Scissor Tutorial

Duanchen Sun

2020-05-27

Contents

1 Introduction

This Scissor R package contains the proposed Scissor algorithm (function `Scissor`), which is a novel single-cell data analysis approach that utilizes disease phenotypes that have been collected for bulk assays to identify the cell subpopulations that most highly correlate with a given phenotype. Scissor provides a novel framework to identify the biologically and clinically relevant cell subpopulations from single-cell assays by leveraging the wealth of phenotypes and bulk-omics datasets. In this tutorial, we used several examples to help users executing Scissor in real applications.

We first load in the required package:

```
library(Scissor)
```

2 Scissor implementation example

2.1 Input data preparation

There are two necessary input data sources of Scissor: a single-cell expression matrix and a bulk profiling data with sample phenotype, which can be a binary group indicator vector or clinical survival data. In this tutorial, the application of lung cancer cell in our manuscript was used as an example. First, let's load in the processed Lung Adenocarcinoma (LUAD) TPM quantification data and the corresponding survival information downloaded from The Cancer Genome Atlas (TCGA).

```
load(url('https://xialab.s3-us-west-2.amazonaws.com/Duanchen/Scissor_data/bulk_dataset.RData'))
load(url('https://xialab.s3-us-west-2.amazonaws.com/Duanchen/Scissor_data/bulk_survival.RData'))
```

In this bulk expression matrix, each row is a gene and each column is a sample. The dimensions of LUAD bulk dataset are:

```
dim(bulk_dataset)
```

```
## [1] 56716   471
```

which indicate there are 56716 genes and 471 samples in total. Besides, all of these samples have clinical outcomes:

```
head(bulk_survival)
```

```
##   TCGA_patient_barcode OS_time Status
## 1         TCGA-05-4249    1158      0
## 2         TCGA-05-4250     121      1
## 3         TCGA-05-4382     607      0
## 4         TCGA-05-4384     426      0
## 5         TCGA-05-4389    1369      0
## 6         TCGA-05-4390    1126      0
```

```
all(colnames(bulk_dataset) == bulk_survival$TCGA_patient_barcode)
```

```
## [1] TRUE
```

Note that Scissor requires a three-column format of clinical data, if the users want to identify a cell subpopulation that are correlated with good or bad survival. The first column contains the sample ID (observations), which should in the same order with the columns in bulk expression matrix. The second column is the survival time and the third column stands for the event of interest (e.g.recurrence of cancer or death), which is indicated by the variable "Status" with 0 means no event (censored) and 1 means event.

Then, we load in the LUAD single-cell RNA-seq raw counts data:

```
load(url('https://xialab.s3-us-west-2.amazonaws.com/Duanchen/Scissor_data/sc_dataset.RData'))
```

Same with the bulk expression matrix, the rows and columns are genes and cells, respectively. The dimensions of LUAD single-cell data are:

```
dim(sc_dataset)
```

```
## [1] 33694  4102
```

The users do not need to keep the common genes between bulk and single-cell expression datasets, which can be automatically achieved by the preprocessing steps of Scissor.

2.2 Identify the most relevent cell subpopulations by Scissor

Given the above inputs, now we can use Scissor to select the most relevent cell subpopulations with the guidence of phenotype (survival outcomes), which is fitted by a Cox regression model (`family = 'cox'`). We set the ratio between the L1 norm and Laplacian regularization to 0.05 (`alpha = 0.05`):

```
infos <- Scissor(alpha = 0.05, family = 'cox', Given_Inputs = NULL,
                 bulk_dataset = bulk_dataset, sc_dataset = sc_dataset, cor_type = 'pearson',
                 survival_info = bulk_survival, save_file = 'test.RData')
```

```
## [1] "Scissor output information: Positive cells: 201. Negative cells: 4."
```

The output message of Scissor indicates that there are 201 Scissor+ cells and 4 Scissor- cells. The information of selected cells is saved in variable `infos`:

```
names(infos)
```

```
## [1] "Coefs"         "Cell_positive" "Cell_negative" "Seurat_data"
```

```
length(infos$Cell_positive)
```

```
## [1] 201
```

```
infos$Cell_positive[1:4]
```

```
## [1] "AAAGTAGAGGAGCGAG_19" "AACCATGCATCTCCCA_19" "AACCGCGAGCTGCGAA_20" "AACTCAGTCCGCGGTA_19"
```

```
length(infos$Cell_negative)
```

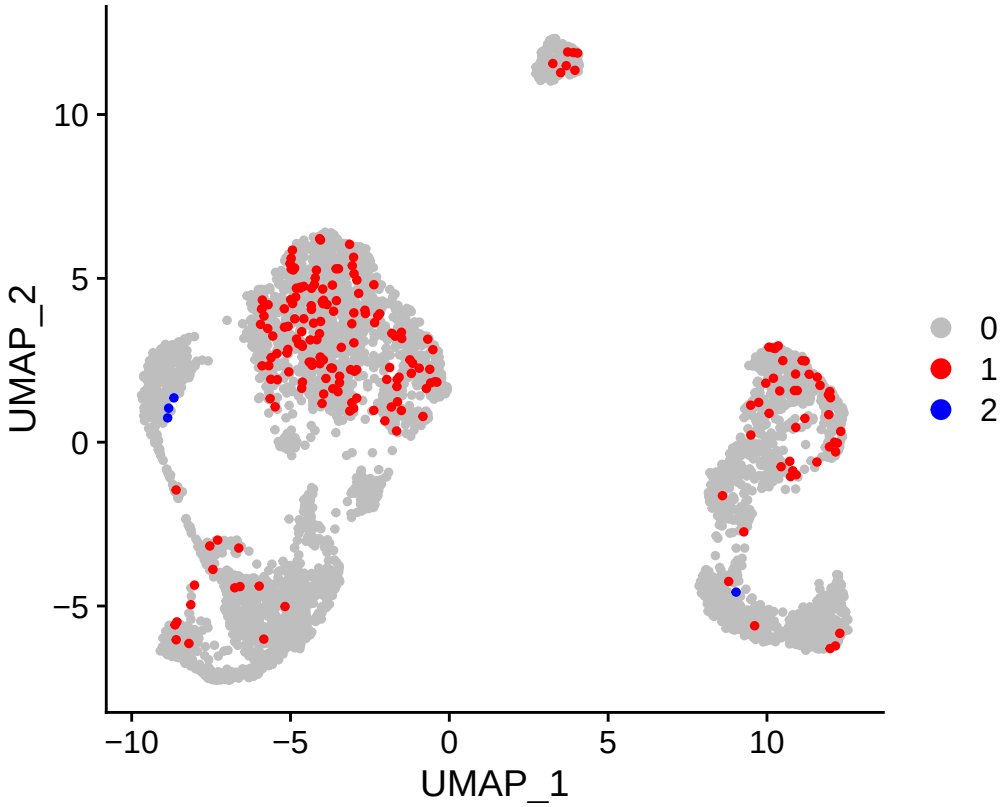
```
## [1] 4
```

```
infos$Cell_negative
```

```
## [1] "ACGCCAGTCCTCCTAG_20" "ACGGGCTAGTGGCACA_20" "CCGGTAGGTACCCAAT_15" "GACGCGTAGTGGTCCC_20"
```

Besides, the output of Scissor also contains a preprocessed single-cell Seurat object `Seurat_data`, which can directly accept the functions in `Seurat` package for analysis and visualization. The internal function in Scissor will automatically save the selected cells into a metadata in `Seurat_data`. Therefore, users can visualize the selected cells as:

```
DimPlot(object = infos$Seurat_data, reduction = 'umap', group.by = 'scissor',
        cols = c('grey','red','blue'), pt.size = 1, order = c(2,1))
```



in which `1` stand for the Scissor+ cells and `2` stand for the Scissor- cells.

2.3 Parameter tuning in Scissor

Usually, users cannot obtain the most informative selected cell at the first trial. In applications, Scissor will automatically save the preprocessed inputs for regression into a RData (`save_file = 'test.RData`), which

is convenient for users to directly execute the parameter tuning step. Suppose users use the above file name `test.RData`, Scissor can reselect the cells with a new alpha (`alpha = 0.01`) as:

```
infos <- Scissor(alpha = 0.01, family = 'cox', Given_Inputs = 'test.RData',
                 bulk_dataset, sc_dataset, bulk_survival)
```

```
## [1] "Scissor output information: Positive cells: 573. Negative cells: 20."
```

A smaller alpha leads to a less sparse selection result. At this time, the output message of Scissor indicates that there are 573 Scissor+ cells and 20 Scissor- cells. Usually, we recommand that the total selected cells of Scissor should not exceed the minimum number of 10% of total cells in single-cell RNA-seq data and 1000. In practice, users can change the alpha with `Given_Inputs` equals to the saved file name to meet their different goals.

3 Reference

Duanchen Sun and Zheng Xia (2020): Phenotype-guided subpopulation identification from single-cell sequencing data