Gene Set Building and CAMML Analysis of GSE72056 Melanoma Data

Courtney Schiebout, H. Robert Frost

Load Libraries

Libraries "CAMML" (Schiebout and Frost 2022) and "Seurat" (Satija et al. 2015) need to be loaded to carry out this vignette, in addition to several other libraries for data processing and gene set development (Robinson, McCarthy, and Smyth 2010; Carlson 2020; Liberzon et al. 2011). Packages will also load additional libraries they depend on.

```
library(CAMML)
library(Seurat)
library(edgeR)
library(org.Hs.eg.db)
library(msigdbr)
```

Load and Process the GSE72056 Dataset

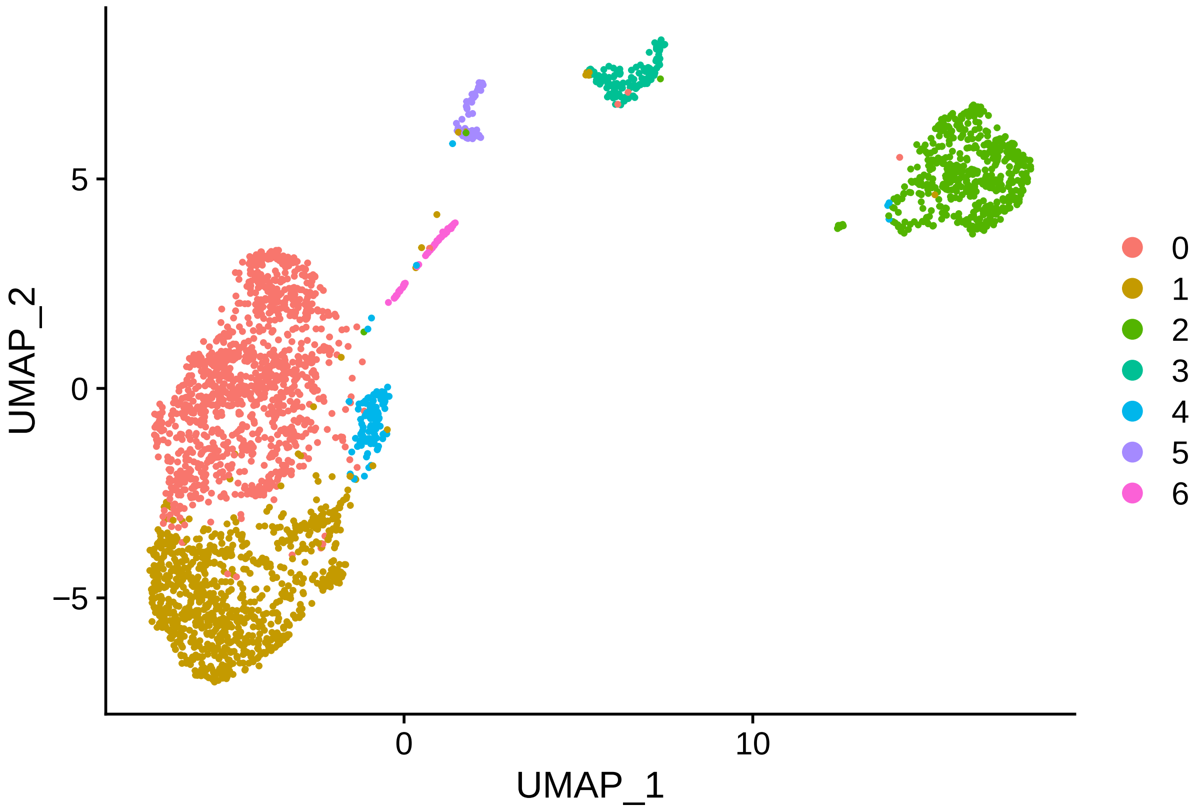
The methods used to create and classify the data used in this vignette can be found at the following DOI: 10.1126/science.aad0501 (Tirosh et al. 2016). The data can be accessed using the Gene Expression Omnibus (GEO) at acccession number GSE72056 (Tirosh et al. 2016; Edgar, Domrachev, and Lash 2002). This data is not originally formatted in the matrix style required for Seurat and must be slightly altered prior to analysis. The altered data structure can than be processed and normalized using the Seurat pipeline (Satija et al. 2015).

```
#access data1
ground_truth <- read.delim("GSE72056_melanoma_single_cell_revised_v2.txt")

#save cell classifications
cells <- ground_truth[1:3,]
cells <- unlist(matrix(cells[3,-1]))

#remove cell classifications from count matrix
gt <- ground_truth[-c(1:3),]

#remove duplications
gt <- gt[-c(which(duplicated((gt[,1])) == TRUE)),]

#reset rownames
rownames(gt) <- gt[,1]
gt <- gt[,-1]
gt <- data.frame(gt[,which(cells!=0)])
```

```r
#change counts to numeric
gt[] <- lapply(gt, function(x) {
  if(is.factor(x)) as.numeric(as.character(x)) else x
})

#Seurat pipeline
gse72056 <- CreateSeuratObject(gt, project = "gse72056",
                           min.cells=100, min.features=500,
                           num.var.features=2000)

#data filtering and normalization
gse72056[["percent.mt"]] <- PercentageFeatureSet(gse72056, pattern = "^MT-")
gse72056 <- subset(gse72056, subset = percent.mt < 5)
gse72056 <- NormalizeData(gse72056)
gse72056 <- FindVariableFeatures(gse72056, selection.method = "vst", nfeatures = 2000)

# Data clustering
gse72056 <- ScaleData(gse72056)
gse72056 <- RunPCA(gse72056)
gse72056 <- FindNeighbors(gse72056, dims = 1:30)
gse72056 <- FindClusters(gse72056, resolution = 0.25)
```

```
## Modularity Optimizer version 1.3.0 by Ludo Waltman and Nees Jan van Eck
##
## Number of nodes: 2887
## Number of edges: 111976
##
## Running Louvain algorithm...
## Maximum modularity in 10 random starts: 0.9148
## Number of communities: 7
## Elapsed time: 0 seconds
```

```r
gse72056 <- RunUMAP(gse72056, dims = 1:30)
UMAPPlot(gse72056)
```



Gene Set Development

Gene sets are developed by running differential expression of cell types from "celldex" and intersecting those with cell type gene sets from MSigDB's C8 Collection (Robinson, McCarthy, and Smyth 2010; Liberzon et al. 2011).

```
#access human reference data
reference <- celldex::HumanPrimaryCellAtlasData()

#set labels
labs <- unique(reference$label.main)
labs <- sort(labs)

#isolate labels for this dataset
labs <- labs[c(2,9,12,35,28,20)]
labs <- sort(labs)

#define relevant columns and counts
colcount <- reference@assays@data$logcounts
counts <- reference$label.main
colcount <- colcount[,which(counts %in% labs)]
counts <- counts[which(counts %in% labs)]

#edgeR DE analysis pipeline
```

```r
v <- data.frame()

for (i in 1:length(labs)){
  d <- DGEList(counts=exp(colcount), group = ifelse(counts == labs[i],1,0))
  d <- calcNormFactors(d)
  d1 <- estimateCommonDisp(d, verbose=T)
  d1 <- estimateTagwiseDisp(d1)
  et12 <- exactTest(d1, pair = c(1,2))

  gp <- et12$table
  gp <- gp[order(gp[,1], decreasing = T),]

  #save gene symbols
  r <- rownames(gp[gp[,1]>5,])

  #save log fc
  gw <- (gp[gp[,1]>5,1])

  v <- rbind(v,cbind(rep(labs[i], length(r)), r,gw))
}
```

```
## Disp = 0.68524 , BCV = 0.8278
## Disp = 0.60415 , BCV = 0.7773
## Disp = 0.70597 , BCV = 0.8402
## Disp = 0.57566 , BCV = 0.7587
## Disp = 0.74497 , BCV = 0.8631
## Disp = 0.61096 , BCV = 0.7816
```

```r
#incorporate and intersect with C8
x <- c()
m <- msigdbr(category = "C8")

#B-cell genes
r <- c(m$gene_symbol[which(m$gs_name == "HAY_BONE_MARROW_FOLLICULAR_B_CELL")])
r <- intersect(r, v[which(v[,1] == "B_cell"),2])
x <- rbind(x,cbind(rep("B_cell", length(r)), r))

#T-cell genes
r <- c(m$gene_symbol[which(m$gs_name == "HAY_BONE_MARROW_CD8_T_CELL")])
r <- intersect(r, v[which(v[,1] == "T_cells"),2])
x <- rbind(x,cbind(rep("T_cells", length(r)), r))
r <- c(m$gene_symbol[which(m$gs_name == "HAY_BONE_MARROW_NAIVE_T_CELL")])
r <- intersect(r, v[which(v[,1] == "T_cells"),2])
x <- rbind(x,cbind(rep("T_cells", length(r)), r))

#NK cells
r <- c(m$gene_symbol[which(m$gs_name == "HAY_BONE_MARROW_NK_CELLS")])
r <- intersect(r, v[which(v[,1] == "NK_cell"),2])
x <- rbind(x,cbind(rep("NK_cell", length(r)), r))

#Macrophages
r <- c(m$gene_symbol[which(m$gs_name == "HAY_BONE_MARROW_MONOCYTE")])
r <- intersect(r, v[which(v[,1] == "Macrophage"),2])
```

```r
x <- rbind(x,cbind(rep("Macrophage", length(r)), r))

#Fibroblasts
r <- c(m$gene_symbol[which(m$gs_name == "CUI_DEVELOPING_HEART_C3_FIBROBLAST_LIKE_CELL")])
r <- intersect(r, v[which(v[,1] == "Fibroblasts"),2])
x <- rbind(x,cbind(rep("Fibroblasts", length(r)), r))

#Endothelial
r <- c(m$gene_symbol[which(m$gs_name == "CUI_DEVELOPING_HEART_C4_ENDOTHELIAL_CELL")])
r <- intersect(r, v[which(v[,1] == "Endothelial_cells"),2])
x <- rbind(x,cbind(rep("Endothelial_cells", length(r)), r))

#merge C8 and DE data
v <- data.frame(v)
df <- data.frame(x)
df <- merge(df, v, by = c("r","V1"),all.x = T)
```

Convert Gene Sets to Ensembl IDs

The gene set development steps use gene symbols which need to be converted to Ensembl IDs for later analyses (Carlson 2020).

```r
#convert gene symbols to Ensembl IDs
# Get the gene symbols that are mapped to an Entrez
symbol2entrez = mappedkeys(org.Hs.egSYMBOL2EG)
# Convert to a list
symbol2entrez = as.list(org.Hs.egSYMBOL2EG[symbol2entrez])
# Convert Gene Symbols to Entrez IDs
gene.symbols = (df$r)
num.ids = length(gene.symbols)
entrez.ids = rep(NA, num.ids)

for (i in 1:num.ids) {
  entrez.id = gene.symbols[i]
  id.index = (which(names(symbol2entrez) == entrez.id))
  if (length(id.index > 0)) {
    # only use the first mapped ensembl id
    entrez.ids[i] =(symbol2entrez[[id.index]][1])
  }
}

# Get the entrez gene IDs that are mapped to an Ensembl ID
entrez2ensembl = mappedkeys(org.Hs.egENSEMBL)
# Convert to a list
entrez2ensembl = as.list(org.Hs.egENSEMBL[entrez2ensembl])

num.ids = length(entrez.ids)
ensembl.ids = rep(NA, num.ids)

for (i in 1:num.ids) {
  entrez.id = entrez.ids[i]
  id.index = (which(names(entrez2ensembl) == entrez.id))
  if (length(id.index > 0)) {
    # only use the first mapped ensembl id
    ensembl.ids[i] =(entrez2ensembl[[id.index]][1])
  }
}

df$ensembl.id = ensembl.ids
colnames(df)[colnames(df) == "r"] <- "gene.symbol"
colnames(df)[colnames(df) == "V1"] <- "cell.type"
colnames(df)[colnames(df) == "gw"] <- "gene.weight"
```

Run CAMML

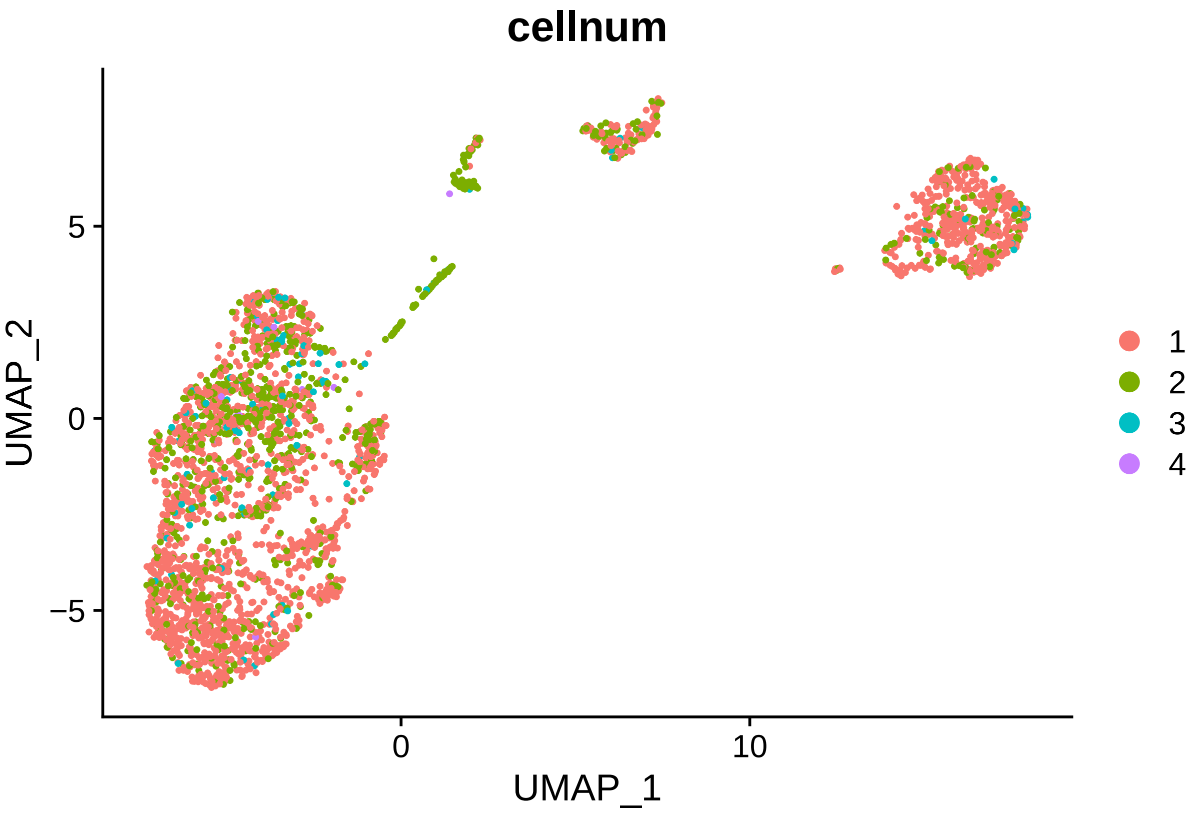
CAMML needs the Seurat Object and gene set data frame to run weighted VAM. CAMML will return the updated Seurat Object with weighted VAM CDFs which can then by inputed into GetCAMMLLabels to return one of several lists that classify cells with differing metrics: top 1, top 2, labels with scores above twice the mean of all scores for that cell, and cell types with scores of at least 90% of the top cell type score. In this case, we take cell types according to this final classification metric and visualize how they compare to the ground truth. As can be seen, cells in transition areas between cell types tend to have more cell type classifications.

```
gse72056 <- CAMML(gse72056,df)

#visualize results
gse72056@assays$CAMML@data[c(1:5),c(1:5)]
```

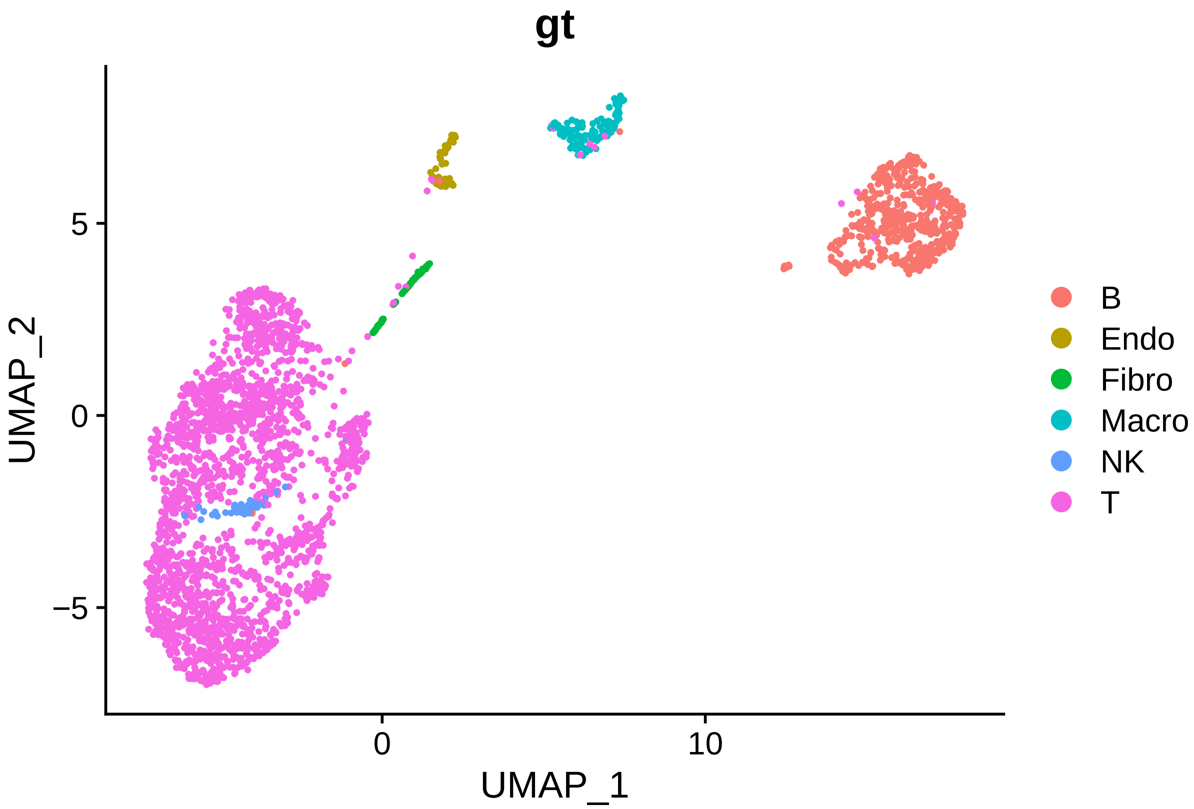
```
## 5 x 5 sparse Matrix of class "dgCMatrix"
##                   Cy72_CD45_H02_S758_comb CY58_1_CD45_B02_S974_comb
## B-cell                         0.98738076               0.009121001
## Endothelial-cells              0.55466484               0.143507296
## Fibroblasts                    0.03911469               .
## Macrophage                     0.17598610               0.101272819
## NK-cell                        .                        0.822440012
##                   Cy72_CD45_D09_S717_comb Cy74_CD45_A03_S387_comb
## B-cell                         0.09312815              .
## Endothelial-cells              0.14857579              0.05145502
## Fibroblasts                    .                       .
## Macrophage                     0.03755774              0.03385605
## NK-cell                        0.01480976              0.84754919
##                   Cy74_CD45_F09_S453_comb
## B-cell                        0.000312117
## Endothelial-cells             0.254476201
## Fibroblasts                   .
## Macrophage                    0.030566030
## NK-cell                       0.981055710
```

```
CAMML.results <- GetCAMMLLabels(gse72056,labels = "top10p")

#look at the number of cell types called for those > 90% of the max score
sizetest <- c()
for (i in 1:length(CAMML.results)){
```

```
sizetest[i] <- nrow(CAMML.results[[i]])
}
#visualize how cell number relates to transitioning states in the data
gse72056$cellnum <- sizetest
UMAPPlot(gse72056, group.by = "cellnum")
```



```
cells[which(cells == 1)] <- "T"
cells[which(cells == 2)] <- "B"
cells[which(cells == 3)] <- "Macro"
cells[which(cells == 4)] <- "Endo"
cells[which(cells == 5)] <- "Fibro"
cells[which(cells == 6)] <- "NK"
gse72056$gt <- cells[which(cells!=0)]
UMAPPlot(gse72056, group.by = "gt")
```



References

Carlson, Marc. 2020. *Org.Hs.eg.db: Genome Wide Annotation for Human.*

Edgar, Ron, Michael Domrachev, and Alex E. Lash. 2002. "Gene Expression Omnibus: NCBI Gene Expression and Hybridization Array Data Repository." *Nucleic Acids Research* 30 (1): 207–10. https://doi.org/10.1093/nar/30.1.207.

Liberzon, Arthur, Aravind Subramanian, Reid Pinchback, Helga Thorvaldsdóttir, Pablo Tamayo, and Jill P. Mesirov. 2011. "Molecular Signatures Database (MSigDB) 3.0." *Bioinformatics* 27 (12): 1739–40. https://doi.org/10.1093/bioinformatics/btr260.

Robinson, Mark D, Davis J McCarthy, and Gordon K Smyth. 2010. "edgeR: A Bioconductor Package for Differential Expression Analysis of Digital Gene Expression Data." *Bioinformatics* 26 (1): 139–40. https://doi.org/10.1093/bioinformatics/btp616.

Satija, Rahul, Jeffrey A Farrell, David Gennert, Alexander F Schier, and Aviv Regev. 2015. "Spatial Reconstruction of Single-Cell Gene Expression Data." *Nature Biotechnology* 33 (5): 495–502. https://doi.org/10.1038/nbt.3192.

Schiebout, Courtney, and H. Robert Frost. 2022. "CAMML: Multi-Label Immune Cell-Typing and Stemness Analysis for Single-Cell RNA-Sequencing." *Pacific Symposium on Biocomputing.*

Tirosh, Itay, Benjamin Izar, Sanjay M. Prakadan, Marc H. Wadsworth, Daniel Treacy, John J. Trombetta, Asaf Rotem, et al. 2016. "Dissecting the Multicellular Ecosystem of Metastatic Melanoma by Single-Cell RNA-Seq." *Science (New York, N.Y.)* 352 (6282): 189–96. https://doi.org/10.1126/science.aad0501.