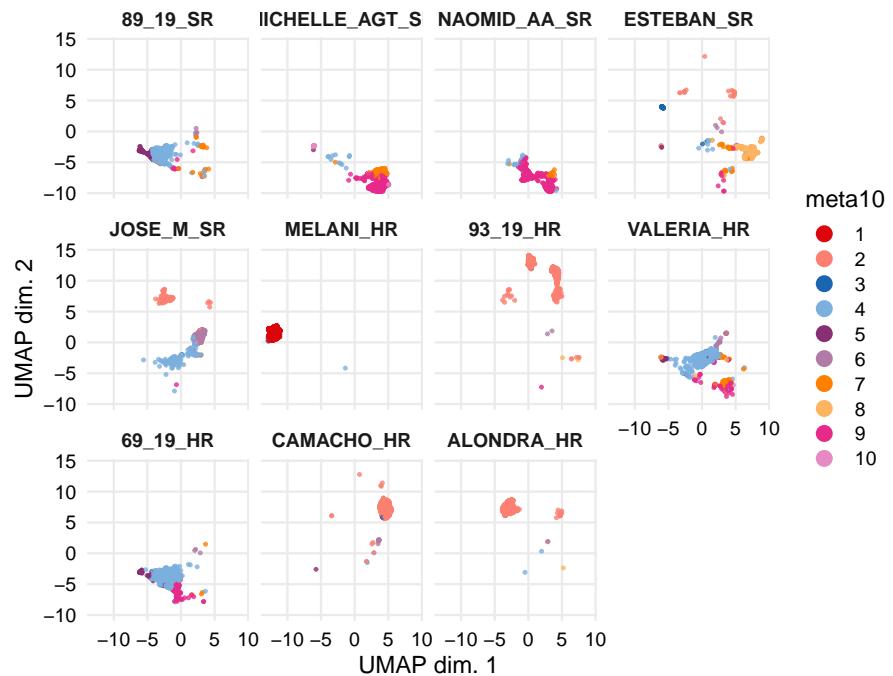


```
plotDR(sce,
       "TSNE",
       scale = TRUE,
       facet_by = "condition",
       color_by = sample(rownames(sce), 6),
       a_pal = rev(hcl.colors(10, "Spectral")))
```

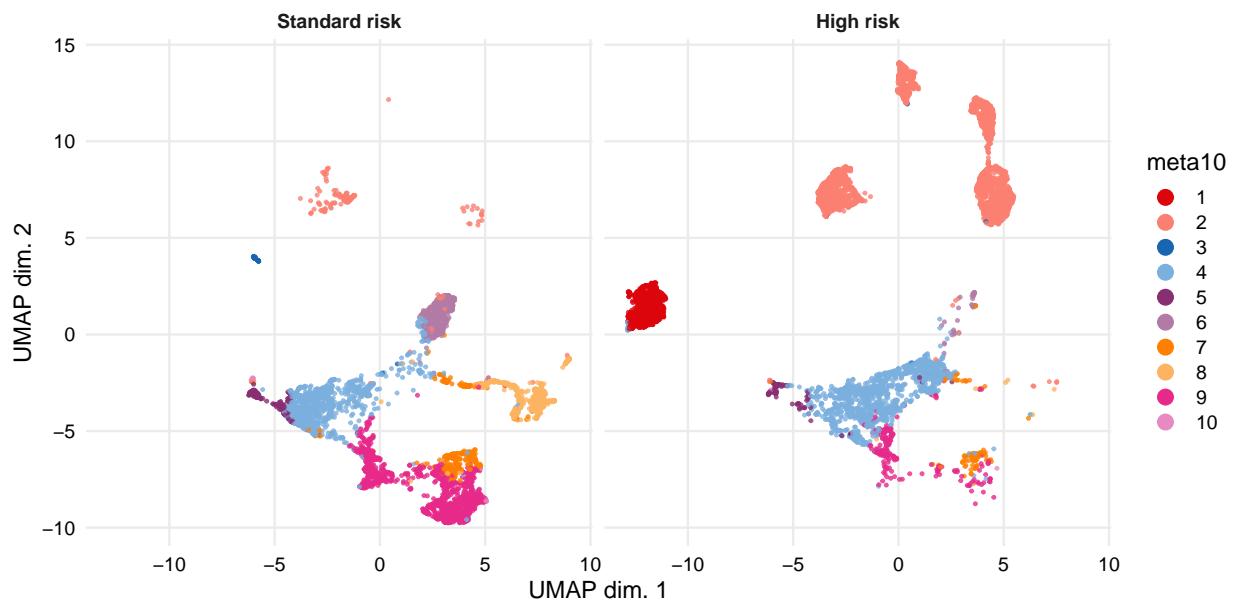


```
set.seed(6666)
sce <- runDR(sce, "UMAP", cells = 1000, features = "type")

plotDR(sce, "UMAP",
       color_by = "meta10",
       facet_by = "sample_id")
```

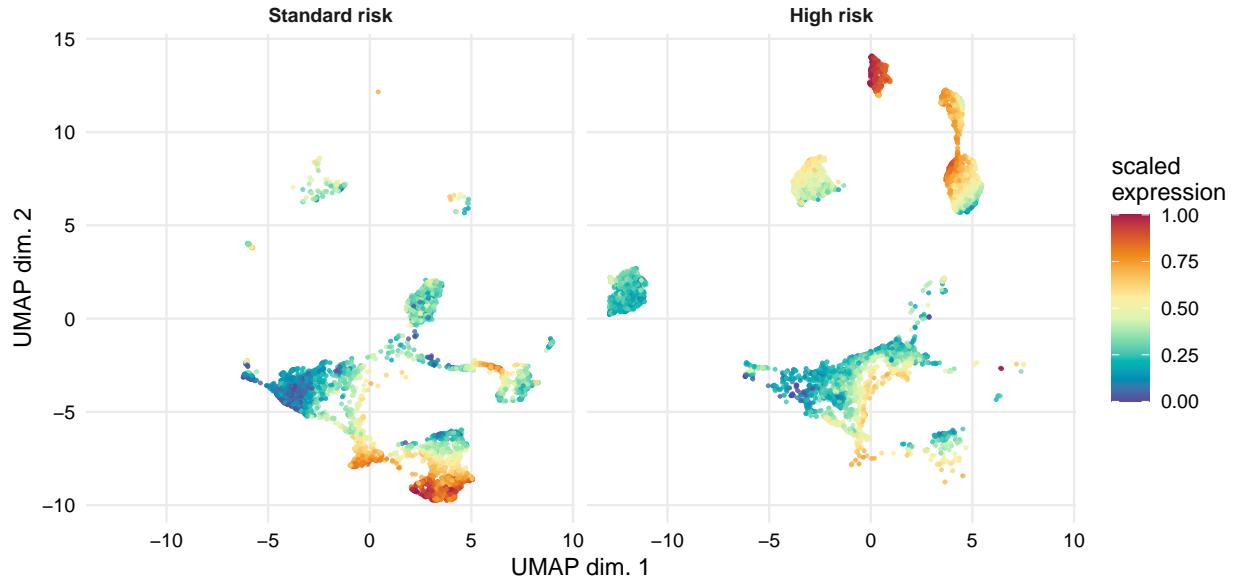


```
plotDR(sce, "UMAP",
       color_by = "meta10",
       facet_by = "condition")
```

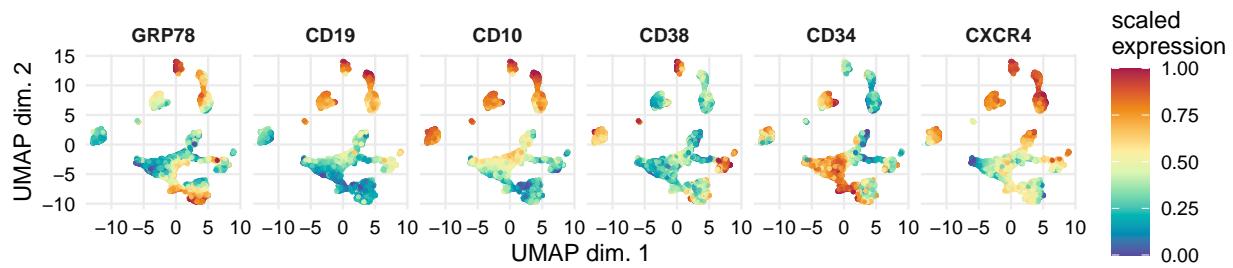


```
plotDR(sce, "UMAP", color_by = "GRP78",
       facet_by = "condition",
       ncol = 4,
       a_pal = rev(hcl.colors(10, "Spectral")))
```

GRP78



```
plotDR(sce, "UMAP", color_by = rownames(sce), ncol = 6,
       a_pal = rev(hcl.colors(10, "Spectral")))
```



```
plotDR(sce,
       "UMAP",
       scale = TRUE,
       facet_by = "condition",
       color_by = sample(rownames(sce), 6),
       a_pal = rev(hcl.colors(10, "Spectral")))
```

