**Drop-seq Pipeline**

by Liu-cy

1.目前从输入到表达矩阵的流程“跑通”了；

2.但我的输入理解错了，Dropseq不像smartseq可以抽出几个细胞来跑，我以为我是选了几个细胞，实际上相当于进行了一次伪降采样，只取了部分reads来跑的；(而且我用的是tenx的数据，我觉得tenx和dropseq两个共通)

3.所以目前得到的表达矩阵是否正确，结果不能验证；

4.××××××××××流程中有几个步骤的原理和作用不能解释×××××××××；

5.引物sequence的确切数据不能100%确定，我从tenx官网的protocal手册示意图里copy的；

6.这个东西在alpha上运行得有点特慢，我相当于只用了不到100M的数据，从头到尾的时间都能有3个小时多了，不知道是不是就应该这样，还是一些其他原因导致的；

8.Dropseq的(非tenx)7.9G的数据，第一步提cellbarcode跑了17个小时都没完，就取消了；

9.正在尝试将多个fastq转bam；(因为一个数据里有多个fastq)

10.正在跑整合的代码，不知能否通(多步整合,software里自带的,需要修改部分代码)。

11.如果目前的流程是正确的话，那么验证存在的问题就是(如果使用Dropseq的数据，太大了，遥遥无期)(如果使用tenx的数据，adapter引物序列不确定，结果可能略有差)。

**Software**

**Drop-seq tools-1.13**

<https://paste.ubuntu.com/p/kGh55nM85J/>

(需要翻墙才能完整下载；需要java环境)

**STAR**

<https://paste.ubuntu.com/p/WFTShtp2kj/>

**文档**

**Drop-seq tools Cookbook**

<http://mccarrolllab.com/wp-content/uploads/2016/03/Drop-seqAlignmentCookbookv1.2Jan2016.pdf>

**Picard Help**

<http://broadinstitute.github.io/picard/command-line-overview.html>

**Refdata**

[Human and mouse reference dataset](https://paste.ubuntu.com/p/fRZg6GfRMy/)

<https://paste.ubuntu.com/p/fRZg6GfRMy/>

**数据**

100 1:1 Mixture of Fresh Frozen Human (HEK293T) and Mouse (NIH3T3) Cells

<https://support.10xgenomics.com/single-cell-gene-expression/datasets/2.1.0/hgmm_100>

(网站上有数据报告，使用的Tenx的数据，应该也可以作为Drop-seq的原始输入)

GSE63473/GSM1544799 (HEK and 3T3 mix, McCarroll Lab)

<https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSM1544799>

Macosko, et al. "Highly Parallel Genome-wide Expression Profiling of Individual Cells Using Nanoliter Droplets." *Cell* 161.5(2015):1202-14.

**流程**

**Pre process:**

Reference中genome.fa需要生成一个dictionary方便后续工具使用

Picard-CreateSequenceDictionary

<https://paste.ubuntu.com/p/SyhbvjFnQk/>

参数说明:

R:RefData中的genome.fa文件

O:生成的dictionary

(注意:生成的.dict文件名需与距.fa文件相同，否则后续工具不能检测到)

**Raw data Input:**

标准输入: Unmapped BAM

通常输入: fastq，需要使用Picard-FastqToSam

Picard-FastqToSam

<https://paste.ubuntu.com/p/HKY354nx8j/>

参数说明:

F1:单端测序的Read 或 双端测序的Read1

F2:双端测序的Read2

O:Unmapped BAM

**Step 1:**

标注cell barcode

TagBamWithReadSequenceExtended

<https://paste.ubuntu.com/p/cBMzt4hbXc/>

参数说明:

INPUT=my\_unaligned\_data.bam

OUTPUT=unaligned\_tagged\_Cell.bam

BASE\_RANGE=1-16: cell barcode的长度，Tenx通常14, Drop-seq通常12, 视实验而定；

BASE\_QUALITY: 质量控制的base score阈值

NUM\_BASES\_BELOW\_QUALITY: 容错为1

**Step 2:**

标注UMI barcode

TagBamWithReadSequenceExtended

<https://paste.ubuntu.com/p/wrZyTvZvdd/>

参数说明:

INPUT=unaligned\_tagged\_Cell.bam

OUTPUT=unaligned\_tagged\_CellMolecular.bam

BASE\_RANGE=17-26: UMI barcode的长度，tenx通常为10, Drop-seq通常为8, 视实验而定；

DISCARD\_READ=True: 去掉低质量的reads。

**Step 3:**

Filter bam 去掉cell和UMI barcode里有低质量碱基的reads

FilterBam

<https://paste.ubuntu.com/p/n6sGmYmZ3Z/>

参数说明:

INPUT=unaligned\_tagged\_CellMolecular.bam

OUTPUT=unaligned\_tagged\_filtered.bam

**Step 4:**

去掉adapter序列

TrimStartingSequence

<https://paste.ubuntu.com/p/smRdct6HWp/>

参数说明：

INPUT=unaligned\_tagged\_filtered.bam

OUTPUT=unaligned\_tagged\_trimmed\_smart.bam

SEQUENCE:Adapter的序列，(Tenx通常为GTGACTGGAGTTCAGACGTGTGCTCTTCCGATCT，Dropseq通常为AAGCAGTGGTATCAACGCAGAGTGAATGGG，视实验而定)；

MISMATCHES=0

NUM\_BASES=5: 连续5个碱基和Adapter的序列差別为0就视为Adapter。

**Step 5:**

去掉 polyA

PolyATrimmer

<https://paste.ubuntu.com/p/XNBJshyBmR/>

参数设置：

INPUT=unaligned\_tagged\_trimmed\_smart.bam

OUTPUT=unaligned\_mc\_tagged\_polyA\_filtered.bam

MISMATCHES=0

NUM\_BASES=6: 连续6个碱基和PolyA的序列差別为0就视为PolyA。

**Step 6：**

将bam转成fastq以进行比对

Picard-SamToFastq

<https://paste.ubuntu.com/p/w6RdxMStYH/>

参数说明：

INPUT=unaligned\_mc\_tagged\_polyA\_filtered.bam

FASTQ=unaligned\_mc\_tagged\_polyA\_filtered.fastq

**Step 7:**

做比对Align

STAR

<https://paste.ubuntu.com/p/KSHdTg8v2b/>

参数说明：

--genomeDir: Refdata里的star文件夹

--readFilesIn: unaligned\_mc\_tagged\_polyA\_filtered.fastq

--outFileNamePrefix: star (前缀)

(输出比对结果starAligned.out.sam)

**Step 8:**

比对结果排序

Picard-SortSam

<https://paste.ubuntu.com/p/dvqNyxkMMP/>

参数说明：

I=starAligned.out.sam

O=aligned.sorted.bam

**Step 9:**

将比对后的reads与比对前的reads合并，修正比对过程中损失的reads

[Picard-MergeBamAlignment](https://paste.ubuntu.com/p/CQrMWKRD3q/)

<https://paste.ubuntu.com/p/CQrMWKRD3q/>

参数说明：

REFERENCE\_SEQUENCE:Refdata的.fa文件(此处要用到.dict)

UNMAPPED\_BAM=unaligned\_mc\_tagged\_polyA\_filtered.bam

ALIGNED\_BAM=aligned.sorted.bam

OUTPUT=merged.bam

**Step 10:**

标记覆盖exon的reads

TagReadWithGeneExon

<https://paste.ubuntu.com/p/4vTBgry82v/>

参数说明：

I=merged.bam

O=star\_gene\_exon\_tagged.bam

ANNOTATIONS\_FILE:Refdata中的.gtf文件

**Step 11:**

检测合成错误

DetectBeadSynthesisErrors

<https://paste.ubuntu.com/p/7jdvRPfS7F/>

参数说明：

I=star\_gene\_exon\_tagged.bam

O=out\_gene\_exon\_tagged.bam

NUM\_BARCODES: (细胞数的两倍)

PRIMER\_SEQUENCE=GTGACTGGAGTTCAGACGTGTGCTCTTCCGATCT (视实验而定)

**Step 12:**

生成表达矩阵

DigitalExpression

<https://paste.ubuntu.com/p/MDQDWJPtzh/>

参数说明：

I=out\_gene\_exon\_tagged.bam

O=out\_gene\_exon\_tagged.dge.txt.gz

NUM\_CORE\_BARCODES:细胞数