Introduction to Single-cell RNA-seq Analysis - 3

by Jiajia Li

Research School of Biology

20 Aug 2025

Australian National University

Learning Objectives of Today

- Make predictions to cell types using reference.
- Visualise the prediction.
- Add predictions to the Seurat metadata table
- See cell type annotation differences between sample.
- Map cell type to the UMAP cluster.

- Try a different reference and compare the annotation results.

- How PC numbers influence the UMAP?

Cell type annotation

Cell type annotation is the process of **assigning biological labels** (e.g., "T cell", "astrocyte", "fibroblast") to individual cells or clusters of cells based on their molecular profiles - most commonly their **gene expression patterns**.

Methods for assigning cell types usually fall into one of two categories:

- **Manual annotation with marker genes**
- **Automated, reference-based annotation**

Manual Annotation

General steps:

- Identify clusters of transcriptionally similar cells.
- Look for genes that are significantly upregulated in each cluster.
- Compare those marker genes to known cell type markers from literature or databases.
- Assign a label to each cluster.

Databases:

- PanglaoDB
- CellMarker 2.0
- Human Protein Atlas
- Azimuth

Automated, Reference-based Annotation

Instead of manually looking up marker genes, you use a **reference dataset** where cell types have already been annotated.

Your new (query) dataset is then compared to that reference, and labels are assigned automatically – usually per cell or per cluster.

Common references:

- Azimuth
- Human Cell Atlas
- Blueprint/ENCODE
- Tabula Muris
- Tabula Sapiens
- Mouse Cell Atlas

Automated, Reference-based Annotation

SingleR is one of the most widely used reference-based cell type annotation tools in single-cell RNA-seq analysis because it's simple, fast, and works well across many datasets.

In this workshop, we will demonstrate how to annotate cell types using SingleR.

First, let's load the libraries we need:

```
library(SingleR)
library(celldex)
library(Seurat)
library(cowplot)
```

Automated, Reference-based Annotation

If you don't have SingleR or celldex installed, please follow this to install Bioconductor first.

```
if (!require("BiocManager", quietly = TRUE))
  install.packages("BiocManager")
```

This is to check if you have "BiocManager", which is Bioconductor, already installed on your computer. If not, it will install automatically.

Then, when it finishes, run:

```
BiocManager::install(version = "devel")
```

This installs the dev version of Bioconductor. You will be prompted with questions asking if you want to update packages, please answer "yes" or "all".

Automated, Reference-based Annotation

Install `SingleR`:

```
BiocManager::install("SingleR")
```

Install `celldex`:

```
BiocManager::install("celldex")
```

Specifying reference dataset

This first step to do is to read in the previously saved Seurat object "rep135_clustered.rds".

```
merged <- readRDS("rep135_clustered.rds")
```

Specify which reference to use:

```
ref_immgen <- celldex::ImmGenData()
```

When we write `**celldex::ImmGenData()**` it means we want to use the `ImmGenData()` function from the package `celldex`.

`**celldex**` is a package that provides a collection of **reference expression datasets** with curated cell type labels, for use in procedures like automated annotation of single-cell data or deconvolution of bulk RNA-seq.

Specifying reference dataset

`**ImmGenData()**` function downloads and cache the normalised expression values of 830 microarray samples of **pure mouse immune cells**, generated by the Immunologic Genome Project (ImmGen).

This dataset consists of **20 broad cell types** ("label.main") and **253 finely resolved cell subtypes** ("label.fine"). The subtypes have also been mapped to the Cell Ontology ("label.ont", if cell.ont is not "none"), which can be used for further programmatic queries.

Calling the ImmGenData() function returns a SummarizedExperiment object containing a matrix of log-expression values with sample-level labels.

Specifying reference dataset

```
> ref_immgen
class: SummarizedExperiment
dim: 22134 830
metadata(0):
assays(1): logcounts
rownames(22134): Zglp1 Vmn2r65 ... Tiparp Kdm1a
rowData names(0):
colnames(830):
  GSM1136119_EA07068_260297_MOGENE-1_0-ST-V1_MF.11C-11B+.LU_1.CEL
  GSM1136120_EA07068_260298_MOGENE-1_0-ST-V1_MF.11C-11B+.LU_2.CEL ...
  GSM920654_EA07068_201214_MOGENE-1_0-ST-V1_TGD.VG4+24ALO.E17.TH_1.CEL
  GSM920655_EA07068_201215_MOGENE-1_0-ST-V1_TGD.VG4+24ALO.E17.TH_2.CEL
colData names(3): label.main label.fine label.ont
```

Specifying reference dataset

Now let's see what each of the labels look like:

```
head(ref_immgen$label.main, n=10)
```

```
[1] "Macrophages" "Macrophages" "Macrophages" "Macrophages" "Macrophages"
[6] "Macrophages" "Monocytes"   "Monocytes"   "Monocytes"   "Monocytes"
```

From the main labels, we can see that we get general cell types such as Macrophages and Monocytes.

Specifying reference dataset

```
head(ref_immgen$label.fine, n=10)
```

```
[1] "Macrophages (MF.11C-11B+)" "Macrophages (MF.11C-11B+)"
[3] "Macrophages (MF.11C-11B+)" "Macrophages (MF.ALV)"
[5] "Macrophages (MF.ALV)"      "Macrophages (MF.ALV)"
[7] "Monocytes (MO.6+I-)"       "Monocytes (MO.6+2+)"
[9] "Monocytes (MO.6+2+)"       "Monocytes (MO.6+2+)"
```

From the fine labels, we can see that we start to **subtype** the more general cell types we saw above.

So rather than seeing 6 labels for Macrophages we now see specific Macrophage types such as **Macrophages (MF.11C-11B+)**.

Specifying reference dataset

```
head(ref_immgen$label.ont, n=10)
```

```
[1] "CL:0000235" "CL:0000235" "CL:0000235" "CL:0000583" "CL:0000583"
[6] "CL:0000583" "CL:0000576" "CL:0000576" "CL:0000576" "CL:0000576"
```

From the ont labels, we can see that the subtypes are now mapped to Cell Ontology IDs.

Applying the ImmGen cell reference to our data

`SingleR()` function returns the best annotation for **each cell** in a test dataset, given a labelled reference dataset.

```
predictions_main <- SingleR(test = GetAssayData(merged),
                            ref = ref_immgen,
                            labels = ref_immgen$label.main)
```

And we can predict using different labels by changing the `labels =` option.

```
predictions_fine <- SingleR(test = GetAssayData(merged),
                            ref = ref_immgen,
                            labels = ref_immgen$label.fine)
```

Applying the ImmGen cell reference to our data

Let's take a look of the result we get:

```
> head(predictions_main)
DataFrame with 6 rows and 4 columns
                                                      scores      labels
                                                    <matrix> <character>
Rep1_ICBdT_AAACCTGAGCCAACAG-1 0.4156037:0.4067582:0.2845856:...         NKT
Rep1_ICBdT_AAACCTGAGCCTTGAT-1 0.4551058:0.3195934:0.2282272:...     B cells
Rep1_ICBdT_AAACCTGAGTACCGGA-1 0.0717647:0.0621878:0.0710026:... Fibroblasts
Rep1_ICBdT_AAACCTGCACGGCCAT-1 0.2774994:0.2569566:0.2483387:...    NK cells
Rep1_ICBdT_AAACCTGCACGGTAAG-1 0.3486259:0.3135662:0.3145100:...     T cells
Rep1_ICBdT_AAACCTGCATGCCACG-1 0.0399733:0.0229926:0.0669236:... Fibroblasts
                              delta.next pruned.labels
                               <numeric>   <character>
Rep1_ICBdT_AAACCTGAGCCAACAG-1  0.0124615         NKT
Rep1_ICBdT_AAACCTGAGCCTTGAT-1  0.1355124     B cells
Rep1_ICBdT_AAACCTGAGTACCGGA-1  0.1981683   Fibroblasts
Rep1_ICBdT_AAACCTGCACGGCCAT-1  0.0577608    NK cells
Rep1_ICBdT_AAACCTGCACGGTAAG-1  0.1038542     T cells
Rep1_ICBdT_AAACCTGCATGCCACG-1  0.2443470   Fibroblasts
```

Each row here is a cell.

And there are 4 columns showing various information.

Applying the ImmGen cell reference to our data

`$scores` column contains a **matrix** for each cell that corresponds to how confident SingleR is in assigning each cell type to the cell.

We can take a further look by: `predictions_main$scores |> View()`

	B cells	B cells, pro	Basophils	DC	Endothelial cells	Eosinophils	Epithelial cells
1	0.41560372	0.40675819	0.28458563	0.35902435	0.17686165	0.258423099	0.21370163
2	0.45510585	0.31959340	0.22822721	0.28552281	0.14564842	0.231320630	0.12756365
3	0.07176469	0.06218776	0.07100260	0.13944442	0.33222247	0.066564904	0.26645829
4	0.27749942	0.25695663	0.24833873	0.24323448	0.13688218	0.233915120	0.11923855
5	0.34862589	0.31356617	0.31450997	0.33118269	0.18668629	0.289118582	0.18318870

Applying the ImmGen cell reference to our data

`$labels` column is the most confident assignment SingleR has for that cell.

We can look at how many cells are assigned to each label by:

```
table(predictions_main$labels)
```

```
          B cells       B cells, pro         Basophils                DC Endothelial cells
             3253                  3                37                 295                  71
Epithelial cells        Fibroblasts               ILC        Macrophages         Mast cells
             1238                589               763                 459                  11
       Monocytes        Neutrophils          NK cells                NKT         Stem cells
              633                 92               565                2249                   2
   Stromal cells            T cells               Tgd
               18              12714               193
```

Applying the ImmGen cell reference to our data

`$delta.next` column contains the "delta" value for each cell, which is the gap, or the difference between the **score of the assigned label** and the **next-best score**.

If the **delta is small**, this indicates that the cell matches all labels with the same confidence, so the assigned label is **not very meaningful**.

SingleR can discard cells with low delta values, so in the `$pruned.labels` column, cells that have low delta value will be marked as **NA**.

How many cells have low delta values or assigned NAs in the `$pruned.labels`?

Applying the ImmGen cell reference to our data

```
> summary(is.na(predictions_main$pruned.labels))
   Mode    FALSE     TRUE
logical    23001      184
```

184 out of 23,185 cells have low delta values, which means not confident in assigning any labels.

Let's have a look of the fine labels result:

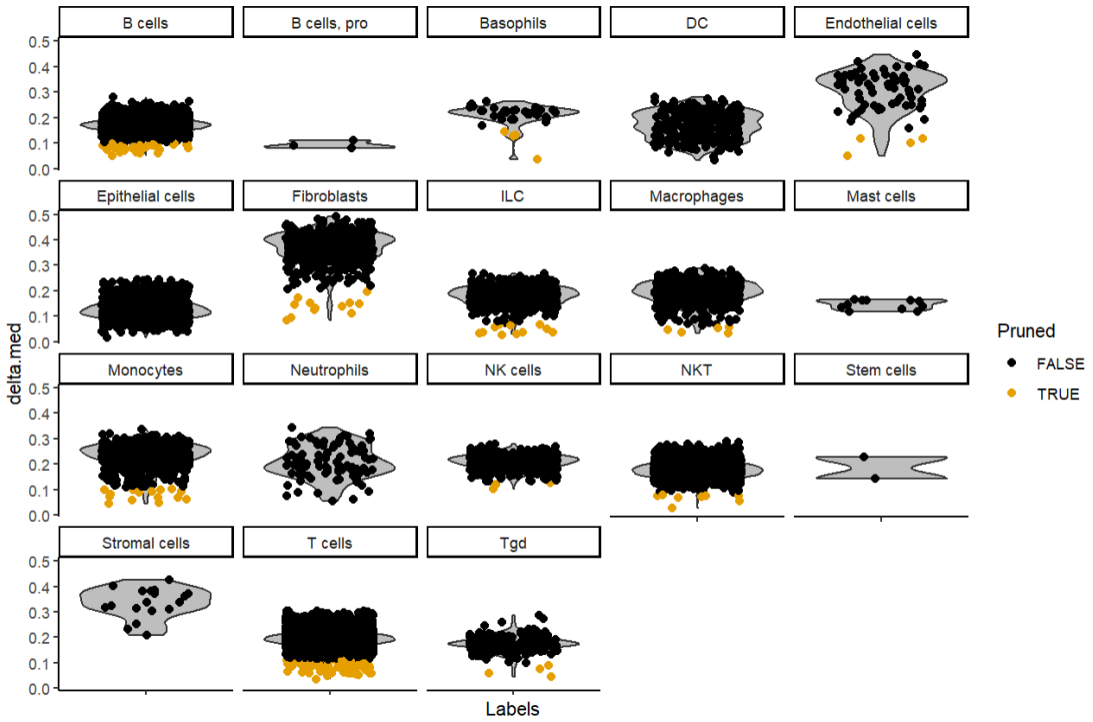
```
> summary(is.na(predictions_fine$pruned.labels))
   Mode    FALSE     TRUE
logical    23005      180
```

180 out of 23,185 cells are assigned as NAs.

Now that we understand what the SingleR dataframe looks and what the data contains, let's begin to visualise it.

Visualise the assigned labels

```
plotDeltaDistribution(predictions_main)
```



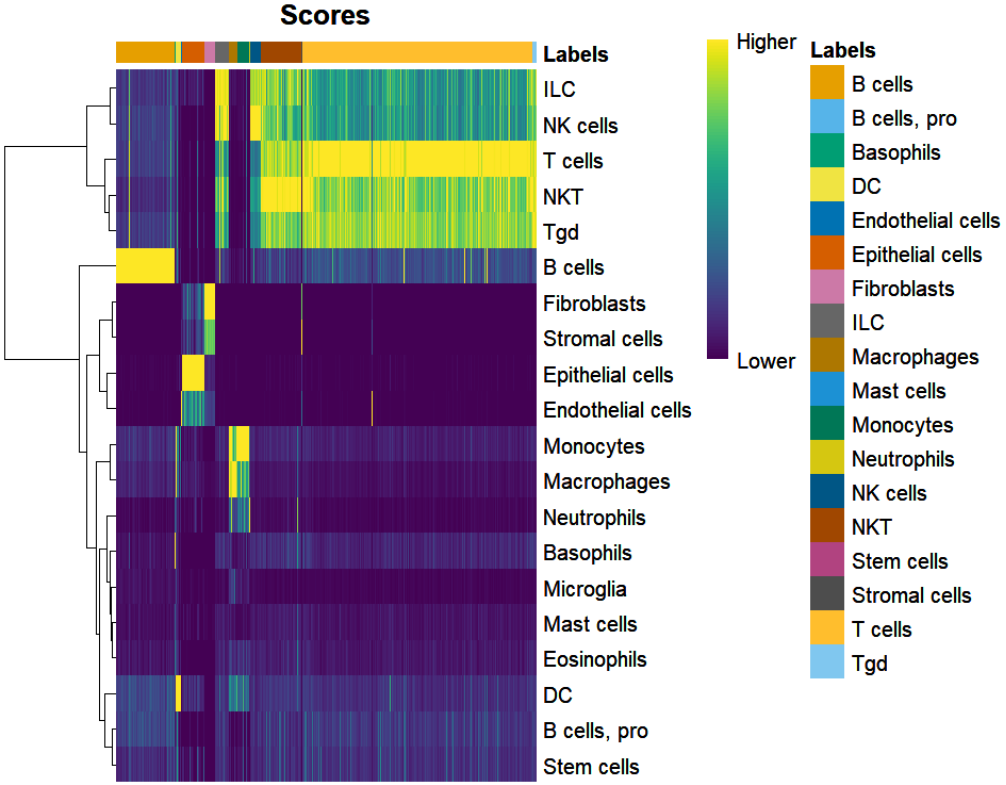
`**delta.med**` is the difference between the score for the assigned label and the median across all labels for each cell.

Our assumption is that most of the labels in the reference are not relevant to any given cell.

Thus, the median across all labels can be used as a measurement of the baseline correlation, while the gap from the assigned label to this baseline can be used as a measure of the **assignment confidence**.

Visualise the assigned labels

```
plotScoreHeatmap(predictions_main)
```



Here, the key is to examine the spread of scores within each cell, i.e., down the columns of the heatmap.

Similar scores for a group of labels indicates that the assignment is uncertain for those columns, though this may be acceptable if the uncertainty is distributed across closely related cell types.

Add labels to Seurat object

Rather than only working with the SingleR dataframe, we can add the labels to our Seurat object as a metadata field.

```
merged[['immgen_singler_main']] <- rep('NA', ncol(merged))
merged$immgen_singler_main[rownames(predictions_main)] <- predictions_main$labels
```

	G2M.Score	Phase	immgen_singler_main
Rep1_ICBdT_AAACCTGAGCCAACAG-1	0.196399610	S	NKT
Rep1_ICBdT_AAACCTGAGCCTTGAT-1	-0.132307049	G1	B cells
Rep1_ICBdT_AAACCTGAGTACCGGA-1	-0.177809316	G1	Fibroblasts
Rep1_ICBdT_AAACCTGCACGGCCAT-1	-0.064005976	G1	NK cells
Rep1_ICBdT_AAACCTGCACGGTAAG-1	-0.066370931	G1	T cells
Rep1_ICBdT_AAACCTGCATGCCACG-1	-0.212300146	G1	Fibroblasts
Rep1_ICBdT_AAACCTGGTCTTGTCC-1	0.001735468	G2M	T cells
Rep1_ICBdT_AAACCTGGTTCCGTCT-1	-0.032053224	G1	Neutrophils
Rep1_ICBdT_AAACCTGTCTCATTCA-1	-0.202545555	G1	Fibroblasts
Rep1_ICBdT_AAACGGGAGCAATATG-1	-0.065991311	G1	NKT
Rep1_ICBdT_AAACGGGAGCGCTCCA-1	-0.088748725	G1	DC

Add labels to Seurat object

Exercise: Similarly, please add the fine labels to the metadata table.

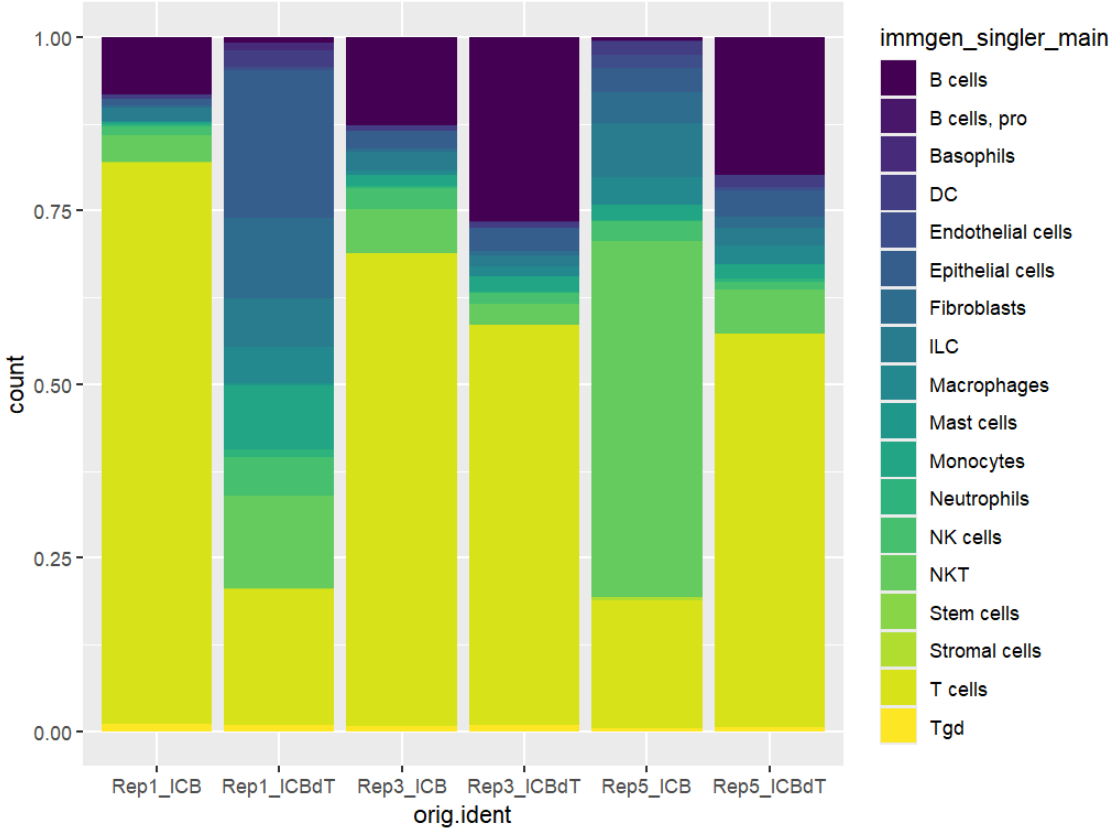
Visualise cell types within sample

How do our samples differ in their relative cell composition?

```
library(viridis)
library(ggplot2)

ggplot(merged[[]], aes(x = orig.ident, fill = immgen_singler_main)) +
  geom_bar(position = "fill") +
  scale_fill_viridis(discrete = TRUE)
```

Visualise cell types within sample

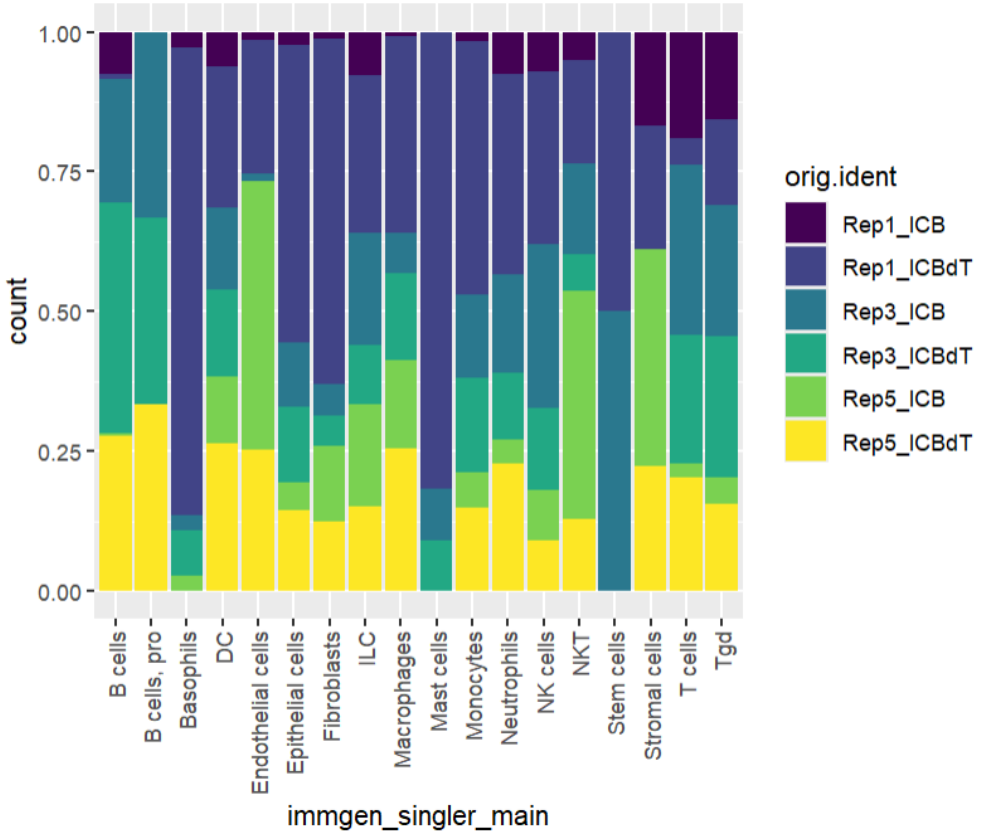


Visualise cell types within sample

We can also flip the sample label and cell label:

```
ggplot(merged[[]], aes(x = immgen_singler_main, fill = orig.ident)) +
  geom_bar(position = "fill") +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust = 1)) +
  scale_fill_viridis(discrete = TRUE)
```

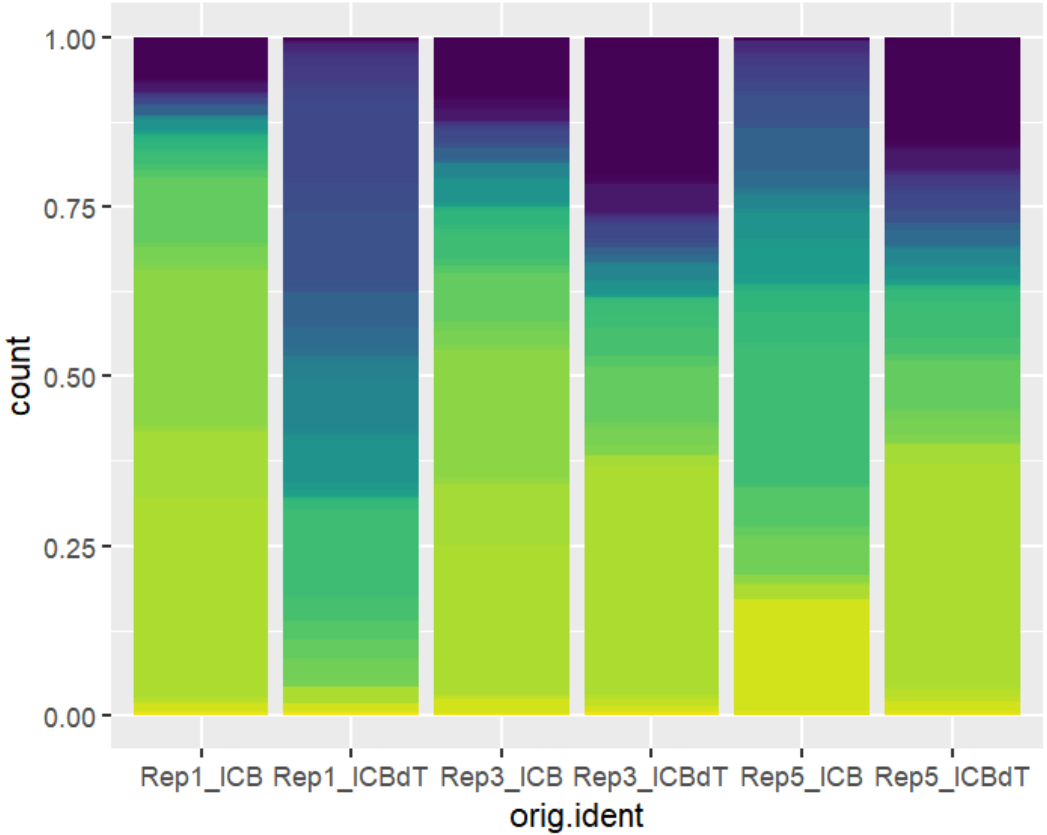
Visualise cell types within sample



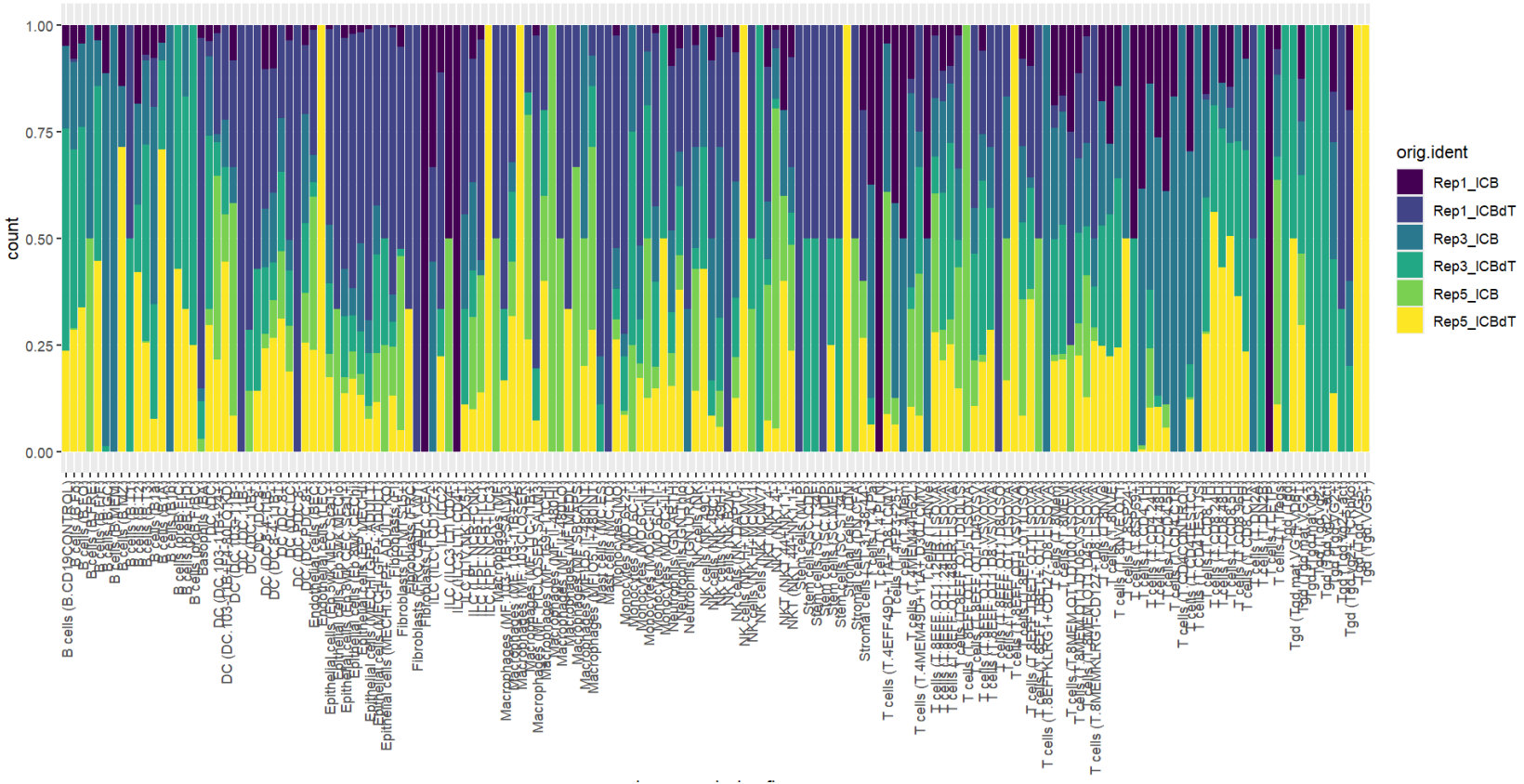
Visualise cell types within sample

Exercise: create these two plots for fine labels too.

Visualise cell types within sample



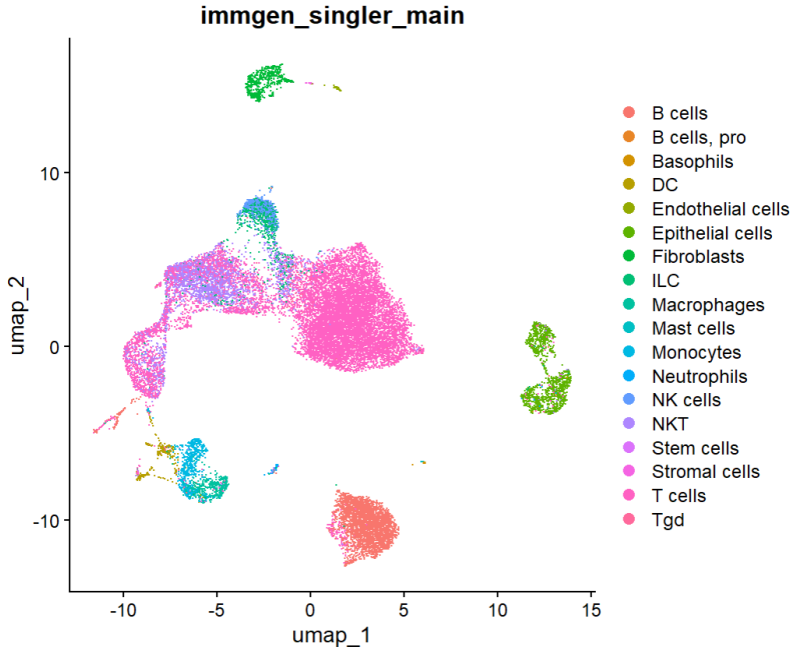
Visualise cell types within sample



Cell type map to clusters

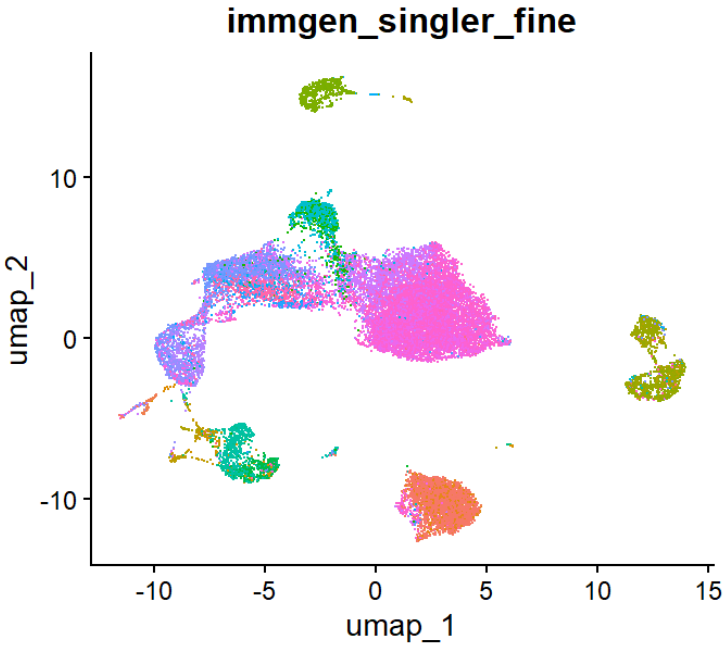
How do our cell type annotations map to our clusters we defined previously?

```
DimPlot(merged, group.by = c("immgen_singler_main"))
```



Cell type map to clusters

Exercise: plot for fine labels as well.



Use a different reference

How do our cell annotations differ if we use a different reference set?

We previously used the ImmGen dataset, let's try a different one.

```
> celldex::listReferences()
[1] "blueprint_encode"           "dice"                       "hpca"
[4] "immgen"                     "monaco_immune"              "mouse_rnaseq"
[7] "novershtern_hematopoietic"
```

This function lists all available references in the celldex package.

Let's try using the "**mouse_rnaseq**" one and see how our labels differ.

Use a different reference

The dataset contains 358 mouse RNA-seq sample annotated to 18 main cell types ("label.main"). These are split further into 28 subtypes ("label.fine"). The subtypes have also been mapped to Cell Ontology as with the ImmGen reference.

```
ref_mouserna <- celldex::MouseRNAseqData()
```

```
> ref_mouserna
class: SummarizedExperiment
dim: 21214 358
metadata(0):
assays(1): logcounts
rownames(21214): Xkr4 Rp1 ... LOC100039574 LOC100039753
rowData names(0):
colnames(358): ERR525589Aligned ERR525592Aligned ... SRR1044043Aligned
  SRR1044044Aligned
colData names(3): label.main label.fine label.ont
```

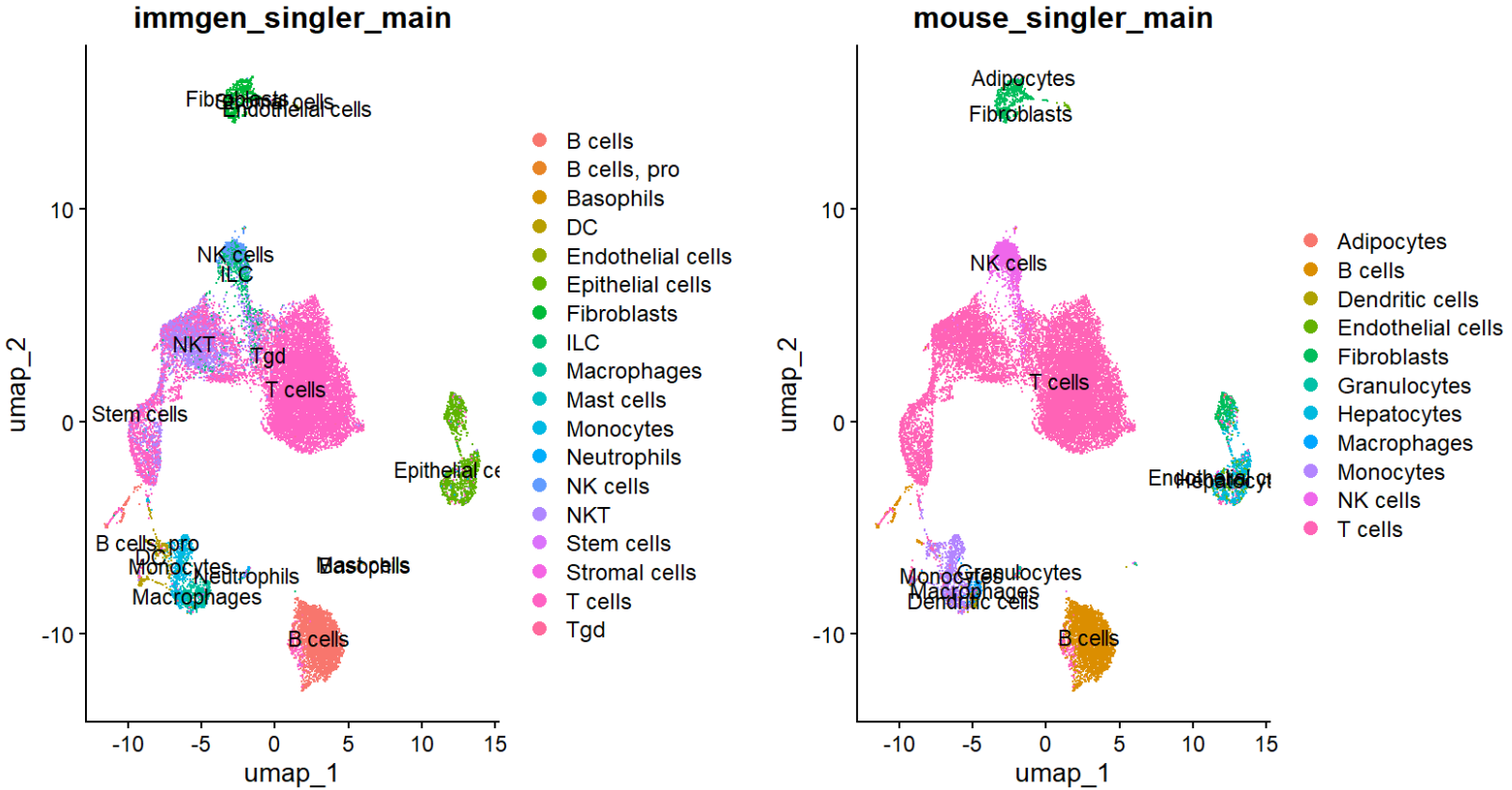
Use a different reference

The `**ref_mouserna**` object should have a similar structure to the `**ref_immgen**` we had before.

Exercise:

1. **get the predictions for our cells with both main and fine labels.**
2. **add the labels to Seurat object metadata table.**
3. **map the cell annotation to UMAP.**
4. **compare the result with ImmGen.**

Use a different reference



How to pick reference?

Your results depend heavily on which reference dataset you pick. To avoid mislabelling or missing rare cases, **it's better to use a reference that covers a wider range of possible labels than just the ones you anticipate in your own data.**

When you use reference samples from other researchers, you usually have to trust their labelling without independent verification. This blind trust is **risky** because labelling might not always be accurate.

It's also not surprising that some reference datasets perform better than others, since their underlying sample preparation may have been done with more care or higher quality standards.

How to pick reference?

It's generally best to use a reference dataset produced with the same experimental methods as your own data. However, if you're using `SingleR()` to annotate cell types that are already well separated, these technical differences are usually not a big problem.

You can also use your **own reference**, if you provide log-transformed expression data and labels for each cell.

How to pick reference?

It's generally best to use a reference dataset produced with the same experimental methods as your own data. However, if you're using `SingleR()` to annotate cell types that are already well separated, these technical differences are usually not a big problem.

You can also use your **own reference**, if you provide log-transformed expression data and labels for each cell.

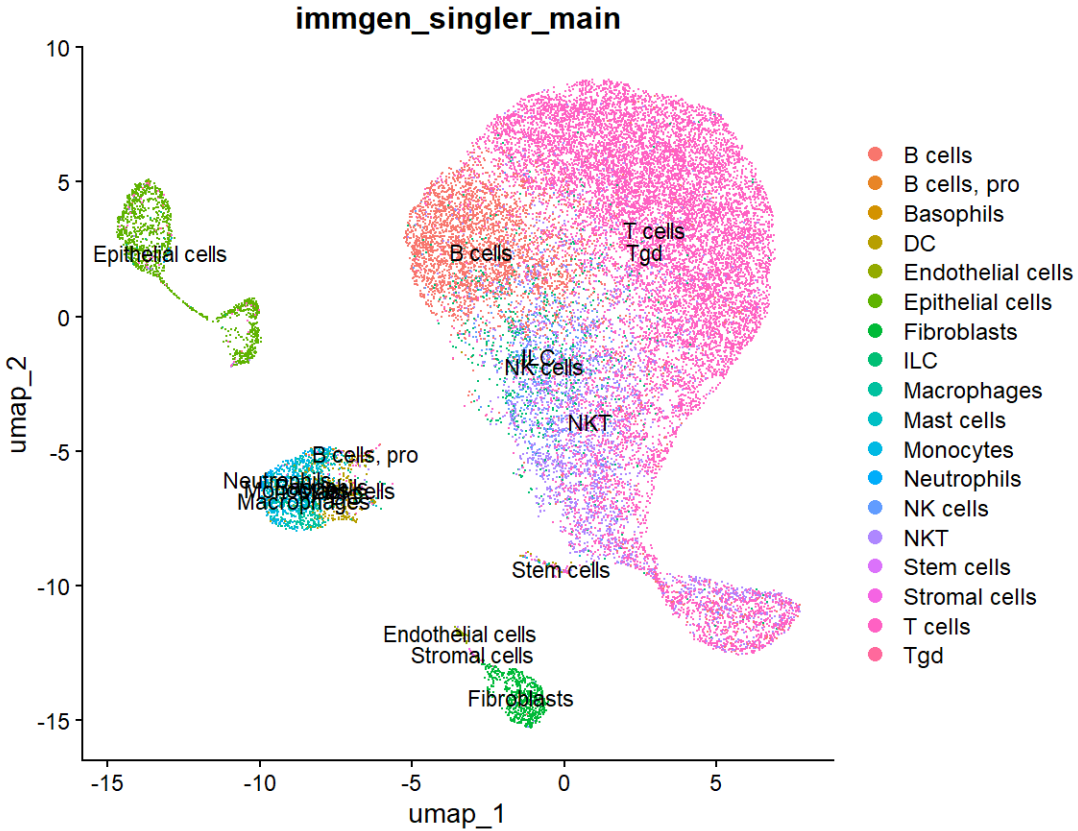
Understand clustering using cell annotation

We know that the UMAP shape changes when we use different number of PCs but why does the UMAP shape change?

Let's try creating a UMAP with only 5 PCs.

```
merged_5PC <- RunUMAP(merged, dims = 1:5)
DimPlot(merged_5PC, label = TRUE, group.by = 'immgen_singler_main')
```

Understand clustering using cell annotation



We can see that the cells form much more general clusters. For example, the immune cells are just in one big blob.

When we increase our PCs, we increase the amount of information that can be used to tease apart more specific cell types.

The distinct B cell cluster we saw earlier was only possible because we provided enough genetic expression information in the PCs we chose.

Saving data

```
saveRDS(merged, file = 'preprocessed_object.rds')
```

Thank you

Contact us

Jiajia Li
Research School of Biology

RN Robertson Building, 46 Sullivan's Creek Rd
The Australian National University
Canberra ACT 2600

E jiajia.li1@anu.edu.au

Australian National University

TEQSA PROVIDER ID: PRV12002 (AUSTRALIAN UNIVERSITY)
CRICOS PROVIDER CODE: 00120C