library(Seurat)

library(dplyr)

library(patchwork)

library(Matrix)

# Load the PBMC dataset

BradleyDCD56neg <- Read10X(data.dir = "bradley\_cr\_output/outs/filtered\_feature\_bc\_matrix/")

# Initialize the Seurat object with the raw (non-normalized data)

BradleyNeg <- CreateSeuratObject(counts = BradleyDCD56neg, project = "CD56neg", min.cells = 3, min.features = 200)

# The [[ operator can add columns to object metadata. This is a great place to stash QC stats

BradleyNeg[["percent.mt"]] <- PercentageFeatureSet(BradleyNeg, pattern = "^MT-")

# Visualize QC metrics as a violin plot

BradleyNegVlnplot1 <- VlnPlot(BradleyNeg, features = c("nFeature\_RNA", "nCount\_RNA", "percent.mt"), ncol = 3)

ggsave("BradleyNegVlnplot1.pdf", BradleyNegVlnplot1)

# FeatureScatter is typically used to visualize feature-feature relationships, but can be used

# for anything calculated by the object, i.e. columns in object metadata, PC scores etc.

plot1 <- FeatureScatter(BradleyNeg, feature1 = "nCount\_RNA", feature2 = "percent.mt")

plot2 <- FeatureScatter(BradleyNeg, feature1 = "nCount\_RNA", feature2 = "nFeature\_RNA")

BradleyNegPlot1and2 <- plot1 + plot2

ggsave("BradleyNegPlot1and2.pdf", BradleyNegPlot1and2)

# Check plots and adjust values below accordingly

#BradleyNeg <- subset(BradleyNeg, subset = nFeature\_RNA > 200 & nFeature\_RNA < 2500 & percent.mt < 5)

BradleyNeg <- subset(BradleyNeg, subset = nFeature\_RNA > 200 & nFeature\_RNA < 4000 & percent.mt < 10)

BradleyNeg <- NormalizeData(BradleyNeg, normalization.method = "LogNormalize", scale.factor = 10000)

# or BradleyNeg <- NormalizeData(BradleyNeg)

BradleyNeg <- FindVariableFeatures(BradleyNeg, selection.method = "vst", nfeatures = 2000)

# Identify the 10 most highly variable genes

top10 <- head(VariableFeatures(BradleyNeg), 10)

# plot variable features with and without labels

plot3 <- VariableFeaturePlot(BradleyNeg)

plot4 <- LabelPoints(plot = plot3, points = top10, repel = TRUE)

BradleyNegPlot3and4 <- plot3 + plot4

ggsave("BradleyNegPlot4.pdf", plot4)

ggsave("BradleyNegPlot3and4.pdf", BradleyNegPlot3and4)

all.genes <- rownames(BradleyNeg)

BradleyNeg <- ScaleData(BradleyNeg, features = all.genes)

BradleyNeg <- RunPCA(BradleyNeg, features = VariableFeatures(object = BradleyNeg))

# Examine and visualize PCA results a few different ways

print(BradleyNeg[["pca"]], dims = 1:5, nfeatures = 5)

#PC\_ 1

#Positive: CFD, AIF1, SERPINA1, SPI1, MS4A7

#Negative: GNLY, CTSW, CCL5, CMC1, IL32

#PC\_ 2

#Positive: SELL, GZMK, FCER1G, KLRC1, AREG

#Negative: CCL5, CCL4L2, S100A6, CCL3L1, CCL4

#PC\_ 3

#Positive: SPON2, GZMB, PRF1, MYOM2, ALOX5AP

#Negative: RPS18, RPL13A, RPS2, RPS12, EEF1A1

#PC\_ 4

#Positive: MT-CO1, MT-CO2, MT-CO3, NEAT1, MT-ND4

#Negative: TYMS, PCLAF, STMN1, TK1, MKI67

#PC\_ 5

#Positive: RPL28, RPL13, RPLP1, RPS12, RPS24

3Negative: MT-CO3, MT-CO2, MT-ND4, MT-CO1, TYMS

VizDimLoadings(BradleyNeg, dims = 1:2, reduction = "pca")

BradleyNegPCA <- DimPlot(BradleyNeg, reduction = "pca")

ggsave("BradleyNegPCA.pdf", BradleyNegPCA)

DimHeatmap(BradleyNeg, dims = 1, cells = 500, balanced = TRUE)

DimHeatmap(BradleyNeg, dims = 1:15, cells = 500, balanced = TRUE)

# NOTE: This process can take a long time for big datasets, comment out for expediency. More

# approximate techniques such as those implemented in ElbowPlot() can be used to reduce

# computation time

BradleyNeg <- JackStraw(BradleyNeg, num.replicate = 100)

BradleyNeg <- ScoreJackStraw(BradleyNeg, dims = 1:20)

#PCA cluster number definitions

JackStrawPlot(BradleyNeg, dims = 1:15)

BradleyNegElbow <- ElbowPlot(BradleyNeg)

ggsave("BradleyNegElbow.pdf", BradleyNegElbow)

BradleyNeg <- FindNeighbors(BradleyNeg, dims = 1:15)

BradleyNeg <- FindClusters(BradleyNeg, resolution = 0.5)

# Look at cluster IDs of the first 5 cells

head(Idents(BradleyNeg), 5)

# If you haven't installed UMAP, you can do so via reticulate::py\_install(packages =

# 'umap-learn')

BradleyNeg <- RunUMAP(BradleyNeg, dims = 1:15)

# note that you can set `label = TRUE` or use the LabelClusters function to help label

# individual clusters

BradleyNegUMAP <- DimPlot(BradleyNeg, reduction = "umap", label = TRUE)

ggsave("BradleyNegUMAP.pdf", BradleyNegUMAP)

str(BradleyNeg)

#BradleyNegUMAPIdent <- DimPlot(BradleyNeg, reduction = "umap", group.by = "orig.ident")

#ggsave("BradleyNegUMAPIdent.pdf", BradleyNegUMAPIdent)

# Plot UMAP with Bozzano associated gene expression; Perforin (PRF1), CD94 (KLRD1), NKG2C (KLRC2), CD57 (B3GAT1), NKp30 (NCR3) and DNAM-1

features <- c("PRF1", "KLRD1", "KLRC2", "B3GAT1", "NCR3")

BradleyNegGeneRidgePlot <- RidgePlot(BradleyNeg, features = features, ncol = 3)

BradleyNegGeneViolinPlot <- VlnPlot(BradleyNeg, features = features)

BradleyNegGeneUMAP <- FeaturePlot(BradleyNeg, features = features)

ggsave("BradleyNegGeneRidgePlot.pdf", BradleyNegGeneRidgePlot)

ggsave("BradleyNegGeneViolinPlot.pdf", BradleyNegGeneViolinPlot)

ggsave("BradleyNegGeneUMAP.pdf", BradleyNegGeneUMAP)

features2 <- c("PRF1", "KLRD1")

BradleyNegGeneRidgePlot2 <- RidgePlot(BradleyNeg, features = features2, ncol = 2)

BradleyNegGeneViolinPlot2 <- VlnPlot(BradleyNeg, features = features2)

BradleyNegGeneViolinPlot2b <- VlnPlot(BradleyNeg, features = features2, pt.size = 0)

BradleyNegGeneUMAP2 <- FeaturePlot(BradleyNeg, features = features2, ncol = 1)

ggsave("BradleyNegGeneRidgePlot2.pdf", BradleyNegGeneRidgePlot2)

ggsave("BradleyNegGeneViolinPlot2.pdf", BradleyNegGeneViolinPlot2)

ggsave("BradleyNegGeneViolinPlot2b.pdf", BradleyNegGeneViolinPlot2b)

ggsave("BradleyNegGeneUMAP2.pdf", BradleyNegGeneUMAP2)

saveRDS(BradleyNeg, file = "BradleyNeg1.rds")

save.image(file="BradleyNeg1.RData")

# Combined clusters 6+9, and clusters 0, 1, 2, 3, 4, 5, 7, and 8, into two meta-clusters for comparison of gene expression

new.cluster.ids <- c("Perf-Pos", "Perf-Pos", "Perf-Pos", "Perf-Pos", "Perf-Pos", "Perf-Pos", "Perf-Neg", "Perf-Pos", "Perf-Pos", "Perf-Neg")

names(new.cluster.ids) <- levels(BradleyNeg)

BradleyNegMeta <- RenameIdents(BradleyNeg, new.cluster.ids)

BradleyNegGeneUMAPMeta <- DimPlot(BradleyNegMeta, reduction = "umap", label = TRUE, pt.size = 0.5) + NoLegend()

BradleyNegGeneMetaViolinPlot <- VlnPlot(BradleyNegMeta, features = features2, pt.size = 0)

ggsave("BradleyNegGeneUMAPMeta.pdf", BradleyNegGeneUMAPMeta)

ggsave("BradleyNegGeneMetaViolinPlot.pdf", BradleyNegGeneMetaViolinPlot)

# Create heatmap of top 20 positive differentiating gene expression between clusters

pbmc.markers <- FindAllMarkers(BradleyNegMeta, only.pos = TRUE, min.pct = 0.25, logfc.threshold = 0.25)

pbmc.markers %>% group\_by(cluster) %>% top\_n(n = 2, wt = avg\_logFC)

top20 <- pbmc.markers %>% group\_by(cluster) %>% top\_n(n = 20, wt = avg\_logFC)

BradleyNegHeatmap <- DoHeatmap(BradleyNegMeta, features = top20$gene, label = FALSE)

#+ NoLegend()

ggsave("BradleyNegHeatmap.pdf", BradleyNegHeatmap)

# Create volcano plots of Perforin-positive and -negative subset defining gene expression

pbmc.markersBland <- FindMarkers(BradleyNegMeta, ident.1 = "Perf-Pos", only.pos = FALSE, min.pct = 0.25, logfc.threshold = 0.25)

#pbmc.markersBland %>% group\_by(cluster) %>% top\_n(n = 2, wt = avg\_logFC)

#top20bland <- pbmc.markersBland %>% group\_by(cluster) %>% top\_n(n = 20, wt = avg\_logFC)

BradleyBAplot <- ggplot(pbmc.markersBland, aes(x=-log10(p\_val\_adj), y=avg\_logFC)) + geom\_point()+ ylim(c(-3,3)) + ggtitle("Variation in gene expression of Perf-Pos CD56neg cells compared to Perf-Neg subset") + xlab("log10 adjusted p-value") + ylab("Fold Difference")

ggsave("BradleyBAplot.pdf", BradleyBAplot)

ggsave("BradleyBAplotB.pdf", BradleyBAplot, width = 900, height = 300, units = "mm")

# want x axis to be average gene expression of Perf-Pos, colour to be based on p-value (the smaller the darker the colour), node size increasing with difference from 0 fold change, and labelled with gene name

BradleyBAplot2 <- ggplot(pbmc.markersBland, aes(x=pct.1, y=avg\_logFC, size = avg\_logFC^2, colour = p\_val\_adj)) + geom\_point() + scale\_colour\_gradient(low="red", high="blue", trans = scales::logit\_trans()) + ylim(c(-3,3)) + ggtitle("Variation in gene expression of Perf-Pos CD56neg cells compared to Perf-Neg subset") + xlab("Adjusted Gene Expression") + ylab("Fold Difference") + NoLegend() + geom\_text(label = row.names(pbmc.markersBland))

ggsave("BradleyBAplot2.pdf", BradleyBAplot2)

# No points just gene names, legend included

BradleyBAplot3 <- ggplot(pbmc.markersBland, aes(x=pct.1, y=avg\_logFC, size = avg\_logFC^2, colour = p\_val\_adj)) + scale\_colour\_gradient(low="red", high="blue", trans = scales::logit\_trans()) + ylim(c(-3,3)) + ggtitle("Variation in gene expression of Perf-Pos CD56neg cells compared to Perf-Neg subset") + xlab("Adjusted Gene Expression") + ylab("Fold Difference") + geom\_text(label = row.names(pbmc.markersBland)) + geom\_hline(yintercept=c(-1,1),color="black",linetype="dashed",size=0.5) + theme(panel.background = element\_rect(fill = "transparent"), plot.background = element\_rect(fill = "transparent", color = NA), panel.grid.major = element\_blank(), panel.grid.minor = element\_blank(), legend.background = element\_rect(fill = "transparent"), legend.box.background = element\_rect(fill = "transparent"))

BradleyBAplot3 <- ggplot(pbmc.markersBland, aes(x=pct.1, y=avg\_logFC, size = avg\_logFC^2, colour = p\_val\_adj)) + scale\_colour\_gradient(low="red", high="blue", trans = scales::logit\_trans()) + ylim(c(-3,3)) + ggtitle("Variation in gene expression of Perf-Pos CD56neg cells compared to Perf-Neg subset") + xlab("Adjusted Gene Expression") + ylab("Fold Difference") + geom\_text(label = row.names(pbmc.markersBland)) + geom\_hline(yintercept=c(-1,1),color="black",linetype="dashed",size=0.5) + theme(panel.background = element\_rect(fill = "transparent"), plot.background = element\_rect(fill = "transparent", color = NA), legend.background = element\_rect(fill = "transparent"), legend.box.background = element\_rect(fill = "transparent"))

ggsave("BradleyBAplot3.pdf", BradleyBAplot3, width = 450, height = 150, units = "mm", bg = "transparent")

# Import CD56bright and CD56dim cells, process them for comparison to CD56neg subsets

BradleyDCD56bright <- Read10X(data.dir = "bradley\_cr\_output3/outs/filtered\_feature\_bc\_matrix/")

BradleyDCD56dim <- Read10X(data.dir = "bradley\_cr\_output4/outs/filtered\_feature\_bc\_matrix/")

# Initialize the Seurat object with the raw (non-normalized data)

BradleyBright <- CreateSeuratObject(counts = BradleyDCD56bright, project = "CD56Bright", min.cells = 3, min.features = 200)

BradleyDim <- CreateSeuratObject(counts = BradleyDCD56dim, project = "CD56Dim", min.cells = 3, min.features = 200)

# The [[ operator can add columns to object metadata. This is a great place to stash QC stats

BradleyBright[["percent.mt"]] <- PercentageFeatureSet(BradleyBright, pattern = "^MT-")

BradleyDim[["percent.mt"]] <- PercentageFeatureSet(BradleyDim, pattern = "^MT-")

# Visualize QC metrics as a violin plot

BradleyBrightVlnplot1 <- VlnPlot(BradleyBright, features = c("nFeature\_RNA", "nCount\_RNA", "percent.mt"), ncol = 3)

ggsave("BradleyBrightVlnplot1.pdf", BradleyBrightVlnplot1)

BradleyDimVlnplot1 <- VlnPlot(BradleyDim, features = c("nFeature\_RNA", "nCount\_RNA", "percent.mt"), ncol = 3)

ggsave("BradleyDimVlnplot1.pdf", BradleyDimVlnplot1)

# FeatureScatter is typically used to visualize feature-feature relationships, but can be used

# for anything calculated by the object, i.e. columns in object metadata, PC scores etc.

Brightplot1 <- FeatureScatter(BradleyBright, feature1 = "nCount\_RNA", feature2 = "percent.mt")

Brightplot2 <- FeatureScatter(BradleyBright, feature1 = "nCount\_RNA", feature2 = "nFeature\_RNA")

BradleyBrightPlot1and2 <- Brightplot1 + Brightplot2

ggsave("BradleyBrightPlot1and2.pdf", BradleyBrightPlot1and2)

Dimplot1 <- FeatureScatter(BradleyDim, feature1 = "nCount\_RNA", feature2 = "percent.mt")

Dimplot2 <- FeatureScatter(BradleyDim, feature1 = "nCount\_RNA", feature2 = "nFeature\_RNA")

BradleyDimPlot1and2 <- Dimplot1 + Dimplot2

ggsave("BradleyDimPlot1and2.pdf", BradleyDimPlot1and2)

# Check plots and adjust values below accordingly

BradleyBright <- subset(BradleyBright, subset = nFeature\_RNA > 200 & nFeature\_RNA < 4000 & percent.mt < 10)

BradleyDim <- subset(BradleyDim, subset = nFeature\_RNA > 200 & nFeature\_RNA < 3000 & percent.mt < 10)

BradleyBright <- NormalizeData(BradleyBright, normalization.method = "LogNormalize", scale.factor = 10000)

BradleyDim <- NormalizeData(BradleyDim, normalization.method = "LogNormalize", scale.factor = 10000)

# or BradleyNeg <- NormalizeData(BradleyNeg)

BradleyBright <- FindVariableFeatures(BradleyBright, selection.method = "vst", nfeatures = 2000)

BradleyDim <- FindVariableFeatures(BradleyDim, selection.method = "vst", nfeatures = 2000)

# Identify the 10 most highly variable genes

Brighttop10 <- head(VariableFeatures(BradleyBright), 10)

Dimtop10 <- head(VariableFeatures(BradleyDim), 10)

# plot variable features with and without labels

Brightplot3 <- VariableFeaturePlot(BradleyBright)

Brightplot4 <- LabelPoints(plot = Brightplot3, points = Brighttop10, repel = TRUE)

BradleyBrightPlot3and4 <- Brightplot3 + Brightplot4

ggsave("BradleyBrightPlot4.pdf", Brightplot4)

ggsave("BradleyBrightPlot3and4.pdf", BradleyBrightPlot3and4)

Dimplot3 <- VariableFeaturePlot(BradleyDim)

Dimplot4 <- LabelPoints(plot = Dimplot3, points = Dimtop10, repel = TRUE)

BradleyDimPlot3and4 <- Dimplot3 + Dimplot4

ggsave("BradleyDimPlot4.pdf", Dimplot4)

ggsave("BradleyDimPlot3and4.pdf", BradleyDimPlot3and4)

Brightall.genes <- rownames(BradleyBright)

BradleyBright <- ScaleData(BradleyBright, features = Brightall.genes)

Dimall.genes <- rownames(BradleyDim)

BradleyDim <- ScaleData(BradleyDim, features = Dimall.genes)

BradleyBright <- RunPCA(BradleyBright, features = VariableFeatures(object = BradleyBright))

BradleyDim <- RunPCA(BradleyDim, features = VariableFeatures(object = BradleyDim))

# Examine and visualize PCA results a few different ways

print(BradleyBright[["pca"]], dims = 1:5, nfeatures = 5)

PC\_ 1

Positive: GNLY, GZMK, IL7R, FOS, JUNB

Negative: TYMS, STMN1, ZWINT, TK1, PCLAF

PC\_ 2

Positive: SPI1, LYZ, IFI30, AIF1, MNDA

Negative: GNLY, NKG7, TYMS, ZWINT, STMN1

PC\_ 3

Positive: TYMS, SPINK2, IFIT2, TOP2A, ZWINT

Negative: GZMA, GZMH, NKG7, GZMB, CCL5

PC\_ 4

Positive: EEF1A1, RPS4X, RPS2, RPS18, RPS12

Negative: MT-CO2, MT-CO1, MT-ND4, MT-CYB, MT-ATP6

PC\_ 5

Positive: S100A4, S100A6, LTB, IL32, LGALS3

Negative: MT-CO3, GZMM, MT-CO1, MT-CO2, MT-ND4

print(BradleyDim[["pca"]], dims = 1:5, nfeatures = 5)

PC\_ 1

Positive: TYMS, PCLAF, STMN1, FCER1G, TK1

Negative: CD3E, CD3D, CCL5, KLRC2, CD52

PC\_ 2

Positive: SPON2, FCER1G, KLRB1, GZMB, IGFBP7

Negative: CD3E, HLA-DRB1, CD52, TYMS, PCLAF

PC\_ 3

Positive: TYMS, PCLAF, GNLY, TK1, STMN1

Negative: LST1, AIF1, SPI1, SERPINA1, CD68

PC\_ 4

Positive: TYMS, PCLAF, TK1, RRM2, MKI67

Negative: EEF1A1, RPS2, RPLP1, RPS12, RPS18

PC\_ 5

Positive: SELL, ZFP36, JUNB, DUSP1, GPR183

Negative: CCL4L2, S100A4, S100A6, CCL4, CCL3L1

VizDimLoadings(BradleyBright, dims = 1:2, reduction = "pca")

BradleyBrightPCA <- DimPlot(BradleyBright, reduction = "pca")

ggsave("BradleyBrightPCA.pdf", BradleyBrightPCA)

VizDimLoadings(BradleyDim, dims = 1:2, reduction = "pca")

BradleyDimPCA <- DimPlot(BradleyDim, reduction = "pca")

ggsave("BradleyDimPCA.pdf", BradleyDimPCA)

DimHeatmap(BradleyBright, dims = 1, cells = 500, balanced = TRUE)

DimHeatmap(BradleyBright, dims = 1:15, cells = 500, balanced = TRUE)

DimHeatmap(BradleyDim, dims = 1, cells = 500, balanced = TRUE)

DimHeatmap(BradleyDim, dims = 1:15, cells = 500, balanced = TRUE)

# Combine QC‚Äôd data into all NK cell plot

BradleyAllNK <- merge(BradleyNegMeta, y = c(BradleyBright, BradleyDim), add.cell.ids = c("CD56neg", "CD56bright", "CD56dim"), project = "Bradley")

BradleyAllNK <- FindVariableFeatures(BradleyAllNK, selection.method = "vst", nfeatures = 2000)

BradleyAllNKplot3 <- VariableFeaturePlot(BradleyAllNK)

BradleyAllNKtop10 <- head(VariableFeatures(BradleyAllNK), 10)

BradleyAllNKplot4 <- LabelPoints(plot = BradleyAllNKplot3, points = BradleyAllNKtop10, repel = TRUE)

BradleyDimPlot3and4 <- BradleyAllNKplot3 + BradleyAllNKplot4

ggsave("BradleyAllNKplot4.pdf", BradleyAllNKplot4)

ggsave("BradleyDimPlot3and4.pdf", BradleyDimPlot3and4)

BradleyAllNKall.genes <- rownames(BradleyAllNK)

BradleyAllNK <- ScaleData(BradleyAllNK, features = BradleyAllNKall.genes)

BradleyAllNK <- RunPCA(BradleyAllNK, features = VariableFeatures(object = BradleyAllNK))

VizDimLoadings(BradleyAllNK, dims = 1:2, reduction = "pca")

BradleyAllNKPCA <- DimPlot(BradleyAllNK, reduction = "pca")

ggsave("BradleyAllNKPCA.pdf", BradleyAllNKPCA)

BradleyAllNK <- RunUMAP(BradleyAllNK, dims = 1:15)

BradleyAllNKUMAP <- DimPlot(BradleyAllNK, reduction = "umap", label = TRUE, pt.size = 0.5)

ggsave("BradleyAllNKUMAP.pdf", BradleyAllNKUMAP)

BradleyAllNKpbmc.markers <- FindAllMarkers(BradleyAllNK, only.pos = TRUE, min.pct = 0.25, logfc.threshold = 0.25)

write.csv(BradleyAllNKpbmc.markers, "BradleyAllNKGenes.csv")

saveRDS(BradleyNeg, file = "BradleyNeg1.rds")

save.image(file="BradleyNeg1.RData")

# Create heatmap of top 20 positive differentiating gene expression between all NK clusters

NK.markers <- FindAllMarkers(BradleyAllNK, only.pos = TRUE, min.pct = 0.25, logfc.threshold = 0.25)

NK.markers %>% group\_by(cluster) %>% top\_n(n = 2, wt = avg\_logFC)

topNK20 <- NK.markers %>% group\_by(cluster) %>% top\_n(n = 20, wt = avg\_logFC)

BradleyAllNKHeatmap <- DoHeatmap(BradleyAllNK, features = topNK20$gene, label = FALSE)

#+ NoLegend()

ggsave("BradleyAllNKHeatmap.pdf", BradleyAllNKHeatmap)

# Create volcano plots of Perforin-positive and -negative subset defining gene expression

pbmc.markersBland <- FindMarkers(BradleyNegMeta, ident.1 = "Perf-Pos", only.pos = FALSE, min.pct = 0.25, logfc.threshold = 0.25)

#pbmc.markersBland %>% group\_by(cluster) %>% top\_n(n = 2, wt = avg\_logFC)

#top20bland <- pbmc.markersBland %>% group\_by(cluster) %>% top\_n(n = 20, wt = avg\_logFC)

BradleyBAplot <- ggplot(pbmc.markersBland, aes(x=-log10(p\_val\_adj), y=avg\_logFC)) + geom\_point()+ ylim(c(-3,3)) + ggtitle("Variation in gene expression of Perf-Pos CD56neg cells compared to Perf-Neg subset") + xlab("log10 adjusted p-value") + ylab("Fold Difference")

ggsave("BradleyBAplot.pdf", BradleyBAplot)

ggsave("BradleyBAplotB.pdf", BradleyBAplot, width = 900, height = 300, units = "mm")

# want x axis to be average gene expression of Perf-Pos, colour to be based on p-value (the smaller the darker the colour), node size increasing with difference from 0 fold change, and labelled with gene name

BradleyBAplot2 <- ggplot(pbmc.markersBland, aes(x=pct.1, y=avg\_logFC, size = avg\_logFC^2, colour = p\_val\_adj)) + geom\_point() + scale\_colour\_gradient(low="red", high="blue", trans = scales::logit\_trans()) + ylim(c(-3,3)) + ggtitle("Variation in gene expression of Perf-Pos CD56neg cells compared to Perf-Neg subset") + xlab("Adjusted Gene Expression") + ylab("Fold Difference") + NoLegend() + geom\_text(label = row.names(pbmc.markersBland))

ggsave("BradleyBAplot2.pdf", BradleyBAplot2)

# No points just gene names, legend included

BradleyBAplot3 <- ggplot(pbmc.markersBland, aes(x=pct.1, y=avg\_logFC, size = avg\_logFC^2, colour = p\_val\_adj)) + scale\_colour\_gradient(low="red", high="blue", trans = scales::logit\_trans()) + ylim(c(-3,3)) + ggtitle("Variation in gene expression of Perf-Pos CD56neg cells compared to Perf-Neg subset") + xlab("Adjusted Gene Expression") + ylab("Fold Difference") + geom\_text(label = row.names(pbmc.markersBland)) + geom\_hline(yintercept=c(-1,1),color="black",linetype="dashed",size=0.5) + theme(panel.background = element\_rect(fill = "transparent"), plot.background = element\_rect(fill = "transparent", color = NA), panel.grid.major = element\_blank(), panel.grid.minor = element\_blank(), legend.background = element\_rect(fill = "transparent"), legend.box.background = element\_rect(fill = "transparent"))

BradleyBAplot3 <- ggplot(pbmc.markersBland, aes(x=pct.1, y=avg\_logFC, size = avg\_logFC^2, colour = p\_val\_adj)) + scale\_colour\_gradient(low="red", high="blue", trans = scales::logit\_trans()) + ylim(c(-3,3)) + ggtitle("Variation in gene expression of Perf-Pos CD56neg cells compared to Perf-Neg subset") + xlab("Adjusted Gene Expression") + ylab("Fold Difference") + geom\_text(label = row.names(pbmc.markersBland)) + geom\_hline(yintercept=c(-1,1),color="black",linetype="dashed",size=0.5) + theme(panel.background = element\_rect(fill = "transparent"), plot.background = element\_rect(fill = "transparent", color = NA), legend.background = element\_rect(fill = "transparent"), legend.box.background = element\_rect(fill = "transparent"))

ggsave("BradleyBAplot3.pdf", BradleyBAplot3, width = 450, height = 150, units = "mm", bg = "transparent")