

Kohnke_et_al_2019_needs_R_kernel

July 19, 2019

1 Code to analyze and visualize scRNAseq data - R

1.1 Cluster cells and find defining genes per cluster

Packages required: cellrangerRkit, NBClust, edgeR, GOplot, devtools

References:

<http://cf.10xgenomics.com/supp/cell-exp/cellrangerrkit-PBMC-vignette-knitr-2.0.0.pdf>

<https://www.bioconductor.org/packages/release/bioc/vignettes/edgeR/inst/doc/edgeRUsersGuide.pdf>

Locate file path that contains files produced from 10x sequencing (including gene BC matrix and mouse genome info), set as working directory

```
[ ]: setwd("~/your/file/path")
```

Load gene BC matrix (contains info about sample barcodes, expression counts per cell) and analysis results (10X performs some preliminary analysis)

```
[ ]: gbm<-load_cellranger_matrix("./")
      analysis_results<-load_cellranger_analysis_results
```

1.1.1 Find optimal number of clusters for the dataset

Use coordinates for all cells in tSNE plot: 'tsne_kmeans.csv' (in Github repository)

```
[ ]: tsne_kmeans<-read.csv('tsne_kmeans.csv')
      fviz_nbclust(tsne_kmeans,kmeans,method="silhouette",k.max=15)
```

See Supplementary Figure 1d for output

1.1.2 Find top defining genes per cluster

Use 9 K-means clusters based on silhouette analysis. Can change 'n' to the number of defining genes required. Output file will be in the working directory

```
[ ]: cells_to_plot<-order_cell_by_clusters(gbm,
      ↳analysis_results$clustering$kmeans_9_clusters$Cluster)
      prioritized_genes<-prioritize_top_genes(gbm,
      ↳analysis_results$clustering$kmeans_9_clusters$Cluster, "sseq", min_mean=0.5)
      write_cluster_specific_genes(prioritized_genes, output_folder, n_genes=50)
```

Use new .csv file that only contains experimental condition and gene expression data from a single cluster. See 'Cluster1_edgeR.txt' (in Github repository). The stringency set in rowSums is based on trial and error for similar experiments in our lab. Can change 'n' to the number of DEGs required.

Packages required: cellrangerRkit, ggplot2, tidyr

1.2.1 UMI counts mapped to tSNE plot

```
[ ]: tsne_proj<-analysis_results$tsne
visualize_umi_counts(gbm,tsne_proj[c("TSNE.1","TSNE.
→2")],limits=c(3,4),marker_size=0.05)
```

See Supplementary Figure 1c for output

1.2.2 Visualize cluster assignments on tSNE plot

Based on 9 clusters determined by silhouette analysis. Cluster names were assigned based on defining genes.

```
[ ]: visualize_clusters(analysis_results$clustering$kmeans_9_clusters$Cluster,tsne_proj[c("TSNE.
→1","TSNE.2")],
→colour=c("green","blue","red","purple","yellow","orange","pink","gray","black"),
→legend_anno = c("tanycyte","astrocyte","oligodendrocyte",
→1,"microglia","neuron",
→"oligodendrocyte 2","VLMC","ependymocyte","endothelial"))
```

See Figure 1b for output

1.2.3 Create heatmap of top gene expression per cluster

Can change n_genes to the number of top genes required

```
[ ]: gbm_pheatmap(log_gene_bc_matrix(gbm,base=2),prioritized_genes,cells_to_plot,n_genes=5,colour=
→c("green","blue","red","purple","yellow","orange","pink","gray","black"),limits=c(-1,2))
```

See Figure 1c for output

1.2.4 Create violin plot of defining genes per cluster

Genes chosen based on defining genes per cluster. Use .csv file with expression values of the chosen genes for all cells. See 'violingen2.csv'(in Github repository).

```
[ ]: nineclusters<-read.csv("violingen2.csv")
nineclusters$cluster<-as.factor(nineclusters$cluster)
nineclusterslong<-gather(nineclusters, key="measure", value="value", c("Rax",
→"Agt","Ernm","C1qc","Snhg11","Cd9","Dcn","Elof1","Itm2a"))
nineclusterslong$measure_f=factor(nineclusterslong$measure,levels=c("Rax",
→"Agt","Ernm","C1qc","Snhg11","Cd9","Dcn","Elof1","Itm2a"))
ggplot(nineclusterslong, aes(x=cluster, y=value,fill=cluster))+
→geom_violin(scale="width")+
→facet_wrap(~measure_f, scales="free_x", nrow=1)+
→coord_flip()+
→theme_classic(base_size=7)+
→scale_x_discrete(limits=unique(rev(nineclusterslong$cluster)))+
→scale_fill_manual(values=c("green","blue","red","purple","yellow",
→"orange","pink","gray","black"))+
```

```
labs(y="UMI")
```

See Figure 1d for output

1.2.5 Create bubble plot of pathways changed between experimental conditions

Use .csv files with regulated pathways for each cluster: see 'IPAOPCbubble.csv', 'NFOLIPAbubble.csv', and 'MOLIPAbubble.csv' (in Github repository). Use .csv files with lists of genes in each regulated pathway with logFC values and p values: see 'opcgenelist.csv', 'nfolgenelist.csv', and 'molgenelist.csv' (in Github repository).

```
[ ]: opc_ipa<-read.csv('IPAOPCbubble.csv')
opcgenelist<-read.csv('opcgenelist.csv')
opccirc<-circle_dat(opc_ipa,opcgenelist)
GOBubble(opccirc, labels=3)
opcreduced_circ<-reduce_overlap(opccirc,overlap=0.8)
GOBubble(opcreduced_circ,colour=c("yellow","light green"),labels=2.05)

nfol_ipa<-read.csv('NFOLIPAbubble.csv')
nfolgenelist<-read.csv('nfolgenelist.csv')
nfolcirc<-circle_dat(nfol_ipa,nfolgenelist)
GOBubble(nfolcirc, labels=3)
nfolreduced_circ<-reduce_overlap(nfolcirc,overlap=0.8)
GOBubble(nfolreduced_circ,colour=c("yellow","light green"),labels=4.8)

mol_ipa<-read.csv('MOLIPAbubble.csv')
molgenelist<-read.csv('molgenelist.csv')
molcirc<-circle_dat(mol_ipa,molgenelist)
GOBubble(molcirc, labels=3)
molreduced_circ<-reduce_overlap(molcirc,overlap=0.8)
GOBubble(molreduced_circ,colour=c("yellow","light green"),labels=5.2)
```

See Figure 4b for output