Oct 24, 2024

Human scRNA-seq Alignment Pipeline Version 1.0

DOI
https://dx.doi.org/10.17504/protocols.io.bp2l6dk7zvqe/v1

Michael Morley[1]

[1]Upenn

LungMap2 Consortium
Tech. support email: lungmap2dcc@gmail.com
Click here to message tech. support

Michael Morley
Upenn

DOI: https://dx.doi.org/10.17504/protocols.io.bp2l6dk7zvqe/v1

Protocol Citation: Michael Morley 2024. Human scRNA-seq Alignment Pipeline Version 1.0. **protocols.io** https://dx.doi.org/10.17504/protocols.io.bp2l6dk7zvqe/v1

Protocol status: Working
We use this protocol and it's working

Created: October 24, 2024

Last Modified: October 24, 2024

Protocol Integer ID: 110848

Keywords: raw sequencing reads from scrna, specific protocol for scrna, efficient alternative to 10x genomics cellranger, 10x genomics cellranger, raw sequencing read, sequencing read, human reference genome, alignment with starsolo, ambient rna, cell rna, read alignment, rna, transcript quantification, scrna, genome, gene expression per cell, gene expression, gene expression matrice, identifies cell, transcript, 10x genomics whitelist, transcript annotation, standard features like gene, achieving cellranger, gene, starsolo algorithm, containing gene expression, alignment, matrix similar to cellranger, starsolo command, cell matrix generation, starsolo tool, unique molecular identifier, cellranger, resulting gene

Abstract

The Morrisey and Basil Labs at the University of Pennsylvania use a specific protocol for scRNA-seq (single-cell RNA sequencing) alignment that involves the STARsolo tool (version 2.9.7a) to generate outputs comparable to those produced by the 10x Genomics Cell Ranger v3.0 pipeline.

Alignment with STARsolo (version 2.9.7a):
The STARsolo algorithm is used as a fast, memory-efficient alternative to 10x Genomics CellRanger.
STARsolo directly aligns raw sequencing reads from scRNA-seq data to the human reference genome.

Like CellRanger, STARsolo performs:
Barcode and UMI (Unique Molecular Identifier) extraction: Identifies cell barcodes and UMIs from the raw sequencing reads.
Read alignment: Reads are aligned to the genome using STAR.
Transcript quantification: Assigns reads to genes based on the alignment to compute gene expression per cell.
Gene-cell matrix generation: Produces a matrix similar to CellRanger's output, containing gene expression counts across cells.

Achieving CellRanger v3.0 Output:
To match the output of CellRanger v3.0, specific parameters are set within the STARsolo command, particularly:
--soloType CB_UMI_Simple: Defines the type of input for the barcodes and UMIs.
--soloCBwhitelist: Uses the 10x Genomics whitelist to ensure correct identification of valid barcodes.
--soloUMIdedup: UMI deduplication is set to handle potential PCR artifacts by counting each transcript only once per UMI.
--soloFeatures: This is set to generate gene expression matrices, capturing standard features like gene and transcript annotations.
--soloCellFilter EmptyDrops_CR
Post-Processing:
Once the alignment and quantification are completed, the resulting gene-cell matrix is further processed using SoupX to remove ambient RNAs

Align fastq files using STARsolo v2.7.9a. This version uses human genome build GRCh38 and gene annotation GENCODE v32/Ensembl98

1

```
~/bin/STAR --soloType CB_UMI_Simple \
--soloUMIlen 12 \
--soloFeatures Gene Velocyto \
--soloCellFilter EmptyDrops_CR \
--soloUMIfiltering MultiGeneUMI \
--soloCBmatchWLtype 1MM_multi_pseudocounts \
--soloCBwhitelist ~/NGSshare/3M-february-2018.txt \
--readFilesIn  ###### ADd Files ############ \
--genomeDir ~/NGSshare/GRCh38_STAR \
--runThreadN 64 \
--readFilesCommand zcat \
--soloBarcodeReadLength 0 \
--outSAMtype None \
--outFileNamePrefix STARsolo/


gzip STARsolo/Solo.out/Gene/filtered/*
gzip STARsolo/Solo.out/Gene/raw/*
```

2 Filter ambient RNAs using SoupX using the following script.

```
library(SoupX)
library(Seurat)
library(optparse)
library(magrittr)

options(future.globals.maxSize = 100000 * 1024^2)

option_list = list(
  make_option(c("-n", "--name"), type="character", default=NULL,
              help="dataset file name", metavar="character"),
  make_option(c("-i", "--input10x"), type="character",
default="outs",
              help="10x input dirs [default= %default]",
metavar="character"),
  make_option(c("-c", "--input10x_h5"), action = 'store_true',
default=FALSE,
              help="Is the the file an H5 [default= %default]"),
  make_option(c("-o", "--outdir"), type="character", default="./",
              help="outpur dir [default= %default]",
metavar="character")
)

opt_parser = OptionParser(option_list=option_list);
opt = parse_args(opt_parser);

outdir<- opt$outdir # All results will be saved here plots, RData
file
#Should only take a single argument now. Don't support H5 files
yet.
input10x<- stringr::str_split(opt$input10x,",")[[1]]

dir.create(outdir,recursive = TRUE)


# Read in count data ------------------------------------------
---------

if(opt$input10x_h5){
  print(paste0(input10x,'/filtered_feature_bc_matrix.h5'))

  filt <-
Read10X_h5(paste0(input10x,'/filtered_feature_bc_matrix.h5'))
  raw<- Read10X_h5(paste0(input10x,'/raw_feature_bc_matrix.h5'))
  filt <-  filt$`Gene Expression`
```

```
  raw <- raw$`Gene Expression`

}else{
  filt <- Read10X(paste0(input10x,'/filtered/'))
  raw<- Read10X(paste0(input10x,'/raw/'))
}

sc = SoupChannel(raw, filt)


#SoupX performs bettter with cluster info, do a basic analysis.
s <- CreateSeuratObject(counts = filt) %>%
  SCTransform(verbose = F) %>%
  RunPCA(verbose = F) %>%
  RunUMAP(dims = 1:30, verbose = F) %>%
  FindNeighbors(dims = 1:30, verbose = F) %>%
  FindClusters(verbose = T)

meta    <- s@meta.data
umap    <- s@reductions$umap@cell.embeddings
sc  <- setClusters(sc, setNames(meta$seurat_clusters,
rownames(meta)))
sc  <- setDR(sc, umap)
#Might need change forceAccept to true if get an error, should
pass this as a cmdline argument
sc  <- autoEstCont(sc,forceAccept = T)
adj.matrix  <- adjustCounts(sc, roundToInt = T)

#DropletUtils:::write10xCounts("SoupTest", adj.matrix)

 DropletUtils:::write10xCounts(
   paste0(outdir,'/Filtered.h5'),
   adj.matrix,
   type = 'HDF5',
   version = '3'
 )
```