PCA

Vlatka Antolovic

Dependencies

```
library(dplyr) # %in%
library(plotfunctions)
library(RColorBrewer)
library(gridExtra)
library(ggplot2)
```

Set working directory (full path of the 'scRNAseq_Pipeline' folder).

```
setwd("C:/Users/Vlatka/Documents/scRNAseq_Pipeline")
```

Import functions

```
# source('Functions/Fun_variableGenes.R')
source('Functions/Fun_prcomp_expressionPlot.R')
source('Functions/Fun_densityPlot.R')
```

1 Data import

Import UMI counts. Import gene features.

```
counts <- read.csv("./Data/norm_counts.csv", row.names = 1)

# Gene Info
info <- read.csv('./Data/geneInfo.csv', header = T, row.names = 1)
```

log10(UMI counts)

```
# log10 expression. <1 equaled to 1, resulting in min(log10(x))=0
# For gene expression overlay; doesn't matter if the count is 0 or 1.
logCounts <- counts %>% replace(., . < 1, 1) %>% log10()
```

2 PCA

Gene selection

```
#---------GENE SELECTION-----------
# A) Only keep genes with mean expression above the threshold
exprThreshold <- 0.01 # 9698
data <- counts[rowMeans(counts) > exprThreshold,]

#--- B) Only keep variable genes
# Using 'distance from median' (DM) value as a variance measure
# from Kolodziejczyk et al. 2015 (doi: 10.1016/j.stem.2015.09.011)
# exprThreshold <- 0.01
# perc <- 0.5 # % of genes to keep (ordered by maximum DM)
# geneNames = variableGenes(counts, exprThreshold, perc)
# data <- counts[geneNames,]
```

PCA

```
#---------PCA-----------
# expression>0.01 => 14 min (i7-4790 CPU, 16GB, 4 cores/ 8 logical cores)
t0 <- Sys.time()
pca = prcomp(log10(t(data) + 0.001), center = T, scale = T) #log10, standardised
Sys.time() - t0
```

```
## Time difference of 13.68093 mins
```

3 Plot gene expression

```
#---------Plot individual gene-----------
# Simplified plot functions
plot_pca <- function(dim1, dim2, gene, if_legend, bgcolor = 'white'){
  # par(mfrow = c(1,1), oma = c(0,0,0,0))
  par(mar = c(3,3,1,1))
  plotExpr(dim1, dim2, gene, colPal, steps, psize1, psize2, if_legend,
           if_panels = F, bgcol = bgcolor)
}

#---------Plot gene set------------------
plot_pcaSet <- function(dim1, dim2, gene_set, thresh, bgcolor = 'white'){
  # Above threshold in at least one cell
  over <- rowSums(counts>thresh) > 0
  set <- info[ ,gene_set] & over
  if (sum(set)==0) return()

  # Mean gene expression values
  arg = colMeans(logCounts[set, ])

  # Plot
  par(mar = c(3,3,1,1))
  plotVector(dim1, dim2, arg, colPal, steps, psize1, psize2, if_legend = F,
             if_panels = F, bgcol = bgcolor)
  # Title
  title(main = paste0(gene_set,' > ',thresh), cex.main = 0.9)
}
```

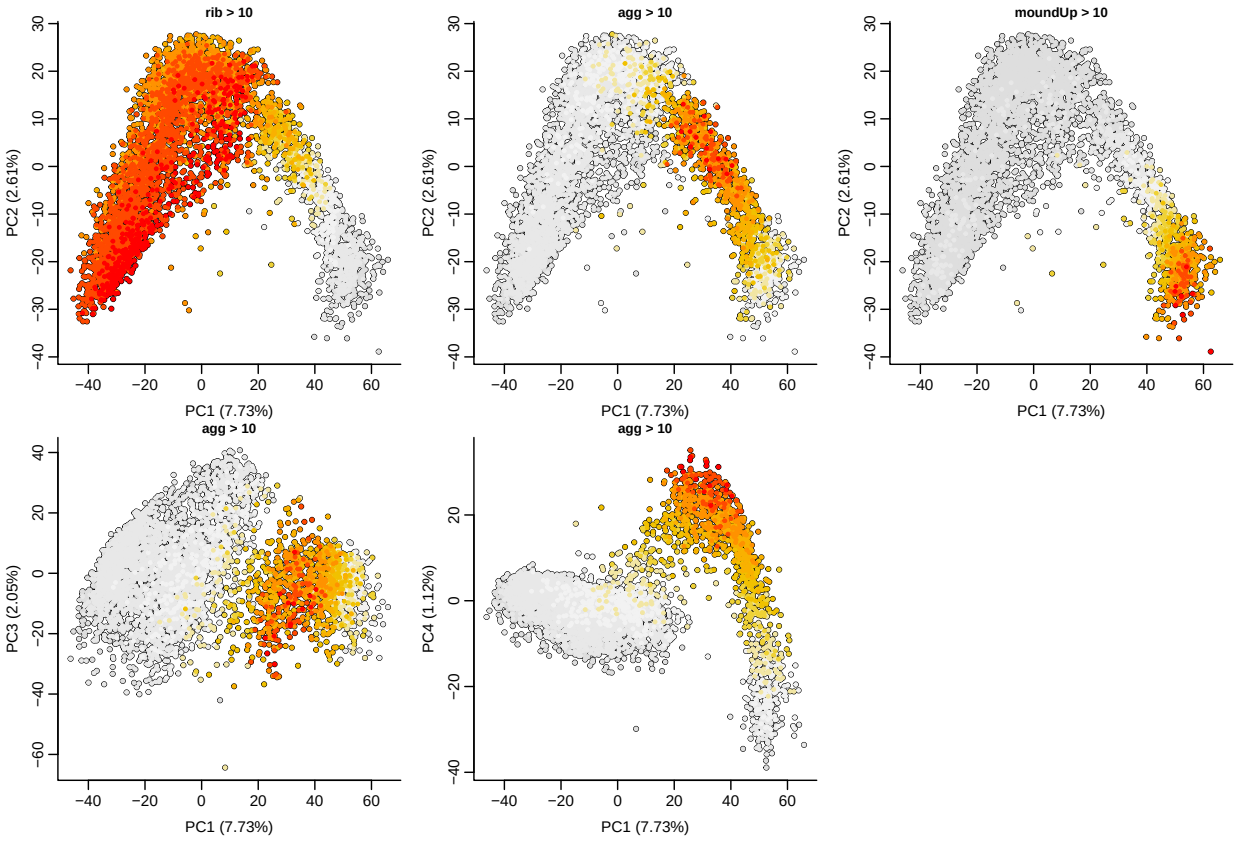
```
#---------Plot settings-----------
# Colour palette
colPal = colorRampPalette(c('gray87', 'whitesmoke', 'gold2', 'orange', 'red'));
steps = 10 #for colour-coding the data

# Outer ring, inner circle
psize1 = 1; psize2 = .75
```

Plot gene sets

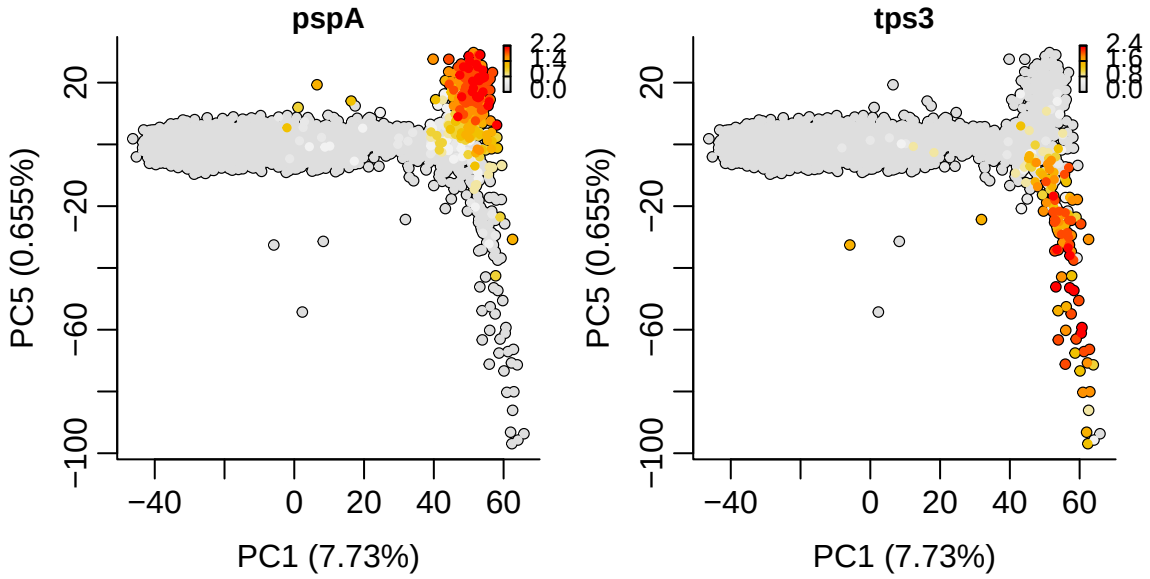
```
# Genes subselection: more than (threshold) UMIs in at least one cell
th <- 10

# Plot
par(mfrow = c(2,3))
plot_pcaSet(1, 2, 'rib', th)
plot_pcaSet(1, 2, 'agg', th)
plot_pcaSet(1, 2, 'moundUp', th) # Rosengarten et al. 2015
plot_pcaSet(1, 3, 'agg', th)
plot_pcaSet(1, 4, 'agg', th)
```



Plot individual genes

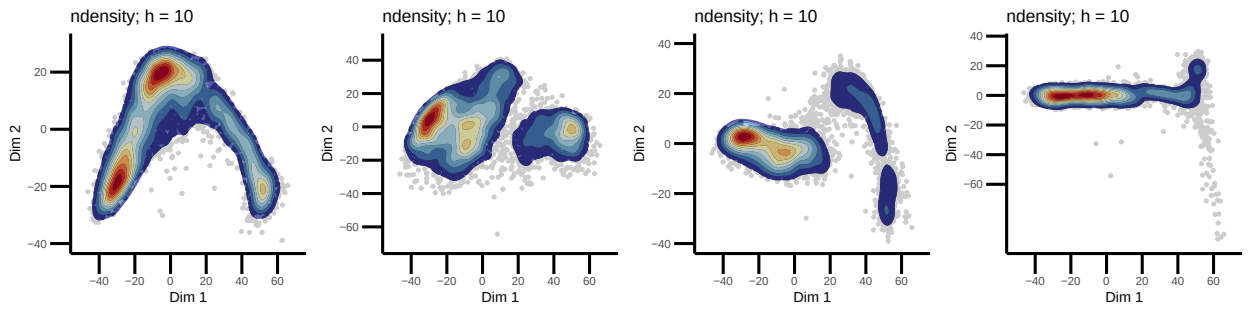
```
par(mfrow = c(1,2))
plot_pca(1, 5, 'pspA', T)
plot_pca(1, 5, 'tps3', T)
```



4 Plot cell density

```
#---------Colour palette----------
# Adjust discrete palette length & binNum
# Reverse & add white as a baseline
mypalette = rev(RColorBrewer::brewer.pal(11, 'RdYlBu')); mypalette =  c('white', mypalette)

#---------Plot----------
grid.arrange(
  plotContour_pca('ndensity', dim_2 = 2, y_lim = c(-40, 30)),
  plotContour_pca('ndensity', dim_2 = 3, y_lim = c(-70, 50)),
  plotContour_pca('ndensity', dim_2 = 4, y_lim = c(-40, 40)),
  plotContour_pca('ndensity', dim_2 = 5, y_lim = c(-100, 35)),
  nrow = 1)
```



Session information

```
sessionInfo()
```

```
## R version 4.1.1 (2021-08-10)
## Platform: x86_64-w64-mingw32/x64 (64-bit)
## Running under: Windows 10 x64 (build 19045)
##
## Matrix products: default
##
## locale:
## [1] LC_COLLATE=English_United States.1252
## [2] LC_CTYPE=English_United States.1252
## [3] LC_MONETARY=English_United States.1252
## [4] LC_NUMERIC=C
## [5] LC_TIME=English_United States.1252
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
## [1] ggplot2_3.3.6      gridExtra_2.3      RColorBrewer_1.1-3 plotfunctions_1.4
## [5] dplyr_1.0.8
##
## loaded via a namespace (and not attached):
##  [1] highr_0.9        pillar_1.7.0     compiler_4.1.1   tools_4.1.1
##  [5] digest_0.6.29    evaluate_0.15    lifecycle_1.0.1  tibble_3.1.6
##  [9] gtable_0.3.0     pkgconfig_2.0.3  rlang_1.0.6      cli_3.1.1
## [13] rstudioapi_0.13  yaml_2.3.5       xfun_0.35        fastmap_1.1.0
## [17] withr_2.5.0      stringr_1.4.0    knitr_1.41       generics_0.1.2
## [21] vctrs_0.3.8      isoband_0.2.5    grid_4.1.1       tidyselect_1.1.2
## [25] glue_1.6.1       R6_2.5.1         fansi_1.0.2      rmarkdown_2.18
## [29] farver_2.1.0     purrr_0.3.4      magrittr_2.0.2   MASS_7.3-54
## [33] codetools_0.2-18 scales_1.2.0     ellipsis_0.3.2   htmltools_0.5.4
## [37] colorspace_2.0-3 utf8_1.2.2       stringi_1.7.6    munsell_0.5.0
## [41] crayon_1.5.1
```

```
Sys.Date()
```

```
## [1] "2023-09-14"
```