从头构建cellranger sparse matrix

# 1. 简介

在分析单细胞数据时大家一定经常遇到这三个文件barcodes.tsv.gz, features.tsv.gz和matrix.mtx.gz。 这三个文件就是cellranger输出的单细胞counts数据格式的其中一种，在GEO数据库中经常看到这种格式的数据，应该是使用频率最高的一种。今天小编就为大家讲解一下怎么这三个文件存放的什么内容，以及怎么从一个matrix构建这三个文件。看到最后文末有福利哦！

# 2. 三个文件存放的什么数据

barcode.tsv.gz存放的是细胞的barcode序列（Figure 1 A），就是一窜ATCG组成的序列，是每个细胞的唯一标识。feature.tsv.gz是放的特征数据（Figure 1 B），单细胞数据基本就是基因ID， single cell ATAC就是每个fragment的ID。如果是统计的是transcript或者exon水平的UMI count，那么feature.tsv.gz存放的就是transcript或者exon ID。 matrix.mtx文件内容稍微复杂一些（Figure 1 C），第一行是header，%%MatrixMarket表示这是一个matrix market format格式的文件。matrix表示这个数据存放的是一个matrix。coordinate表示数据是按照行列数据的形式存放的。integer表示存放的数据是整数。general表示数据是非对称的。

第二行的三个数据分别表示这个样本有36773个features（或者基因）,3904个细胞，以及15139930个非零数值。

最下面的三列分别是非零数值所在的行id, 列id以及这个非零数据。

|  |  |
| --- | --- |
| A  A | B  B |

|  |  |
| --- | --- |
| C  C |  |

Figure 1: 三种格式内容展示。(A)barcode文件内容。 (B)feature文件内容。(C)matrix.mtx文件内容。

# 3. 怎么从一个matrix构建这三个文件

我们从GEO下载数据的时候有时候会碰到作者上传的是一个matrix。如果我们想合并多个数据集的数据集，其中有的作者上传的是matrix格式的数据，而有的作者提供的是cellranger输出的数据，这时我们就希望能把matrix转换成cellranger输出的格式。下面我们就用一个真实的数据展示一下怎么用matrix数据构建cellranger输出格式的数据。实例数据来源于GEO数据库中的GSM3823939样本。

library(magrittr)  
library(Matrix)  
library(rio)  
  
GSM3823939 <- readRDS("GSM3823939\_control.s1.dgecounts.rds")  
  
# 这个数据包含了intro, exon，intro + exon的UMI count和read count数据  
# 这里我么使用intron + exon的UMI count数据进行演示  
mtx <- GSM3823939[[1]][[2]][[1]] %>% as.matrix

Warning in asMethod(object): sparse->dense coercion: allocating vector of size  
1.1 GiB

看一下这个数据长什么样

mtx[1:3, 1:3]

AAACCTGAGTGTTAGA AAACCTGCAAGCGCTC AAACCTGCACCAGATT  
ENSG00000000003 0 0 0  
ENSG00000000005 0 0 0  
ENSG00000000419 0 0 0

# 4. 提取features和barcodes数据

下面我们就从这个数据中提取feature和barcode数据，其实就是这个矩阵的行名和列名。

features <- mtx %>% rownames  
barcodes <- mtx %>% colnames  
  
# 使用rio export函数在到处的时候加上gz后缀，在保存的时候会被自动压缩  
rio::export(x = data.frame(features),   
 file = "sparse/features.tsv.gz",   
 col.names = F,   
 format = "tsv")  
  
rio::export(data.frame(barcodes),   
 file = "sparse/barcodes.tsv.gz",   
 col.names = F,   
 format = "tsv")

# 5. 将matrix保存成Matrix Martket Format

首先需要将矩阵转换成稀疏矩阵CsparseMatrix，CsparseMatrix是Matrix包提供的一个class。

mtx <- as(mtx, "CsparseMatrix")  
# 将数据保存成matrix.mtx需要使用Matrix包提供的writeMM函数  
writeMM(obj = mtx, file = "sparse/matrix.mtx")

NULL

# 6. 将上述新构建的矩阵读取进来，看一下与最开始的矩阵是否一样

library(Seurat)

mtx1 <- Read10X(data.dir = "sparse", gene.column = 1)  
  
identical(mtx, mtx1)

[1] TRUE

# 7. 彩蛋

为了一次性解决此类问题，我们专门开发了一个函数，这个函数可以批量将matrix转化成cellranger输出的格式。你只需要准备matrix list和样本名，它就可以将每个matrix分别保存到指定的目录下面，并且每个文件是压缩好的。

# 8. 总结

今天小编为大家展示了经常见到的cellranger sparse matrix保存的是什么数据，以及每个数据中意义。此外，我们还写个一个个性化的函数，希望大家以后遇到这样的问题时不必烦恼。

**本公众号专注于单细胞、空间转录组、空间代谢组、空间蛋白质组以及宏基因组方向个性化分析和方案设计，欢迎大家交流咨询。后续我们还会更新更多空间转录组、空间蛋白质组、空代间谢组以及单细胞方面的内容，欢迎关注交流**。

