Introduction:

Cell Ranger provides prebuilt human and mouse reference packages for use with the pipeline, downloadable here (https://support.10xgenomics.com/single-cell-vdj/software/downloads/latest). Our reference packages are based on the T cell receptor (TRA, TRB) and B cell immunoglobin (IGH, IGL, IGK) gene annotations in Ensembl version 94 for the human and mouse references. Our reference also includes multiple corrections to various V, D, J, and C genes based on empirical observations and to correct clear errors such as frameshifts, leader peptide truncations, and nucleotides that are never observed in rearrangements. These changes are documented in the release notes of each version of Cell Ranger. Also see Prebuilt References for more information about how these references were created.

If you would like to use your own genome FASTA or gene GTF annotations, Cell Ranger supports the use of customer-generated Ensembl-based references. Cell Ranger also includes support for generating a V(D)J reference from the IMGT database.

There are two ways to generate a V(D)J reference:

Making a Genome-based Reference Package (e.g., using Ensembl)
Making a V(D)J Segment-based Reference Package (e.g., using IMGT)

Making a Genome-based Reference Package
The cellranger mkvdjref tool can be used to generate a custom reference package from a genome sequence FASTA File and a gene annotation GTF.

Download cellranger:

Under: /share/data/software/cellranger/

```
curl -o cellranger-6.1.2.tar.gz

"https://cf.10xgenomics.com/releases/cell-exp/cellranger-6.1.2.tar.gz?Expires=1648621580&Policy=eyJTdGF0ZW1lbnQiOlt7IlJlc291cmNlIjoiaHR0cHM6Ly9jZi4xMHhnZW5vbWljcy5jb20vcmVsZWFzZXMvY2VsbC1leHAvY2VsbHJhbmdlci02LjEuMi50YXIuZ3oiLCJDb25kaXRpb24iOnsiRGF0ZUxlc3NUaGFuIjp7IkFXUzpFcG9jaFRpbWUiOjE2NDg2MjE1ODB9fX1dfQ__&Signature=RAATXYUck9peRrJIRrXGAA3Oz2mrK-j-WTpkFAMflWKdDAK3GFImviaj4XqsTx64GtCVhWvfds30E-
```

RGScYMVATqnyYm7yTkcA~ObrKTe2Tb8ob6DoVuWAaUNBxnO9-2e9ql2Ka7FbvRDgJjPWTyfwpMsYHEybBWM86epljqJijmftjTU-XjakwR9NMdfYraOyYVpRci7yPTxqZZKSmh6Z2mCPVFHYlCufsfcGK4ldIO2obdTk6SKOWjvMdNWJ3D9bTrOxJv8V6TRN~92Ge4n44PzT3zUFcAU2fiZSgPr5jW4axMWlfFm0wRQnGh1-2p9lpTCfZGppIGJ-qekWvrbw__&Key-Pair-Id=APKAI7S6A5RYOXBWRPDA"

Downloading Pre-built Indexes and a sample dataset for testing cellranger

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

```
Download sample data and genome index as follows:

wget https://cf.10xgenomics.com/samples/cell-exp/3.0.0/pbmc_1k_v3/pbmc_1k_v3_fastqs.tar

tar -xvf pbmc_1k_v3_fastqs.tar

pbmc_1k_v3_fastqs/
pbmc_1k_v3_fastqs/pbmc_1k_v3_S1_L001_R2_001.fastq.gz
pbmc_1k_v3_fastqs/pbmc_1k_v3_S1_L002_I1_001.fastq.gz
pbmc_1k_v3_fastqs/pbmc_1k_v3_S1_L001_R1_001.fastq.gz
pbmc_1k_v3_fastqs/pbmc_1k_v3_S1_L002_R1_001.fastq.gz
pbmc_1k_v3_fastqs/pbmc_1k_v3_S1_L002_R2_001.fastq.gz
pbmc_1k_v3_fastqs/pbmc_1k_v3_S1_L001_I1_001.fastq.gz

wget https://cf.10xgenomics.com/supp/cell-exp/refdata-cellranger-GRCh38-3.0.0.tar.gz
tar -zxvf refdata-cellranger-GRCh38-3.0.0.tar.gz
```

Create your own genome index using cellranger:

Under : /share/data/RNA_Seq/reference_genomes/10X_genomics

https://support.10xgenomics.com/single-cell-gene-expression/software/release-notes/build#grch38_1.2.0

```
wget ftp://ftp.ensembl.org/pub/release-84/fasta/homo_sapiens/dna/Homo_sapiens.GRCh38.dna.primary_assembly.fa.gz
```

```
gunzip Homo_sapiens.GRCh38.dna.primary_assembly.fa.gz
```

```
wget ftp://ftp.ensembl.org/pub/release-84/gtf/homo_sapiens/Homo_sapiens.GRCh38.84.gtf.gz
```

```
gunzip Homo_sapiens.GRCh38.84.gtf.gz
```

```
cellranger mkgtf Homo_sapiens.GRCh38.84.gtf Homo_sapiens.GRCh38.84.filtered.gtf \
                --attribute=gene_biotype:protein_coding \
                --attribute=gene_biotype:lincRNA \
                --attribute=gene_biotype:antisense \
                --attribute=gene_biotype:IG_LV_gene \
                --attribute=gene_biotype:IG_V_gene \
                --attribute=gene_biotype:IG_V_pseudogene \
                --attribute=gene_biotype:IG_D_gene \
                --attribute=gene_biotype:IG_J_gene \
                --attribute=gene_biotype:IG_J_pseudogene \
                --attribute=gene_biotype:IG_C_gene \
                --attribute=gene_biotype:IG_C_pseudogene \
                --attribute=gene_biotype:TR_V_gene \
                --attribute=gene_biotype:TR_V_pseudogene \
                --attribute=gene_biotype:TR_D_gene \
                --attribute=gene_biotype:TR_J_gene \
                --attribute=gene_biotype:TR_J_pseudogene \
                --attribute=gene_biotype:TR_C_gene
```

Run cellranger command:

```
cellranger mkref --genome=GRCh38 \
            --fasta=Homo_sapiens.GRCh38.dna.primary_assembly.fa \
            --genes=Homo_sapiens.GRCh38.84.filtered.gtf \
            --ref-version=1.2.0
```

Output log from the above command:

```
cellranger mkref --genome=GRCh38 \
>             --fasta=Homo_sapiens.GRCh38.dna.primary_assembly.fa \
>             --genes=Homo_sapiens.GRCh38.84.filtered.gtf \
>             --ref-version=1.2.0

['/share/data/software/cellranger/cellranger-6.1.2/bin/rna/mkref', '--genome=GRCh38', '--
fasta=Homo_sapiens.GRCh38.dna.primary_assembly.fa', '--
genes=Homo_sapiens.GRCh38.84.filtered.gtf', '--ref-version=1.2.0']

Creating new reference folder at
/share/data/RNA_Seq/reference_genomes/10X_genomics/GRCh38/GRCh38
...done

Writing genome FASTA file into reference folder...
...done

Indexing genome FASTA file...
...done

Writing genes GTF file into reference folder...
...done
```

```
Generating STAR genome index (may take over 8 core hours for a 3Gb genome)...
Mar 29 15:41:13 ..... started STAR run
Mar 29 15:41:13 ... starting to generate Genome files
Mar 29 15:42:42 ... starting to sort Suffix Array. This may take a long time...
Mar 29 15:42:51 ... sorting Suffix Array chunks and saving them to disk…
```

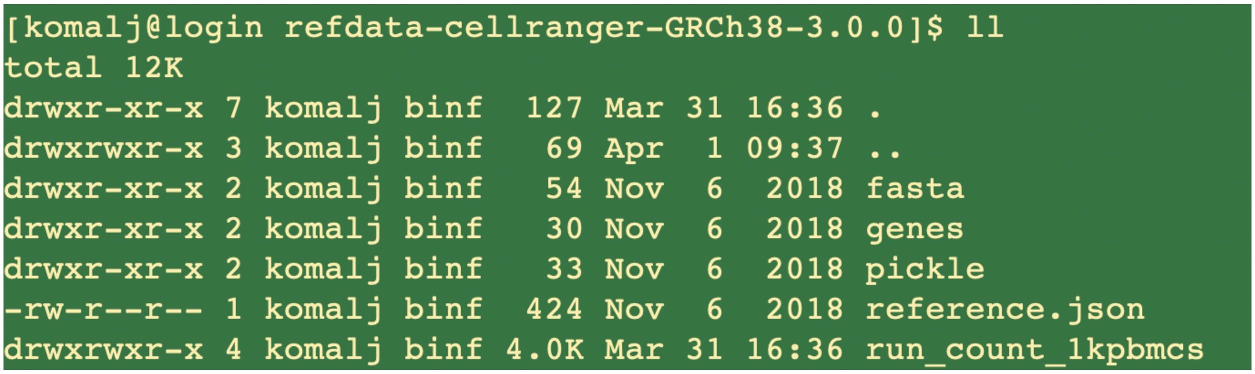
<u>Run cellranger using the test dataset:</u>

Working commands:

```
cellranger count --id=run_count_1kpbmcs --fastqs=/share/data/RNA_Seq/10X/Raw_fastq/pbmc_1k_v3_fastqs/ --sample=pbmc_1k_v3 --transcriptome=/share/data/RNA_Seq/reference_genomes/10X_genomics/human/refdata-cellranger-GRCh38-3.0.0/
```

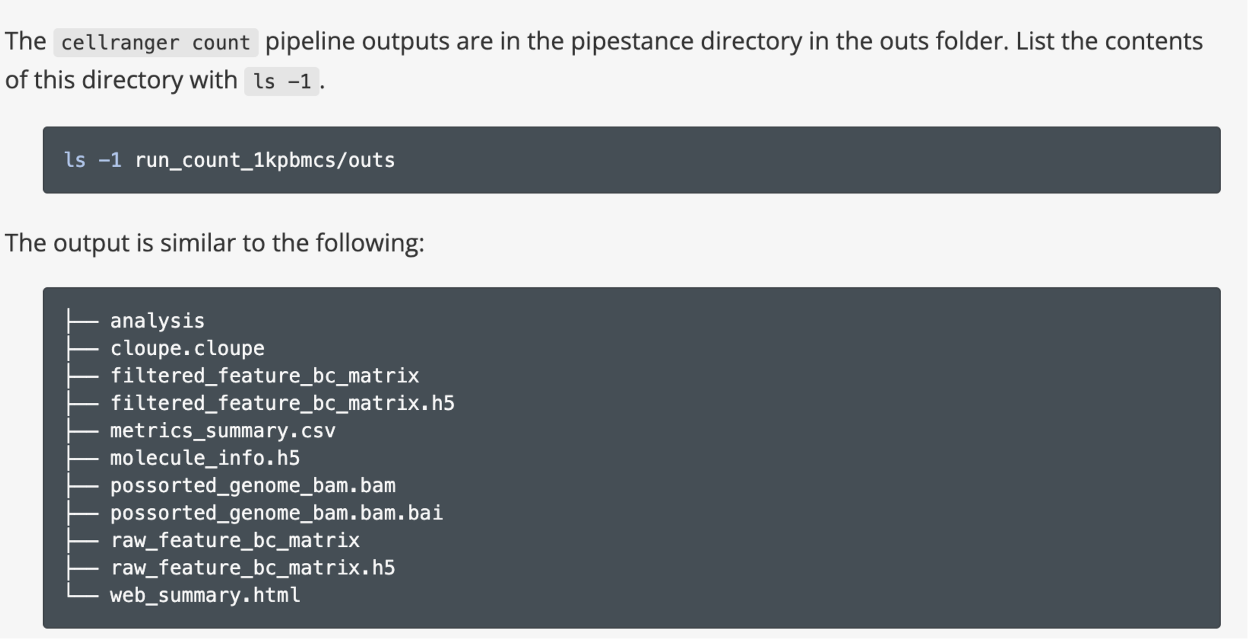
```
echo /share/data/software/cellranger/cellranger-6.1.2/bin/cellranger count --id=sample_name_5k --transcriptome=/share/data/RNA_Seq/reference_genomes/10X_genomics/human/refdata-cellranger-GRCh38-3.0.0/ --fastqs=/share/data/RNA_Seq/10X/Raw_fastq/pbmc_1k_v3_fastqs/ --expect-cells=5000 | qsub -S /bin/sh -V -N pbmc_cellranger -o /share/data/RNA_Seq/10X/Raw_fastq/pbmc_1k_v3_fastqs/ -e /share/data/RNA_Seq/10X/Raw_fastq/ -j y -cwd -l h_vmem=10G -pe smp 16
```

Id name is used to dump the output in the output directory: In this case, it is run_count_1kpbmcs. Use a meaningful name as per your project requirements.



```
[komalj@login refdata-cellranger-GRCh38-3.0.0]$ ll
total 12K
drwxr-xr-x 7 komalj binf  127 Mar 31 16:36 .
drwxrwxr-x 3 komalj binf   69 Apr  1 09:37 ..
drwxr-xr-x 2 komalj binf   54 Nov  6  2018 fasta
drwxr-xr-x 2 komalj binf   30 Nov  6  2018 genes
drwxr-xr-x 2 komalj binf   33 Nov  6  2018 pickle
-rw-r--r-- 1 komalj binf  424 Nov  6  2018 reference.json
drwxrwxr-x 4 komalj binf 4.0K Mar 31 16:36 run_count_1kpbmcs
```

Note: The command generates the output in the directory from where cellranger is run. Make sure to run cellranger in the project directory. Inside the script, cd into the directory.



The `cellranger count` pipeline outputs are in the pipestance directory in the outs folder. List the contents of this directory with `ls -1`.

```
ls -1 run_count_1kpbmcs/outs
```

The output is similar to the following:

```
├── analysis
├── cloupe.cloupe
├── filtered_feature_bc_matrix
├── filtered_feature_bc_matrix.h5
├── metrics_summary.csv
├── molecule_info.h5
├── possorted_genome_bam.bam
├── possorted_genome_bam.bam.bai
├── raw_feature_bc_matrix
├── raw_feature_bc_matrix.h5
└── web_summary.html
```

Result interpretation of cellranger count (Useful links)

https://assets.ctfassets.net/an68im79xiti/163qWiQBTVi2YLbskJphQX/e90bb82151b1cdab6d7e9b6c845e6130/CG000329_TechnicalNote_InterpretingCellRangerWebSummaryFiles_RevA.pdf

https://support.10xgenomics.com/single-cell-gene-expression/software/pipelines/latest/output/summary

https://bauercore.fas.harvard.edu/10x-chromium-system

https://kb.10xgenomics.com/hc/en-us/articles/360019665352-What-is-the-difference-between-TotalSeq-A-B-and-C-

TotalSeq data analysis:

https://www.biolegend.com/en-us/totalseq/mas

https://assets.ctfassets.net/an68im79xiti/7KPpQNrqSsAeSGYDr6Xeub/06fa19442477db836fde8928c89d5dda/CG000383_TechNote_ChromiumNextGEMSingle_Cell_3___v3.1_CellMultiplexing_Rev_A.pdf

Hashing using antibodies:

https://genomebiology.biomedcentral.com/articles/10.1186/s13059-018-1603-1