10X Transcriptomic Data Cloud Processing

Ho Lab

2019-11-27

Contents

Chapter 1

Prerequisites

1. Amazon Web Services (AWS) Account:

Create free Amazon Web Services (AWS) Account

https://portal.aws.amazon.com/billing/signup?redirect_url=https%3A%2F%2Faws.amazon.com%2Fregistration-confirmation#/start

2. Command Line Interface (CLI):

MacOS or Linux Operating System.

For Windows Users: Install Windows Subsystem for Linux

https://docs.microsoft.com/en-us/windows/wsl/install-win10

Chapter 2

Introduction

Amazon Web Services provides cloud computing capabilities, which allows on-demand compute power, database, storage, applications, and other IT resources via the internet. This allows extremely flexibile and customisable of usage of their products depending on your specific demands/requirements.

This could be in the form of computing power, i.e. CPU cores and memory, or data storage space. All requested resources from AWS can be rescaled for your business operations, thus optimising efficiency and cost savings. Set up and usage is also extremely fast and simple, available for usage for all user background types.

In this workshop, we will exploit AWS's cloud computing service, Elastic Compute Cloud (EC2), to perform single cell 10X genomics RNA-sequencing data processing. Specifically, the mapping of raw transcript reads to an annotated human genome, which is generally a computationally demanding task, requiring more than 32GBs of RAM and numerous threads for efficient/timely processing.

10X genomics single cell RNA-sequencing (scRNA-seq) technology is becoming the most predominant type of scRNA-seq performed due to its high sequencing depth and library preparation technique to capture UMI/cell barcodes. This technology has enabled sequencing on the scale of thousands to millions of individual cells, which generates raw data files much larger than previous bulk sequencing experiements. For this reason, the average local computer generally does not hold enough computing power to perform analysis on this big data.

```
**Quiz**
1. What are UMIs and cell barcodes? and why are they beneficial?
2. What format are raw transcript reads stored as?
(a) Fasta (b) Fastq (c) Fastx (d) BAM (e) SAM
3. Estimate the file size of
(a) Raw transcript file
```

```
(b) Aligned reads file (binary compressed format)
(c) Feature count matrix
```

Chapter 3

AWS EC2 Instance

3.1 Introduction

An AWS EC2 instance is equivalent to a portable, customisable, intangible computer which is accessible through your personal local computer with a command line interface. We are able to specify memory requirements, CPU power (computing cores/threads), data storage size, and the operation system (e.g. Linux, Ubuntu, Windows) of this intangible computer.

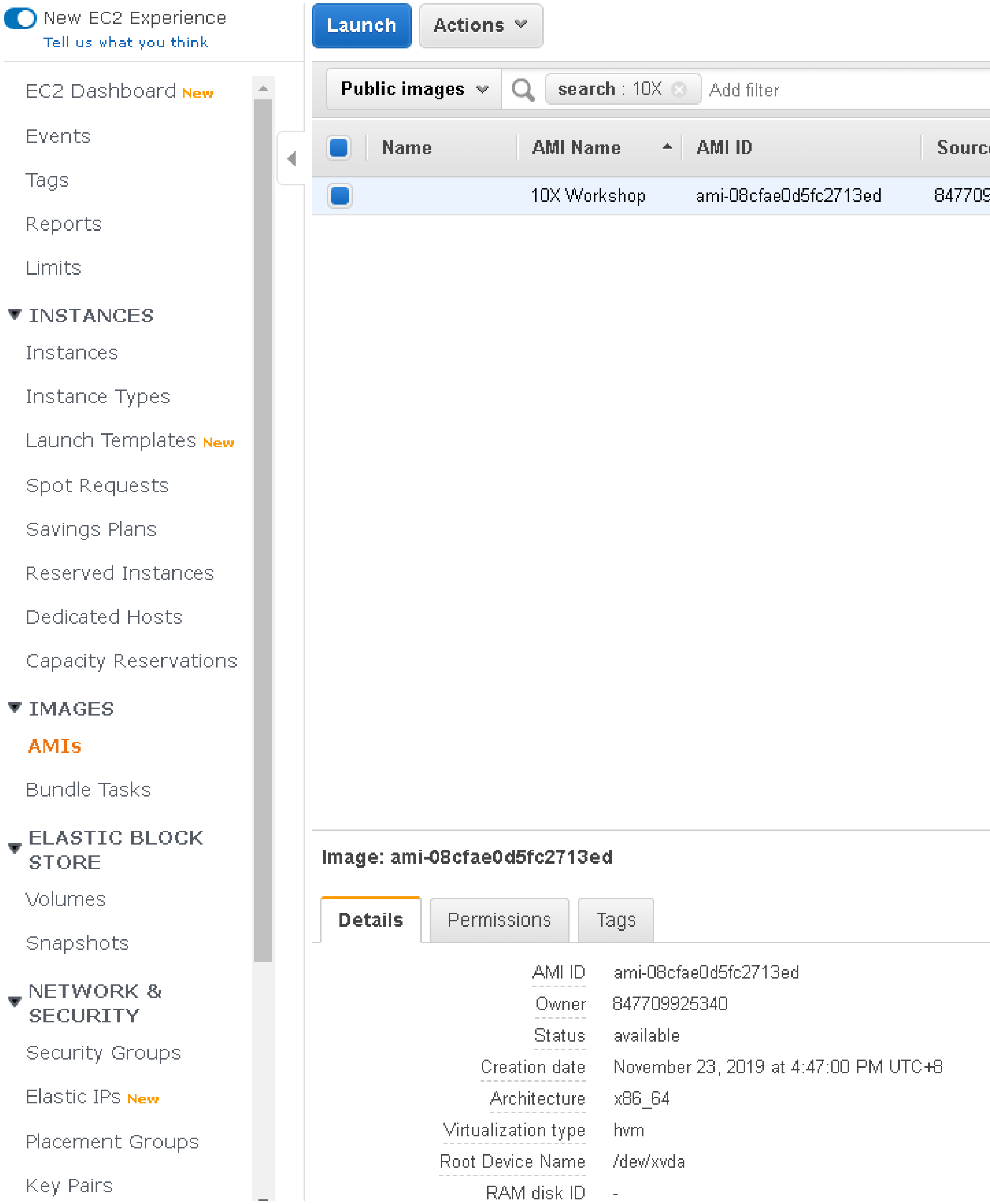
3.2 Set Up

1. Log into your AWS Account.

2. Navigate to EC2 under Amazon's Services tab.

3. Select AMIs under IMAGES tab.

4. Change filter bar to “Public Images” and search for “10X Workshop” and launch.



☐	Name	AMI Name	AMI ID	Sourc
☐		10X Workshop	ami-08cfae0d5fc2713ed	847709

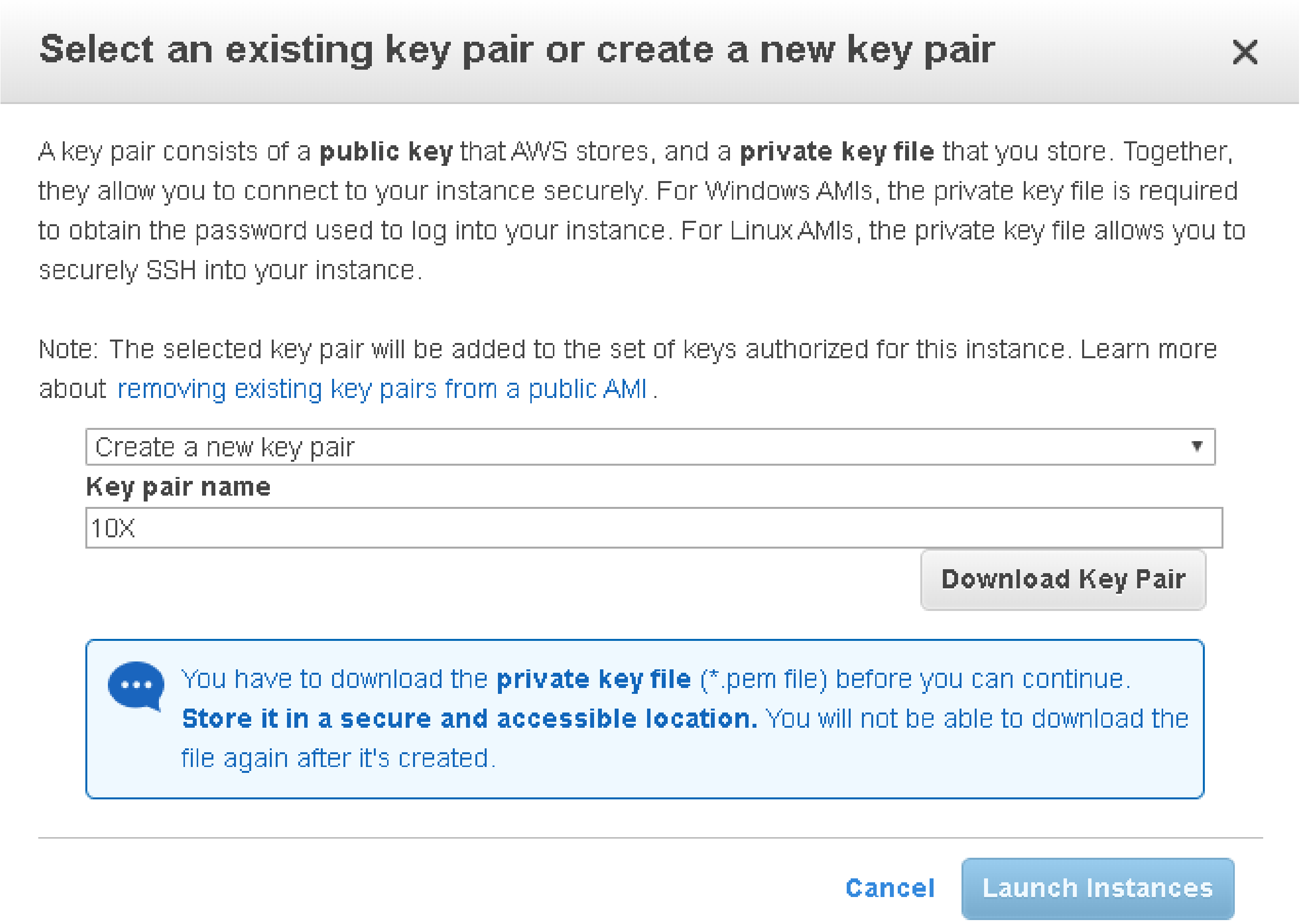
Image: ami-08cfae0d5fc2713ed

AMI ID	ami-08cfae0d5fc2713ed
Owner	847709925340
Status	available
Creation date	November 23, 2019 at 4:47:00 PM UTC+8
Architecture	x86_64
Virtualization type	hvm
Root Device Name	/dev/xvda
RAM disk ID	-

4. Launch specfication to use: `m5.4xlarge` instance type and `64GBs` of storage space. (Leave other options to default)

5. Click Launch.

6. When given the “Key pair” prompt: select “Create a new key pair”. Name your key pair “10X” and download. Then “Launch Instances”



A key pair consists of a **public key** that AWS stores, and a **private key file** that you store. Together, they allow you to connect to your instance securely. For Windows AMIs, the private key file is required to obtain the password used to log into your instance. For Linux AMIs, the private key file allows you to securely SSH into your instance.

Note: The selected key pair will be added to the set of keys authorized for this instance. Learn more about removing existing key pairs from a public AMI .

Create a new key pair ▾

Key pair name

10X

You have to download the **private key file** (*.pem file) before you can continue. **Store it in a secure and accessible location.** You will not be able to download the file again after it's created.

```
**Quiz**
1. What is an AMI?
2. What is the purpose of a key pair?
```

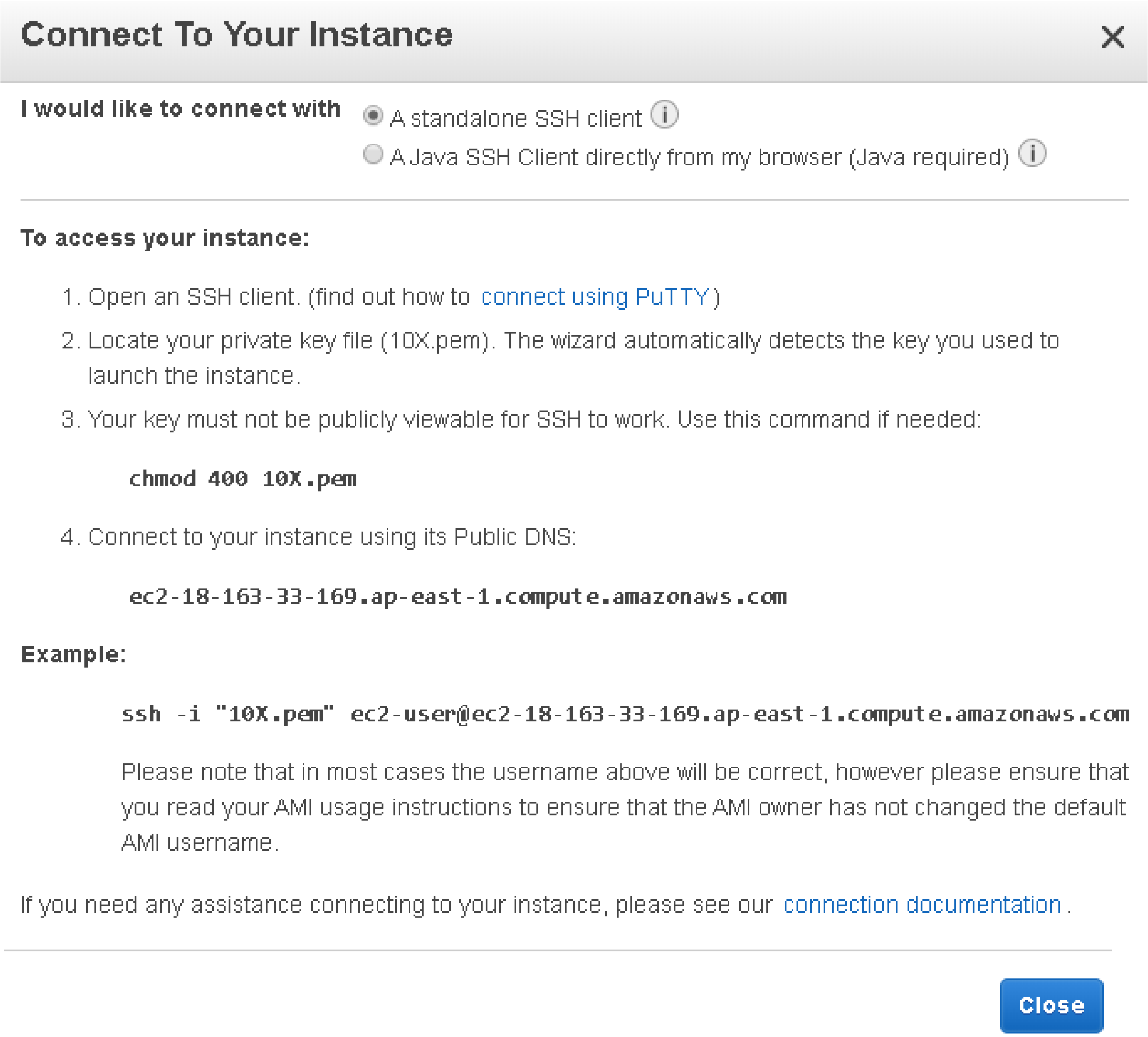
3.3 Connecting to your EC2 instance

1. Open your command line interface

2. Connect to your EC2 instance by entering the following command.

```
ssh -i "10X.pem" ec2-user@ec2-XX-XXX-XX-XXX.ap-east-1.compute.amazonaws.com
```

“10X.pem” should be the location of where you have stored your key pair. The X’s is the ip address of your personal EC2 instance. This can be found by going to EC2 Dashboard > Running Instances > Selecting your running instance > Click Connect.



I would like to connect with ● A standalone SSH client ○ A Java SSH Client directly from my browser (Java required)

To access your instance:

1. Open an SSH client. (find out how to connect using PuTTY)
2. Locate your private key file (10X.pem). The wizard automatically detects the key you used to launch the instance.
3. Your key must not be publicly viewable for SSH to work. Use this command if needed:

```
chmod 400 10X.pem
```

4. Connect to your instance using its Public DNS:

```
ec2-18-163-33-169.ap-east-1.compute.amazonaws.com
```

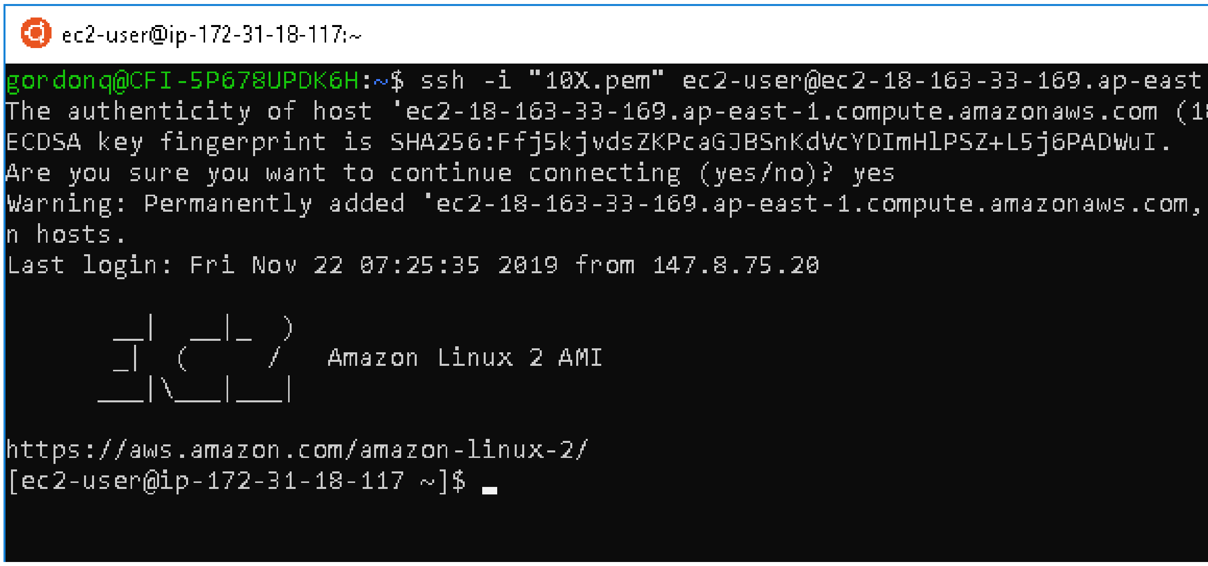
Example:

```
ssh -i "10X.pem" ec2-user@ec2-18-163-33-169.ap-east-1.compute.amazonaws.com
```

Please note that in most cases the username above will be correct, however please ensure that you read your AMI usage instructions to ensure that the AMI owner has not changed the default AMI username.

If you need any assistance connecting to your instance, please see our connection documentation .

3. Type “yes” if prompted.



ec2-user@ip-172-31-18-117:~

```
gordonq@CFI-5P678UPDK6H:~$ ssh -i "10X.pem" ec2-user@ec2-18-163-33-169.ap-east
The authenticity of host 'ec2-18-163-33-169.ap-east-1.compute.amazonaws.com (1
ECDSA key fingerprint is SHA256:Ffj5kjvdsZKPcaGJBSnKdVcYDImHlPSZ+L5j6PADWuI.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'ec2-18-163-33-169.ap-east-1.compute.amazonaws.com,
n hosts.
Last login: Fri Nov 22 07:25:35 2019 from 147.8.75.20

       __|  __|_  )
       _|  (     /   Amazon Linux 2 AMI
      ___|\___|___|

https://aws.amazon.com/amazon-linux-2/
[ec2-user@ip-172-31-18-117 ~]$ ▂
```

```
**Quiz**
1. What is ssh?
```

Chapter 4

Processing 10X RNA-seq data

4.1 Introduction

Raw RNA-sequencing data will be in the format of a fastq file. This format describes the read ID, read sequence and sequencing quality scores. Represented in the following format:

```
>ReadID
READ SEQUENCE
+
SEQUENCING QUALITY SCORES
```

Generally fastq files are pre-processed using quality control tools, such as FastQC. This outputs a series of metrics assessing the quality of sequence reads. We will skip this step as we are using a public (pre-checked) scRNA-seq dataset, and limited in time. Some of these metrics include:

```
1. Per base sequence quality
2. Per sequence quality scores
3. Per base sequence content
4. Per sequence GC content
5. Per base N content
6. Sequence Duplication Levels
7. Overrepresented Sequences
8. Adapter Content
9. Kmer Content
```

For more details: https://rtsf.natsci.msu.edu/genomics/tech-notes/fastqc-tutorial-and-faq/

4.2 Read Alignment

Once confirming adequate sequencing quality of your library, the next goal is to align individual reads to the reference genome whilst retaining information about where the read originated from. The cell barcode from transcript reads capture this information. With the input of a gene annotation file (i.e. GTF file), not only can we decipher the location a read maps to in the genome, but the gene name it corresponds to. This in theory, this allows the quantification/count of reads which align to each gene/feature of the genome. This is important for downstream analysis where comparitive analysis between single cells occurs and these counts uncover differentially expressed genes. This can eventually lead to inferences such as unique cell type populations.

4.2.1 Building a reference genome index

Alignment to a reference genome first requires the generation of an genome index to facilitate the mapping process. This allows looking up parts of the genome in a much faster manner which is neccessary when trying to mapping millions of read sequences. This process saves alots of time and memory when aligning. An analogy to this process is reflected in book indexes. To find a specific part of a book, it is alot faster to look at the chapter indexes first to locate the region of interest instead of looking through every page of the book from top down.

Here we can provide an annotation file (.gtf) of the reference genome to provide information about where genes lie within the genome.

This step was performed in advanced to save time as it can take up to 1 hour to build a reference index for the human genome. The following command was performed:

```
STAR --runMode genomeGenerate --runThreadN 16 --genomeDir genome_index/ --genomeFastaF
```

The genome index output is located in the folder `genome_index`.

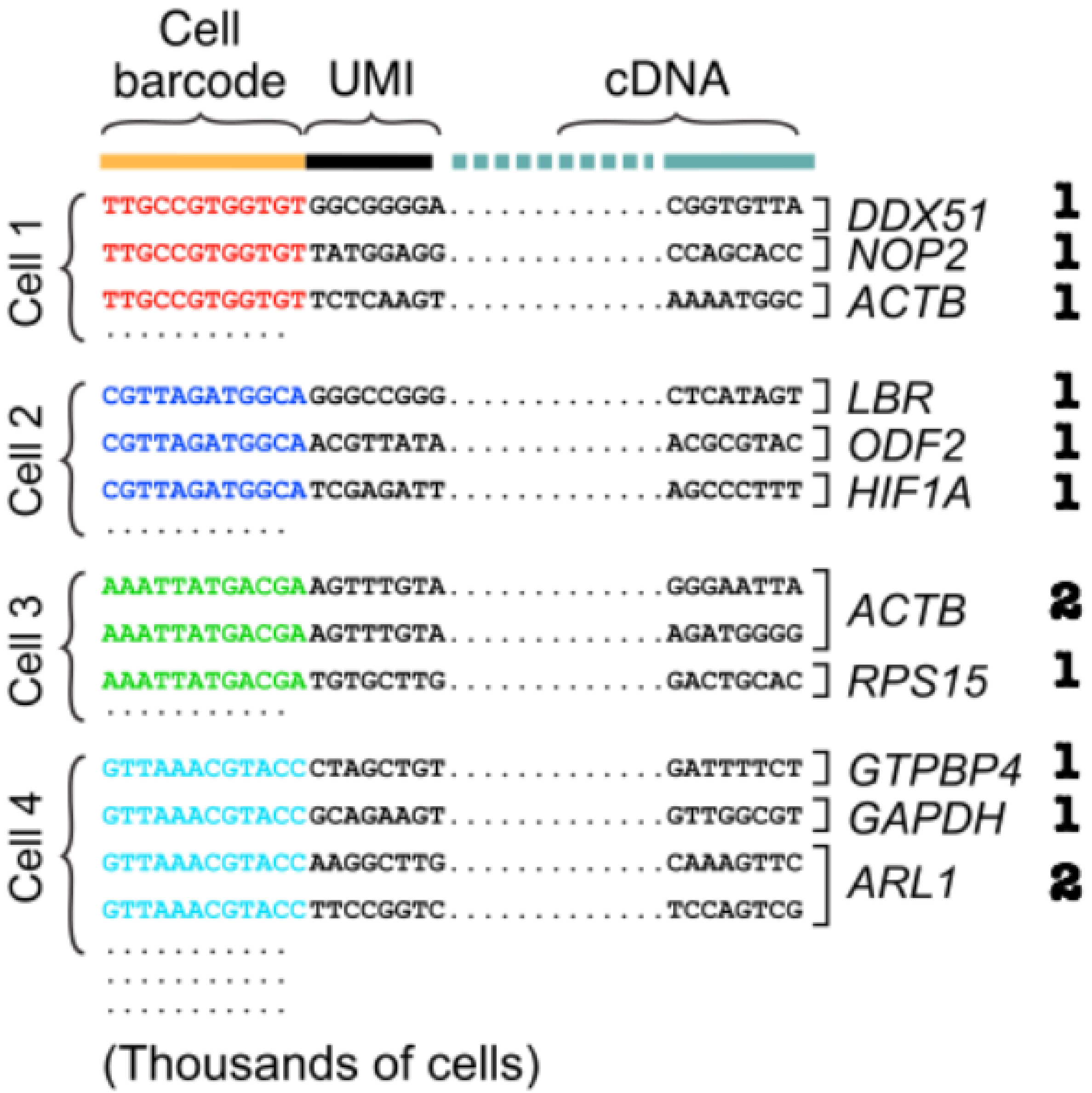
```
**Quiz**
1. How many base pairs are there in the (a) human (b) mouse genome?
2. What is the file size of the whole human genome DNA sequence?
3. What format is this file in?
```

4.2.2 Alignment of reads

The next step is to map the raw fastq files to the reference genome. In this tutorial we will be using Human CD45+ cells from human melanoma samples, which were sequenced via 10X Genomics and Chromium™ Single Cell

3' Reagent Kit (v2). More information about this dataset can be found at https://www.ncbi.nlm.nih.gov/sra/SRX6872900.

Paired-end sequencing outputs 2 fastq files corresponding to the 5' and 3' direction of sequencing. It is important to recognised the library preparation chemistry used for sequencing in order to determine cell barcode and UMI barcode sequence length and location. This allows the mapping algorithm to distinguish which sequences are barcodes and which are transcript sequences.


Cell barcode
UMI
cDNA
Cell 1
TTGCCGTGGTGT GGCGGGGA.............CGGTGTTA DDX51 1
TTGCCGTGGTGT TATGGAGG.............CCAGCACC NOP2 1
TTGCCGTGGTGT TCTCAAGT.............AAAATGGC ACTB 1
Cell 2
CGTTAGATGGCA GGGCCGGG.............CTCATAGT LBR 1
CGTTAGATGGCA ACGTTATA.............ACGCGTAC ODF2 1
CGTTAGATGGCA TCGAGATT.............AGCCCTTT HIF1A 1
Cell 3
AAATTATGACGA AGTTTGTA.............GGGAATTA
AAATTATGACGA AGTTTGTA.............AGATGGGG ACTB 2
AAATTATGACGA TGTGCTTG.............GACTGCAC RPS15 1
Cell 4
GTTAAACGTACC CTAGCTGT.............GATTTTCT GTPBP4 1
GTTAAACGTACC GCAGAAGT.............GTTGGCGT GAPDH 1
GTTAAACGTACC AAGGCTTG.............CAAAGTTC
GTTAAACGTACC TTCCGGTC.............TCCAGTCG ARL1 2
(Thousands of cells)

```
**Quiz**
1. What is the difference between a cell barcode and UMI barcode? and what are their s
2. What are the lengths of the cell barcode and UMI barcode used in our dataset?
3. Are these barcodes located on the 5' or 3' read file?
```

Generally 10X Genomics scRNA-Seq reads are aligned using the tool Cell

Ranger. It is a wrapper to the open source alignment tool, STAR by Alexander Dobin, which optimises algorithms to handle the sequencing chemistry of 10X genomic library preparation. This includes UMI counting and calling cell barcodes. More information can be found at https://support.10xgenomics.com/single-cell-gene-expression/software/pipelines/latest/algorithms/overview.

In this workshop we will be using a recently developed application of STAR, STARsolo, which is an effective solution to handle droplet-based scRNA-seq data analysis. It provides an output similar to Cell Ranger which is important for downstream analysis packages such as Seurat. STARsolo also performs extremely fast read alignment of single cell raw reads from 10X genomics. It claims to be 10 times faster than Cell Ranger. More details can be found in the latest version of STAR manual: https://github.com/alexdobin/STAR/blob/master/doc/STARmanual.pdf.

Some of the key functions of STARsolo are:

```
1. Error correction and demultiplexing of cell barcodes using user-input whitelist.
2. Mapping the reads to the reference genome using the standard STAR spliced read alignment algor
3. Error correction and collapsing (deduplication) of Unique Molecular Identifiers (UMIs).
4. Quantification of per-cell gene expression by counting the number of reads per gene.
```

Create an output directory called “starsolo_out”.

```
mkdir starsolo_out
```

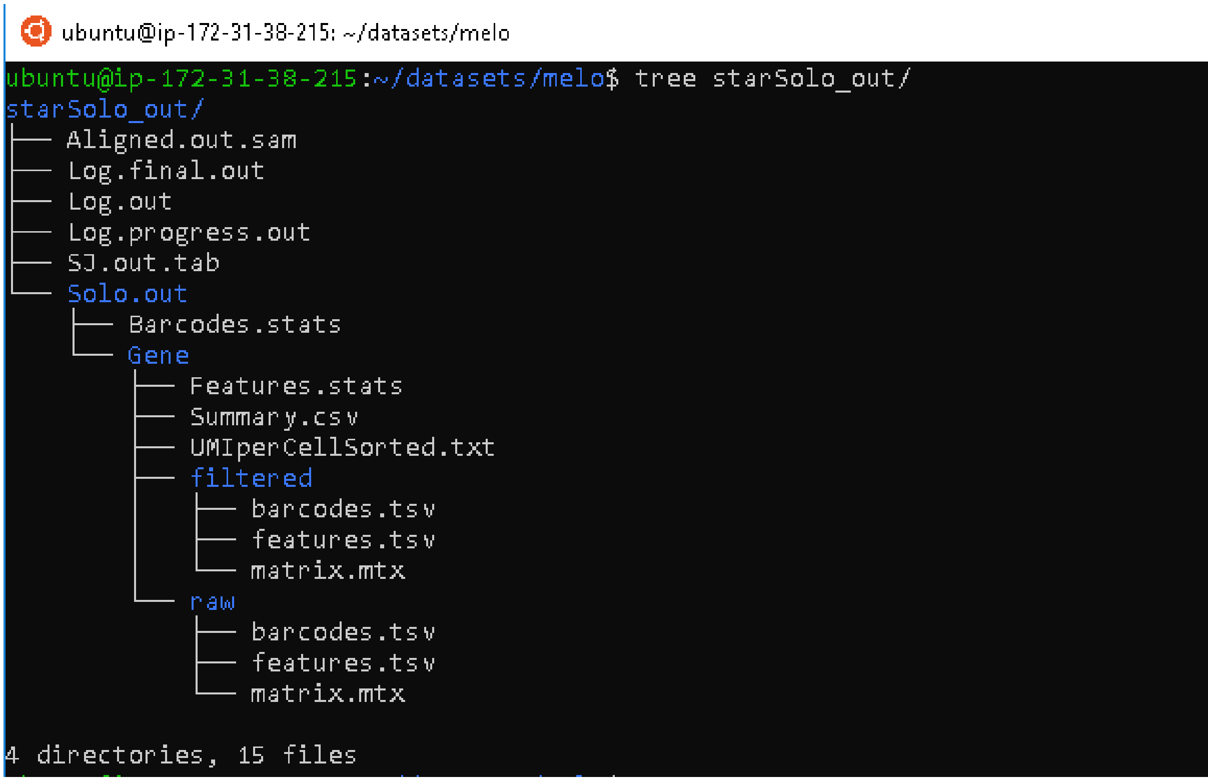
Run the following command to perform read alignment:

```
STAR --genomeDir genome_index/ --soloType Droplet --soloCBwhitelist barcode_whitelist/10X_v2.txt
```

This step can take several minutes. Please work through the quiz questions in the meanwhile and ask questions!

```
**Quiz**
1. Describe the function of each parameter in the sequence alignment command.
2. What are some different types of scRNA-sequencing methods other than droplet based? And descri
```

4.3 STARsolo Output



```
ubuntu@ip-172-31-38-215:~/datasets/melo$ tree starSolo_out/
starSolo_out/
├── Aligned.out.sam
├── Log.final.out
├── Log.out
├── Log.progress.out
├── SJ.out.tab
└── Solo.out
    ├── Barcodes.stats
    └── Gene
        ├── Features.stats
        ├── Summary.csv
        ├── UMIperCellSorted.txt
        ├── filtered
        │   ├── barcodes.tsv
        │   ├── features.tsv
        │   └── matrix.mtx
        └── raw
            ├── barcodes.tsv
            ├── features.tsv
            └── matrix.mtx

4 directories, 15 files
```

The STARsolo program outputs a large amount of files reflecting details of the read alignment process. We will only discuss some of the key files important for downstream analysis.

1. The **BAM file**, contains information about mapped reads, in a binary compressed format. When de-compressed into a SAM file, information is stored as a tab seperated table where the columns corresponds to:

```
QNAME : read name (generally will include UMI barcode if applicable)
FLAG : number tag indicating the "type" of alignment, link to explanation of all possil
RNAME : reference sequence name (i.e. chromosome read is mapped to).
POS : leftmost mapping position
MAPQ : Mapping quality
CIGAR : string indicating the matching/mismatching parts of the read (may include soft-
RNEXT : reference name of the mate/next read
PNEXT : POS for mate/next read
TLEN : Template length (length of reference region the read is mapped to)
SEQ : read sequence
```

```
QUAL : read quality
```

BAM files can be viewed using SAMtools.

```
samtools view output.bam
```

2. **Alignment summary** files, Features.stat and Summary.csv, contains information about basic mapping details. This can serve as an easy preliminary quality control check of the alignment process.

3. The **feature matrix** file, matrix.mtx, contains information about the counts of genes mapped in each individual cell. The column names corresponds to each individual cell barcode, and the row names corresponds to all annotated genes. Due to the large size of this data, it is stored as a sparse matrix.

4. **Auxiliary files**, barcodes.tsv and features.tsv, provide extra metadata important for downstream analysis. These files along with the matrix file are required for analysis in gold standard scRNA-seq data analysis package, seurat.

Chapter 5

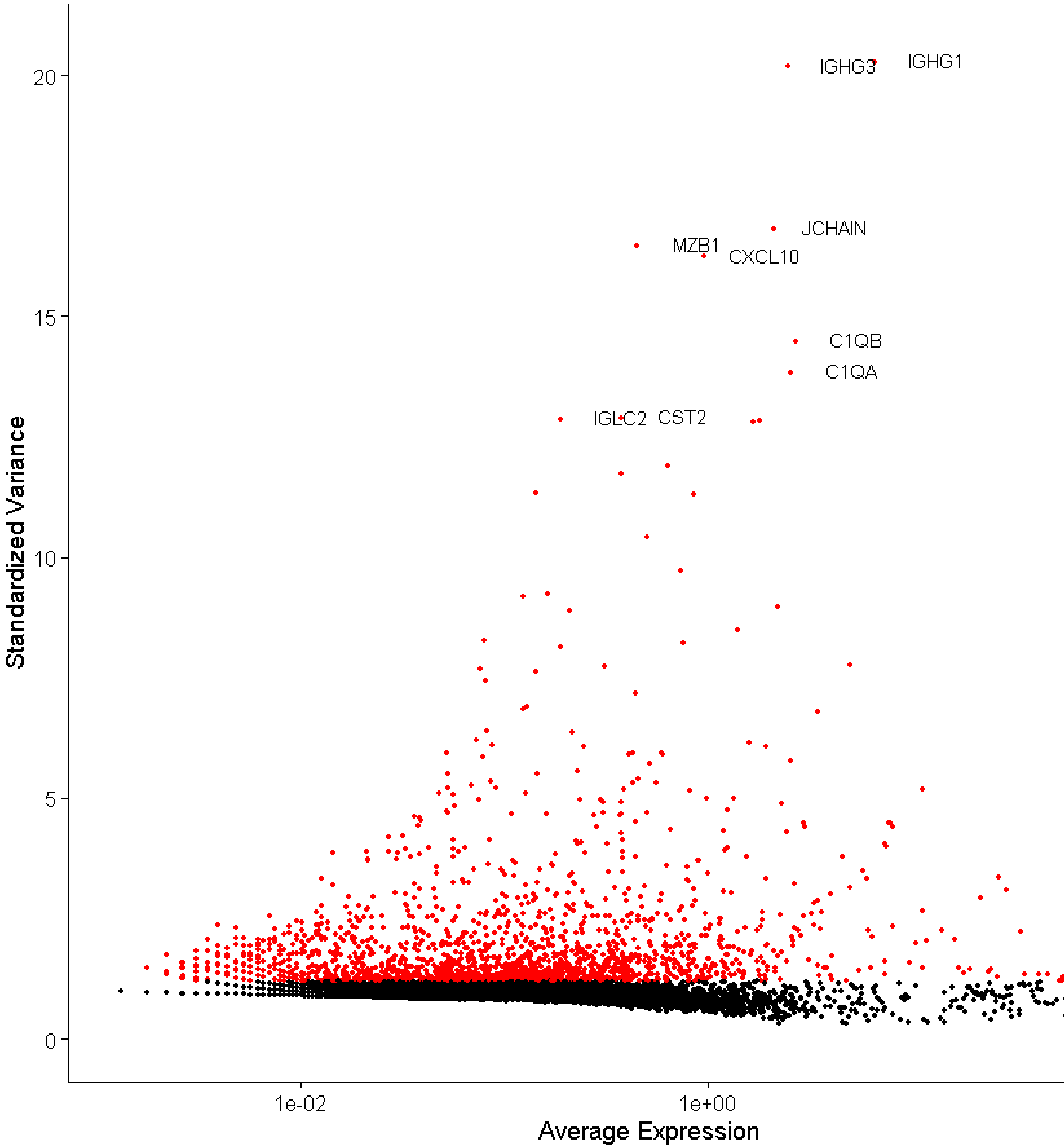
scRNA-seq Downstream Analysis

5.1 Seurat Package

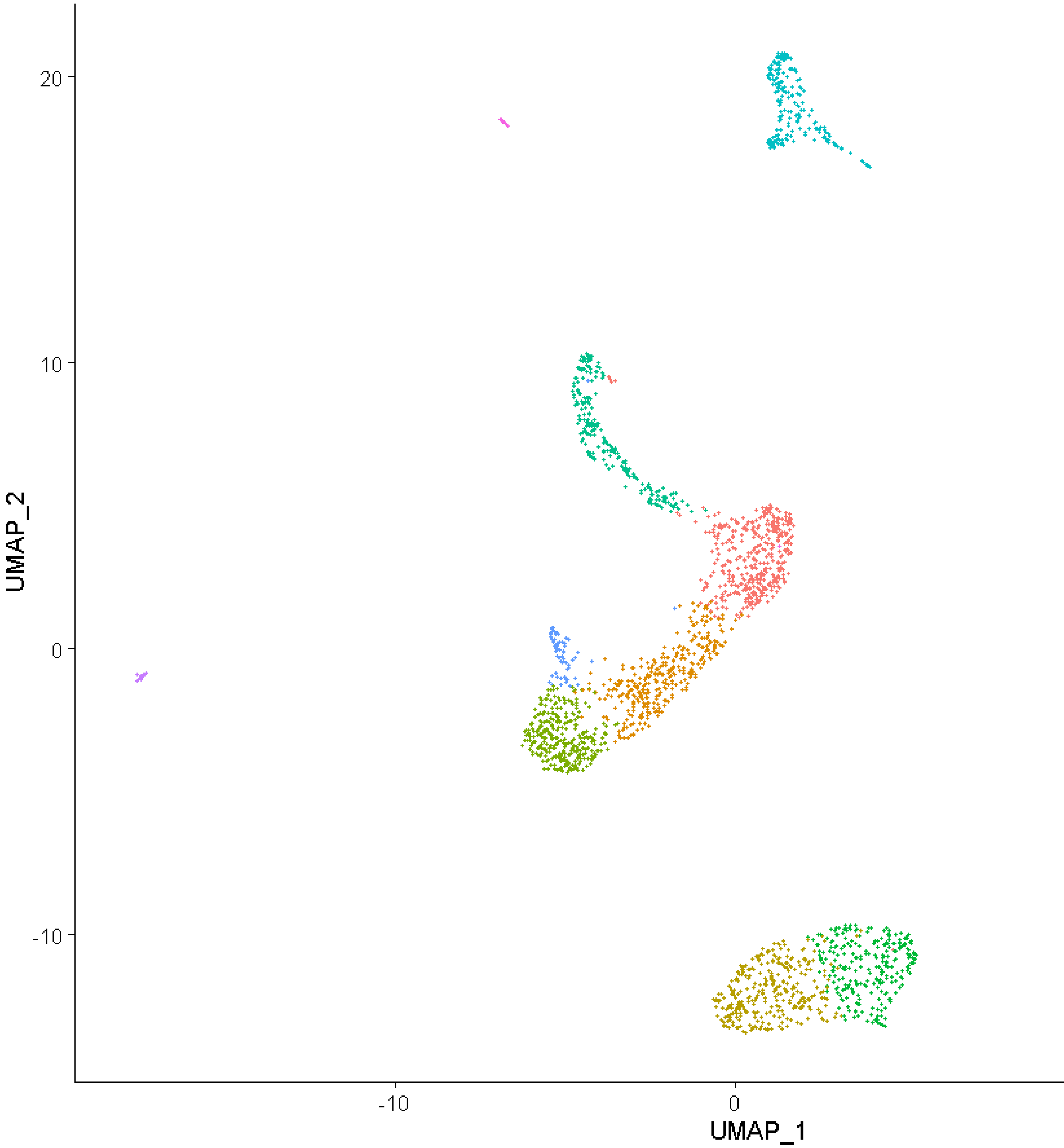
Seurat is an R package designed for QC, analysis, and exploration of single-cell RNA-seq data. Seurat aims to enable users to identify and interpret sources of heterogeneity from single-cell transcriptomic measurements, and to integrate diverse types of single-cell data.

The format of output results from both Cell Ranger and STARsolo stream nicely into the Seurat defualt analysis pipeline. We will not demonstrate this process but only highlight the types of insights/plots you can achieve from this data analysis. More information can be found at https://satijalab.org/seurat.

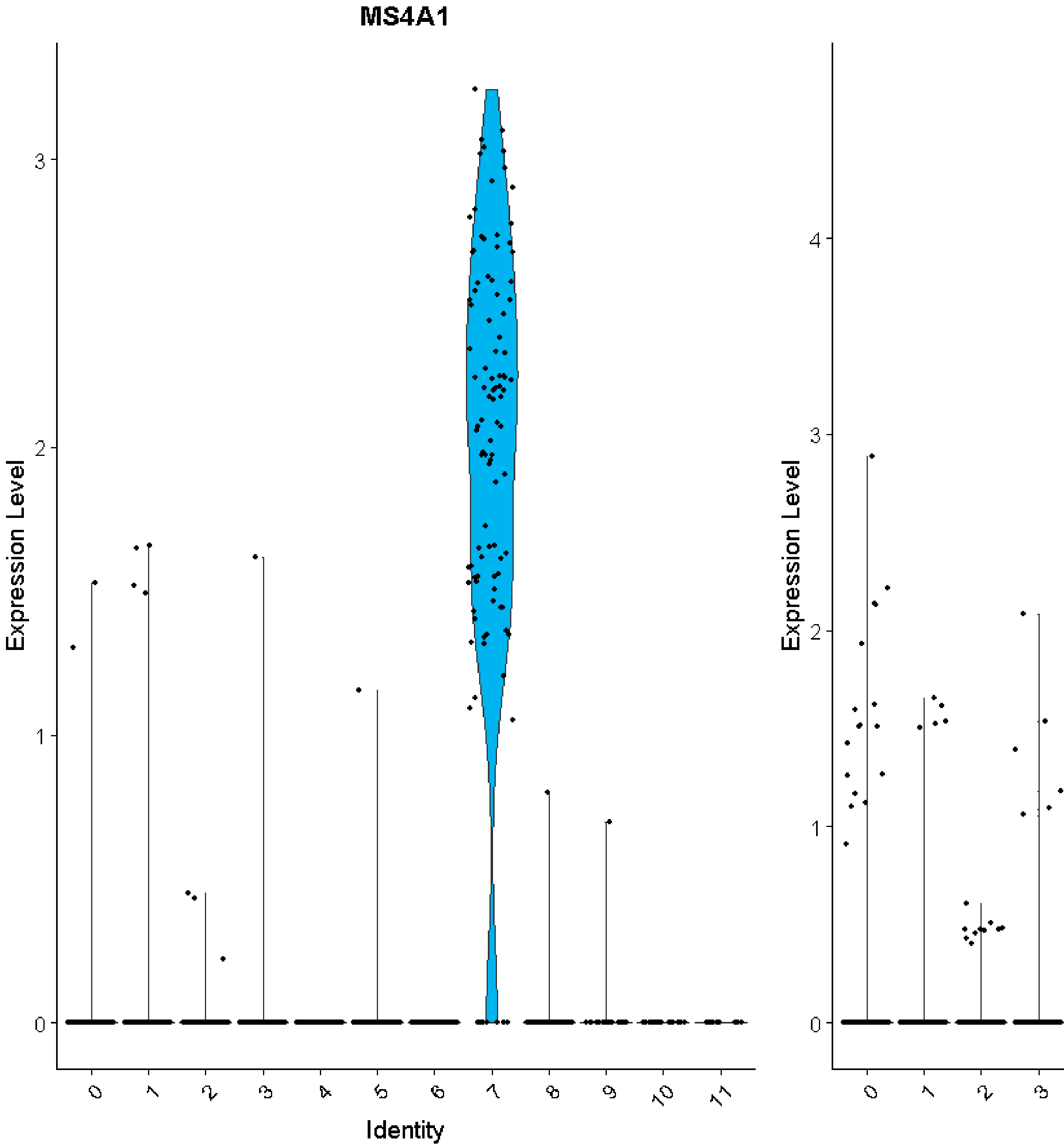
5.1.1 Differentially expressed Genes

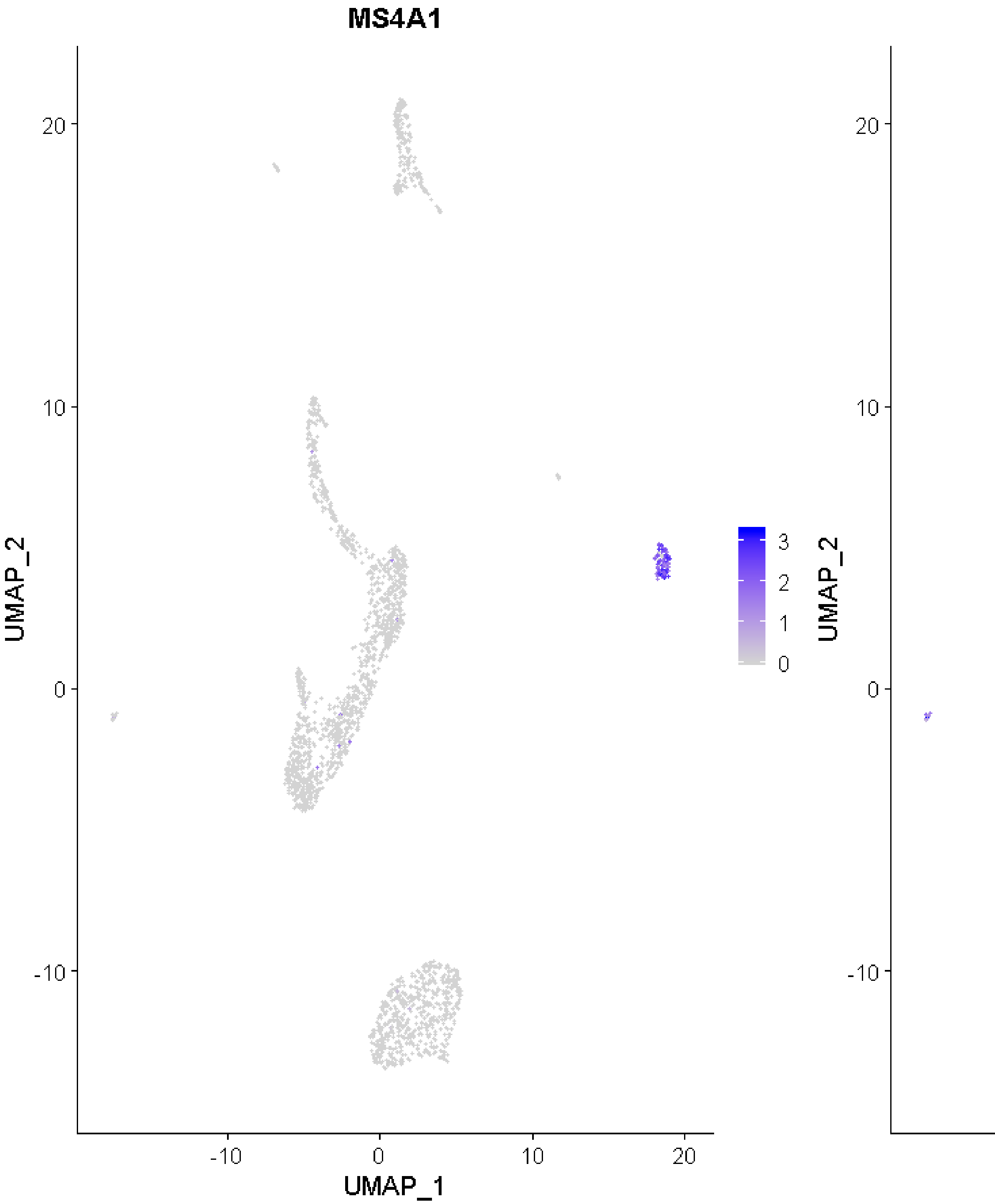


5.1.2 Cell Clustering

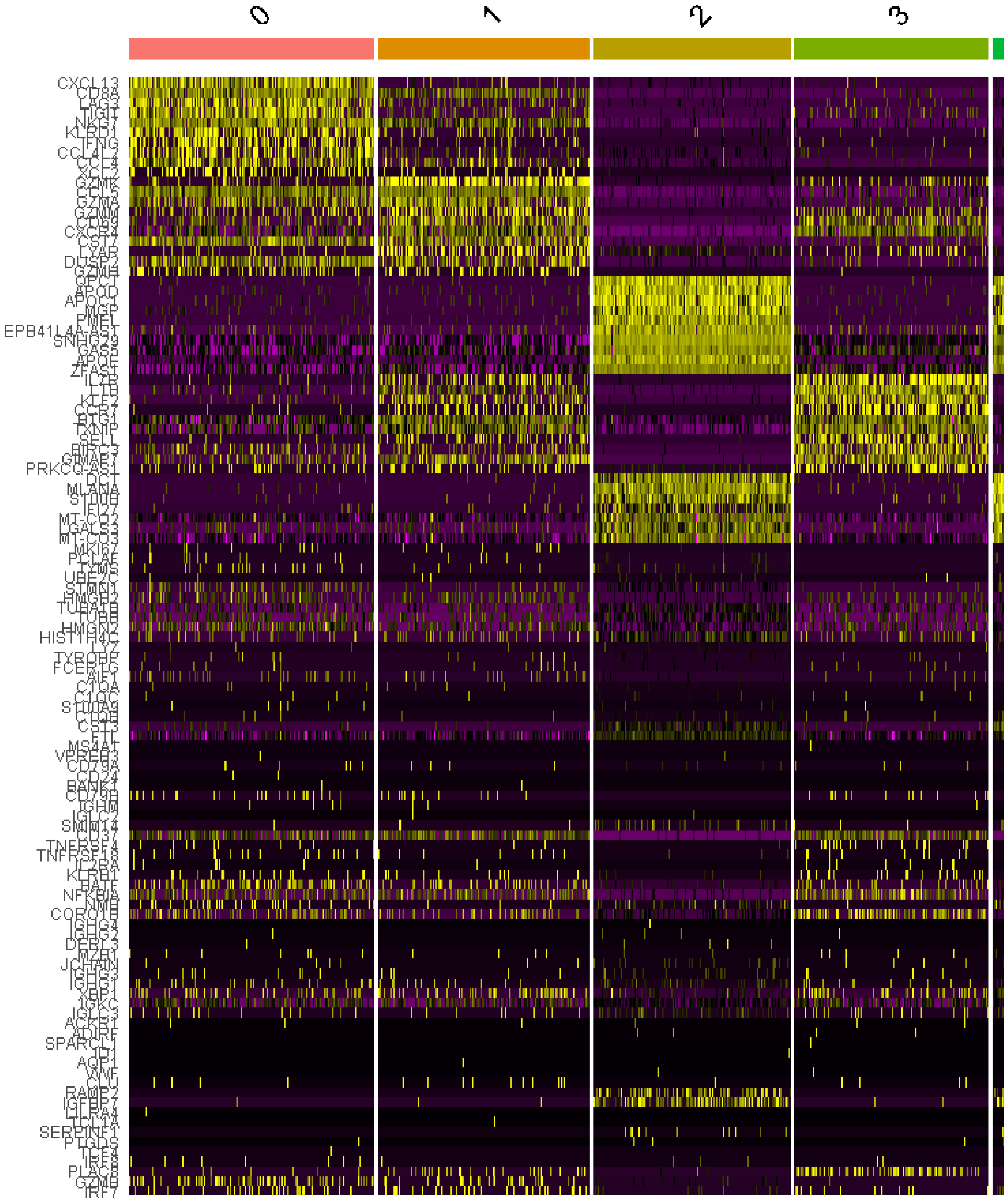


5.1.3 Analysis of Variable Gene Markers




MS4A1
UMAP_2
UMAP_1
20
10
0
-10
-10
0
10
20
3
2
1
0
UMAP_2
20
10
0
-10

5.1.4 Heatmap of gene expression by clusters



Chapter 6

References

Dobin, A., Davis, C. A., Schlesinger, F., Drenkow, J., Zaleski, C., Jha, S., … Gingeras, T. R. (2013). STAR: Ultrafast universal RNA-seq aligner. Bioinformatics, 29(1), 15–21. https://doi.org/10.1093/bioinformatics/bts635