## ##Query to human reference datasets####

library(SeuratDisk)

library(anndata)

library(SummarizedExperiment)

library(TabulaMurisData)

library(patchwork)

library(scater)

library(ExperimentHub)

library(SingleCellExperiment)

library(iSEE)

library(scran)

library(GEOquery)

library(Seurat)

library(RcmdrMisc)

library(dplyr)

library(reshape2)

setwd("~/Desktop/Reference\_Datasets/Mouse\_to\_human")

####Query - Mouse (upper case gene names) to human (upper case gene name)####

experiment.uc <- LoadH5Seurat("experiment.uc.H5Seurat")

query <- experiment

head(query[[]])

####Reference - Human (upper case gene names)####

##1) Human bone marrow

hu.BM.1 <- readRDS("./Human\_BM/AMLs\_Scano\_projected.rds")

head(hu.BM.1[[]])

reference <- hu.BM.1

####Process data and qquery in the same way####

##LogNormalization

DefaultAssay(reference) <- "RNA"

DefaultAssay(query) <- "RNA"

reference <- NormalizeData(reference)

reference <- FindVariableFeatures(reference)

reference <- ScaleData(reference)

query <- NormalizeData(query)

query <- FindVariableFeatures(query)

query <- ScaleData(query)

DefaultAssay(reference) <- "RNA"

DefaultAssay(query) <- "RNA"

##SCTransform

DefaultAssay(reference) <- "RNA"

DefaultAssay(query) <- "RNA"

reference <- SCTransform(reference, verbose = TRUE)

query <- SCTransform(query, verbose = TRUE)

DefaultAssay(reference) <- "SCT"

DefaultAssay(query) <- "SCT"

####Find Anchors between query and reference

##LogNormalization

anchors <- FindTransferAnchors(reference = reference, query = query, dims = 1:25, approx = FALSE)

predictions <- TransferData(anchorset = anchors,refdata = reference$ct)

query <- AddMetaData(object = query, metadata = predictions)

head(query[[]])

##SCTransform

anchors <- FindTransferAnchors(reference = reference, query = query, normalization.method = "SCT", dims = 1:25, approx = FALSE)

predictions <- TransferData(anchorset = anchors,refdata = reference$ct)

query <- AddMetaData(object = query, metadata = predictions)

head(query[[]])

####Visualise results####

##Colours

my.colors <- viridisLite::viridis(n = 100, option = "D")

colfunc <- colorRampPalette(c("white" ,"navy"))

colors3<-colfunc(100)

##Dim, Feature and VlnPlots

DimPlot(query, label = TRUE, group.by = "predicted.id", repel = TRUE, reduction = "wnn.umap") + ggtitle("Query")

FeaturePlot(query,"prediction.score.Pre.B.cells", reduction = "wnn.umap")

VlnPlot(query,"prediction.score.Pre.B.cells", pt.size = 0 )+NoLegend()

##Heatmap: Mean prediction scores per cluster

query@meta.data[,'Annot'] <-Idents(query)

predictions$prediction.score.max

Data<-query@meta.data

Data$orig.ident

Forest <-Data %>%

group\_by(`Annot`) %>%

summarize\_all(mean)

Forest<-as.data.frame(Forest)

rownames(Forest)<-Forest$Annot

Forest<-Forest[,c(20:62)]

final<-t(Forest)

pheatmap::pheatmap(final, cluster\_rows = T, cluster\_cols = T,show\_rownames = T, show\_colnames = T,

cellwidth = 10,cellheight = 10, angle\_col = 45)

##Heatmap: Percent oi cells associated with wthe reference cell type per cluster

number\_perCluster<- table(query$Annot,query$predicted.id)

percents<-rowPercents(number\_perCluster)

dim(percents)

percents<-percents[,c(-22,-23)]

pheatmap::pheatmap(t(percents), cluster\_rows = T, cluster\_cols = T,show\_rownames = T, show\_colnames = T,

cellwidth = 10,cellheight = 10, angle\_col = 45, color = colors3)

##Tables

number\_perPredicted.id <- table(query@meta.data$predicted.id,query@meta.data$orig.ident)

number\_perCluster <- table(query@meta.data$seurat\_clusters ,query@meta.data$orig.ident)

##Bar charts

DF1 <- melt(number\_perPredicted.id, id.var="Cluster")

ggplot(DF1, aes(Var2,value , fill = Var1)) +

geom\_bar(stat= "identity", position = "fill") +

xlab("Cluster") + ylab("Proportion of cluster")