Supplementary Information 2: Benchmarking *SingleR*

Contents

Benchmarking *SingleR* **1**

Case study 1: GSE74923 – Kimmerling et al.[1] **1**
Obtaining data . . . 1
SingleR analysis . . . 2
SingleR score is equivalent to the number of non-zero genes . . . 5

Case study 2: 10X datasets – Zheng et al.[2] **7**
Obtaining data . . . 7
SingleR analysis . . . 7
Effect of fine-tuning . . . 12
Granular analysis of B-cells . . . 14
Comparison to other reference classification methods . . . 15
Identifying rare events . . . 17
SingleR score is association with the number of non-zero genes . . . 18
Confidence of annotations . . . 20

Case study 3: Simulating number of non-zero genes **21**

***SingleR* web tool** **23**

References **23**

Benchmarking *SingleR*

In the case studies below we use the Seurat package to process the scRNA-seq data and perform the t-SNE analysis. All visualizations are readily available through the *SingleR* web tool – http://comphealth.ucsf.edu/SingleR. The web app allows viewing the data and interactive analysis.

Case study 1: GSE74923 – Kimmerling et al.[1]

Obtaining data

A data set that was created to test the C1 platform. 194 single-cell mouse cell lines were analyzed using C1: 89 L1210 cells, mouse lymphocytic leukemia cells, and 105 mouse CD8+ T-cells. 5 cells with less than 500 non-zero genes were omitted.

The data was downloaded from GEO and read to R using the following code:

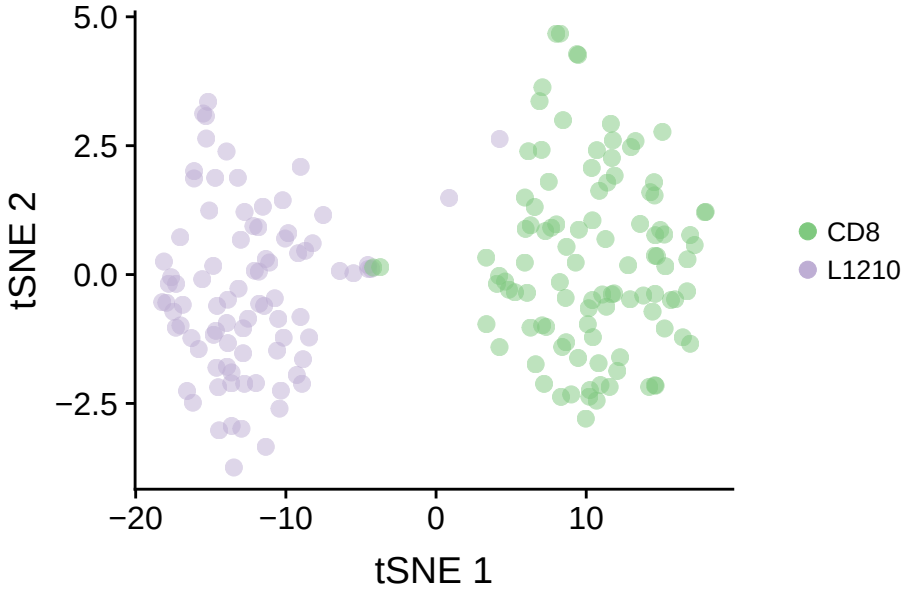
```
counts.file = 'GSE74923_L1210_CD8_processed_data.txt'
# This file was probably proccessed with Excel as there are duplicate gene names
                                        # (1-Mar, 2-Mar, etc.). They were removed manually.
annot.file = 'GSE74923_L1210_CD8_processed_data.txt_types.txt' # a table with two columns
                                      # cell name and the original identity (CD8 or L1210)
singler = CreateSinglerSeuratObject(counts.file, annot.file, 'GSE74923',
                                      variable.genes='de', regress.out='nUMI',
```

```
                                    technology='C1', species='Mouse',
                                    citation='Kimmerling et al.', reduce.file.size = F,
                                    normalize.gene.length = T)
save(singler,file='GSE74923.RData'))
```

SingleR analysis

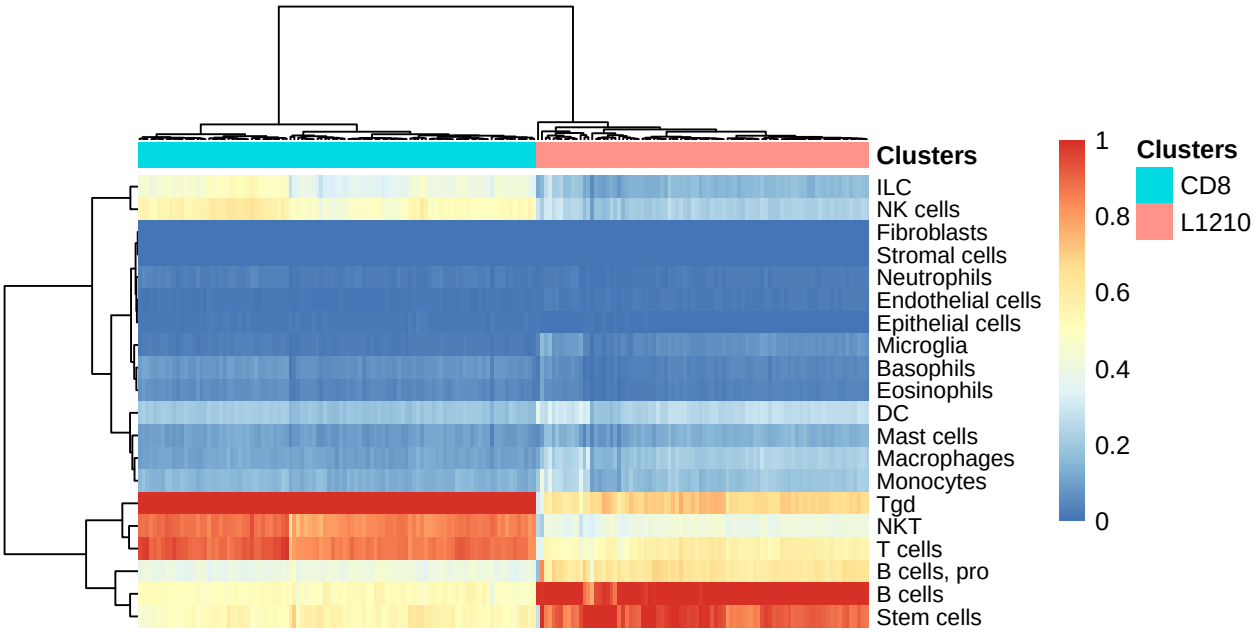
First, we look at the t-SNE plot colored by the original identities:

```
# singler$singler[[1]] is the annotations obtained by using ImmGen dataset as reference.
# singler$singler[[2]] is based on the Mouse-RNAseq datasets.
load (file.path(path,'GSE74923.RData'))
out = SingleR.PlotTsne(singler$singler[[1]]$SingleR.single,
      singler$meta.data$xy, do.label = FALSE, do.letters = F,
      labels=singler$meta.data$orig.ident,label.size = 6,
      dot.size = 3)
out$p
```



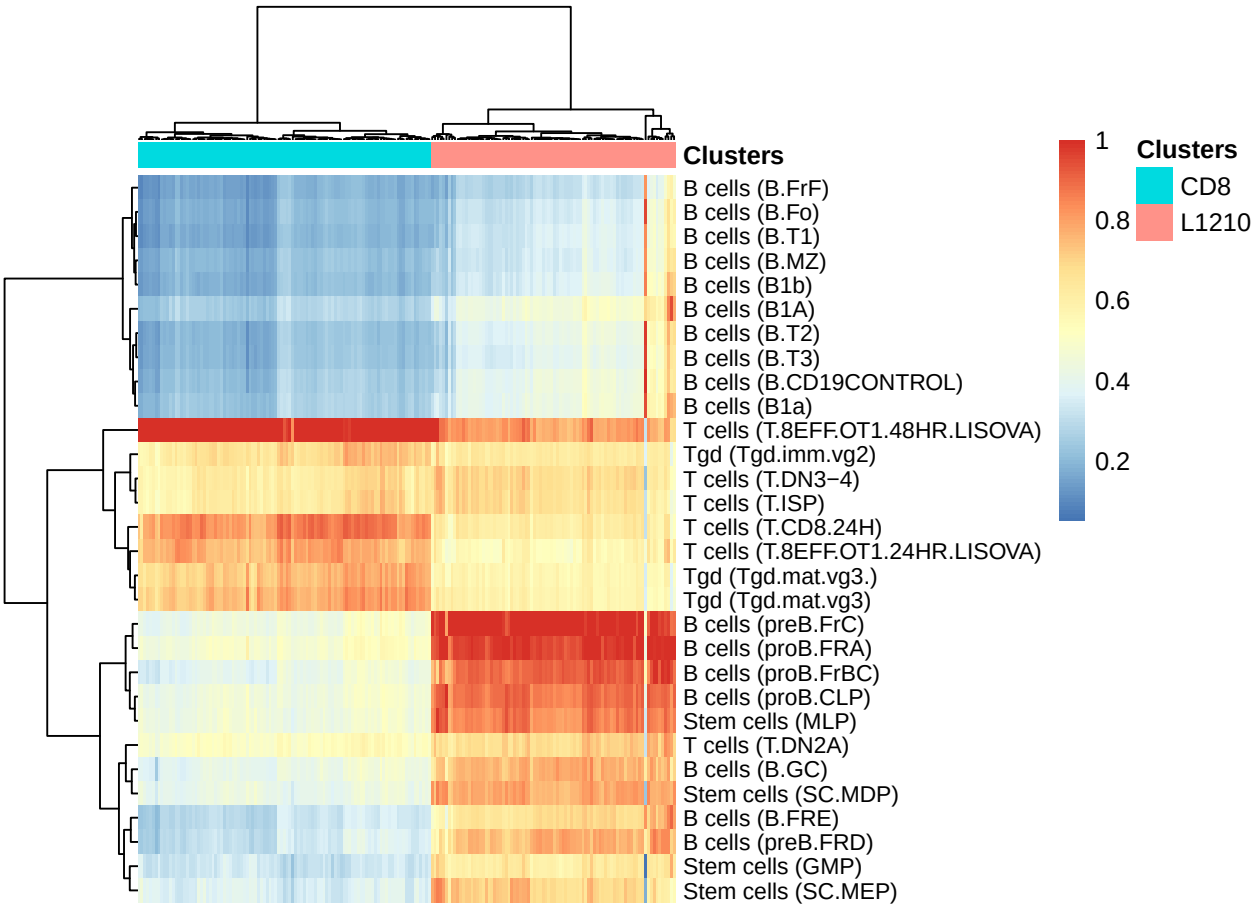
We can then observe the classification by a heatmap of the aggregated scores using *SingleR* with reference to Immgen. These scores are before fine-tuning. We can view this heatmap by the main cell types:

```
SingleR.DrawHeatmap(singler$singler[[1]]$SingleR.single.main, top.n = Inf,
                    clusters = singler$meta.data$orig.ident)
```



Or by all cell types (presenting the top 30 cell types):

```
SingleR.DrawHeatmap(singler$singler[[1]]$SingleR.single, top.n = 30,
                    clusters = singler$meta.data$orig.ident)
```



We can see that the L1210 cells were classified strongly to (mostly) 2 types of B-cell progenitors. We can see that the CD8 cells were mostly correlated with a specific activation of effector CD8+ T-cells. Interestingly, SingleR suggests that one L1210 cell is now more similar to fully differentiated B-cells than to a progenitor.

Another interesting application of this heatmap is the ability to cluster the cells, not by their gene expression profile, but by their similarity to all cell types in the database:

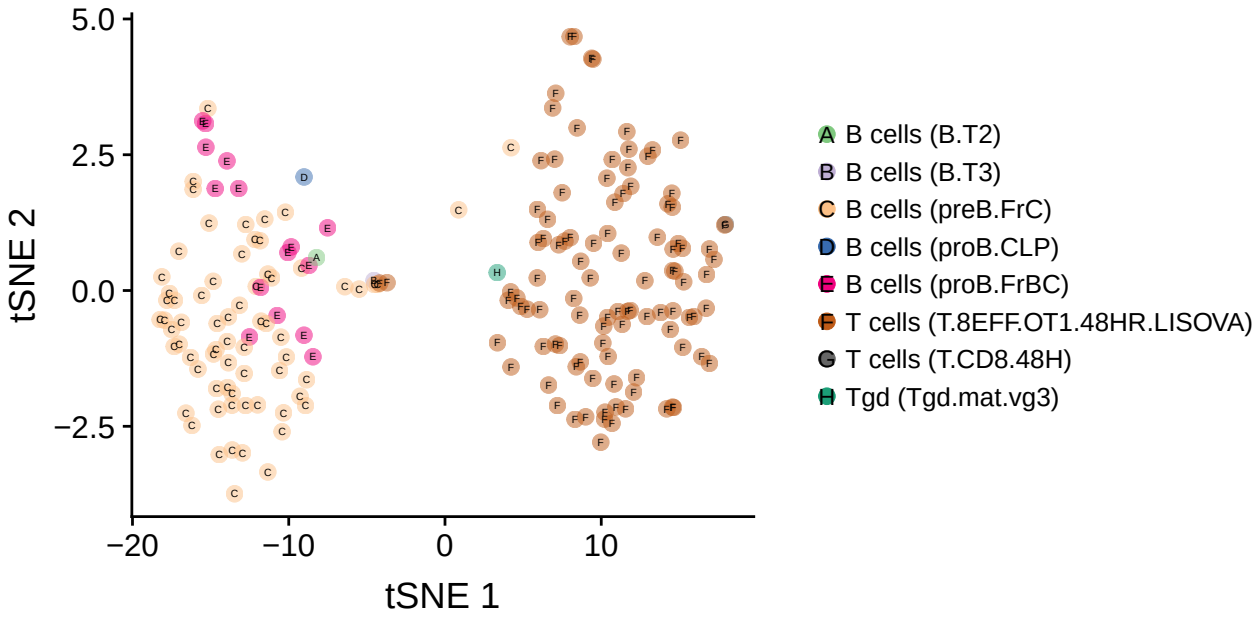
```
K = SingleR.Cluster(singler$singler[[1]]$SingleR.single,num.clusts = 2)
kable(table(K$cl,singler$meta.data$orig.ident),row.names = T)
```

	CD8	L1210
1	0	86
2	103	0

Here it is not very interesting, but we will see later that this clustering capability may be useful, especially for identifying new cell types. We use it in the manuscript and find an intermediate, uncharacterized macrophage state.

Finally, we present the annotations in a t-SNE plot:

```
out = SingleR.PlotTsne(singler$singler[[1]]$SingleR.single,
        singler$meta.data$xy,do.label=FALSE,
        do.letters = T,labels = singler$singler[[1]]$SingleR.single$labels,
        label.size = 4, dot.size = 3)
out$p
```



We can see that *SingleR* correctly annotated all the L1210 as types of B-cell, almost exclusively as B-cell progenitors. On the other hand, all CD8 cells were correctly annotated to CD8+ T-cells. It is important to remember that there are 253 types that *SingleR* could have chosen from, but it correctly chose the most relevant cell types. Interestingly, the tSNE plot incorrectly positioned cells in the wrong cluster, but *SingleR* was not affected by this.

It is often useful to view the annotations in a table compared to the original identities:

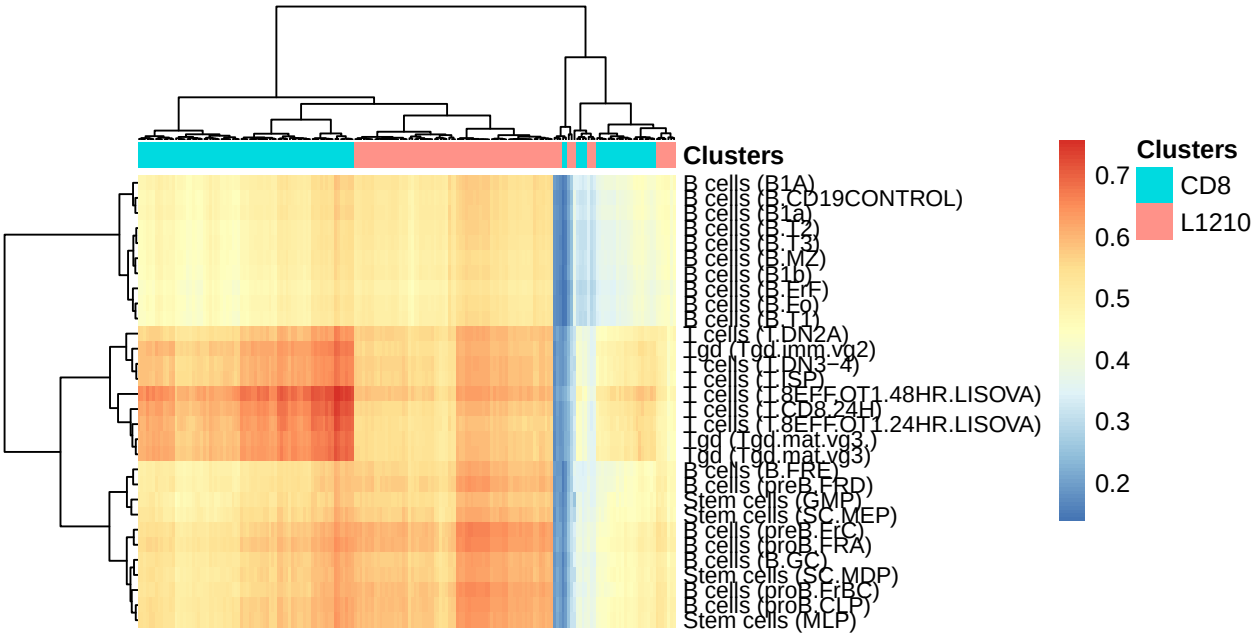
```
kable(table(singler$singler[[1]]$SingleR.single$labels,singler$meta.data$orig.ident))
```

	CD8	L1210
B cells (B.T2)	0	1
B cells (B.T3)	0	1
B cells (preB.FrC)	0	68
B cells (proB.CLP)	0	1
B cells (proB.FrBC)	0	15
T cells (T.8EFF.OT1.48HR.LISOVA)	101	0
T cells (T.CD8.48H)	1	0
Tgd (Tgd.mat.vg3)	1	0

SingleR score is correlated with the number of nonzero genes

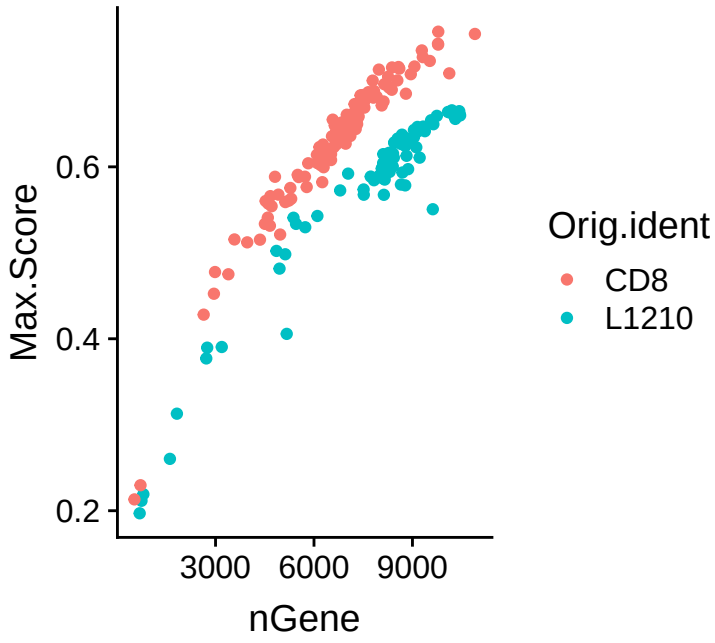
The heatmap presented above may be misleading, since each column is normalized to a score between 0 and 1. Thus, a single cell may have low correlations with multiple cell types, and none of them are the accurate cell type, but in the heatmap they will all be red. Without normalization the data looks like this:

```
SingleR.DrawHeatmap(singler$singler[[1]]$SingleR.single,top.n = 30,
        normalize = F,clusters = singler$meta.data$orig.ident)
```



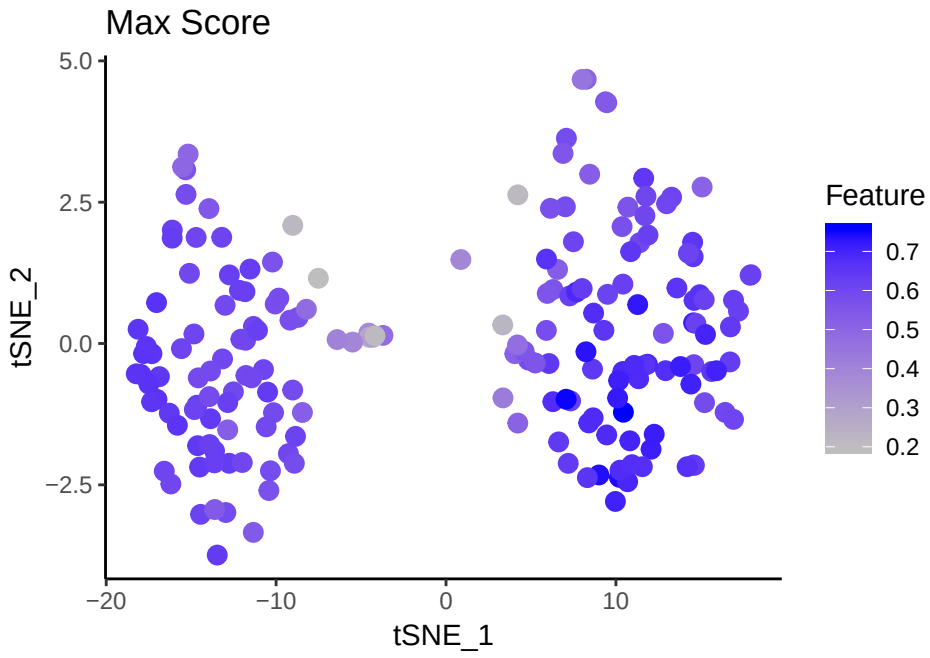
We can see that there is a group of cells that have lower scores with all cell types. What causes these cells to have low correlations?

```
df = data.frame(Max.Score=apply(singler$singler[[1]]$SingleR.single$scores,1,max),
                nGene=singler$seurat@meta.data$nGene,Orig.ident=singler$meta.data$orig.ident)
ggplot(df,aes(x=nGene,y=Max.Score,color=Orig.ident))+geom_point()
```



This plot of the number of non-zero genes (nGene) vs. the top *SingleR* score in a cell shows that the SingleR score is dependent on nGene. However, as we have seen above SingleR was able to correctly annotate those cells, despite the lower number of nGene. This is in contrast to the Seurat t-SNE plot, which misplaced those cells as can be seen in the plot below, which colors the cells based on the Max.Score:

```
SingleR.PlotFeature(singler$singler[[1]]$SingleR.single,singler$seurat,'MaxScore',dot.size=3)
```



In the next case study we will examine SingleR's ability to correctly annotate cells using a criterion of correlation "outlierness" that is independent of nGene.

Case study 2: 10X datasets – Zheng et al.[2]

Obtaining data

Here we analyzed a unique human dataset of sorted immune cell types that were analyzed using the 10X platform. We obtained this data from https://support.10xgenomics.com/single-cell-gene-expression/datasets, and processed it with the *SingleR* pipeline. To reduce computation times and to make the analyses simpler, we randomly selected 1000 cells with >200 non-zero genes from 10 cell populations, using the following code:

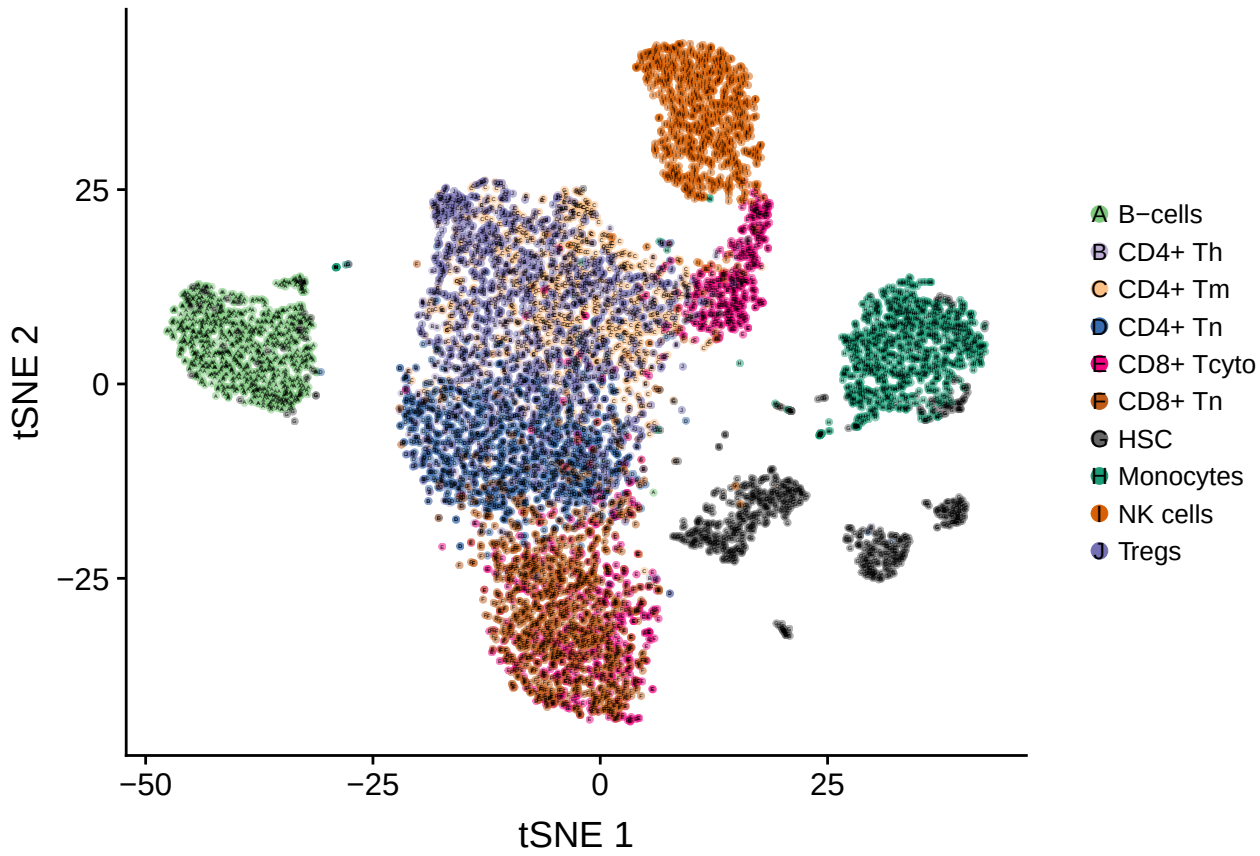
```
# path/10X/ contains a directory for each of the expriments, each with the three 10X files.
dirs = dir(paste0(path,'/10X'),full.names=T)
tenx = Combine.Multiple.10X.Datasets(dirs,random.sample=1000,min.genes=200)
singler = CreateSinglerSeuratObject(tenx$sc.data, tenx$orig.ident, 'Zheng',
                                    variable.genes='de',regress.out='nUMI',
                                    technology='10X', species='Human',
                                    citation='Zheng et al.', reduce.file.size = F,
                                    normalize.gene.length = F)
save(singler,file='10x (Zheng) - 10000cells.RData'))
```

SingleR analysis

First, we plot the original identities:

Note: When there are more than a handful of cell types, its is hard to distinguish them by color. Shapes are also hard to distinguish in small sizes. Thus, we use distinct letters for each cell type/color. However, since the cells are small, this requires substantial zooming in. An alternative is to use our SingleR web tool or running the code and plotting with ggplotly, which both allow to hover over the cells and see its label.

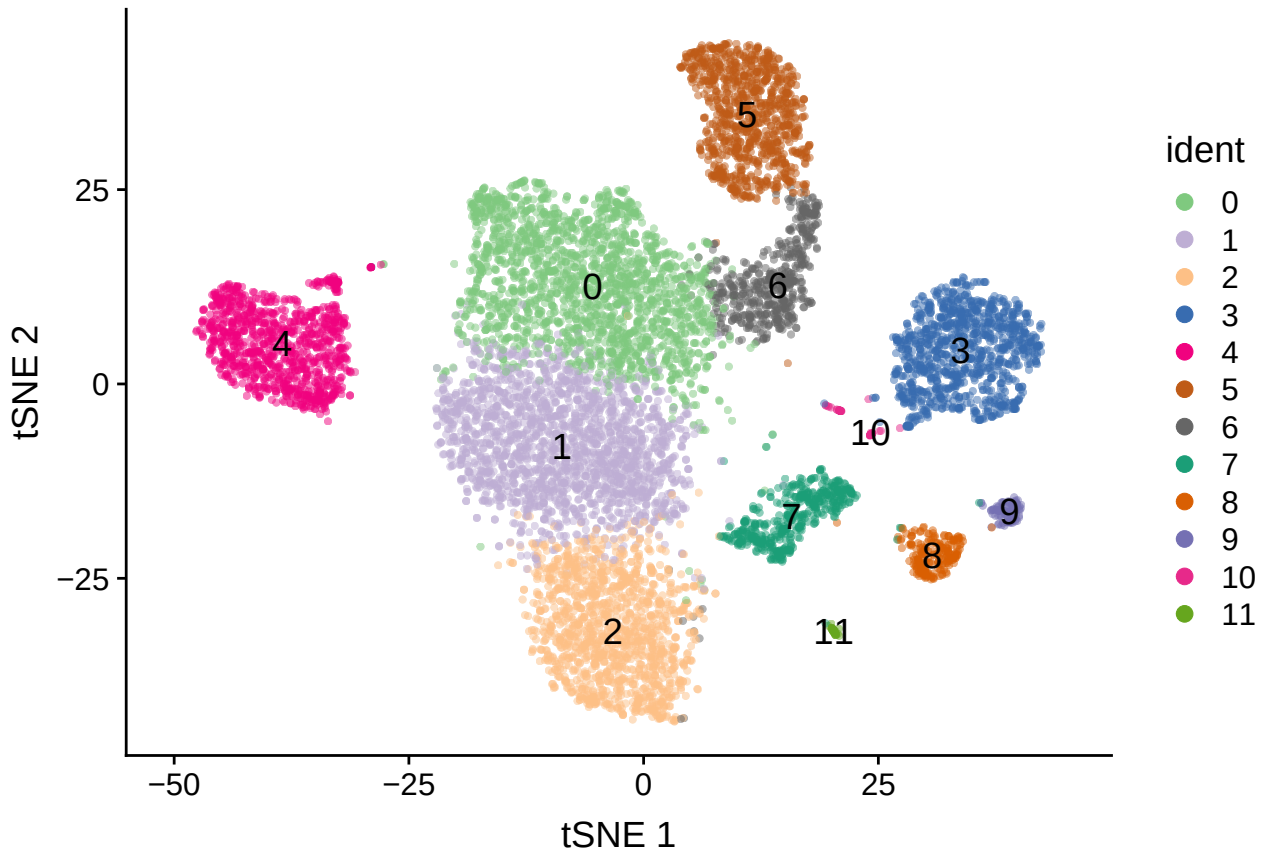
```
load (file.path(path,'SingleR.Zheng.200g.RData'))
out = SingleR.PlotTsne(singler$singler[[1]]$SingleR.single,
                       singler$meta.data$xy,do.label = F,
                       do.letters = T,labels = singler$meta.data$orig.ident,
                       dot.size = 1.3,alpha=0.5,label.size = 6)
out$p
```



We can see that the tSNE plots allows to distinguish most cell types, but the CD4+ T-cell subsets are blurred together.

We next look at the Seurat clustering:

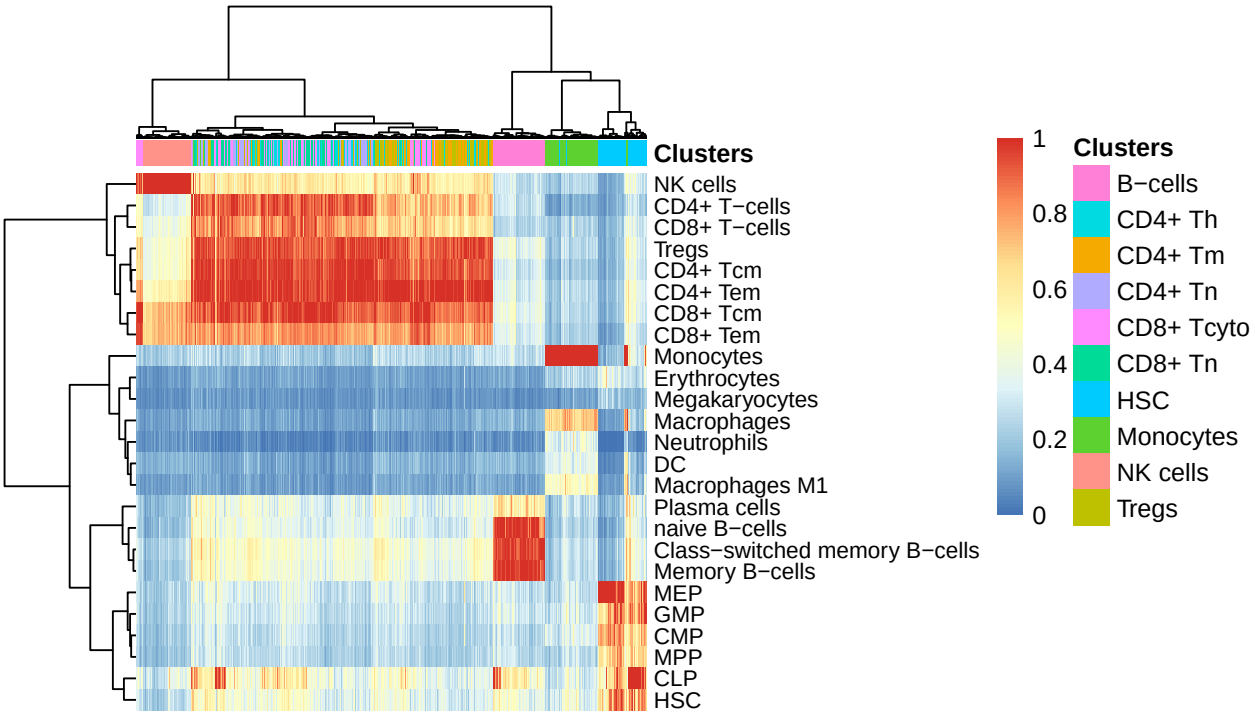
```
out = SingleR.PlotTsne(singler$singler[[1]]$SingleR.single,
                       singler$meta.data$xy,do.label = T,
                       do.letters = F,labels=singler$seurat@ident,
                       dot.size = 1.3,label.size = 5,alpha=0.5)
out$p
```



We can see that the clustering performs relatively well; however, regulatory T-cells are completely dissolved in the memory T-cells cluster.

SingleR using Blueprint+ENCODE (BE) as reference produced the following annotations before fine-tuning:

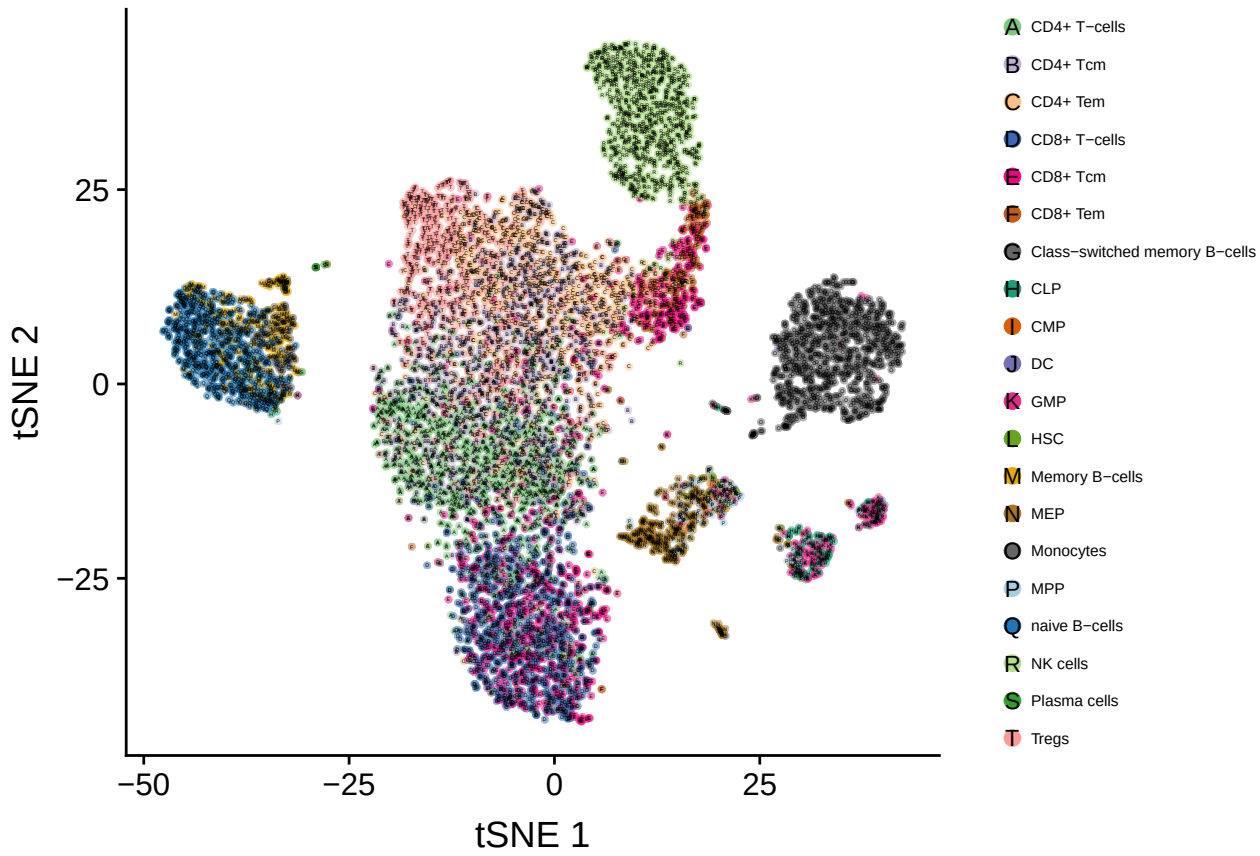
```
# Note the use of the second iterm in the the singler$singler list to use
# the Blueprint+ENCODE reference. The first item is HPCA.
# use singler$singler[[i]]$about for meta-data on the reference.
SingleR.DrawHeatmap(singler$singler[[2]]$SingleR.single,top.n=25,
                    clusters = singler$meta.data$orig.ident)
```



We can see that before fine-tuning, there is strong blurring between T-cells states, which cannot be distinguished.

However, with fine-tuning we obtain the following annotations:

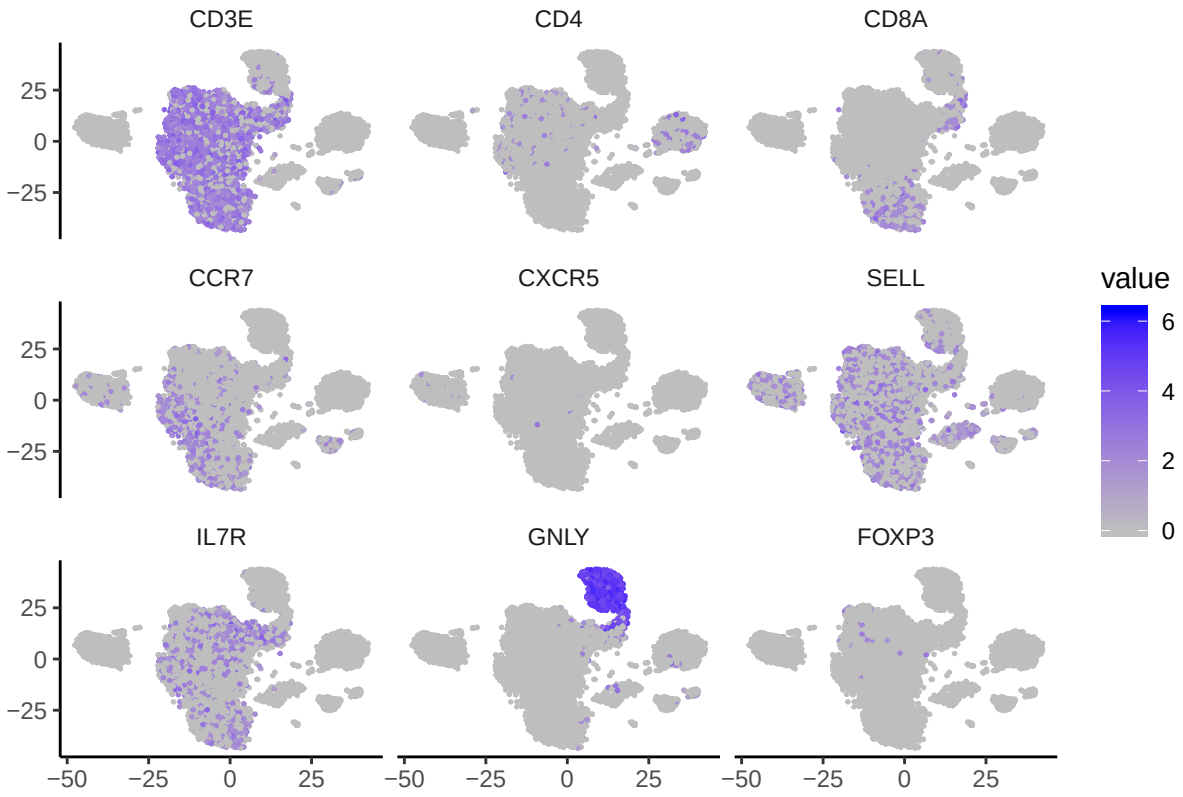
```
out = SingleR.PlotTsne(singler$singler[[2]]$SingleR.single,
                       singler$meta.data$xy,do.label=FALSE,
                       do.letters =T,labels=singler$singler[[2]]$SingleR.single$labels,
                       dot.size = 1.3, font.size = 6)
out$p
```



By observing the colors we can see that the CD4+ T-cell cluster can roughly be divided in to 4 states, from naive CD4+ T-cells on the bottom (green), to central memory and effector memory CD4+ T-cells in the middle (purple and orange) and Tregs in the top (pink), in accordance with the original identities. While it is not perfect, it provides us a much more granular view of the cell states without the need to go over many markers that might not be in the data at all and whose interpretation is often confusing; for example:

```
genes.use = c('CD3E','CD4','CD8A','CCR7','CXCR5','SELL','IL7R','GNLY','FOXP3')

df = data.frame(x=singler$meta.data$xy[,1],
                y=singler$meta.data$xy[,2],
                t(as.matrix(singler$seurat@data[genes.use,])))
df = melt(df,id.vars = c('x','y'))
ggplot(df,aes(x=x,y=y,color=value)) +
  geom_point(size=0.3)+scale_color_gradient(low="gray", high="blue") +
  facet_wrap(~variable,ncol=3) +theme_classic()+xlab('')+ylab('')+
  theme(strip.background = element_blank())
```

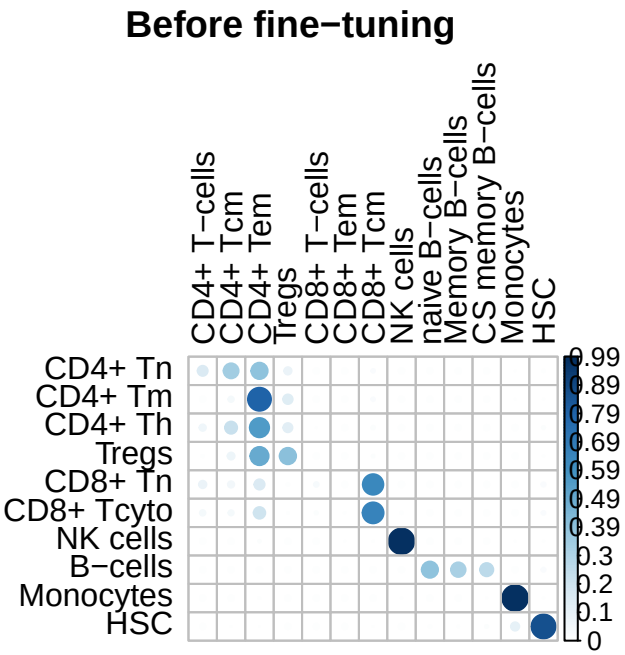


Effect of fine-tuning

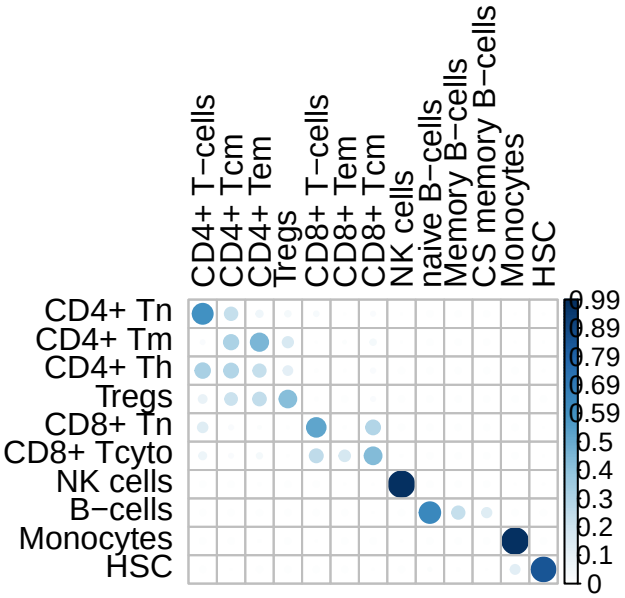
To compare the results before and after fine-tuning we can look at the following plots (rows - original identities, columns - *SingleR* labels):

```
singler$singler[[2]]$SingleR.single$labels1 =
  gsub('Class-switched','CS',singler$singler[[2]]$SingleR.single$labels1)
singler$singler[[2]]$SingleR.single$labels =
  gsub('Class-switched','CS',singler$singler[[2]]$SingleR.single$labels)
hsc = c('CLP','CMP','GMP','MEP','MPP')
singler$singler[[2]]$SingleR.single$labels1[
  singler$singler[[2]]$SingleR.single$labels1 %in% hsc] = 'HSC'
singler$singler[[2]]$SingleR.single$labels[
  singler$singler[[2]]$SingleR.single$labels %in% hsc] = 'HSC'

order1 = c('CD4+ T-cells','CD4+ Tcm','CD4+ Tem','Tregs','CD8+ T-cells',
           'CD8+ Tem','CD8+ Tcm','NK cells','naive B-cells',
           'Memory B-cells','CS memory B-cells','Monocytes','HSC')
order2 = c('CD4+ Tn','CD4+ Tm','CD4+ Th','Tregs','CD8+ Tn','CD8+ Tcyto',
           'NK cells','B-cells','Monocytes','HSC')
a = table(singler$meta.data$orig.ident,singler$singler[[2]]$SingleR.single$labels1)
a = a[order2,order1]
corrplot(a/rowSums(a),is.corr=F,tl.col='black',title = 'Before fine-tuning',
         mar=c(0,0,2,0))
```




```
b = table(singler$meta.data$orig.ident,singler$singler[[2]]$SingleR.single$labels)
b = b[order2,order1]
corrplot(b/rowSums(b),is.corr=F,tl.col='black',title = 'After fine-tuning',
         mar=c(0,0,2,0))
```



We can see that before fine-tuning many CD8+ T-cells were annotated as CD4+ (46.5% of CD8+ T-cells). This phenomenon that cells showed highest correlation with CD4+ T-cells has also been reported at the original Zheng et al. manuscript. After the fine-tuning this number is reduced to 19.25%. In addition, naive CD8+ T-cells were not annotated as such before fine-tuning but have been correctly annotated after fine-tuning.

{#CorrectLabeling}In summary, SingleR annotated 84.3% of the cells to main cell types in accordance with the original identities, not far from the expected purity of the sorting:

```
ident = as.character(singler$meta.data$orig.ident)
ident[grepl('CD4',ident)]='CD4+ T-cells'
ident[ident=='Tregs']='CD4+ T-cells'
ident[grepl('CD8',ident)]='CD8+ T-cells'
kable(table(singler$singler[[2]]$SingleR.single.main$labels,ident))
```

	B-cells	CD4+ T-cells	CD8+ T-cells	HSC	Monocytes	NK cells
B-cells	999	2	0	25	8	0
CD4+ T-cells	0	2743	127	1	1	0
CD8+ T-cells	1	1245	1868	4	3	15
DC	0	0	0	1	1	0
HSC	0	6	0	850	1	2
Monocytes	0	2	0	119	981	1
NK cells	0	2	5	0	5	982

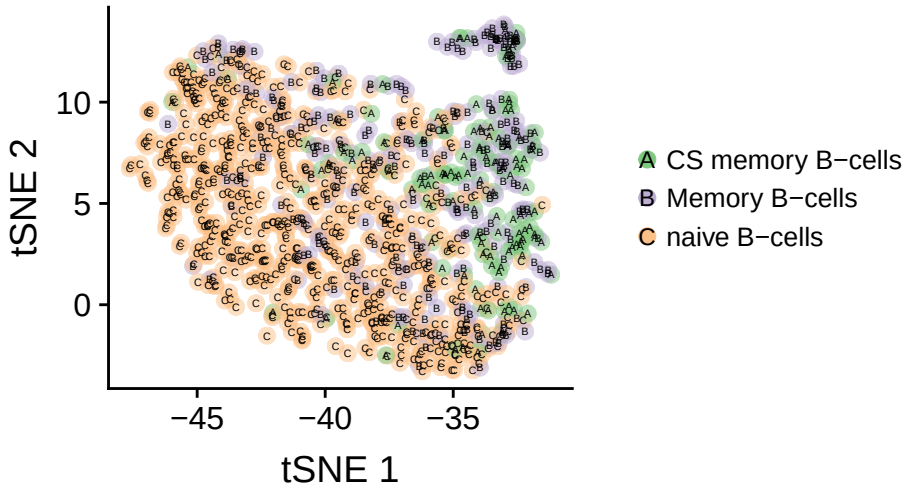
```
sum(ident==singler$singler[[2]]$SingleR.single.main$labels)/10000
```

```
## [1] 0.8423
```

Granular analysis of B-cells

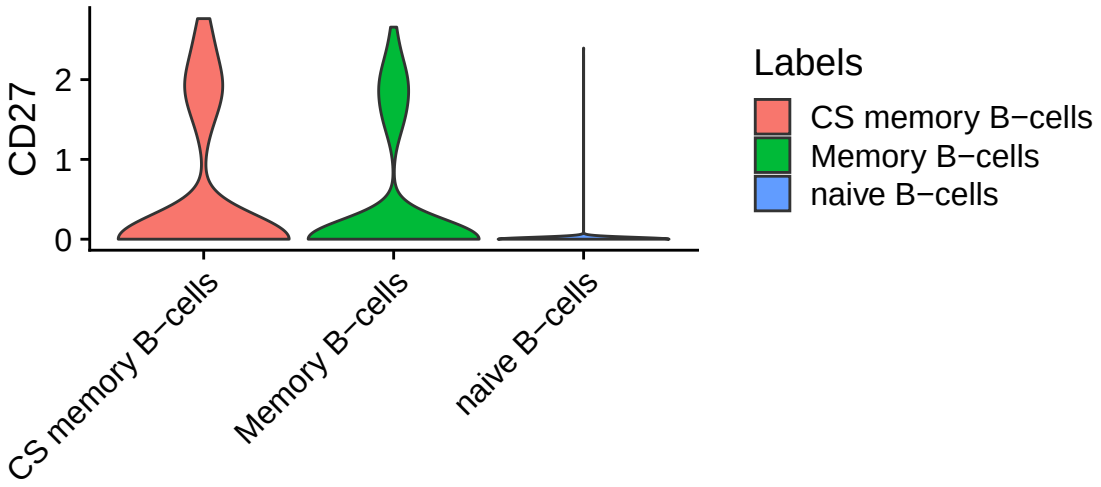
Interestingly, *SingleR* suggests a more granular view of the B-cell cluster, splitting it to naive and memory B-cells, which seems to agree with the t-SNE plot structure:

```
bcells = SingleR.Subset(singler,grepl('B-cells',
                                      singler$singler[[2]]$SingleR.single$labels))
out = SingleR.PlotTsne(bcells$singler[[2]]$SingleR.single,
                       bcells$meta.data$xy,
                       dot.size = 3,alpha=0.5)
out$p
```



And by CD27 expression (a marker of memory B-cells):

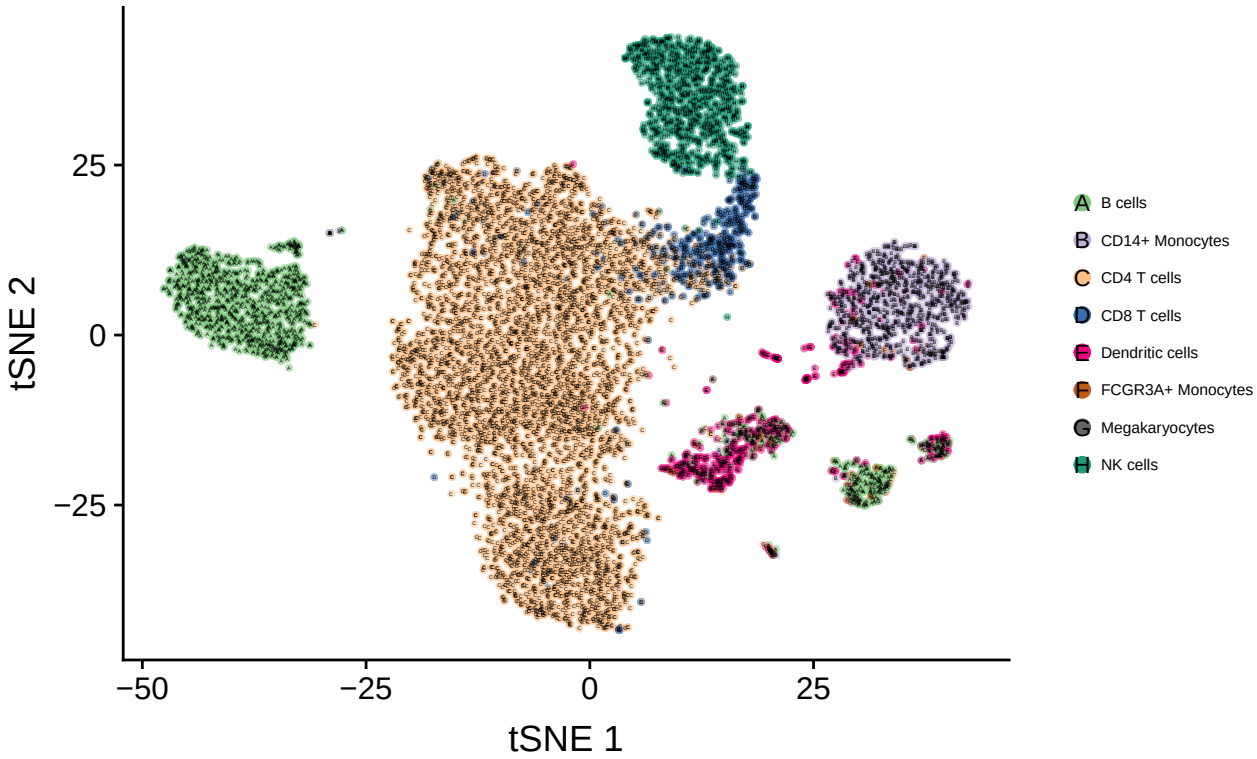
```
df = data.frame(CD27=bcells$seurat@data['CD27',],
                Labels=bcells$singler[[2]]$SingleR.single$labels)
ggplot(df,aes(x=Labels,y=CD27,fill=Labels))+geom_violin(scale='width') +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))+xlab('')
```



Comparison to other reference classification methods

Kang et al., Nature Biotechnology[3] used sets of markers learned from scRNA-seq PBMCs (from Zheng et al.) to annotate single-cells:

```
out = SingleR.PlotTsne(singler$singler[[2]]$SingleR.single,
                       singler$meta.data$xy,do.label=FALSE,
                       do.letters =T,labels=singler$other[,'Kang'],
                       dot.size = 1.3, font.size = 6)
out$p
```



We can see that this method has limited usability, as it cannot differentiate CD4+ and CD8+ T-cells. Note that data generated with 10X was used to create the reference matrix in this method, and it was specifically

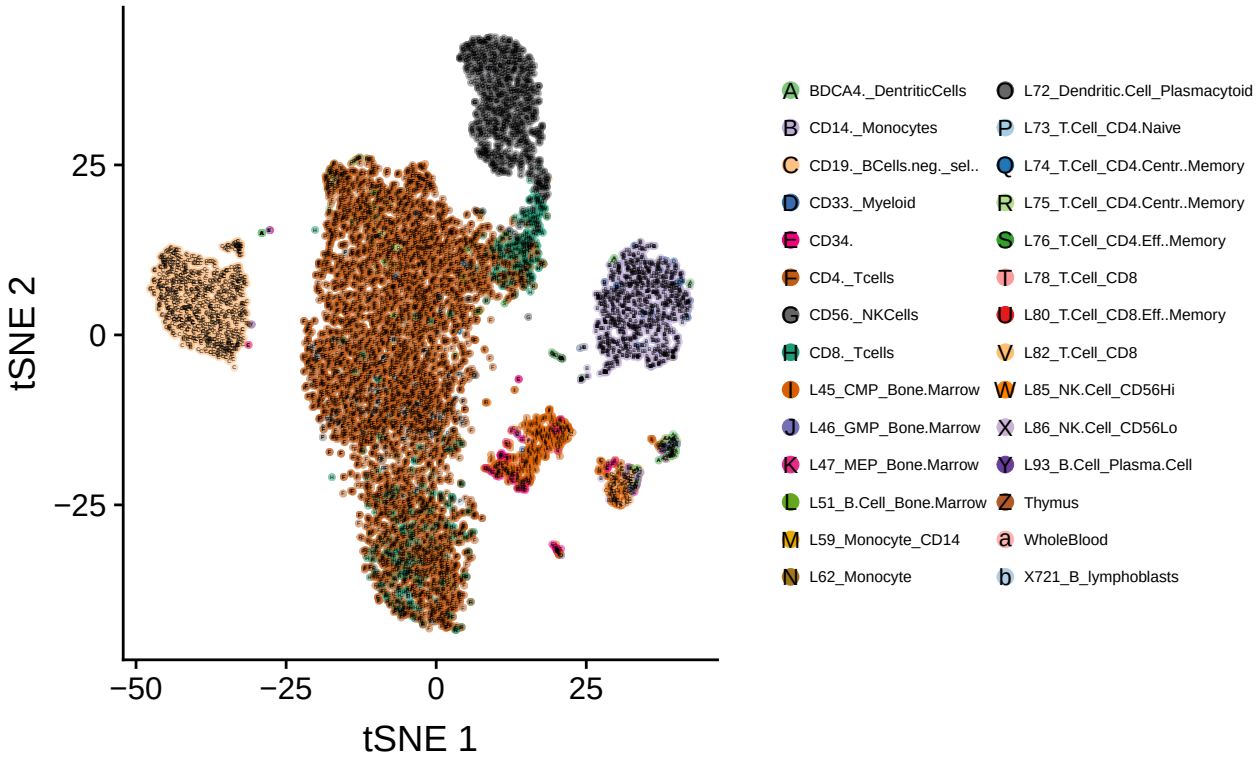
trained for immune subsets in blood.

A bulk reference-based method by Li et. al, Nature Genetics[4] used a reference-based approach, but without fine-tuning:

```
library(RCA)
tpm_data = TPM(as.matrix(singler$seurat@data),human_lengths)
data_obj = dataConstruct(tpm_data);
data_obj = geneFilt(obj_in = data_obj);
data_obj = cellNormalize(data_obj);
data_obj = dataTransform(data_obj,"log10");
rownames(data_obj$fpkm_transformed) = toupper(rownames(data_obj$fpkm_transformed))
data_obj = featureConstruct(data_obj,method = "GlobalPanel");
scores = data_obj$fpkm_for_clust
RCA.annot = rownames(scores)[apply(scores,2,which.max)]
n = table(RCA.annot)
RCA.annot[RCA.annot %in% names(n)[n<5]] = 'Other (N<5)'
names(RCA.annot) = colnames(tpm_data)

singler$other = cbind(singler$other,RCA.annot)
colnames(singler$other) = c('Kang','RCA')
```

```
out = SingleR.PlotTsne(singler$singler[[2]]$SingleR.single,
                       singler$meta.data$xy,do.label=FALSE,
                       do.letters =T,labels=singler$other[,'RCA'],
                       dot.size = 1.3, font.size = 6)
out$p
```

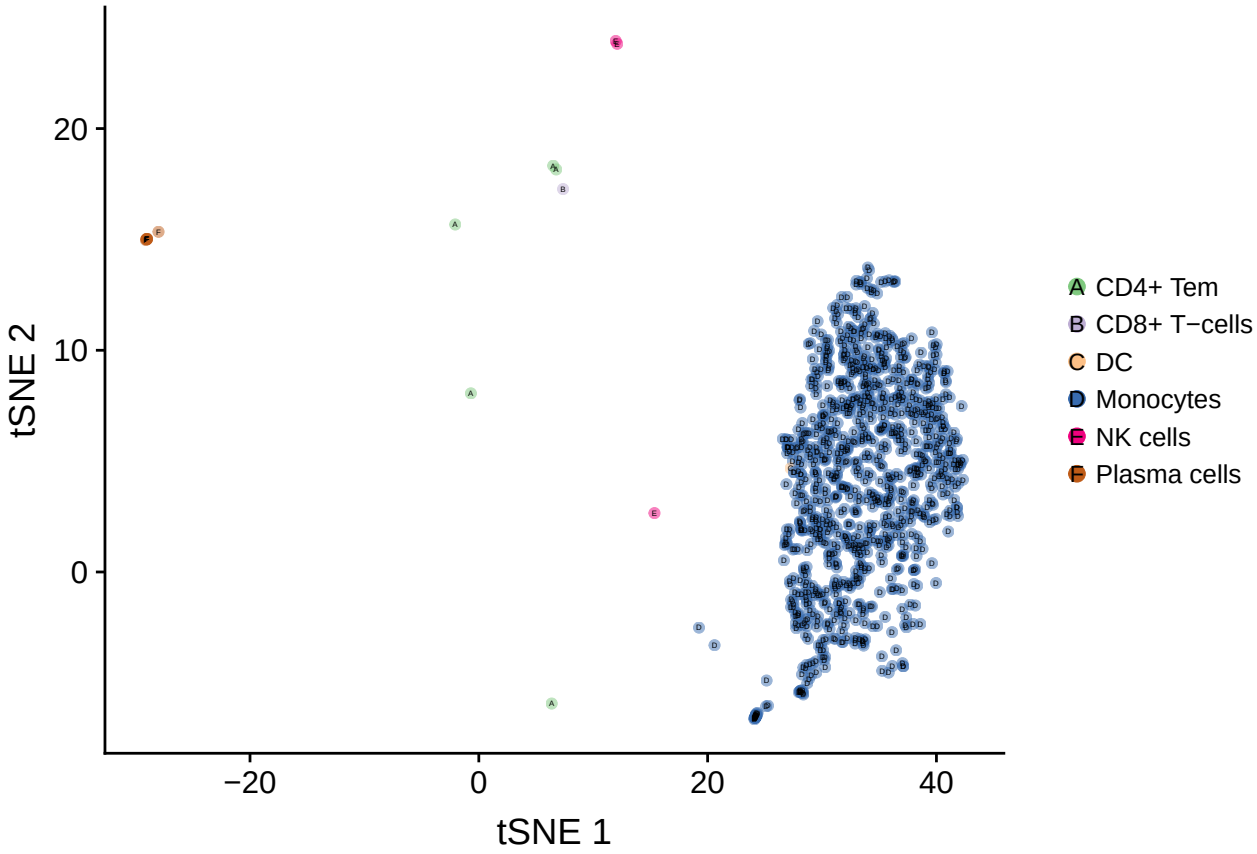


Again, we can see that the reference is not able to distinguish CD4+ and CD8+ T-cells without fine-tuning.

Identifying rare events

SingleR also allows to detect rare events. For example, lets take a deeper look at the sorted monocytes:

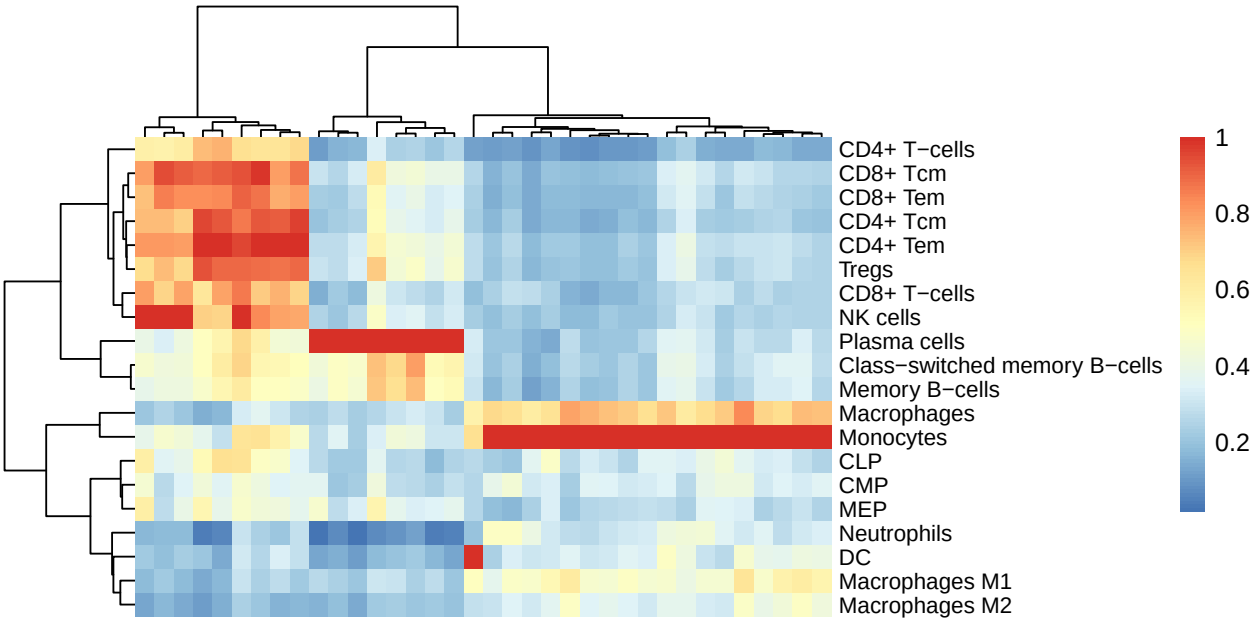
```
monocytes = SingleR.Subset(singler,singler$meta.data$orig.ident=='Monocytes')
out = SingleR.PlotTsne(monocytes$singler[[2]]$SingleR.single,
                       monocytes$meta.data$xy,do.label=F,
                       do.letters =T, dot.size = 2)
out$p
```



We can see that the t-SNE plot already suggests that there are 18 cells that are not part of the main cluster (which means that the sorting purity was ~98%). *SingleR* detected those cells to be plasma cells (8 cells), T-cells (2 cells), 3 NK cell and 1 DC. Is *SingleR* correct?

For presentation purposes we only plot 18 cells from the main cluster + the 18 other cells:

```
cells.use = c(sample(which(monocytes$singler[[2]]$SingleR.single$labels=='Monocytes'),18),
              which(monocytes$singler[[2]]$SingleR.single$labels!='Monocytes'))
SingleR.DrawHeatmap(monocytes$singler[[2]]$SingleR.single,top.n = 20,cells.use=cells.use)
```

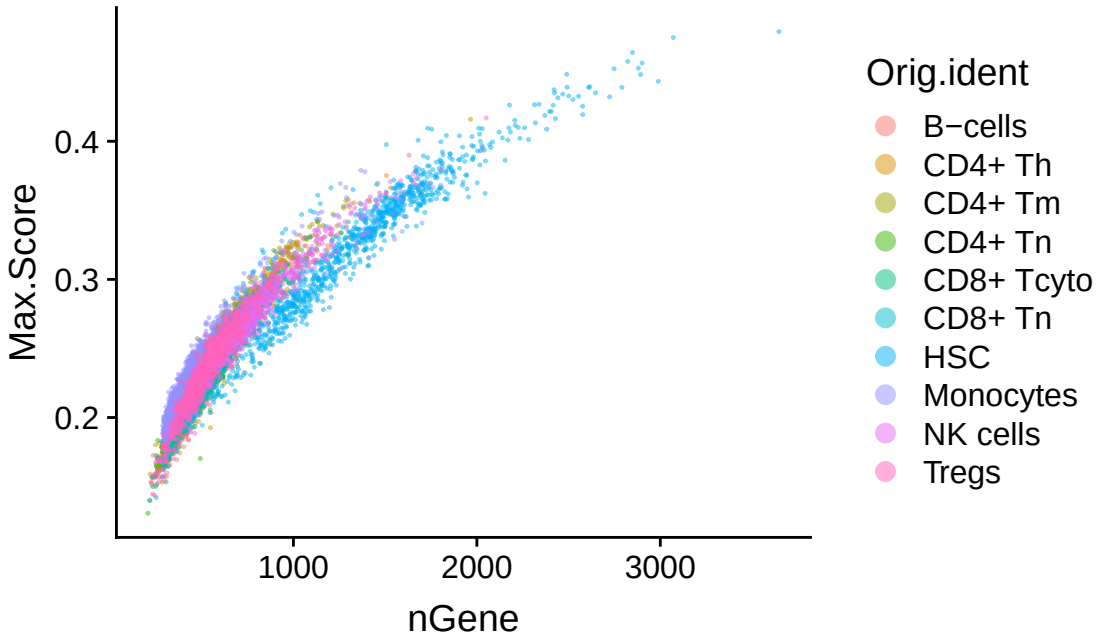


We can see that *SingleR* is quite convinced in its calls, giving low monocyte scores to those cells. Using markers for rare cell types (at least in the monocytes sorted cells) is problematic, since marker-based analysis is focused on clusters and not on single cells.

SingleR score is association with the number of non-zero genes

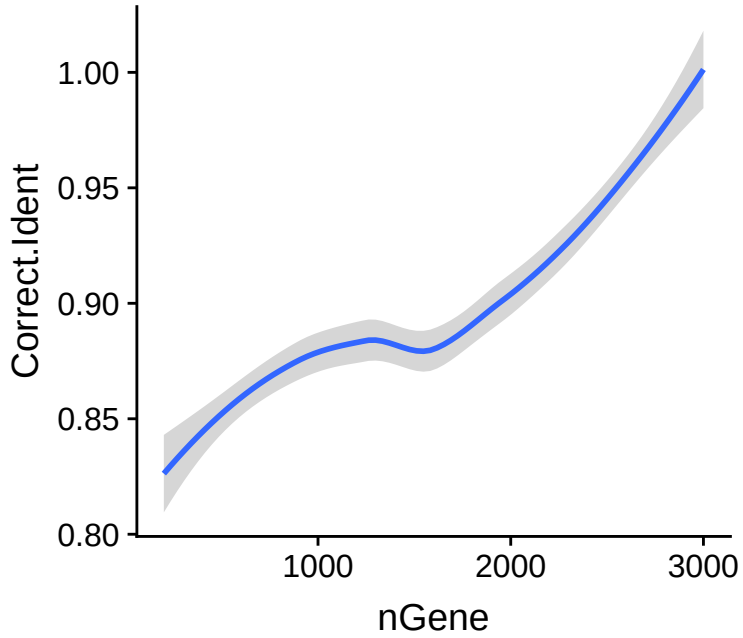
Here we used a threshold of 200 non-zero genes (nGenes). As in case study 1, there is a strong correlation between nGenes and the max score per cell:

```
nGene=singler$seurat@meta.data$nGene
df = data.frame(Max.Score=apply(singler$singler[[1]]$SingleR.single$scores,1,max),
                nGene=nGene,Orig.ident=singler$meta.data$orig.ident)
ggplot(df,aes(x=nGene,y=Max.Score,color=Orig.ident))+geom_point(size=0.2,alpha=0.5)+
  guides(color = guide_legend(override.aes = list(size = 3)))
```



The question is whether cells with low 'max-score' are less reliable. We see that for some degree this is true - using the metric of 'correct' labeling of main cell types, we can see that with more nGenes the annotations tend to be more accurate:

```
Correct.Ident = unlist(lapply(seq(from=200,to=3000,by=50),
FUN=function(x) {
  A=nGene>=x
  sum(ident[A]==singler$singler[[2]]$SingleR.single.main$labels[A])/sum(A)}
))
df = data.frame(nGene=seq(from=200,to=3000,by=50),Correct.Ident)
ggplot(df,aes(x=nGene,y=Correct.Ident))+geom_smooth(method = 'loess')
```



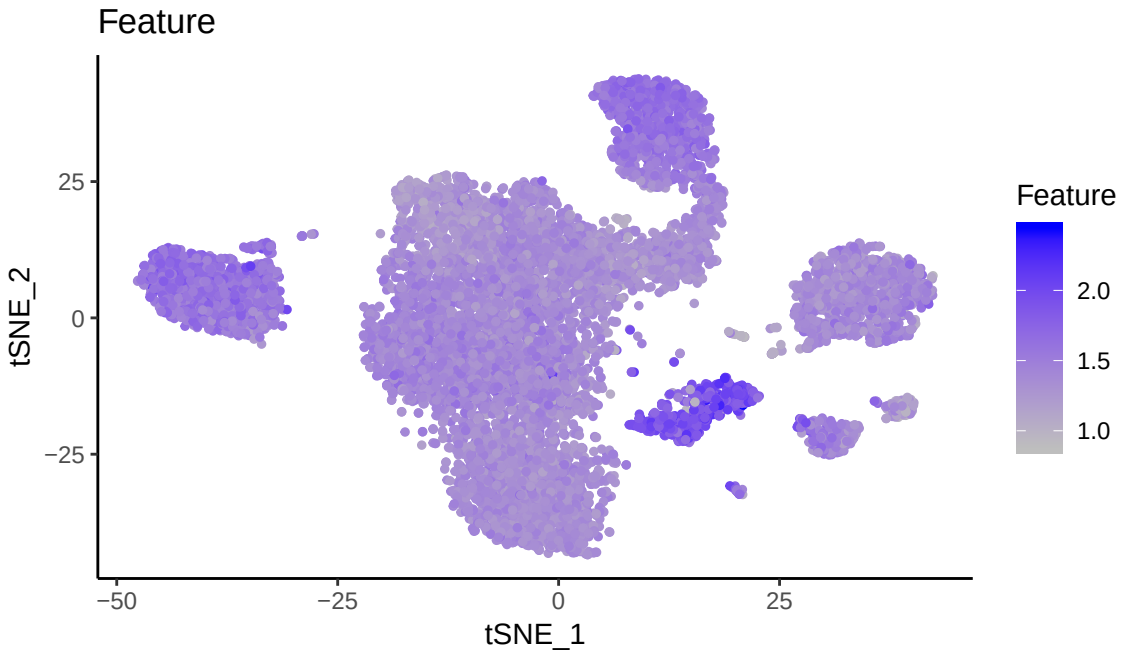
We continue to explore the ability of *SingleR* to correctly annotate cells as a function of the number non-zero genes in case study 3.

Confidence of annotations

Can we determine a significance test for the confidence of the annotations?

A possible approach, according to the plot above, is to use a threshold on scores. However, we can see that even low scores are mostly reliable. This is because the *SingleR* scores are associated with nGenes, but for a give single-cell the annotation is relative for the cell types in the reference data. Thus we introduce a significance test that tests whether the scores for the top cell types are different from the majority of low scored cell types. We do that using a chi-squared outliers test for the top score, where the null hypothesis is that it is not an outlier. This test does not provide confidence for the fine-tuned annotation, but can suggest if a single-cell does not have sufficient information to be annotated. We can see the t-SNE plot with -log10(p-value):

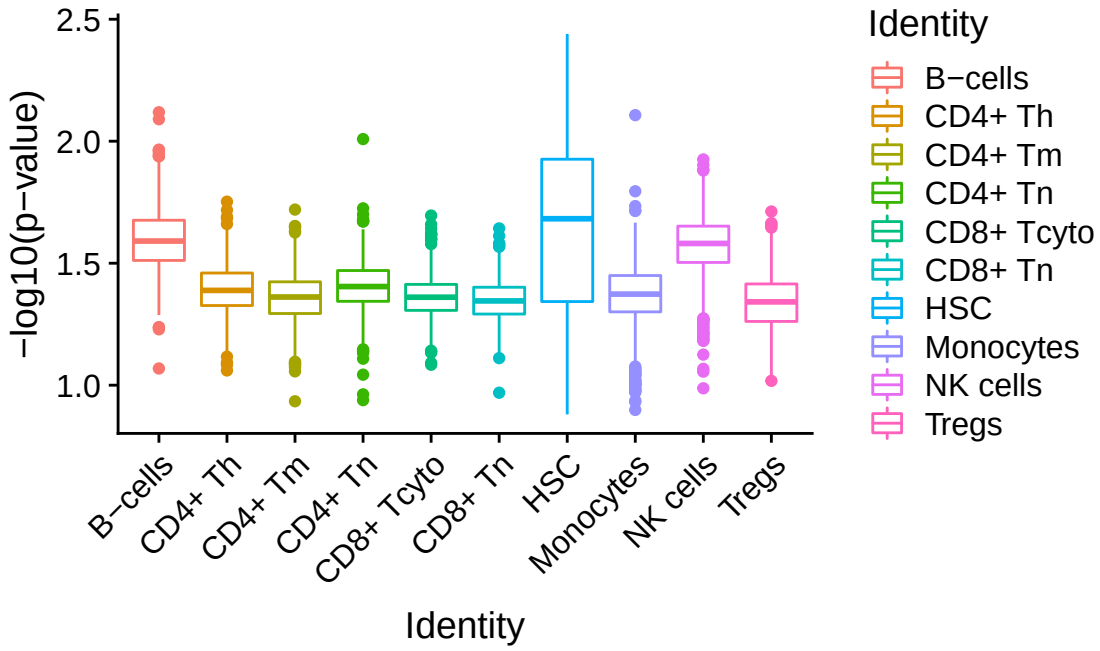
```
SingleR.PlotFeature(singler$singler[[2]]$SingleR.single,singler$seurat,
                    plot.feature = -log10(singler$singler[[2]]$SingleR.single.main$pval))
```



This plot suggests that for one of the HSC clusters the confidence is greater than the other, but also confidence in the NK cells and B-cells annotations:

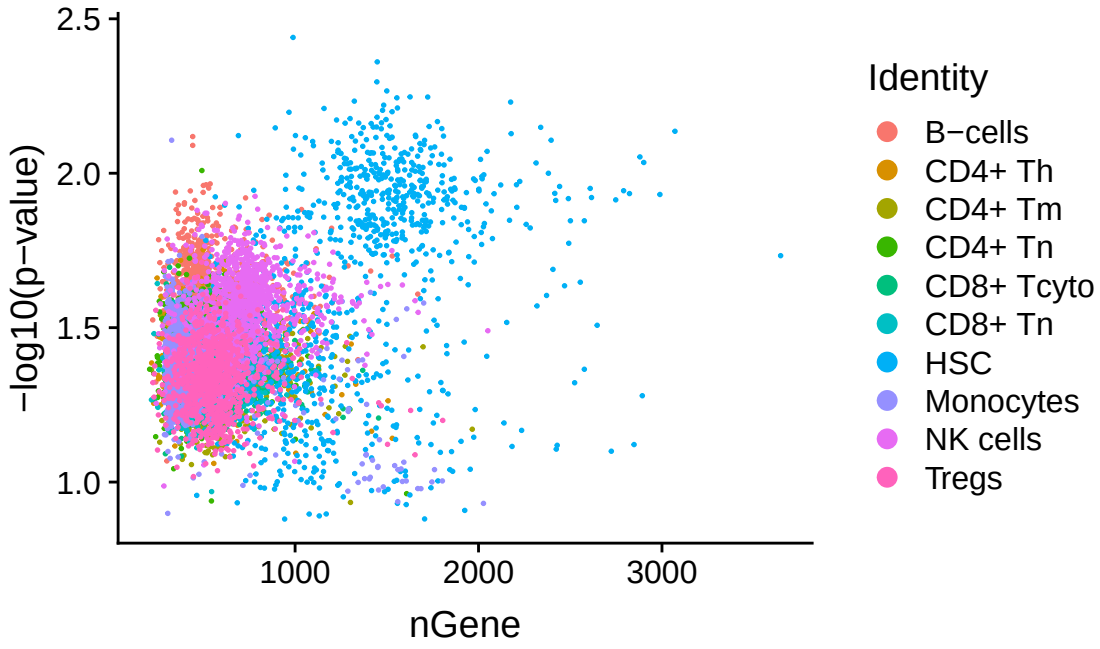
```
df = data.frame(nGene=singler$seurat@meta.data$nGene,
                pval=-log10(singler$singler[[2]]$SingleR.single.main$pval),
                Identity=singler$meta.data$orig.ident)

ggplot(df,aes(x=Identity,y=pval,color=Identity))+geom_boxplot()+
  ylab('-log10(p-value)')+theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



Importantly, we can see that this test is not dependent on nGenes:

```
ggplot(df,aes(x=nGene,y=pval,color=Identity))+geom_point(size=0.3)+
    guides(color = guide_legend(override.aes = list(size = 3)))+
  ylab('-log10(p-value)')
```



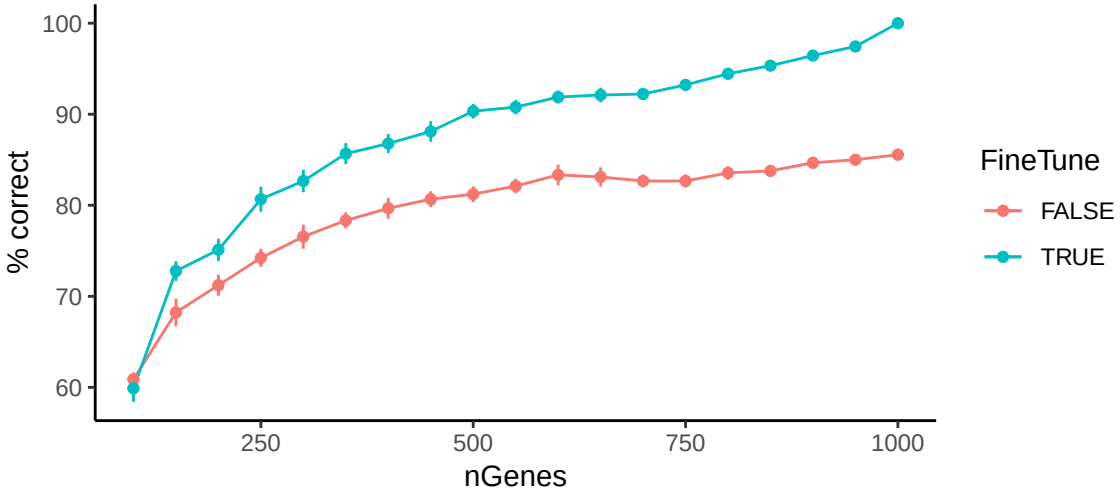
Case study 3: Simulating number of non-zero genes

The number of drop-outs in cells is highly variable and may have strong impact on the ability to correctly annotate cells. Here we further explore this issue by simulating varying number of non-zero genes (nGenes) in cells with known identity.

From the sorted 10X dataset presented in case study 2 (excluding CD4+ helper T-cells, which are represtend by the CD4+ memory T-cells), we randomly chose 10 cells with at least 1000 nGenes that were correcly annotated by SingleR (after fine-tuning).

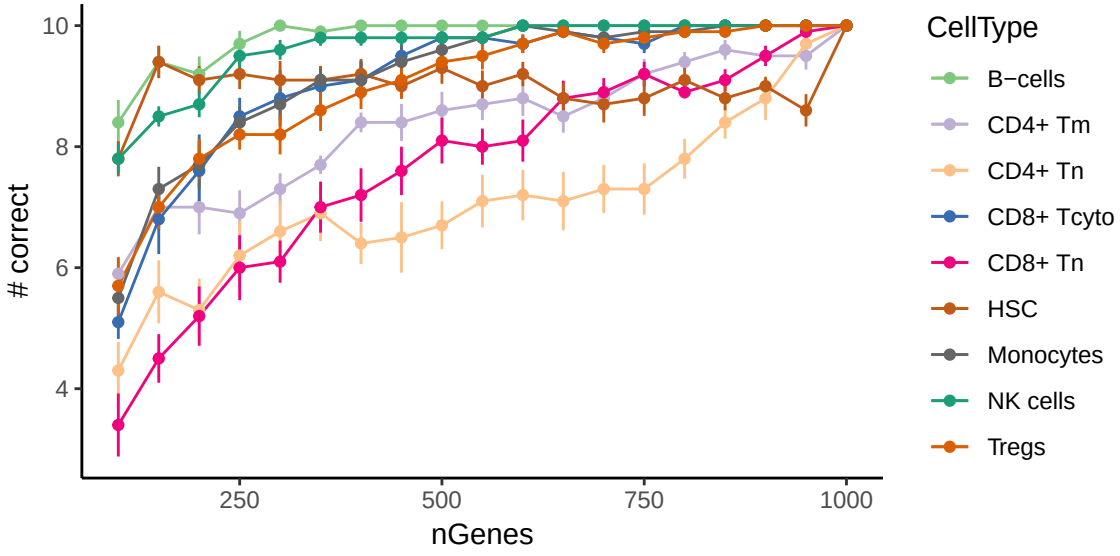
From the non-zero genes in each cell we chose 1000 genes to be non-zero and the rest were switched to 0; thus all cells have exactly 1000 non-zero genes. We counted correct inferences by SingleR, before and after fine-tuning. We then iteratively remove 50 genes, ran SingleR, and counted again the number of correct annotations. We repeated this process 10 times, randomly choosing different genes to remove. The code for this analysis is available in the Github repository.

We plot the percent of correct annotations as a function of nGenes (standard error is shown):



We can see a gradual decline in the accuracy of SingleR as a function of nGenes. This is more evident in the fine-tuned annotations (blue line), which shows the importance of fine-tuning to differentiate closely related cell types, even when there are more genes available. At 500 genes we observe 90%, and the decline is more evident with less genes, supporting our choice to use >500 nGenes in our analysis of mouse lung injury.

The lines per cell types show that as expected, with fewer genes it is harder to differentiate closely related cell types (T-cell subsets in this data):



SingleR web tool

The *SingleR* web tool contains >50 publicly available scRNA-seq datasets. All data has been reprocessed with the tools described above, and the web tool allows the user immediate access to analyze the data and perform further investigations on published single cell data. In addition, we invite users to upload their own scRNA-seq data, which will be analyzed on our servers and sent back to the user. The processed *SingleR* object can then be uploaded and further analyzed on the website (privately-only the user with the object is able to view it). Please visit http://comphealth.ucsf.edu/SingleR for more information.

References

1. Kimmerling, R. J. *et al.* A microfluidic platform enabling single-cell RNA-seq of multigenerational lineages. *Nature Communications* **7**, 10220 (2016).

2. Zheng, C. *et al.* Landscape of Infiltrating T Cells in Liver Cancer Revealed by Single-Cell Sequencing. *Cell* **169**, 1342–1356.e16 (2017).

3. Kang, H. M. *et al.* Multiplexed droplet single-cell RNA-sequencing using natural genetic variation. *Nature Biotechnology* **36**, 89–94 (2017).

4. Li, H. *et al.* Reference component analysis of single-cell transcriptomes elucidates cellular heterogeneity in human colorectal tumors. *Nature Genetics* **49**, 708–718 (2017).