Description

The `RNAseq_pipeline.sh` script integrates the common steps of RNA-seq analysis :

1. Trimming and qc reports -- `trim_galore` and `fastqc`
2. Mapping -- `STAR`
3. sam2bam, sort and index -- `samtools`
4. gene expression quantification -- `featureCounts`
5. synthetic spike-ins mapping and quantification (optional) -- `STAR` and `featureCounts`

Dependences

```
python #v3.7
fastqc #v0.11.9
trim_galore #v0.6.6
STAR #2.7.6a
samtools #1.11
featureCounts #2.0.1
```

Also, **STAR index** and **gene annotation** of your interest organism or reference sequence are required.

- STAR index can be found in `/public/Reference/<organism>/index/star_index/`
- Gene annotation can be found in `/public/Reference/<organism>/annotation/`

注意，STAR index和gene annotation要选择版本匹配的，例如 `GRCm38` 的index对应 `gencode.vM23.annotation.gtf`.

Prerequisite

首先，在你的创建管理本次项目的目录，例如`RNA-seq_20210923`

```
mkdir RNA-seq_20210923 # 创建目录
cd RNA-seq_20210923 # 切换到该目录
```

再依照下面的结构创建以下子目录

```
RNA-seq_20210923/
|-- data
|   -- fastq
|       `*fastq.gz`
|-- results
|-- src
```

```
mkdir -p data/fastq
mkdir results
mkdir src
```

`data/fastq/` 放你的测序fastq文件

`results` 保存结果

`src` 保存要用的脚本, e.g., `RNAseq_pipeline.sh`

再将RNA-seq的分析脚本复制到src目录

```
cp /public/publicUse/script/RNA-seq_pipeline/v2.0/RNAseq_pipeline.sh src/
```

然后，把你所需要分析的测序数据拷贝到项目目录里的`data/fastq/`， 例如：

```
cp /public/DATA/Nova/20210628-WXT YYN
Project_s740g01014_10Samples_20210627_1624774324/Sample_R21071207-WY-W1/*gz
~/RNA-seq_20210923/data/fastq
```

这里可以使用通配符识别多个文件批量复制。还要注意自己当前所在的路径，以上只是示例代码

```
# 批量复制示例
cp -r /public/DATA/Nova/20210628-WXT YYN
Project_s740g01014_10Samples_20210627_1624774324/S*/*gz ~/RNA-
seq_20210923/data/fastq
```

另外，还需要准备一个样本信息表格（**samplesheet.csv**），例如：

sample	fastq_1	fastq_2
NC1	SQ23003392-KU-NC1_combined_R1.fastq.gz	SQ23003392-KU-NC1_combined_R2.fastq.gz
NC4	SQ23003392-KU-NC4_combined_R1.fastq.gz	SQ23003392-KU-NC4_combined_R2.fastq.gz

其中，

`sample`: 为样本的编号，最好采取英文字母和数字的组合。

`fastq_1`, `fastq_2`:分别为两端测序文件的名字，这里不需要输入文件所在路径，只需要名字即可。如果是单端测序，只需要输入`fastq_1`下的内容即可，但是`fastq_2`的表头要保留下来。

上述步骤准备好后，你的项目目录应该是这样的：

```
RNA-seq_20210923/
├── data
│   └── fastq
│       ├── S1_R1.fastq.gz
│       ├── S1_R2.fastq.gz
│       ├── S2_R1.fastq.gz
│       ├── S2_R2.fastq.gz
│       ├── ...
│       └── S20_R2.fastq.gz
├── results
└── src
 ├── samplesheet.csv
 └── RNAseq_pipeline.sh
```

Usage

```
bash RNAseq_pipeline.sh [--pair] \
    -d <project_dir> \
    -i <input_csv> \
    --ref <reference_genome> \
    --gtf <GTF_file> \
    -t <threads> \
    --syn <1 or 2>
```

`-d`: 你的项目目录的（绝对）路径，例如 `~/RNA-seq_20210923`

`-i`: 待分析的样本信息csv文件

`--ref`: 使用的参考基因组index的路径

`--gtf`: 所用的基因注释文件 (GTF)的位置

`--pair`: FLAG, 添加这个参数代表分析的是双端测序数据

`-t`: 使用的线程数，在计算节点上 (`ssh node1/node2`)一般用32，最多不要超过60

`--syn`: 使用这个参数将对synthetic spike-ins进行分析，1代表分析只添加两种spike-ins（NAD，m7G-RNA序列）的情况，2代表添加两种spike-ins以上的情况。不使用这个参数将不进行synthetic spike-ins分析

详细使用方法见以下例子

Example

假设你要对以下这两个样品进行分析：

sample	fastq_1	fastq_2
NC1	SQ23003392-KU-NC1_combined_R1.fastq.gz	SQ23003392-KU-NC1_combined_R2.fastq.gz
NC4	SQ23003392-KU-NC4_combined_R1.fastq.gz	SQ23003392-KU-NC4_combined_R2.fastq.gz

```
$ ll ~/LNlab_project/test/data/fastq/
total 1.9G
-rw-rw-r-- 1 lidean lidean 456M Jun  3 17:16 SQ23003392-KU-NC1_combined_R1.fastq.gz
-rw-rw-r-- 1 lidean lidean 475M Jun  3 17:16 SQ23003392-KU-NC1_combined_R2.fastq.gz
-rw-rw-r-- 1 lidean lidean 472M Jun  3 17:16 SQ23003392-KU-NC4_combined_R1.fastq.gz
-rw-rw-r-- 1 lidean lidean 492M Jun  3 17:17 SQ23003392-KU-NC4_combined_R2.fastq.gz
```

在分析之前确认你已经进入计算节点，并激活了`RNAseq_py3`的`conda`环境

```
ssh node1 # or node2
conda activate RNAseq_py3
```

接着，切换到脚本所在的目录，例如`cd ~/LNlab_project/test/src/`，再执行以下代码:

运行以下代码时，务必确保你的目录包括了`RNAseq_pipeline.sh`

```
nohup bash RNAseq_pipeline.sh --pair \
-d ~/LNlab_project/test \
-i ~/LNlab_project/test/src/samplesheet.csv \
--ref /public/Reference/human/index/star_index/GRCh38_primary_assembly/ \
--gtf /public/Reference/human/annotation/Homo_sapiens.GRCh38.94.chr.gtf \
-t 32 --syn 1 >nohup1.out 2>&1 &
```

--syn 1 只有在你加入两种synthetic spike-ins的时候才使用这个参数；如果加了超过两种，就用`--syn 2`；如果没加或不需要分析synthetic spike-ins则不使用这个参数。

Messages from the program will be directed to file `nohup1.out`, you can use `cat nohup1.out` or `tail nohup1.out` to check.

Output

1. The trimmed `fastqs` will be output in `<input_data_dir>/clean`

```
data/clean/
|-- *.fq.gz
...
```

2. The QC results of trimmed reads are in `<output_dir>/QC/`

```
results/QC/
|-- *_R1_clean_fastqc.html
|-- *_R1_clean_fastqc.zip
...
```

results/QC/ 下有multiqc的汇总报告`multiqc_report.html`下载这个看**测序质量**

3. The mapping results in `<output_dir>/align/`

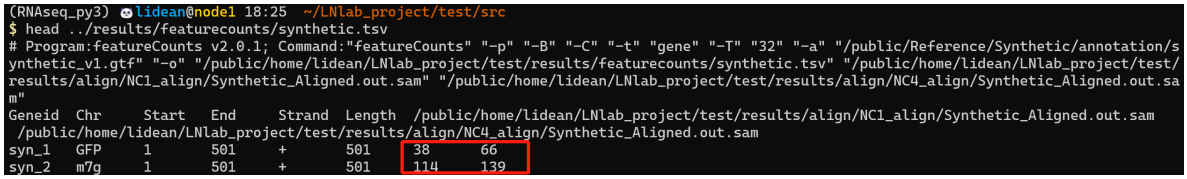
```
results/align/
|-- *_align
|   |-- Aligned.out.sam
|   |-- Log.final.out
|   |-- Log.out
|   |-- Log.progress.out
|   |-- *.sorted.bam
|   |-- *.sorted.bam.bai
|   -- SJ.out.tab
...
```

results/align/ 下有multiqc的汇总报告`multiqc_report.html`下载这个看**比对情况**

4. The gene expression quantification results are in `<output_dir>/featurecounts/`

```
  results/featurecounts/
|-- Counts.csv
|-- *_counts.txt
|`-- *_counts.txt.summary
...
```

如果进行了synthetic spike-ins分析，你应该还会在`results/featurecounts/`找到`synthetic.tsv`文件，其中包含了synthetic RNA的counts，例如：



其中，红框内的才是synthetic RNA的counts，其余都是注释信息。注意这个文件是未经过排序的，你可以手动将它与`Counts.csv`按样本顺序进行合并。

分析完成后，分别下载:

- `results/QC/multiqc_report.html`到本地电脑，可命名为`reads_multiqc_report.html`;
- `results/align/multiqc_report.html`到本地电脑，可命名为`align_multiqc_report.html`;
- `results/featurecounts/Counts.csv`（和`results/featurecounts/synthetic.tsv`）到本地电脑，可放置于本地项目目录的`project_dir/data/`下进行后续分析

- **Version:** 2.0
- **Date:** 2023-06-03
- **Contributors:** Dean Li