Quick Start vignette for scOh

Ki Oh, Luke Torre-Healy, Richard Moffitt

05/05/2022

Contents

The following vignette is to aid in quickly implementing the scOh classifier. The scOh classifier was trained using the singleCellNet package. For a more detailed vignette on the functions and parameters of the single cell classifier architecture and functions, please refer to https://github.com/pcahan1/singleCellNet.

Required Installations (should install on scOh install)

```
library(singleCellNet)
library(dplyr)
library(Seurat)
library(scOH)
```

Quick Start

First, load in the trained classifier and your data set to be labeled. For this vignette, we use the validation set from our submission. You may slot in any PDAC single cell set you'd like into this vignette, so long as you adjust lines 30-35.

```
# Load Classifier, a list with three entries
data("trainedClassifier", verbose = T)
## name=trainedClassifier:   found in Rdata.rds

## If you need to, download the sample set using the following code.
## If not, please change lines 30-35 to point to your single cell dataset.
# wget http://pdacr.bmi.stonybrook.edu/scRNA/classification_sampleset.RData

# Load in data to classify from wherever you downloaded it
load("/mnt/lth-fs1/scRNA/classification_sampleset.RData")

## Convert Seurat data into dgCMatrix.
sampInfo <- ValidationSet@meta.data
sampInfo$sample_name <- rownames(sampInfo)
expMat <- ValidationSet@assays$RNA
```

Predict Cell Type

Once the scRNA data is loaded and represented as a matrix, scn_predict() will generate prediction probabilities for the 6 classes.

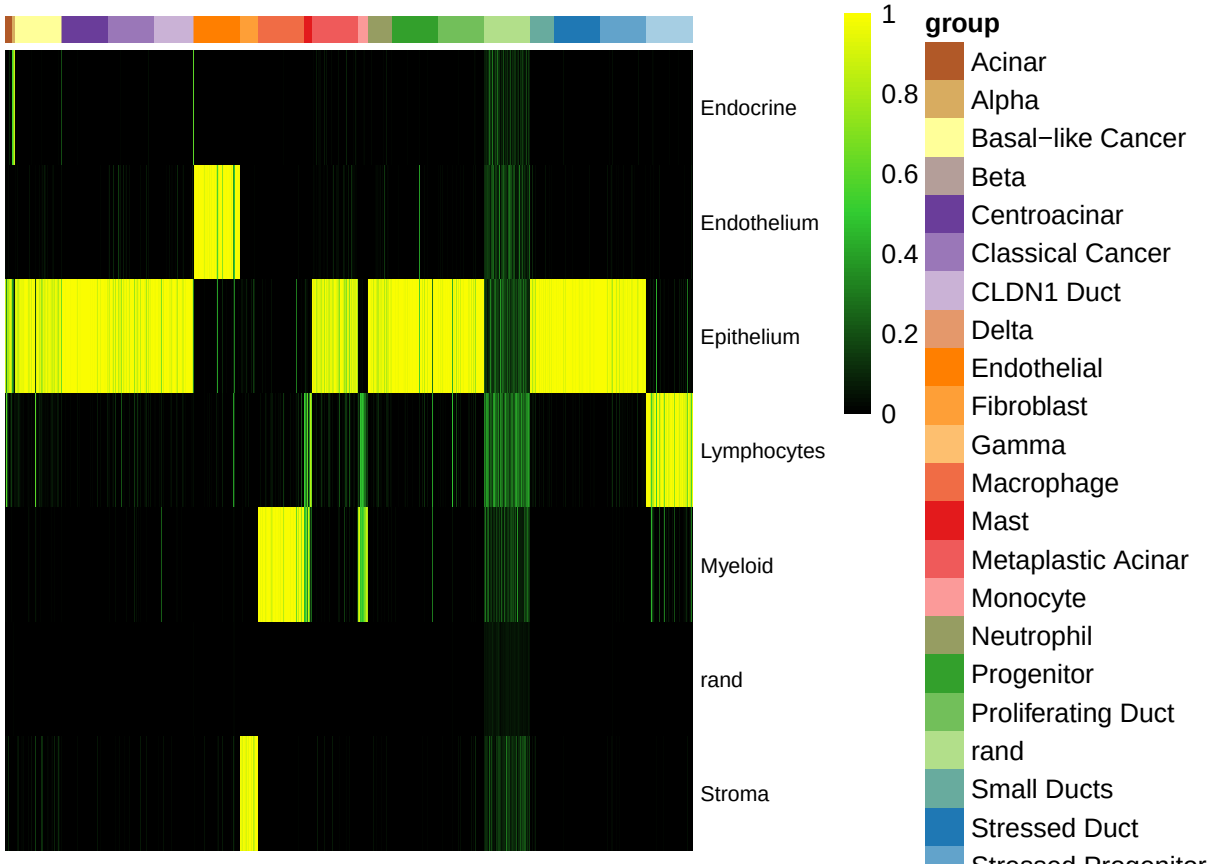
```
## Set number of random profiles to generate for eval process.
## Upper limit = sample size of experiment
nqRand = 2000

## Run your predictions as shown here.
celltypePredictions <- scn_predict(Classifier[['cnProc']], # The classifier
                                   expMat, # Expression values
                                   nrand = nqRand)
## Loaded in the cnProc
## All Done
str(celltypePredictions)
##  'matrix' num [1:7, 1:6278] 0.033 0.023 0.8665 0.0415 0.0125 ...
```

```
##  - attr(*, "dimnames")=List of 2
##   ..$ : chr [1:7] "Endocrine" "Endothelium" "Epithelium" "Lymphocytes" ...
##   ..$ : chr [1:6278] "cell_091-133_1_1" "cell_177-113_1_1" "cell_289-088_1_1" "cell_205-268_1_1" ...
```

Visualize Classifier Results

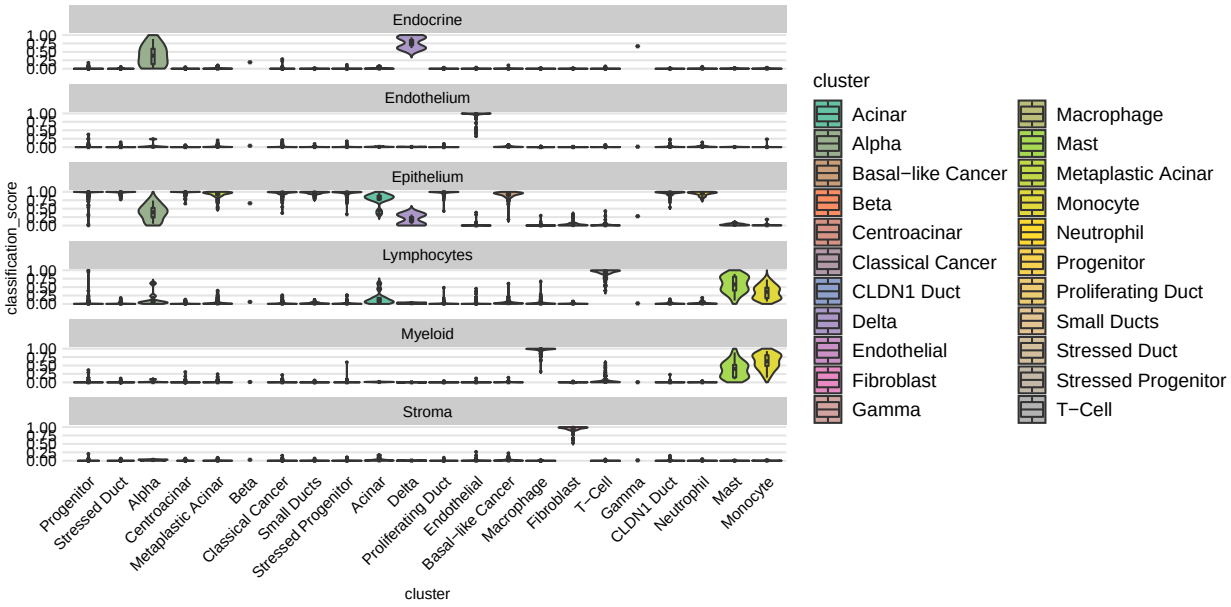
The output of scn_predict() is a matrix with 7 rows (6 classes + 1 rand) and a number of columns equal to nqRand + number of input columns (here, 7 x 6,278). We now generate a vector of column groups for a heatmap to visualize the output of the prediction probabilities.

```
## Make list of column groups for the classifier output heatmap.
##Our validation dataset has a variety of ground truth
## cell type labels ($CellTtype) that we use here.
sgrp = as.vector(sampInfo$CellType)
names(sgrp) = as.vector(sampInfo$sample_name)

## add the random values as columns to predicted values
grpRand =rep("rand", nqRand)
names(grpRand) = paste("rand_", 1:nqRand, sep='')
sgrp = append(sgrp, grpRand)

## Generate heatmap classification result
sc_hmClass(celltypePredictions, # The output of the prediction classifier
           sgrp, # colnames to group by
           maxPerGrp =100, # maximum cells per group to represent
           isBig=TRUE,
           cCol=F, # Cluster the output?
           font=8)
```



Using the sc_violinClass() function, we see that fibroblasts (light pink) are called Stroma (bottom row) and T cells (grey) are called Lymphocytes (fourth row), along with other robust predictions of cell type.

```
sampInfo <- get_cate(classRes = celltypePredictions,
                     sampTab = sampInfo,
                     dLevel = "CellType",
                     sid = "sample_name",
                     nrand = nqRand)
sc_violinClass(sampTab = sampInfo,
               classRes = celltypePredictions,
               sid = "sample_name",
               dLevel = "CellType",
               addRand = nqRand)
```



Join the result back to Original Seurat object

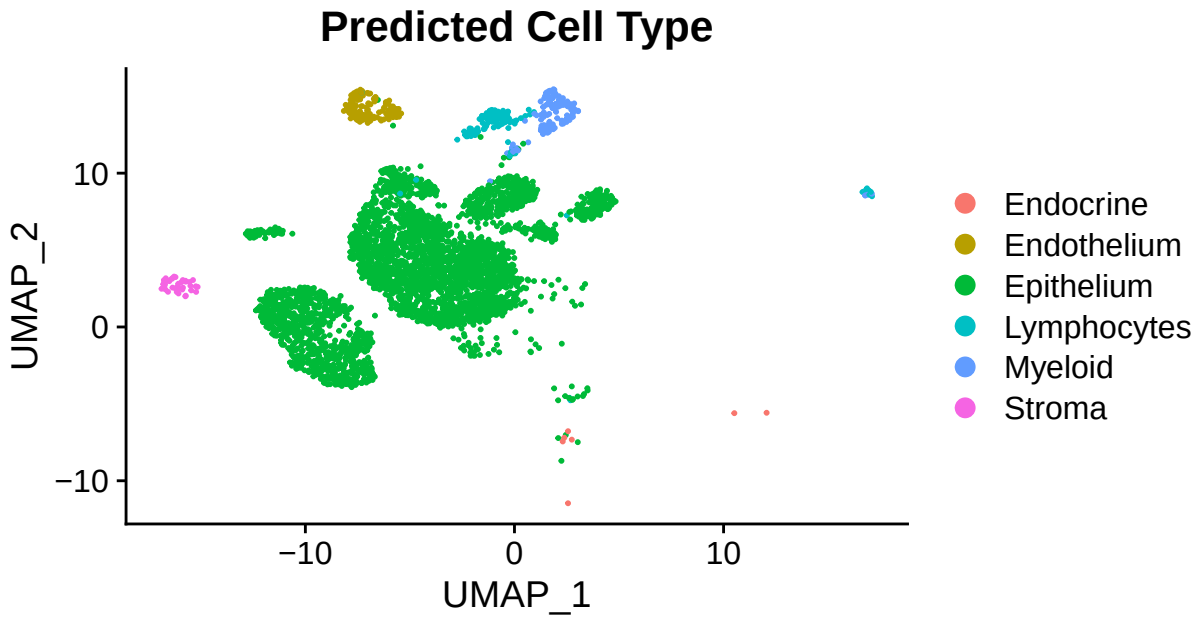
All that is left to do is remove the random entries in the matrix and join the prediction values back into your original Seurat object. This is made easy by the Seurat function AddMetadata() as shown below.

```
## remove the randomly generated cells from the prediction data frame
result = celltypePredictions[,!grepl("^rand_",colnames(celltypePredictions))]

## drop rand row : Classifier has six cell types and random will be at 6th row
result <- result[-which(rownames(result)=="rand"), ]

## Transpose the matrix and make it into dataframe
result_classi <- as.data.frame(t(result))

## Pick maximum result as the result of the Classifier.
result_classi$CellType_cs <- colnames(result_classi)[max.col(result_classi,
                                                             ties.method="first")]
result_classi$sample_name <- row.names(result_classi)

## Join prediction values and classification back into original Seurat Object
ValidationSet = Seurat::AddMetaData(ValidationSet,result_classi)
```

Visualize classifier output on Seurat Object

Below we show the similarity between our predicted cell types and the ground truth labels of the validation data.

```
DimPlot(ValidationSet, group.by = "CellType_cs") + ggtitle("Predicted Cell Type")
```




```
DimPlot(ValidationSet, group.by = "CellType") + ggtitle("Ground Truth Labels")
```