Smart-seq数据分析基本流程

相关软件安装

## FastQC

FastQC是用来对fastq文件进行第一步质量控制的软件。

FastQC的安装需要Java环境，JDK安装直接用rpm就可以。

安装参考<http://www.bubuko.com/infodetail-2086082.html>

直接下载的是编译好的文件，不需要root，加到本地的环境变量里就行。

## Tophat

Tophat用于在bowtie2生成的全基因组索引的基础上对fastq进行mapping。

我安装的是tophat2

安装参考<http://blog.sina.com.cn/s/blog_751bd9440102v4lq.html>

依赖项是bowtie1/2，安装方法见上，下载的是编译好的文件，直接加进变量。

Tophat2同样，编译好的直接加进变量。

## Cufflinks

安装参考<http://blog.sina.com.cn/s/blog_751bd9440102v4lq.html>

直接下载，加进环境变量。

## Boost

Boost是tophat的一个依赖库。安装参考<http://blog.csdn.net/u011641865/article/details/73498533>

下载之后需要编译一下，不需要root

## Samtools

安装参考<http://blog.sina.com.cn/s/blog_751bd9440102v2rr.html>

Samtools是需要自己编译的，需要gcc，但是不需要root权限。编译的过程中注意路径的选取要变成自己的。

## STAR

这个和tophat类似，是RNA-seq的序列比对软件。

安装参考<http://www.bio-info-trainee.com/727.html>

是已经编译的版本，注意选取对应的操作系统加入到环境变量当中

## HISAT2

Hisat2也是序列比对软件，按照JHU的说法是tophat2和HISAT1的替代者，安装方法参考<http://www.bio-info-trainee.com/731.html> 这里面是HISAT1的，但是找到对应链接换成2别的都差不多。

不需要编译，直接解压加入环境变量

## HTSeq

Raw data counts的python库，直接pip install HTSeq就可以，注意使用的时候注意大小写

## SCDE

一个R包，需要巨多的依赖，但是考虑到是生物信息处理的R包，上bioconductor上找了一下果然有，直接安装，连带所有的依赖库就都能基本解决了

处理流程记录

## SRA文件的下载

在GEO网站找到相对应的实验，从SRA的FTP当中找到相应的SRA二进制数据下载。

在我处理的过程中，数据来源于<https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE57872> ，是人类恶性胶质瘤的单细胞RNA-seq测序结果，具体采用的是paired-end smart-seq，一共875个sample，出于计算时间的考量，我选取了其中75个sample进行验证。测序的格式是hg19。

因为它给出的数据是做了adapter trimming之后的，因此没有前面的识别头，可以直接使用。

## SRA文件的转换

输入文件：从GEO上下载的.sra序列文件

输出文件：两个fastq文件，分别命名为XXX\_1.fastq和XXX\_2.fastq，代表了paired-end测序中从3‘-5’和从5‘-3’两种测序方向。

命令行：fastq-dump --split-3 filename

参数：

--split-spot: 将双端测序分为两份,但是都放在同一个文件中

--split-files: 将双端测序分为两份,放在不同的文件,但是对于一方有而一方没有的reads直接丢弃

--split-3 : 将双端测序分为两份,放在不同的文件,但是对于一方有而一方没有的reads会单独放在一个文件夹里

使用SRA网站给出的SRA Toolkit中的fastq-dump功能，将SRA文件转换为fastq文件。注意，如果测序是paired-end的，会生成两份fastq文件，分别是从3到5以及从5到3。

## 使用fastQC进行质量控制

输入文件：上一步生成的fastq文件XXX\_X.fastq

输出文件：一个记录了质量检测结果的html文件和包含了一些图表的zip压缩包

命令：fastqc –o –f fastq filename

重要参数：

-o --outdir FastQC生成的报告文件的储存路径，生成的报告的文件名是根据输入来定的

-t --threads 选择程序运行的线程数

-a --adapters 也是输入一个文件，文件的格式Name [Tab] Sequence，储存的是测序的adpater序列信息，如果不输入，目前版本的FastQC就按照通用引物来评估序列是否有adapter的残留

最后会生成一个压缩包和一个html页面，里面有一些质量控制的报告，之后可以在这些报告的基础上利用一些其它的软件对fastq进行filter

## Bowtie2建立参考基因组的索引

输入文件：参考基因组文件genome.fa

输出文件：bowtie2index文件夹下的6个文件分别叫XXX.1/2/3/4/rev1/rev2.bt2。

命令：bowtie2-build genome.fa bowtie2index/genome>>bowtie2.log &

其中genome.fa是参考基因组的fa文件，生成的索引在bowtie2index文件夹下，会生成几个bt2格式的文件，生成索引之后记得要把fa文件和索引放在一起，才能进行下一步

这里面，根据原数据论文中的说法，他们使用了UCSC hg19的转录本数据，我则使用了UCSC hg19的全基因组数据，从结果上看差别不大

几点注意：

1. Bowtie2index后面的那个genome代表生成文件的前缀；
2. 整个运行过程取决于你使用的索引fa大小（全基因组/部分转录本），一般要一两个小时，运行一次之后就可以一直用，也可以直接上网找到别人算好的index。

## Tophat2进行mapping

输入文件：参考基因组对应的gtf文件；上一步生成的参考基因组索引；待mapping的fastq数据

输出文件：在XXX\_thout文件夹中的文件，包括记录了mapping结果的accepted\_hits.bam；记录了mapping情况的日志align\_summary.txt以及一些其他的文件和日志（主要是一些处理中遇到问题的序列等等）

命令：tophat -p 8 -G genome.gtf -o SRR1274192\_thout bowtie2index/genome SRR1274192\_1.fastq SRR1274192\_2.fastq

参数：

-p ：指定线程数，默认为1

-G ：指定已有的基因组注释信息，gtf或gff文件；

-o ：指定输出目录，默认为”./tophat\_out“；

其中-G跟的是gtf文件，是和前面的参考基因组fa文件一起下载的，最好找相同的版本

-o是输出的文件夹，后面的genome是文件前缀

差不多要两个小时跑一个细胞

接下来有两种选择：1. 接着使用cufflinks系列的软件，可以得到归一化处理的counts matrix；2. 使用HTSeq，得到raw reads count结果

## Route1：使用HTseq计算raw reads count

## Bam转换为sam

输入：tophat生成的accepted\_hits.bam文件

输出：转换格式之后的accepted\_hits.sam文件

命令：samtools view -o accepted\_hits.sam accepted\_hits.bam

这里是处理文件格式，使得HTSeq能够读取文件

## 计算基因表达量

输入：上一步中转换得到的accepted\_hits.sam文件；参考基因组的gtf注释文件

输出：记录了raw reads count的XXX.count文件

python -m HTSeq.scripts.count -s no /home/songshaoming/data/HCA/SRR1274192\_thout/accepted\_hits.sam gencode.v27.chr\_patch\_hapl\_scaff.basic.annotation.gtf > SRR1274192.count

使用HTSeq，输入sam文件和gtf文件，最终生成raw reads count

一些参数：

-f | --format default: sam

设置输入文件的格式，该参数的值可以是sam或bam。

-r | --order default: name

设置sam或bam文件的排序方式，该参数的值可以是name或pos。前者表示按read名进行排序，后者表示按比对的参考基因组位置进行排序。

-s | --stranded default: yes

设置是否是链特异性测序。

-a | --a default: 10

忽略比对质量低于此值的比对结果。

-t | --type default: exon

程序会对该指定的feature（gtf/gff文件第三列）进行表达量计算，而gtf/gff文件中其它的feature都会被忽略。

-i | --idattr default: gene\_id

设置feature ID是由gtf/gff文件第9列那个标签决定的；若gtf/gff文件多行具有相同的feature ID，则它们来自同一个feature，程序会计算这些features的表达量之和赋给相应的feature ID。

-m | --mode default: union

设置表达量计算模式。该参数的值可以有union, intersection-strict and intersection-nonempty。

-o | --samout

输出一个sam文件，该sam文件的比对结果中多了一个XF标签，表示该read比对到了某个feature上。

## Route2：使用Cufflinks系列软件

使用的软件包括cuffmerge、cuffquant和cuffnorm。

具体操作参照<http://blog.sina.com.cn/s/blog_711ea0600102vn5p.html> 以及<http://blog.csdn.net/hugolee123/article/details/44814469>

主要特点是整合了所有的细胞数据，能够进行细胞间的归一化处理计算

其中的一些参考代码：

## Cufflinks组装结果

输入文件：参考基因组的gtf文件；tophat生成的accepted\_hits.bam

输出文件：XXX\_clout当中的文件，有记录了组装结果的transcripts.gtf；有记录了转录本fpkm计算结果的isoforms.fpkm\_tracking；以及记录了基因fpkm结果的genes.fpkm\_tracking

命令：cufflinks -p 8 -G genome.gtf -o SRR1274192\_clout SRR1274192\_thout/accepted\_hits.bam

这里是最基本的组装结果，需要用到gtf文件和tophat生成的bam文件

参数说明：

-p ：指定线程数；

-o ：指定输出文件所在目录；

## 转录本合并——Cuffmerge

输入文件：参考基因组的gtf文件，参考基因组的fa文件，以及所有实验整合到一起的assemblies.txt文件（具体格式见下）

输出文件：merged\_asm文件夹下记录了合并结果的merged.gtf

命令：cuffmerge -g /home/songshaoming/data/HCA/genome.gtf -s /home/songshaoming/data/HCA/bowtie2index/genome.fa -p 8 /home/songshaoming/data/HCA/assemblies.txt

参数说明：-g ：参考基因组注释文件

-s ：参考基因组序列文件

-p ：指定线程数

-o ：指定输出文件merged.gtf所在目录，默认情况下是 merged\_asm

最后assemblies.txt ：一个包含每个样品（重复）拼接后的gtf文件的列表；如下：两个文件分别是在上一步中生成的样品的转录本注释文件。

./s0924fb\_clout/transcripts.gtf

./sCal27\_clout/transcripts.gtf

## 基因和转录本表达定量——cuffquant

输入文件：参考基因组的gtf文件，tophat生成的accepted\_hits.bam文件

输出文件：XXX\_quant文件夹中的abundances.cxb文件，记载了定量的结果

命令：cuffquant -o SRR12952$num"\_quant" -p 8 -u /home/songshaoming/data/HCA/genome.gtf /home/songshaoming/data/HCA/SRR12952$num"\_thout/accepted\_hits.bam"

参数说明：-o ：指定结果输出目录：包含结果文件abundances.cxb

-p ：指定线程数

-u ：指定对比对上基因组上多个位置的reads进行统计分析。

加上参考基因组注释文件：AT.gff

最后加上Tophat2产生的该样本的比对结果文件：accepted\_hits.bam

## 基因和转录本表达水平标准化——cuffnorm

输入文件：参考基因组的gtf，所有实验进行过quant之后生成的abundances.cxb文件

输出文件：cuffnorm文件夹中的记录基因表达水平的\*.fpkm\_table文件和记录基因fragment count values的\*.count\_table文件以及记录per-replicate expression and count data的\*.read\_table文件

命令：cuffnorm -o cuffnorm\_out -p 8 -L SRR1295210,SRR1295211... /home/songshaoming/data/HCA/genome.gtf /home/songshaoming/data/HCA/SRR1295210\_quant/abundances.cxb,/home/songshaoming/data/HCA/SRR1295211\_quant/abundances.cxb,...

2）参数说明：-o ：指定结果输出目录

-p ：指定线程数

-L ：为每个样本（处理）作标记

–total-hits-norm ：计算所有的fragments，包括与所有的参考转录本不容的，默认不激活。

–compatible-hits-norm ：只计算与一些参考转录本相容的fragments,默认激活。

加上参考基因组注释文件，最后加上每个样本（处理）的cuffquant产生的abundances.cxb文件，样本的每个重复之间用逗号”，“分割；样本之间则由空格分割。

## 转录本差异表达分析——Cuffdiff：分析差异表达基因的工具。

输入文件：cuffmerge生成的整合的gtf文件；参考基因组的fa文件；所有实验经过tophat生成的accepted\_hits.bam文件

输出文件：除去cuffnorm生成的文件之外，还多了Differential expression test，用于记录样本之间的表达差异。

命令：cuffdiff -o diff\_out -b bowtie2index/genome.fa -p 8 -L SRR1295210,SRR1295211... -u /home/songshaoming/data/HCA/merged\_asm/merged.gtf /home/songshaoming/data/HCA/SRR1295210\_thout/accepted\_hits.bam，/home/songshaoming/data/HCA/SRR1295211\_thout/accepted\_hits.bam,...

参数说明：-o ：指定输出目录

-b ：参考基因组序列文件

-p ：指定线程数

-L ：为每个样本标上名称 -u：-u命令指cuffdiff对回帖的基因组中多个位置的read进行一个初步的估计，然后加权分配到各个基因组位置。而不是简单的平均分配，其功能与Cufflinks中的u命令相同。

加上合并后的转录本：merged.gtf;由cuffmerge产生。

最后是TopHat产生的样本的bam文件，如果一个样本有多个生物学重复，那么我们需要提供每个重复的bam文件，文件名之间以逗号隔开并且样本名应与-L参数相对应。

## 后端

使用的数据：

/home/songshaoming/data/HCA/GSE75140\_hOrg.fetal.master.data.frame.txt

具体操作参照<http://hms-dbmi.github.io/scw/heterogeneity.html>

介绍一下其中基本的几个操作

首先在处理数据方面，对于低表达的基因和低表达的细胞都进行处理，过滤掉；其中低表达基因有两种：在所有细胞中表达的总reads数太少 or 在所有细胞中有表达的细胞数太少，二者都要进行处理。

PCA。PCA的作用在于降维可视化，尤其是当细胞系分为明显有区别的两类时，效果尤为明显。

tSNE。同样也是用来做降维的，要注意dist(mat)这一操作在矩阵很大的时候要花费大量的时间。

除此之外就是在linux命令行下，输出图像要用png函数和dev.off()

Pathway和基因集的不均衡分布分析，这里要用到scde包

其中pagoda函数可以用来找到基因聚类，识别出亚组，从而进行细胞通路以及GO的富集分析。

同时使用scde还能进行表达的差异分析，以及使用monocle进行伪时序的分析