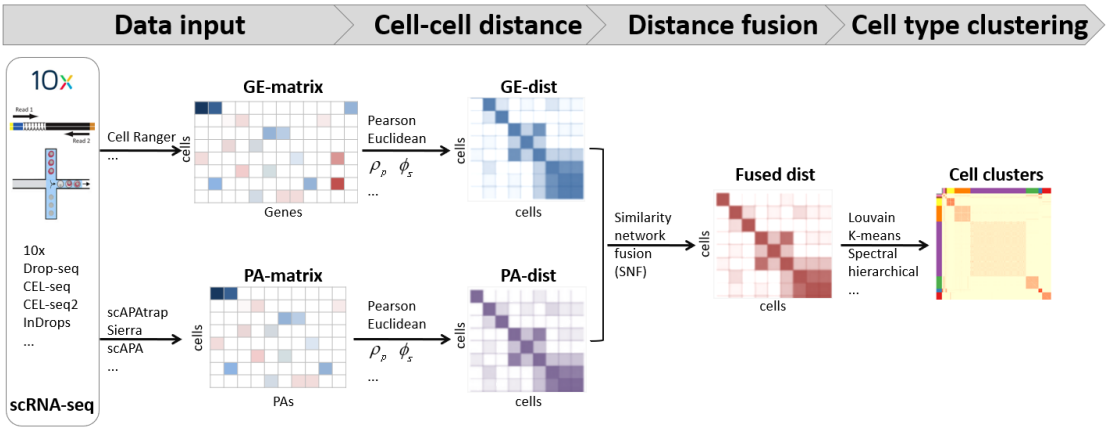
scLAPA: Learning association for single-cell transcriptomics by integrating single-cell profiling of gene expression and alternative polyadenylation

2020-12-08

1 Introduction

High variability and dropout rate inherent in scRNA-seq confounds the reliable quantification of cell-cell associations based on the gene expression profile alone. Lately bioinformatics studies have emerged to capture key transcriptome information on alternative polyadenylation (APA) from standard scRNA-seq and revealed APA dynamics among cell types, suggesting the possibility of discerning cell identities with the APA profile. We proposed a toolkit called scLAPA for learning association for single-cell transcriptomics by combing single-cell profiling of gene expression and alternative polyadenylation. As a comprehensive toolkit, scLAPA provides a unique strategy to learn cell-cell associations, improve cell type clustering and discover novel cell types by the augmentation of gene expression profiles with polyadenylation information, which can be incorporated in most existing scRNA-seq pipelines.

scLAPA mainly consists of four modules: (i) the input module, (ii) cell-cell distance, (iii) distance fusion, (iv) cell type clustering. The input module prepares the input for scLAPA, including a poly(A) site expression matrix (hereinafter referred as PA-matrix) and a gene expression matrix (hereinafter referred as GE-matrix). In the module of cell-cell distance, a cell-cell distance matrix is learned for PA-matrix (called PA-dist) and GE-matrix (called GE-dist), respectively. The module of distance fusion employs similarity network fusion (SNF) to integrate the two distance matrices (PA-dist and GE-dist) into one cell-cell distance matrix. The cell type clustering module clusters cells based on the fused distance matrix with various clustering methods.



2 Installation

You can install scLAPA from github with:

```
install.packages("devtools")
library(devtools)
install_github("BMILAB/scLAPA")
library(scLAPA)
```

3 Preparations

3.1 Identification of poly(A) sites from scRNA-seq

We identified poly(A) sites with scAPAtrap [1]. raw FASTQ reads were mapped with Cell Ranger or STAR and then uniquely mapped reads were obtained with SAMtools. Then UMI-tools [2] was employed to remove polymerase chain reaction (PCR) duplicates and extract unique molecular identifiers (UMIs). The findTails function in the scAPAtrap package was used to determine exact locations of poly(A) sites from reads with A/T stretches and the findPeaks function was adopted to identify all potential peaks of poly(A) sites from the whole genome level. Finally consensus poly(A) sites supported by both of the peak and the tail evidence were used. The featureCounts function in the Subread toolkit [3] was adopted to quantify the expression level for each poly(A) site. Please refer to the scAPAtrap website (https://github.com/BMILAB/scAPAtrap) for detailed usage steps.

Poly(A) site annotation was performed with the movAPA package [4], using the latest genome annotation of the respective species -- TAIR10 for Arabidopsis, mm10 for mouse and GRCh38 for human.

```
Library(movAPA)
library(TxDb.Mmusculus.UCSC.mm10.knownGene)
barcodefile <- './barcode.tsv'
barcode <- read.delim2(barcodefile,header = F)
barcode <- gsub('-[0-9]','',barcode$V1)
input<- generatescExpMa(countsfile, peaksfile, barcode, tails, min.cells = 2,min.count = 0)
coldata<data.frame(group=colnames(input)[7:ncol(input)],row.names=colnames(input)[7:
ncol(input)])
scPACds <- movAPA::readPACds(pacfile = input, coldata = coldata)
txdbnu.10<-TxDb.Mmusculus.UCSC.mm10.knownGene
scPACds <- annotatePAC(scPACds,txdbnu.10)
```

3.2 Preparation of data input for scLAPA

The input module prepares the input for scLAPA, including a poly(A) site expression matrix (hereinafter referred as PA-matrix) and a gene expression matrix (hereinafter referred as GE-matrix). The PA-matrix is generated from raw scRNA-seq with scAPAtrap, which stores expression levels of poly(A) sites, with each row denoting a poly(A) site and each column denoting a cell. The GE-matrix can be obtained from websites like NCBI GEO and 10x Genomics, or generated by various routine scRNA-seq analysis tools like Cell Ranger. In this study, we use the mouse Hypothalamus data as example.

```
> library(scLAPA)
>####loading DE-matrix
> data("gene_Hypothalamus")
> gene.data<-gene_Hypothalamus
> dim(gene.data)
[1] 14374   727
> gene.data[1:3,1:3]
                AAAACAGCGTGA AAAACGCCGTCG AAAAGGTGACTC
0610007C21RIK            0     1.956876     2.293011
0610007L01RIK            0     1.395896     0.000000
0610007N19RIK            0     0.000000     0.000000
>####loading PA-matrix
> data("PA_Hypothalamus")
> PA.data<-PA_Hypothalamus$counts
> dim(PA.data)
[1] 7114  727
> PA.data[5:7,1:3]
       AAAACAGCGTGA AAAACGCCGTCG AAAAGGTGACTC
PA8389     0.000000            0     0.000000
PA8396     1.584963            0     3.584963
PA8640     0.000000            0     0.000000
```

4 Standard analysis work-flow

4.1 Data preprocessing (optional)

To reduce computational time, users can use R package Seurat [5] to remove low quality cells and filter out genes with low expression of GE-matrix and sue R package movAPA to preprocess PA-matrix for downstream analysis.

4.2 Cell-cell distance

A cell-cell distance matrix is learned for *PA-matrix* (called *PA-dist*) and *GE-matrix* (called *GE-dist*), respectively .In this module, various distance metrics can be chosen, including Euclidean distance, Pearson correlation, two metrics of proportionality (ρ_p and $\emptyset_s$) [6], RAFSIL(Random Forest based Similarity Learning) [7] and SIMLR (Single-cell Interpretation via Multikernel Learning) [8].

```
># PA-matrix convert to Pearson similarity matrix(PA-dist)
> PA.pearson<-calculate_distance(PA.data,method = "pearson")
> PA.pearson[1:3,1:3]
                AAAACAGCGTGA AAAACGCCGTCG AAAAGGTGACTC
AAAACAGCGTGA       0.0000000       0.9076725       0.4434940
AAAACGCCGTCG       0.9076725       0.0000000       0.9019272
AAAAGGTGACTC       0.4434940       0.9019272       0.0000000
># GE-matrix convert to Pearson similarity matrix(DE-dist)
> gene.pearson<-calculate_distance(gene.data,method = "pearson")
```

4.3 Distance fusion

After learning PA-dist and GE-dist, similarity network fusion (SNF)[9] is utilized to flexibly integrate the two layers of cell-cell similarities into one distance matrix. First, PA-dist and GE-dist are iteratively and gradually fused to a consensus network, utilizing the non-linear method of message passing theory. Then weak similarities representing potential noise are discarded, and strong similarities are retained. By generating coherent cell-cell similarities from both APA isoform and gene layers, SNF profiles a more comprehensive biological relationship among cells, beyond the scope of methods solely based on GE-matrix.

```
> fusion<-network_fusion(PA.pearson,gene.pearson,K=20,alpha = 0.5,T=10)
> fusion[1:3,1:3]
                AAAACAGCGTGA AAAACGCCGTCG AAAAGGTGACTC
AAAACAGCGTGA 0.0008185671 0.0004694652 0.0333463433
AAAACGCCGTCG 0.0004694652 0.0008264428 0.0005650480
AAAAGGTGACTC 0.0333463433 0.0005650480 0.0008335804
```

4.4 Single-cell clustering

Four widely-used clustering methods were provided in scLAPA to cluster cells on the basis of the fused cell-cell similarity matrix, including Louvain clustering [10] , hierarchical clustering (HC) [11], spectral clustering (SC) [12], and k-means.

```
>library(clues)
>library(psych)
>library(igraph)
>library(flashClust)
>#louvain clustering
```

```
> d.pr<-1/(fusion+1)
> fusion_clu<-Clustering(d.pr,method = "louvain")
> label<-as.character(PA_Hypothalamus$colData$celltype)
>#get clustering index
> fusion_index_louvain<-index_index(fusion_clu,label = label)
> fusion_index_louvain
    purity      Rand        HA        MA        FM   Jaccard       nmi
0.9807428 0.9948049 0.9845614 0.9846390 0.9878699 0.9760239 0.9756830
```

We use three other clustering methods in the scLAPA package besides louvain clustering for cell clustering.

```
>## spectral clustering
>fusion_clu<-Clustering(fusion,c=7,method = "SC")
>fusion_index_SC<-index_index(fusion_clu,label = label)
> fusion_index_SC
    purity      Rand        HA        MA        FM   Jaccard       nmi
0.9821183 0.9759872 0.9260780 0.9264726 0.9421664 0.8887836 0.9419880
>## hierarchical clustering
>fusion_clu<-Clustering(d.pr,c=7,method = "HC")
>fusion_index_HC<-index_index(fusion_clu,label = label)
> fusion_index_HC
    purity      Rand        HA        MA        FM   Jaccard       nmi
0.9834938 0.9777530 0.9315414 0.9319066 0.9465057 0.8965936 0.9473860
>##K-means
>fusion_clu<-Clustering(d.pr,c=7,method = "Kmeans",iter.max = 1e+9, nstart = 1000)
>fusion_index_Kmeans<-index_index(fusion_clu,label = label)
> fusion_index_Kmeans
    purity      Rand        HA        MA        FM   Jaccard       nmi
0.9821183 0.9762714 0.9269895 0.9273789 0.9428724 0.8900999 0.9423603
```

5 Session information

```
sessionInfo()
R version 3.5.3 (2019-03-11)
Platform: x86_64-w64-mingw32/x64 (64-bit)
Running under: Windows 10 x64 (build 19041)
Matrix products: default
locale:
[1] LC_COLLATE=Chinese (Simplified)_China.936
[2] LC_CTYPE=Chinese (Simplified)_China.936
[3] LC_MONETARY=Chinese (Simplified)_China.936
[4] LC_NUMERIC=C
[5] LC_TIME=Chinese (Simplified)_China.936
```

```
attached base packages:
[1] stats4    parallel  stats     graphics  grDevices utils     datasets
[8] methods   base
other attached packages:
[1] SIMLR_1.8.1        SNFtool_2.3.0      flashClust_1.01    igraph_1.2.5    clues_0.5.9
[6] psych_2.0.9        RAFSIL_0.2.5       WGCNA_1.69         scLAPA_0.1.0
loaded via a namespace (and not attached):
[1]rappdirs_0.3.1     ClusterR_1.2.1     prabclus_2.3-2     GGally_2.0.0    coda_0.19-3
[6]tidyr_1.1.0        bit64_0.9-7        knitr_1.2          irlba_2.3.3     dclone_2.3-0    rpart_4.1-15
[12]inline_0.3.15 hwriter_1.3.2 RCurl_1.98-1.2 AnnotationFilter_1.6.0 generics_0.0.2
[17]callr_3.4.3       cowplot_1.0.0      RSQLite_2.2.0      future_1.18.0   bit_1.1-15.2    readxl_1.3.1
[22]caTools_1.16      DBI_1.1.0          geneplotter_1.6    htmlwidgets_1.5.1  reshape_0.8.8
[27]purrr_0.3.4       ellipsis_0.3.1     backports_1.1.8    annotate_1.60.1
```

6 References

1. Xiaohui, W., et al., *scAPAtrap: identification and quantification of alternative polyadenylation sites from single-cell RNA-seq data.*
2. Smith, T., A. Heger, and I.J.G.r. Sudbery, *UMI-tools: modeling sequencing errors in Unique Molecular Identifiers to improve quantification accuracy.* 2017. **27**(3): p. 491.
3. Yang, et al., *featureCounts: an efficient general purpose program for assigning sequence reads to genomic features.* 2014.
4. Ye, W., et al., *movAPA: Modeling and visualization of dynamics of alternative polyadenylation across biological samples.* Bioinformatics, 2020.
5. Stuart, T., et al., *Comprehensive Integration of Single-Cell Data.* 2019. **177**(7): p. 1888-1902.e21.
6. Skinnider, M.A., J.W. Squair, and L.J.J.N.M. Foster, *Evaluating measures of association for single-cell transcriptomics.* 2019. **16**(5).
7. Baran, P.M. and K.J.B. Dennis, *Random forest based similarity learning for single cell RNA sequencing data.* 2018(13): p. i79-i88.
8. Wang, B., et al., *Visualization and analysis of single-cell RNA-seq data by kernel-based similarity learning.* 2017. **14**(4): p. 414.
9. Wang, B., et al., *Similarity network fusion for aggregating data types on a genomic scale.* 2014. **11**(3): p. 333-337.
10. Blondel, V.D., et al., *Fast unfolding of communities in large networks.* 2008.
11. Eisen, M.B., et al., *Eisen, M.B., Spellman, P.T., Brown, P.O. & Botstein, D. Cluster analysis and display of genome-wide expression patterns. Proc. Natl. Acad. Sci. USA 95, 14863–14868.* 1999. **95**(25): p. 14863-14868.
12. Ng, A.Y., M.I. Jordan, and Y. Weiss. *On Spectral Clustering: Analysis and an Algorithm*. in *Proceedings of the 14th International Conference on Neural Information Processing Systems: Natural and Synthetic*. 2001.