SDAT User guide

INSTALL:

You can use: (1) pip install sdat   (2) python setup.py install to install the package.

Usage:

"sdat filter"   #filter the correct reads and rename the reads with cell_id and umi

options:

-b1   barcode1   # Round1 barcode,please input as: 1-8,which means Round1_01-08 and Round1_49-56. default:1-48.

-b2   barcode2   # Round2 barcode,please input as: 1-8,which means Round2_01-08. default:1-96

-b3   barcode3   # Round3 barcode,please input as: 1-8,which means Round3_01-08. default:1-96

-e   0 or 1   # allowed 0 or 1 bp mismatch per barcode. default:1

-nm   # means not mix oligo dT with hexamer. when using '-um', the reads from Round1-48, Round49-96 will be named with a suffix '-0' or '-1' respectively

example:

sdat filter   [options]   -R1 xxxx_R1.fq(.gz)   -R2   xxxx_R2.fq(.gz)

Some details:

if you mix oligo-dT and hexmer in round1 for a sample, which is recommended in the standard protocol, then the reads that have the same barcode2, barcode3 and corresponding oligo-dT and hexamer barcode1 is from same cell. And I will rename the reads with same cell id. For example: the following two reads are from the same cell:

Read1	Round3_01 (barcode3_1)	Round2_01 (barcode2_1)	Round1_01 (barcode1_1)	Cell_1
Read2	Round3_01 (barcode3_1)	Round2_01 (barcode2_1)	Round1_49 (barcode1_49)	Cell_1

There is a “-nm” option in sdat filter. When using this option, the upper reads will be named as:

Read1	Round3_01 (barcode3_1)	Round2_01 (barcode2_1)	Round1_01 (barcode1_1)	Cell_1-0
Read2	Round3_01 (barcode3_1)	Round2_01 (barcode2_1)	Round1_49 (barcode1_49)	Cell_1-1

So, a “-0” suffix means this reads it from oligo-dT, and “-1” means it from hexamer.

if you use Round1_01-08, Round1_49-56 in the first round, Round2_1-96 in the second round and Round3_1-96 as the third round. your command may like this:

```
sdat filter -b1 1-8    -b2 1-96    -b3 1-96    -e 1 -R1 xxxx_R1.fq(.gz)    -R2 xxxx_R2.fq(.gz)
```

if you just use Round1_01 and Round1_49 in the first round, Round2_1-96 in the second round and Round3_1-96 as the third round. your command may like this:

```
sdat filter -b1 1-1    -b2 1-96    -b3 1-96    -e 1 -R1 xxxx_R1.fq(.gz)    -R2 xxxx_R2.fq(.gz)
```

How it work:
First, sdat filter will build a barcode index. It’s in a folder named “barcode” under your input fastq file folder.
Then, sdat filter will split fastq to a series of tmp fastq files and filter the phased reads parallelly for great time saving.
Finally, sdat filter will merge the filtered tmp fastq files and logs file.
The filter fastq file will be named as xxxx_filtered.fq in your input fastq folder.
Also, there will be xxxx_barcode1.data, xxxx_barcode2.data, xxxx_barcode3.data as barcode log file; xxxx.log as overall profile log; xxxx_reads_per_cell.data as the reads number per cell log.

The reads in output fastq file(xxxx_filtered.fq) like this:

```
@Cell_82819:TCCTTCCGTC
GTATTGATGTTAACTATTAATGAGTCAGAAATATTGAAGTGAGGTTAGAGACTTTGCTAGTTGAA
A
+
/AAAAEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEAEAEEEEE
```

First line: @cell_id:umi. Umi is the head 10bp in corresponding read2
The other lines are from the corresponding read1

"sdat align"   #unique align the filtered fastq file to genome

options:

```
        -gtf     filename            # Path to your gtf file
        -o         output                   #/path/to/output/dir/prefix    default: ./xxxx (xxxx means prefix of your input fastq)
        -g                                  #without -g: only keep unique alignments, with -g: keep the unique alignments and the best alignment of multi-alignment reads(5 place)
        -m       alignIntronMax    #maximum intron size default:5000
```

example:

```
sdat align    [options]    -G index_dir    -fq xxxx_filtered.fq
```

sdat align if using STAR for reads alignment. So, if you want to use sdat align, make sure that STAR is installed in your server. And genome index has been built by STAR.
"-o" option is the same as "--outFileNamePrefix" in STAR,
"-g" option is the same as "--outFilterMultimapNmax 5 --outSAMmultNmax 1" in STAR
"-m" option is the same as "--alignIntronMax" in STAR
So, if your command like this:

```
sdat align -g -m 5000 -o output_dir/prefix   -gtf gtf_file   -G index   -fq path/to/xxxx_filtered.fq
```

your command is the same as:

```
STAR --runMode alignReads --runThreadN 16 --genomeDir index --sjdbGTFfile gtf_file --readFilesIn xxxx_filter --outFileNamePrefix output_dir/prefix --outSAMtype BAM SortedByCoordinate --outFilterMultimapNmax 5 --outSAMmultNmax 1 --alignIntronMax 5000
```

"sdat cell"   #remove PCR duplication and split aligned reads to corresponding cell

example:

```
sdat cell  -b xxxx.bam    or    sdat cell  -s xxxx.sam
```

First, sdat cell split input file to tmp file Using chromosome information
Then, sdat cell remove PCR duplication: if reads aligns to the same position of genome and reads is from same cell and have same umi(allowed 1 mismatch), this reads will be recognized as PCR duplication. sdat cell will keep the first one of these reads in xxxx_rmdup.sam, and the others will be keep in xxxx_dup.sam file.
Next, sdat cell split reads to corresponding cell file(named as Cell_1.sam,Cell_2.sam).
Finally, sdat cell merge cell file and log file:
sdat cell build a file named "cell" in your input file folder. Then, the split cell file will be merged and write to corresponding file in cell.
xxxx_rmdup.log: log file of rmdup
Reads_number_per_cell.log: reads number per cell after alignment and rmdup.

"sdat count"   #calculate gene count per cell

Sdat count is designed to get gene count from sdat cell output, before using sdat count, make sure that featureCounts is installed in your server.
And only unique alignment will be count.
options:

```
        -d          INT         # threshold,only reads number of the  cell is above threshold will be count genes number. default:200
        -gtf    filename    # Path to your gtf file
        -p          INT         # process numbers. default:10
```

example:

```
sdat count      [options]      -g Reads_number_per_cell.log    -in cell_folder
```
if your command is like this:
```
sdat count -d 200 -p 20 -gtf    gtf_file -g Reads_number_per_cell.log    -in cell_folder
```

your comman is the same as:
```
nohup featureCounts -g gene_id -T 16 -a gtf_file -o cell_id.count cell_folder/cell_id.sam >/dev/null 2>&1
```