

EE_Analysis

Dependencies

```
library(dplyr) # %in%
library(plotfunctions)
library(ggplot2)
library(ks)
```

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

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

Import functions

```
source('Functions/Fun_EE_expressionPlot.R')
source('Functions/Fun_densityPlot.R')
```

1 Data import

Import UMI counts (counts). Import gene features (info).

Import 2D cell coordinates after dimensionality reduction (XX). Reducing the dimensionality of transcriptome data to two dimensions used PCA first, then elastic embedding on the first 11 principal components (selected based on their variance contribution), as described in: Chen et al. (2019) Bioinformatics. doi: 10.1093/bioinformatics/bty1009.

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

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

# Data after (PCA and) EE
XX <- as.matrix(read.csv('./Data/SCRAN_0_01_EE.csv', header = F))
```

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 Plot expression

```

-----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 individual gene expression

```

-----Functions-----
# simplified plot functions
plotgene <- function(gene_name, background = 'white'){
  eplot(gene_name, colPal, steps, psize1, psize2, if_legend = T, if_title = T, background)
}

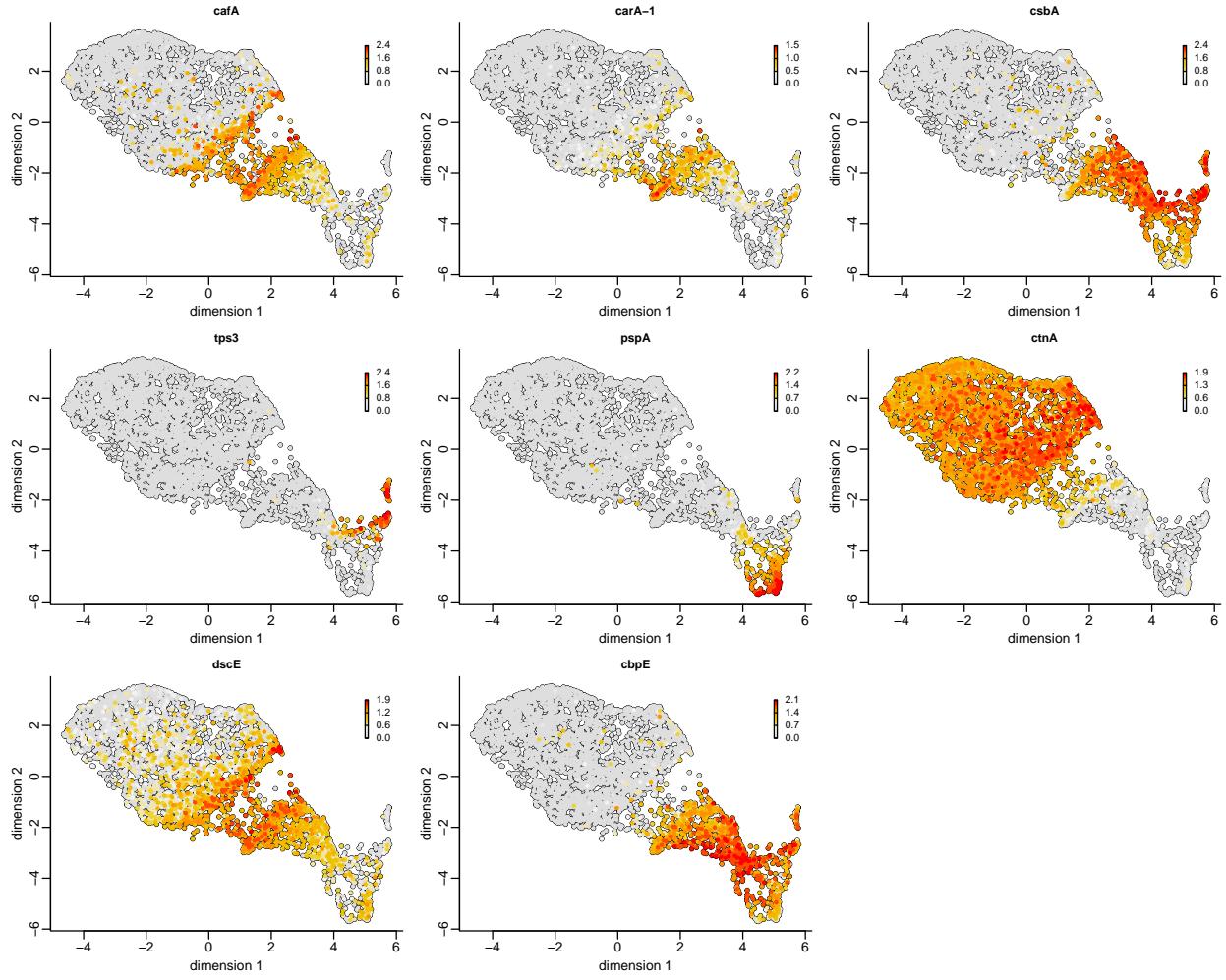
# ... with ID
plotgeneid <- function(gene_id, background = 'white'){
  idplot(gene_id, colPal, steps, psize1, psize2, if_legend = T, if_title = T, background)
}

```

```

-----Plot-----
# Min to max value
par(mfrow = c(3,3))
plotgene('cafA')
plotgene('carA-1')
plotgene('csbA')
plotgene('tps3')
plotgene('pspA')
plotgene('ctnA')
plotgene('dscE')
plotgene('cbpE')

```



Plot average expression of a group of genes

```

#-----Functions-----
# Plot mean expression of a gene set
plotset <- function(gene_set, thresh, if_legend, out, if_title = T){
  # 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
  if (missing(out)) {
    vplot(arg, colPal, steps, psize1, psize2, if_legend, background = 'white')
  } else {
    vplotTop(arg, out, colPal, steps, psize1, psize2, if_legend, 1,
             background = 'white')
  }
  # Title
  if (if_title == T) {title(main = paste0(gene_set, ' > ', thresh), cex.main = 0.9)}
}

```

```

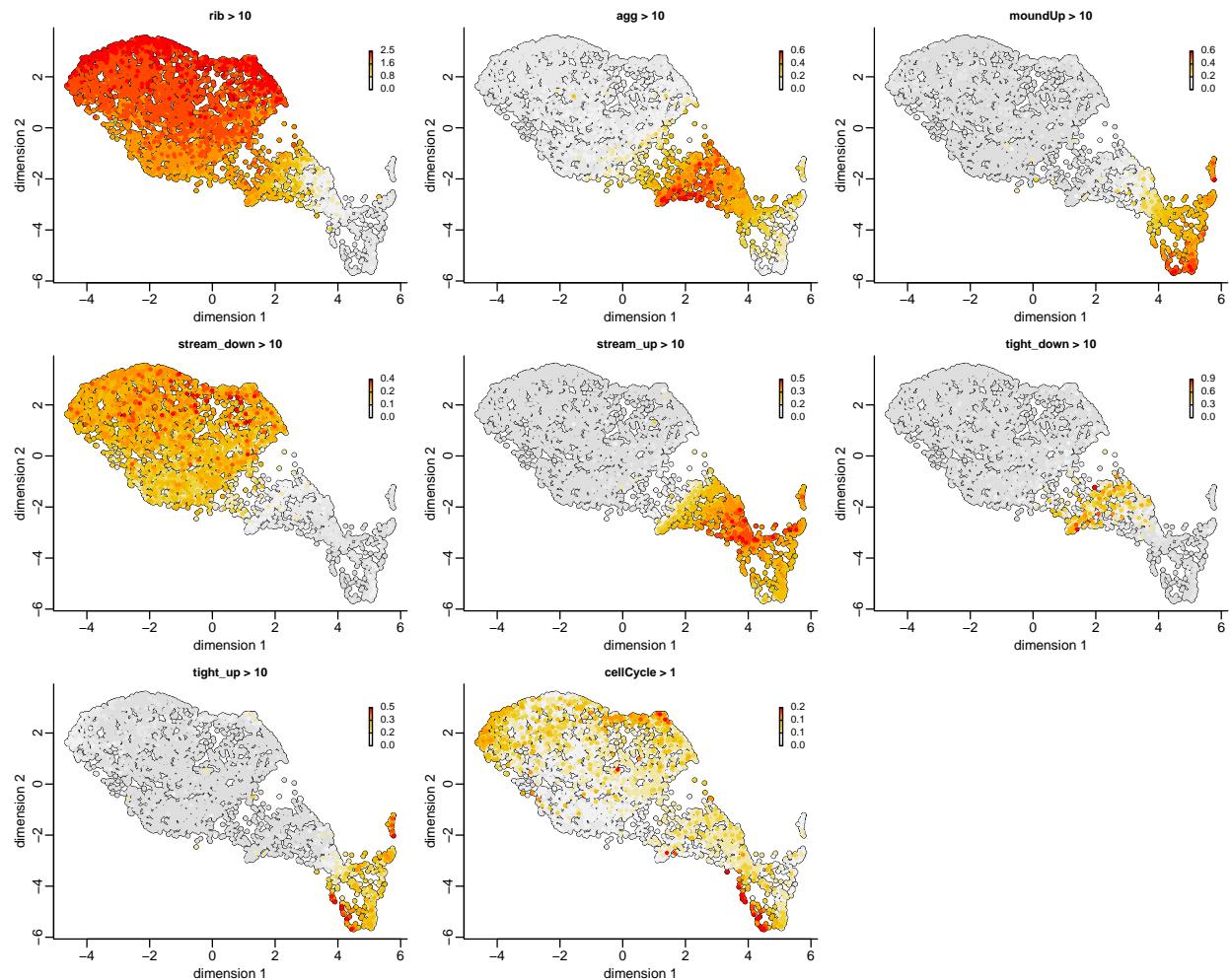
#-----Plot-----
# Genes expressed > threshold in at least one cell
th <- 10

par(mfrow = c(3,3))
plotset('rib', th, T)
plotset('agg', th, T)
plotset('moundUp', th, T)

# 'milestone' morphological change (Katoh-Kurasawa et al. 2021):
plotset('stream_down', th, T)
plotset('stream_up', th, T)
plotset('tight_down', th, T)
plotset('tight_up', th, T)

th <- 1
plotset('cellCycle', th, T, out = .2)

```



3. Cell density plots

3D density plot

```

# Define grid
x_lim = c(-5.5, 6.5)
y_lim = c(-6.5, 4.5)
by <- diff(x_lim) / 500 # resolution

xseq <- seq(x_lim[1] + .02, x_lim[2] + .05, by = by)
yseq <- seq(y_lim[1] - .05, y_lim[2] - .1, by = by)
grid <- expand.grid(xseq, yseq)

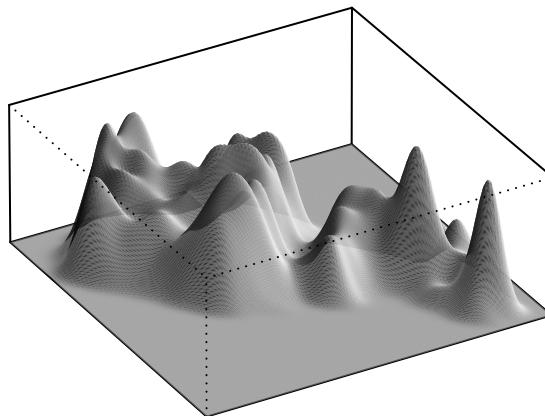
# Bandwidth
hvalue = Hpi(XX)

# Densities
KDE <- ks::kde(XX, H = hvalue, eval.points = grid, binned = T)$estimate
dens <- matrix(KDE, ncol = length(yseq), nrow = length(xseq))

# Plot 3-dimensional density surface
par(mar = c(0,0,1,0))
persp(xseq, yseq, dens, theta = 330, phi = 30, expand = 0.4, r = 90, ltheta = 80,
      shade = 0, border = NA, main = 'Density landscape', axes = F)

```

Density landscape



2D density plot

```

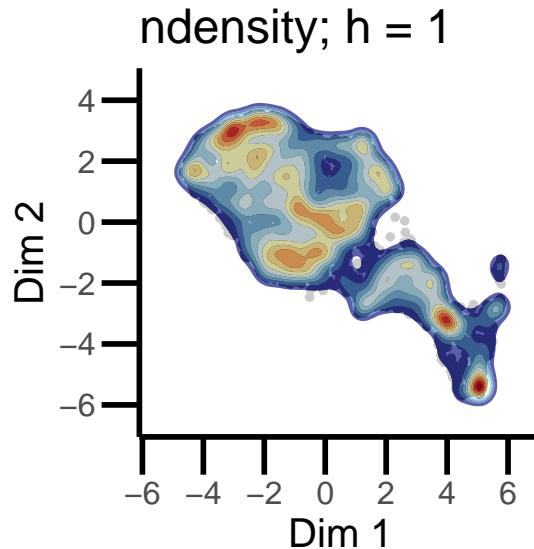
-----Colour palette-----
# Adjust discrete palette length & binNum

# Reverse & add white as a baseline
mypalette = rev(RColorBrewer::brewer.pal(11, 'RdYlBu')); mypalette = c('white', mypalette)

-----PLOT-----
# Define top level for EE density plot
d_ee <- d_plot()

# Plot density
# set or calculate h-value
plotContour(d_ee, 'ndensity')

```



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] ks_1.13.5       ggplot2_3.3.6     plotfunctions_1.4 dplyr_1.0.8
##
## loaded via a namespace (and not attached):
## [1] pracma_2.3.8      RColorBrewer_1.1-3  highr_0.9        pillar_1.7.0
## [5] compiler_4.1.1     tools_4.1.1        mclust_5.4.10    digest_0.6.29
## [9] lattice_0.20-44   evaluate_0.15     lifecycle_1.0.1   tibble_3.1.6
## [13] gtable_0.3.0      pkgconfig_2.0.3    rlang_1.0.6      Matrix_1.3-4
## [17] cli_3.1.1        rstudioapi_0.13   yaml_2.3.5      mvtnorm_1.1-3
## [21] xfun_0.35         fastmap_1.1.0     withr_2.5.0      stringr_1.4.0
## [25] knitr_1.41        generics_0.1.2    vctrs_0.3.8      isoband_0.2.5
## [29] grid_4.1.1        tidyselect_1.1.2   glue_1.6.1       R6_2.5.1
## [33] fansi_1.0.2       rmarkdown_2.18     farver_2.1.0     purrr_0.3.4
```

```
## [37] magrittr_2.0.2      MASS_7.3-54        codetools_0.2-18    scales_1.2.0
## [41] ellipsis_0.3.2       htmltools_0.5.4     colorspace_2.0-3    utf8_1.2.2
## [45] KernSmooth_2.23-20   stringi_1.7.6     munsell_0.5.0       crayon_1.5.1
```

```
Sys.Date()
```

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