Cellranger Multi Pipeline Integration

Jake Sauter

Project Motivations

The initial overall vision for this project is to create a workflow starting with sequencing data (a run directory) and ending with visualization of the pipeline results in Metabase.

- Cell Ranger for Immune Profiling
- Cellranger multi
  - cellranger multi can run, with one command line and an appropriate samplesheet, different types of samples such as 5'GEX, 5'VDJ, and Feature Barcoding

Some steps would include:

- creating a pipeline for running cellranger multi
  - generating a sample sheet for input to cellranger multi
  - saving stats to lims
  - standardizing output of pipeline
- creating a dashboard in metabase for data visualization and analysis

Cellranger Multi -- Summary

Analyzing V(D)J and Gene Expression / Feature Barcode with cellranger multi

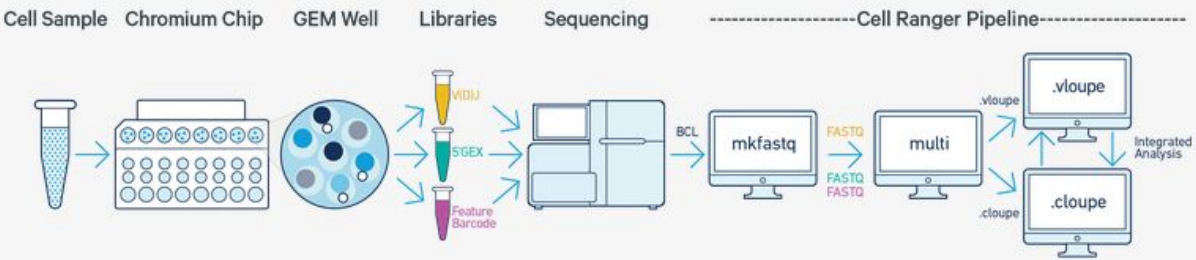
The 5' Chromium Single Cell Immune Profiling Solution with Feature Barcode technology enables simultaneous profiling of V(D)J repertoire, cell surface protein, antigen specificity and gene expression data. The `cellranger multi` pipeline enables the analysis of these multiple library types together. The advantage of using the multi pipeline (as opposed to using `cellranger vdj` and `cellranger count` separately) is that it enables more consistent cell calling between the V(D)J and gene expression data. This involves the following steps:

1. Run `cellranger mkfastq` on the Illumina BCL output folder to generate FASTQ files.
2. Run `cellranger multi` on FASTQ files produced by `cellranger mkfastq`.

Cellranger Multi -- What is it?

Analyze different assays **for the same initial library: VDJ, GEX, Feature Barcoding** with one command

Analysis of these data types can be performed using the latest versions of Cell Ranger. The following illustration depicts how the `cellranger multi` pipeline can be used to analyze FASTQ data derived from these three Immune Profiling library types from the same GEM Well:



Cellranger Multi -- What it is NOT

Initially it was easy to think that cellranger multi could perform analyses on multiple runs at the same time, however this is not the case and cellranger multi should only be configured to run with parallel assays on the **same libraries**.

This is because the heart of cellranger multi is in the filtering of vdj cell barcodes directly through comparison to called cell barcodes under the "gene expression" feature type analysis

More info on next slide

Cellranger Multi -- Intended and Recommended Use

When to use the multi pipeline

VDJ	5' GEX	5' FB	Use multi?
Yes	Yes	Yes	Recommended
Yes	Yes	No	Recommended
Yes	No	Yes	Optional. No effect on cell calling
Yes	No	No	Optional
No	No	Yes	Optional
No	Yes	No	Optional
No	Yes	Yes	Optional

"The gene expression library is representative of the entire pool of poly-adenylated mRNA transcripts captured within each partition (droplet). The TCR or BCR transcripts are then selectively amplified to create the V(D)J library. Therefore, **the gene expression library has more power to detect partitions containing cells compared to the V(D)J library. If the multi pipeline is run with both gene expression and VDJ data, then barcodes which are not called as cells by using the gene expression data will be deleted from the V(D)J cell set**."

Integrated V(D)J and Transcriptome Analysis from Single Cells

10X GENOMICS

Enrichment | Library Prep | Sequence | Pipeline | Visualization

V(D)J* ONLY (Option 1 in User Guide)

B cells
OR
T cells

→ Direct Target Enrichment → V(D)J Enriched Library Prep →  → cellranger vdj → VDJ Analysis Only  Loupe V(D)J Browser

5' Gene Expression and V(D)J* (Option 2 in User Guide)

Whole transcriptome → 5' Gene Exp Library Prep →  → cellranger count

Initial 5' Gene Exp Amp

B cells
AND/OR
T cells

→ Target Enrichment → V(D)J Enriched Library Prep →  → cellranger vdj

Integrated Transcriptome + VDJ Analysis

 Loupe Cell Browser

 Loupe V(D)J Browser

*Requires separate library prep, pipeline instances for B and T cells

0:50

Cellranger Multi Testing

- Testing on Run_id: 201028_A00814_0296_AHVKWTDMXX
- 3 Control
  - VDJ Libraries: CTRL_1-Ig, CTRL_2-Ig, CTRL_3-Ig
  - GEX libraries: CTRL_1-GEX, CTRL_2-GEX, CTRL_3-GEX
- 3 Experimental
  - VDJ Libraries: MYD88_1-Ig, MYD88_2-Ig, MYD88_3-Ig
  - GEX libraries: MYD88_1-gex, MYD88_2-gex, MYD88_3-gex

Show 25 entries Search:)28_A00814_0296_AHVKWTDMXX

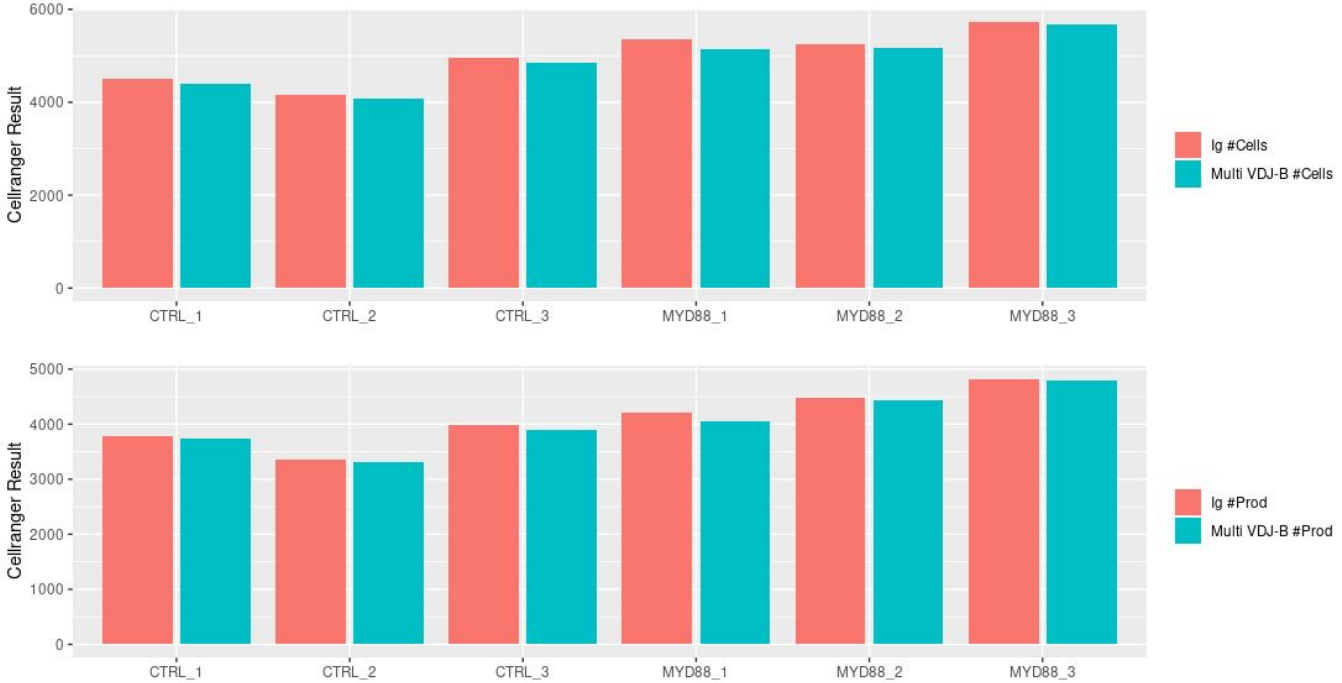
	Run	Staging Area	Status	Jobs
RR SB	201028_A00814_0296_AHVKWTDMXX	epicore09:share004	complete	transfer: 107069, cellranger_mkfastq: 107070 107071 107072 107073 107074 107075, cellranger_count: 107076 107077 107078 107079 107080 107081 107082 107083 107084 107085 107086 107087 107088 107089 107090 107091, cellranger_vdj: 107092 107093 107094 107095 107096 107097

Showing 1 to 1 of 1 entries (filtered from 1,779 total entries)

Cellranger Multi Sample Sheet for CTRL_1

[gene-expression]			
reference	/athena/epicore/ops/scratch/genomes/cellranger/3.0.0/Homo_sapiens/refdata-cellranger-GRCh38-3.0.0		
[vdj]			
reference	/athena/epicore/ops/scratch/genomes/indices/Mus_musculus/refdata-cellranger-vdj-GRCm38-alts-ensembl-5.0.0		
[feature]			
reference	/home/jns4001/piali_feature_barcoding_multi/EC-SC-6362_featureref.csv		
[libraries]			
fastq_id	fastqs	lanes	feature_types
CTRL_1-GEX	/athena/epicore/ops/scratch/analysis/store100/demux_2200422_201028_A00814_0296_AHVKWTDMXX_EC-LV-6398__uid16974/Project_EC-LV-6398/Sample_CTRL_1-GEX		gene expression
CTRL_1-lg	/athena/epicore/ops/scratch/analysis/store100/demux_2200422_201028_A00814_0296_AHVKWTDMXX_EC-LV-6398__uid16974/Project_EC-LV-6398/Sample_CTRL_1-lg		vdj

Comparison to Running VDJ + GEX Individually



Slightly fewer cells called for multi in most experiments, however these were simply cells that were not called in the gene expression "count" analysis

Validating Cellranger Multi Results

In order to determine how cellranger multi is actually filtering the VDJ experiments from the gene expression experiment data, we can observe the cellranger multi output structure and discover that **the filtered gene expression called cell barcodes and vdj called cell barcodes are easily available**

```
[jns4001@epicore08 outs]$ pwd
/scratch001/jns4001/overnight_vdj_runs/CTRL_1-Ig_Gex/outs
[jns4001@epicore08 outs]$ ls
config.csv  count  vdj_b  vdj_reference  web_summary.html
```

```
[jns4001@epicore08 outs]$ head count/filtered_
feature_bc_matrix/barcodes.tsv
AAACCTGAGCTAGTGG-1
AAACCTGAGCTGTTCA-1
AAACCTGAGGAGTTGC-1
AAACCTGAGTTTGCGT-1
AAACCTGCAATCTGCA-1
AAACCTGCAATGGAAT-1
AAACCTGCACCAGATT-1
AAACCTGCACGAAAGC-1
AAACCTGCACTCGACG-1
AAACCTGCAGTACACT-1
```

```
[jns4001@epicore08 outs]$ head vdj_b/cell_barcodes.json
[
    "AAACCTGAGGAGTTGC-1",
    "AAACCTGCAATCTGCA-1",
    "AAACCTGCAATGGAAT-1",
    "AAACCTGCACTCGACG-1",
    "AAACCTGTCAAACAAG-1",
    "AAACCTGTCCGGGTGT-1",
    "AAACCTGTCGCATGAT-1",
    "AAACGGGAGACAAAGG-1",
    "AAACGGGAGTCCATAC-1",
```

```
ctrl_1_full_barcodes <- read.csv(ctrl_1_unfiltered_count_file) %>% unlist()
ctrl_1_cell_barcodes <- read.csv(ctrl_1_count_file) %>% unlist()
ctrl_1_vdj_barcodes <- rjson::fromJSON(file=ctrl_1_vdj_file) %>% unlist()
ctrl_1_multi_vdj_barcodes <- rjson::fromJSON(file=ctrl_1_multi_vdj_file) %>% unlist()
ctrl_1_multi_cell_barcodes <- read.csv(ctrl_1_multi_count_file) %>% unlist()
```

```
cat('Number of unfiltered barcodes: ', length(ctrl_1_full_barcodes))
```

```
Number of unfiltered barcodes:  737279
```

```
cat('Number of cell-associated barcodes: ', length(ctrl_1_cell_barcodes))
```

```
Number of cell-associated barcodes:  9927
```

```
if (length(setdiff(ctrl_1_cell_barcodes, ctrl_1_full_barcodes)) == 0) {
  cat('All cell-associated barcodes found within unfiltered barcodes\n')
}
```

```
All cell-associated barcodes found within unfiltered barcodes
```

```
cat('Number of vdj called cells: ', length(ctrl_1_vdj_barcodes), '\n')
```

```
Number of vdj called cells:  4504
```

```
cat('Number of vdj called cells (with multi): ', length(ctrl_1_multi_vdj_barcodes))
```

```
Number of vdj called cells (with multi):  4408
```

- Was able to validate that ~750,000 unique 10x single cell barcodes are produced within the initial library
- Cellranger count, vdj and multi numbers all matched up with what I found in the web summary reports
- **Filtering pure VDJ analysis by only barcodes called cells in corresponding INDEPENDENT gene expression experiment yielded same results as multi**
    - This shows that multi is exactly only filtering VDJ with gex

Automation: Input -- run id or json

- Given a Run id, we can curl for the "flowcell_design.json", which stores metadata information on the sequencing that occurred for an experiment

curl -o working_flowcelldesign.json https://abc.med.cornell.edu/epilims/rest/SeqmonDatasheet?run_id=201028_A00814_0296_AHVKWTDMXX

Automation: Input -- flowcell_design.json

- Via the previous command shown, we can retrieve a json file that lists all libraries sequenced in a run
- Associated experiment data such as **Library_Name, iLab_Service_ID, Genome Build** can all be accessed via this file


```
"264282": {
  "ID": 264282,
  "Status ID": 1,
  "Status": "Published",
  "User ID": 34,
  "User": "Yushan Li",
  "Library_Made_By": "Yushan Li",
  "Date_Library_Prepared": "2020-10-15",
  "Microbiome_Sequencing_Request": null,
  "Sequencing_Request": 262870,
  "Genome_Build": "mm10",
  "Barcode_Index": "SI-GA-B1",
  "Barcode_Kit": "Chromium i7 Multiplex Kit",
  "Library_assay": "cellranger-vdj",
  "Library_Name": "CTRL_1-Ig",
  "Sample_Number": 1,
  "Library_Type": "cellranger-vdj",
  "Organism": "mouse",
  "PI": "Ari Melnick",
  "Submitter_E-mail": "lev2009@med.cornell.edu",
  "iLab_Service_ID": "EC-LV-6398",
  "Demuxware": "cellranger3.0",
  "Alignment_Requested": "Yes",
  "Data_Processing_Instructions": null
}
```

Automation: Generate CSV Sample Sheet

- Now in order to automate the process of queueing up cellranger multi runs, we must make a sample sheet (as we have seen before) for each experimental pair (as we should not mix experiments that come from different library (barcode) experimental preps)

- I have made a prototype using **Python3** in order to determine possible and appropriate design patterns for this sort of automated process

Current Working Status

- Input: Run id
- Output: Populated directory with cellranger multi sample sheets
  - References are automatically deteremed from "reference genome" json field
  - TODO: Still have not figured out best way to correlate this data with Fastq file locations

```
[jns4001@epicore08 python_generate_sample_sheet]$ ls
cellranger_config_csvs  json_to_csv_sample_sheet.py
[jns4001@epicore08 python_generate_sample_sheet]$ ls cellranger_config_csvs/
cellranger_multi_config_CTRL_1-Ig_CTRL_1-GEX.csv
cellranger_multi_config_CTRL_2-Ig_CTRL_2-GEX.csv
cellranger_multi_config_CTRL_3-Ig_CTRL_3-GEX.csv
cellranger_multi_config_MYD88_1-Ig_MYD88_1-GEX.csv
cellranger_multi_config_MYD88_2-Ig_MYD88_2-GEX.csv
cellranger_multi_config_MYD88_3-Ig_MYD88_3-GEX.csv
[jns4001@epicore08 python_generate_sample_sheet]$ cat cellranger_config_csvs/cell
ranger_multi_config_CTRL_1-Ig_CTRL_1-GEX.csv
[gene-expression],,,
reference,/athena/epicore/ops/scratch/genomes/indices/Mus_musculus,,
[vdj],,,
reference,/athena/epicore/ops/scratch/genomes/indices/Mus_musculus,,
fastq_id,fastqs,lanes,feature_types
CTRL_1-Ig,fastq_for_vdj.fastq, ,vdj
CTRL_1-GEX,fastq_for_gex.fastq, ,gene expression
[jns4001@epicore08 python_generate_sample_sheet]$
```

Seqmon Integration?

I seem to have found the handover instruction file for Seqmon

```
[jns4001@epicore08 doc]$ pwd
/home/aladdin/sequencing_monitor/current/doc
[jns4001@epicore08 doc]$ ls
files  handover.md  img
[jns4001@epicore08 doc]$ head handover.md
# Epicore Sequencing Monitor - Documentation for Handover

## Cluster overview

### Main epicore cluster

The sequencing monitor application (SeqMon) is part of the wider epicore cluster.
 It is comprised of the following machines:

```
epicore03 - SGE interactive node
[jns4001@epicore08 doc]$
```
```

Seqmon Integration?

Job templates and pipelines

Job templates contain the `qsub` scripts and the HTML files needed to show the template options. They are stored within SeqMon:

```
aladdin@epicore09 ~ $ pwd
/home/aladdin/sequencing_monitor/current/job_templates
aladdin@epicore09 ~ $ ls
bcl2fastq       c18              cellranger_mkfastq  external      transfer
bismark_bt_two  c182             demux               ipm_phase1
bwa_aligner     cellranger_count  errbs               star_aligner
```

These templates allow SeqMon to collect variables and pass them on to SGE. Those are customized for each template, see the files for details.

The templates call the analysis applications. These are installed in the aladdin user's home directory and shared across the epicore cluster:

Current Method

It seems like currently for **/home/aladdin/sequencing_monitor/current/job_templates/cellranger_count/cellranger_count.qsub**

the dataset is passed **as input** via a form (I haven't seen yet), so Fastq location was not a problem here

```
# Hello world
echo "Cellranger Count pipeline qsub script"
echo "-------------------------------------"

# Global variables
# Treat unset variables as an error
set -o nounset

# Pipeline variables
echo "run             = ${run}"
echo "datasetUID      = ${datasetUID}"
echo "datasetPath     = ${datasetPath}"
echo "analysisArea    = ${analysisArea}"
echo "refgenome       = ${refgenome}"
echo "project         = ${project}"
echo "sample          = ${sample}"
```

Next Steps?

- If implementing in Python going to have to take more than the run id
  - Some information that can be used to map samples to their dataset location / dataset uid
- Thadeous currently building out new sequencing monitor tool, should we integrate into this instead?
- Any other (potentially more useful) tools that need development?

Course Related Questions

- Defining the project goal: How will the utility I write be properly integrated?
- What sort of deliverable will be due?
    - Should I write a paper with different section about what I've learned. Or something like a manual for specifications of the program (depending on how final integration would be possible)
- Given that Thadeous said he is currently building or thinking about building a new run manager, **what does final use of this tool look like?**

Release notes for Cell Ranger 6.0.0 (March 2, 2021):

New Feature: Cell Multiplexing

1. Cell Ranger 6.0 now supports analysis of Cell Multiplexing data for the 3' Gene Expression, Targeted Gene Expression, and Feature Barcode solutions. Instructions for running the `cellranger multi` subcommand are described in the running multi page. A new Getting Started Tutorial is also available. The Cell Multiplexing algorithms include a new method to call singlets, multiplets, and empty drops. The output file structure has also changed to accomodate multiple samples multiplexed in a single GEM well.

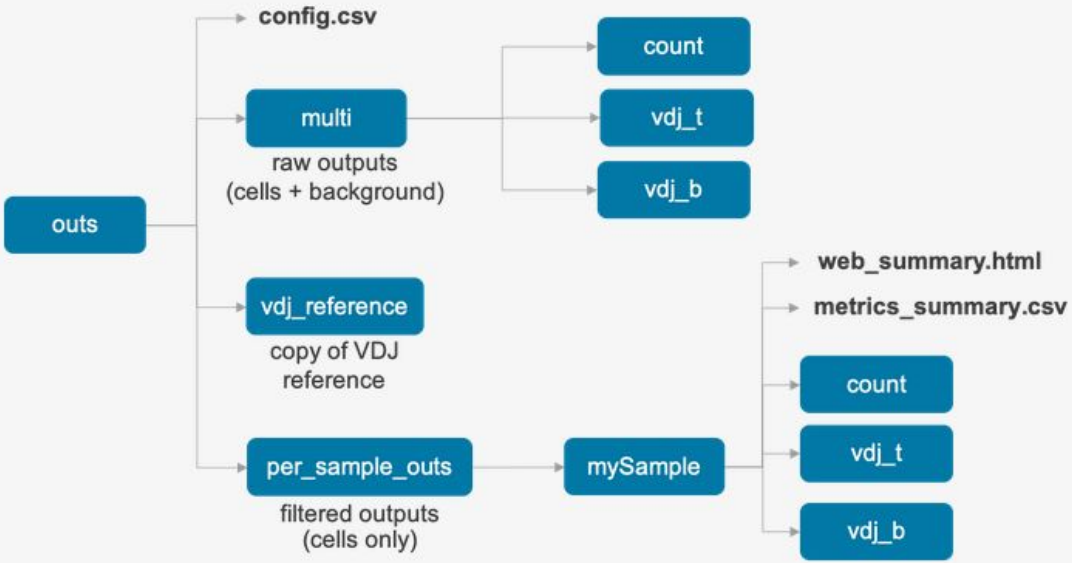
2. The `aggr` subcommand now supports analysis of `cellranger multi` outputs for the 3' Gene Expression, Targeted Gene Expression, and Feature Barcode solutions. Further details are described in the running aggr page.

Changes that apply to 5' Immune Profiling analysis

In Cell Ranger 6.0, the following changes apply to joint analysis of Immune Profiling, Gene Expression, and Feature Barcode data with the `multi` sub-command:

1. The structure of the outs folder has been updated, as described in running cellranger multi.

Upon completion, the `cellranger multi` pipeline will produce an `outs` directory with the following structure:



Current Best Link:
https://support.10xgenomics.com/single-cell-gene-expression/software/pipelines/latest/output/cellplex

3' Cell Multiplexing Outputs

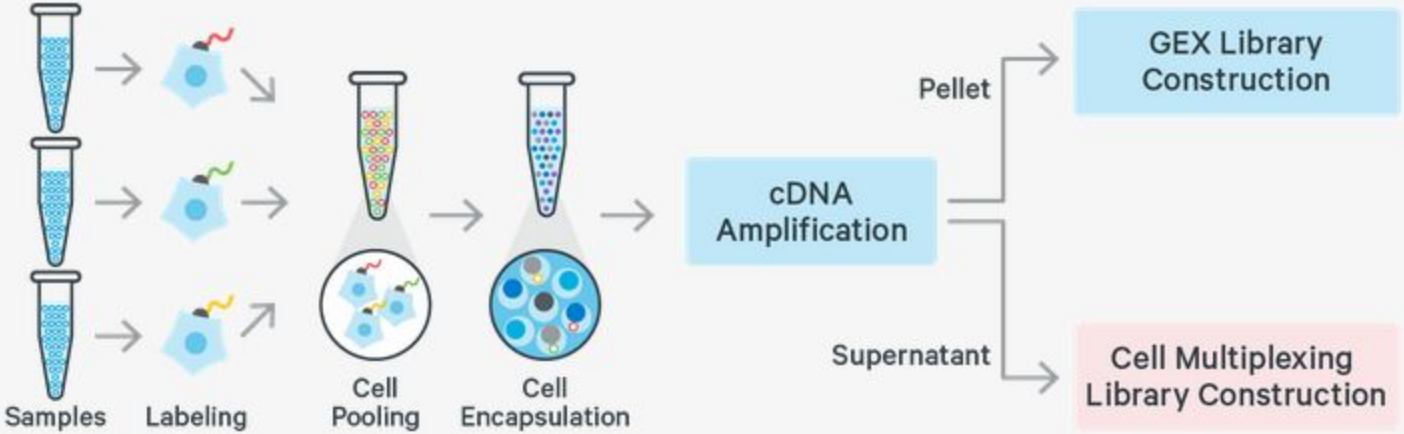
This page describes the output file structure from the `cellranger multi` subcommand specifically for **3' Cell Multiplexing** data. This subcommand was introduced in Cell Ranger 5.0 for joint analysis of 5' gene expression and VDJ (GEX + VDJ) data, and in Cell Ranger 6.0 for 3' Cell Multiplexing data.

3' Cell Multiplexing Outputs

This page describes the output file structure from the `cellranger multi` subcommand specifically for **3' Cell Multiplexing** data. This subcommand was introduced in Cell Ranger 5.0 for joint analysis of 5' gene expression and VDJ (GEX + VDJ) data, and in Cell Ranger 6.0 for 3' Cell Multiplexing data.

What is Cell Multiplexing?

Cell Multiplexing refers to the labeling of a cell or nuclei sample with a molecular tag and subsequently mixing this sample with other labeled samples. This set of multiplexed samples can be processed together in a single GEM well. After cell encapsulation, library preparation, and sequencing, molecular tag information can be assigned to cells. Tag assignment enables identification of droplets that originally contained one (singlet) or more cells (multiplets). Cells assigned a given single tag are binned together, bioinformatically recapitulating the individual samples originally mixed together.



References

https://support.10xgenomics.com/single-cell-vdj/software/pipelines/latest/what-is-cell-ranger

https://support.10xgenomics.com/single-cell-vdj/software/pipelines/latest/using/multi

https://support.10xgenomics.com/single-cell-gene-expression/software/pipelines/6.0/using/multi

https://support.10xgenomics.com/single-cell-gene-expression/software/pipelines/latest/what-is-cellplex