How to analyse single cell SLAMseq data (as of 23.03.2022)

--> global MT tag version

this readme mainly applies to the content of the `scPipeline` *folder*

0. setting up:

make sure to have a conda environment (if desired, python 3.8 tested) and the following packages in working order:

```
pysam
re
pdb
sys
os
subprocess
string

pandas
numpy
```

Only `pysam` needs to be installed, the others should come with your python installation.

`pandas` and `numpy` are not strictly necessary for this particular workflow, but do appear in various analysis scripts that may or may not be used later on. They can be skipped here if desired.

Also have a vaguely recent version of `samtools` installed (via conda or otherwise, v. 1.14 tested)

1. map with cellranger similar to this (cellranger 6.1.1 tested):

```
<location/of/your/cellranger>/cellranger count
--id=<name for the output folder>
--fastqs=/data/junker/runs/<folder with your fastqs>
--transcriptome=<your transcriptome here>
--sample=<sample name from demultiplexing>
--localcores=<20 to 40>
```

Mapping with STARsolo also works, make sure to add gene identifiers using e.g. Rsubread if your run of STAR does not produce them. Supported tags are GN/GX (cellranger-style) or XT (Script in 5.1 only). Adding the MD tag can be skipped if the STARsolo run already added it.

STAR-mapped bulk data with MD tag present can be processed using Steps 4 & 6 for a bulk mutation rate. Bulk mutation rates per gene can be calculated using the scripts in `mutations_per_gene/*` based on the format of your gene tag (GN and XT are currently supported). A SNP-aware version (see 5.1 for requirements) is located in `SNP-filtering/mutation_eff_per_gene_v4_XT-tag_MTglob_SNPs_v4.py` for XT-tags only.

2. extract actual cells (as detected by cellranger) if desired. Can be skipped if cellranger is not to be trusted on your data

enter the `/outs/filtered_feature_bc_matrix/` directory and unzip `barcodes.tsv.gz`

run

```
python <script location>/1_extract_actual_cells_v2.py possorted_genome_bam.bam \
 /outs/filtered_feature_bc_matrix/barcodes.tsv N
```

3. add information about the reference genome for each mismatch using samtools:

```
samtools calmd possorted_genome_bam_actual_cells.bam <path/to/genome>/genome.fa \
 | samtools view -Sb | samtools sort > possorted_genome_bam_actual_cells_MD.bam
```

4. aggregate information on mutations in a custom tag:

```
python <script location>2_create_MT_tag_v4_global_toStdout.py possorted_genome_bam_actual_cells_MD.bam
```

catch the script's output as desired, instructions will appear automatically if the output is not piped.
This guide assumes the output of Step 4 to be called *possorted_genome_bam_actual_cells_MD_MTglob.bam*

index the output

```
samtools index possorted_genome_bam_actual_cells_MD_MTglob.bam
```

5. separate labelled and unlabelled reads:

```
python \
<script location>3_separate_labeled_unlabeled_reads_v5_2mutations_MTglob_enhancedUmiStats_PrimOnly.py \
 possorted_genome_bam_actual_cells_MD_MTglob.bam <mutation sequencing quality> <number of mutations per UMI>
```

This will generate 3 .bam files, one that has all UMIs that satisfy the 'labelled' criteria, one with entirely unlabelled UMIs and the third one with unclear cases that appear when more that 1 mutation per UMI is required in order to determine it to be labelled.

5.1

SNP-filtering:

the above script does not filter for potential SNP positions. If you want to filter some positions there is an alternative script available in `SNP-filtering/3_separate_labeled_unlabeled_reads_v5_2mutations_MTglob_enhancedUmiStats_PrimOnly_SNPs.py`
that takes an additional file with positions to be filtered as an argument. You will also need to chose the minimum coverage for a position to be considered labelled and the maximum percentage of allowed T to C events before it is considered a SNP. E.g. a position for valid labelling events needs to have a minimum coverage of 10 with no more than 25% TtoC. Positions that have less coverage or a higher fraction of TtoC will be considered unlabelled.
Output files created will be similar to point **5**, but note the TtoC fraction cutoff in the file name as well.

The SNP-file needs to be in the following format:
`<chromosome>:<position>\t<coverage>\t<number of TtoC>`
It should contain one line for each position covered by your dataset.

You can e.g. generate one using
`SNP-filtering/SNP_finding_TCperGene_plus_per_position_MTglob-based.py`
The file will end in **coverage_per_T-position_TC-Q20_SNPs.tsv*

6. calculate bulk mutation rates:

run

```
python <script location>/4_count_mutations_SE_MTglob_v6.py \
possorted_genome_bam_actual_cells_MD_MTglob.bam <mutation sequencing quality>
```

This generates one file that has total nucleotice counts (based on the reference in case of substitutions) and one that has total mutation counts. Use to plot as desired.

7. generate labelled / unlabelled count matrices
generate the count matrices directly from the .bam files based on a script based on one that was supplied with STARsolo 2.7.10a (STAR_2.7.10a/extras/scripts/soloCountMatrixFromBAM.awk)

run

```
python <script location>/5_count-matrix_from_bam_v1.py \
possorted_genome_bam_actual_cells_MD_MTglob_(un)labeled_<...> <desired matrix output folder>
```

zip all three files in `<desired matrix output folder>` by e.g. entering it and `gzip *`

The resulting matrices are in `MatrixMarket matrix coordinate integer general` format, similar to the cellranger and STARsolo output

In case of Seurat analysis: feed said `<desired matrix output folder>` to seurat as a 10x directory like this:

```
project.data <- Read10X(data.dir = "<desired matrix output folder>", gene.column=1)
```

and proceed as normal