ROGUE

September 26, 2019

Description

Often, it is not even clear whether a given cluster is uniform in unsupervised scRNA-seq data analyses. Here, we proposed the concept of cluster purity and introduced a conceptually novel statistic, named ROGUE, to examine whether a given cluster is a pure cell population.

Installation instructions for ROGUE

1. Install R (vesion 3.5 or greater).
2. Install R Studio (optional).

Installing dependency package

Before installing ROGUE, the "tidyverse" package should be installed first:

```
install.packages("tidyverse")
```

Installing ROGUE

To install ROGUE, run:

```
if (!requireNamespace("devtools", quietly = TRUE)) install.packages("devtools")
devtools::install_github("PaulingLiu/ROGUE", build_opts = NULL)
```

Vignettes

In this example workflow, we will be analyzing a previously reported dataset of dendritic cells (DCs). Here we provide the expression matrix and meta information (stored in example.data).

Library

```
suppressMessages(library(ROGUE))
suppressMessages(library(ggplot2))
suppressMessages(library(tidyverse))
```

Load the data

```
expr <- readr::read_rds(path = "~/DC.rds.gz")
meta <- readr::read_rds(path = "~/info.rds.gz")
```

For expression matrices, rows should be genes and columns should be cells. The expression value should be UMI counts (droplet-based datasets) or TPM (full-length based datasets).

```
expr[1:5, 1:4]
```

```
##       _p1t1__bcGDSJ _p1t1__bcDRQX _p1t1__bcFPXB _p1t1__bcHVVV
## A2M               0             0             0             0
## A2ML1             0             0             0             0
## AAAS              0             0             0             0
## AACS              0             0             0             0
## AAGAB             0             0             0             0
```

Meta information

The column 'ct' contains corresponding cell subtypes and column 'Patient' contains samples (e.g. patients) to which each cell belongs.

```
head(meta)
```

```
## # A tibble: 6 x 26
##   Patient Tissue `Barcoding emul~ Library Barcode `Total counts`
##   <chr>   <chr>  <chr>            <chr>   <chr>            <dbl>
## 1 p1      tumor  p1t              p1t1    bcGDSJ            4731
## 2 p1      tumor  p1t              p1t1    bcDRQX            1212
## 3 p1      tumor  p1t              p1t1    bcFPXB            2639
## 4 p1      tumor  p1t              p1t1    bcHVVV            2978
## 5 p1      tumor  p1t              p1t1    bcGJVN            1509
## 6 p1      tumor  p1t              p1t1    bcFSSY            3369
## # ... with 20 more variables: `Percent counts from mitochondrial
## #   genes` <dbl>, `Most likely LM22 cell type` <chr>, `Major cell
## #   type` <chr>, ct <chr>, used_in_NSCLC_all_cells <lgl>,
## #   x_NSCLC_all_cells <lgl>, y_NSCLC_all_cells <lgl>,
## #   used_in_NSCLC_and_blood_immune <lgl>, x_NSCLC_and_blood_immune <dbl>,
## #   y_NSCLC_and_blood_immune <dbl>, used_in_NSCLC_immune <lgl>,
## #   x_NSCLC_immune <lgl>, y_NSCLC_immune <lgl>,
## #   used_in_NSCLC_non_immune <lgl>, x_NSCLC_non_immune <lgl>,
## #   y_NSCLC_non_immune <lgl>, used_in_blood <lgl>, x_blood <dbl>,
## #   y_blood <dbl>, CellID <chr>
```

Filtering out low-abundance genes and low-quality cells

The `matr.filter` function allows you to filter out low-abundance genes and low-quality cells based on user-defined criteria.

```
expr <- matr.filter(expr, min.cells = 10, min.genes = 10)
```

Expression entropy model

To apply the S-E model, we calculate the expression entropy for each gene using `SE_fun` function.

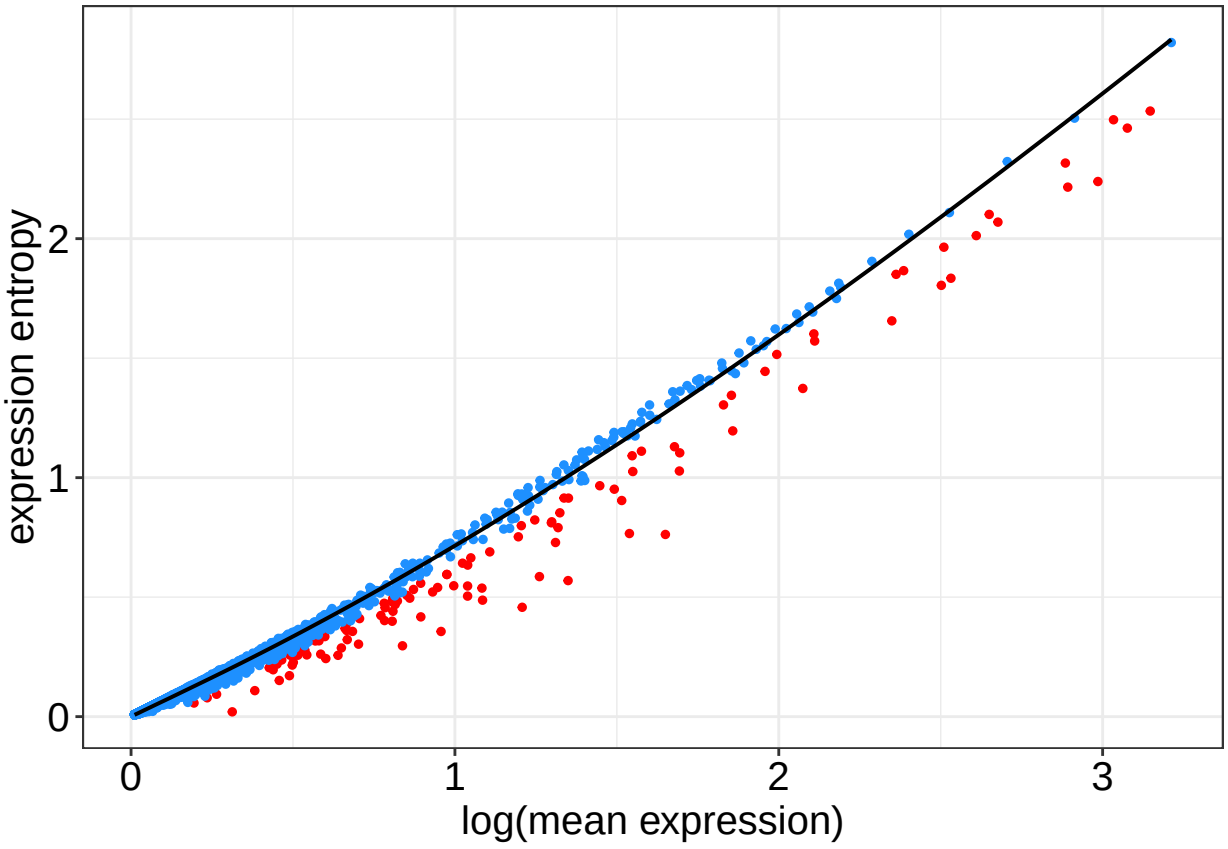
```
ent.res <- SE_fun(expr)
head(ent.res)
```

```
## # A tibble: 6 x 7
##   Gene     mean.expr entropy   fit    ds p.value p.adj
##   <chr>        <dbl>   <dbl> <dbl> <dbl>   <dbl> <dbl>
## 1 LYZ           1.65   0.762 1.27  0.510       0     0
## 2 HLA-DQB2      1.35   0.569 1.01  0.437       0     0
## 3 BIRC3         1.21   0.458 0.886 0.428       0     0
## 4 HSPA1A        1.54   0.766 1.17  0.406       0     0
## 5 HLA-DRB1      2.99   2.24  2.59  0.353       0     0
## 6 GZMB          1.26   0.586 0.931 0.345       0     0
```

S-E plot

We use `SEplot` function to visualize the relationship between S and E.

```
SEplot(ent.res)
```



- The identified highly informative genes could be applied to both clustering and pseudotime analyses.

ROGUE calculation

To access the purity of this DC population, we can calculate the ROGUE value using the `CalculateRogue` function. This population received a ROGUE value of 0.72, thus confirming their heterogeneity.

```
rogue.value <- CalculateRogue(ent.res, platform = "UMI")
rogue.value
```

```
## [1] 0.7219202
```

Calculate the ROGUE value of each putative cluster for each sample

In order to obtain an accurate estimate of the purity of each cluster, we recommend calculating the ROGUE value of each cell type in different samples.

```
rogue.res <- rogue(expr, labels = meta$ct, samples = meta$Patient, platform = "UMI", span = 0.6)
rogue.res
```

```
##         tDC2      tpDC      tDC3      tDC1
## p1 0.8376831 0.8604547 0.8494896 0.8481964
## p2        NA        NA        NA        NA
## p3 0.8028900 0.8941508 0.8995863 0.9150546
## p4 0.8041421 0.8992421 0.8763108 0.8658948
## p5 0.8702724 0.9321946 0.9247687        NA
## p6 0.8596472        NA 0.8892388 0.9280764
## p7 0.9262411 0.9028763 0.8949111 0.9419589
```

Visualize ROGUE values on a boxplot

```
rogue.boxplot(rogue.res)
```