Response to reviewer: running CellAssign on McParland et al. data

Contents

```r
library(tidyverse)
library(SingleCellExperiment)
library(scater)
library(scran)
library(Matrix)
library(Seurat)

library(cellassign)
```

Introduction

We have prepared this notebook to serve as a guide for running CellAssign (on the MacParland et al. human liver dataset).

To see a detailed step-by-step for how to get CellAssign running on this data, please refer to the CellAssign section.

Preprocessing

Data acquisition

To obtain the normalized data used by the authors:

```r
# devtools::install_github("BaderLab/HumanLiver")
library(HumanLiver)

load(system.file("liver/HumanLiver.RData", package = "HumanLiver"))

# Put in a dummy matrix for raw.data to prevent Seurat::Convert from failing
HumanLiverSeurat@raw.data <-
  matrix(0,
         nrow = nrow(HumanLiverSeurat@data),
         ncol = ncol(HumanLiverSeurat@data))
```

```r
rownames(HumanLiverSeurat@raw.data) <- rownames(HumanLiverSeurat@data)
colnames(HumanLiverSeurat@raw.data) <- colnames(HumanLiverSeurat@data)

# Remove reduced dimension object as this causes conversion errors
HumanLiverSeurat@dr <- list()

# Convert Seurat object to SingleCellExperiment
sce_HL <- as.SingleCellExperiment(HumanLiverSeurat)
```

However, this data (both the version in `HumanLiver` and on `GEO`) is missing raw counts. To reproduce the raw counts, we processed the bam files deposited in SRA by McParland et al. with CellRanger v2.1.

```r
# Read matrices from CellRanger v2.1
humanliver_filtered_dirs <-
  Sys.glob(file.path('/datadrive/projects/cellassign-paper/data/baderlab/nick_outputs',
                     '*',
                     'outs',
                     'filtered_gene_bc_matrices',
                     'GRCh38'))
names(humanliver_filtered_dirs) <- basename(humanliver_filtered_dirs)

sce <- DropletUtils::read10xCounts(humanliver_filtered_dirs)

# Include patient as a metadata column
metadata <- data.frame(Sample=humanliver_filtered_dirs) %>%
  dplyr::mutate(patient=paste0("P",
                               str_extract(Sample, "(?<=patient)[0-9]+"),
                               "TLH"))

colData(sce) <- colData(sce) %>%
  data.frame(check.names = FALSE, stringsAsFactors = FALSE) %>%
  dplyr::left_join(metadata) %>%
  DataFrame(check.names = FALSE, stringsAsFactors = FALSE)

sce <- sce %>%
  scater::mutate(cellname=paste(patient,
                                str_replace_all(Barcode, "\\-", "_"),
                                sep = "_"))
```

Merging raw counts with normalized SingleCellExperiment

We'll substitute in the remainder of the data for the matrix, using the values from the `HumanLiver` R package on GitHub created by the authors (`BaderLab/HumanLiver`).

```r
# Use gene symbols instead of Ensembl gene IDs, to imitate McParland et al.
sce_filtered <- sce %>%
  scater::filter(cellname %in% colnames(sce_HL))

gene_ids <- get_ensembl_id(rownames(sce_HL), sce_filtered)

sce_subset <- sce_filtered[gene_ids,]
colnames(sce_subset) <- sce_subset$cellname
rownames(sce_subset) <- rownames(sce_HL)
```

```
cells_intersect <- intersect(colnames(sce_HL),
                             colnames(sce_subset))

sce_merged <- sce_HL[,cells_intersect]
counts(sce_merged) <- counts(sce_subset[,cells_intersect])
sce_merged$patient <- sce_subset[,cells_intersect]$patient
```

We add cell type annotations, using the `viewHumanLiver()` function to determine the authors' mapping between clusters and cell types:

```
celltype_labels <- c(
  '1'='Hepatocytes',
  '2'='ab T cells',
  '3'='Hepatocytes',
  '4'='Macrophages',
  '5'='Hepatocytes',
  '6'='Hepatocytes',
  '7'='Plasma cells',
  '8'='NK cells',
  '9'='gd T cells',
  '10'='Macrophages',
  '11'='LSECs',
  '12'='LSECs',
  '13'='LSECs',
  '14'='Hepatocytes',
  '15'='Hepatocytes',
  '16'='Mature B cells',
  '17'='Cholangiocytes',
  '18'='gd T cells',
  '19'='Erythroid cells',
  '20'='Hepatic Stellate Cells'
)

sce_merged <- sce_merged %>%
  scater::mutate(
    celltype=celltype_labels[as.character(ident)],
    patient=str_extract()
  )
```

We perform dimensionality reduction, since those slots were not properly moved over by `Seurat::Convert`.

```
sce_merged <- runPCA(sce_merged, ntop = 1000, ncomponents = 50)
sce_merged <- runTSNE(sce_merged, use_dimred = 'PCA', n_dimred = 50)
sce_merged <- runUMAP(sce_merged, use_dimred = 'PCA', n_dimred = 50)
```

```
print(sce_merged)
```

```
## class: SingleCellExperiment
## dim: 20007 6408
## metadata(0):
## assays(2): counts logcounts
## rownames(20007): RP11-34P13.7 FO538757.2 ... AC233755.1 AC240274.1
## rowData names(1): gene
## colnames(6408): P2TLH_AAACCTGGTACGCTGC_1 P2TLH_AAACGGGAGCGAGAAA_1
##   ... P5TLH_TTTGTCAGTTTAGGAA_1 P5TLH_TTTGTCATCAGCTTAG_1
## colData names(12): total_counts total_features ... pct_counts_mito
##   pct_counts_ribo
## reducedDimNames(3): PCA TSNE UMAP
## spikeNames(0):
```

CellAssign

Recommendation 1: The error the reviewer encountered was a result of some library sizes in the input data being equal to zero. This occurred because size factors were computed from the subsetted SingleCellExperiment (i.e. with only marker gene rows), rather than the whole SingleCellExperiment. The `NA` values during inference result when the model attempts to compute `log(0)`.

Solution: Size factors should be computed on the full SingleCellExperiment object. Size factors from a subsetted SCE may not be representative of the actual library size/normalization factors that should be applied to cell subsets. Please refer to the following warning message thrown by calling `cellassign`: `It is highly recommended to supply size factors calculated using the full gene set`.

We have additionally updated the GitHub repository for CellAssign to throw an error message when any cells with size factors of <= 0 are provided.

Recommendation 2: Since the experiment consists of data for multiple patients, `patient` should be included as a covariate in the design matrix `X`.

Solution: `patient` is included as a covariate in the design matrix `X`.

Note: The data for patient 1 was not available for this analysis, as 10x Genomics' `CellRanger v2.1` did not finish running on that patient's data at the time of this response.

Size factor computation

We begin by computing size factors on the full SingleCellExperiment:

```
sce_merged <- scran::computeSumFactors(sce_merged, BPPARAM = MulticoreParam(10))
```

We next confirm that none of the size factors are negative or zero.

```
summary(sizeFactors(sce_merged))
```

```
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
##  0.004849  0.463910  0.757003  1.000000  1.269547 15.252579
```

Using the marker gene matrix provided by the reviewer (we use the `marker_list_to_mat` helper function implemented in the `cellassign` package for readability):

```
sce_merged_subset <- sce_merged %>%
  scater::filter(celltype %in% c("Cholangiocytes",
                                 "Mature B cells",
                                 "Hepatocytes"))

# Reviewer's marker gene matrix
marker_list <- list(
  'Cholangiocytes'=c("FXYD2", "KRT7", "TACSTD2", "CLDN10", "KRT19"),
  'Mature B cells'=c("MS4A1", "IGHD", "LINC00926", "TNFRSF13C", "VPREB3"),
  'Hepatocytes'=c("TAT-AS1", "APOA5", "ADH4", "CYP2C9", "LEAP2")
)
```

```r
mgi <- marker_list_to_mat(marker_list, include_other = FALSE)
```

Finally, we run CellAssign. We use a design matrix that includes patient as a covariate as per **Recommendation 2**.

```r
# Design matrix for patient-specific effects
design <- model.matrix(~ patient,
                       colData(sce_merged_subset))

gene_order <- intersect(rownames(mgi), rownames(sce_merged_subset))

fit <- cellassign(
  exprs_obj = sce_merged_subset[gene_order,],
  marker_gene_info = mgi[gene_order,],
  s = sizeFactors(sce_merged_subset),
  X = design,
  shrinkage = TRUE,
  min_delta = 2 # Set minimum LFC for marker genes
  # Higher values are recommended when the 'other' group is not included
  # or when substantial imbalance exists between cell type proportions
)
```

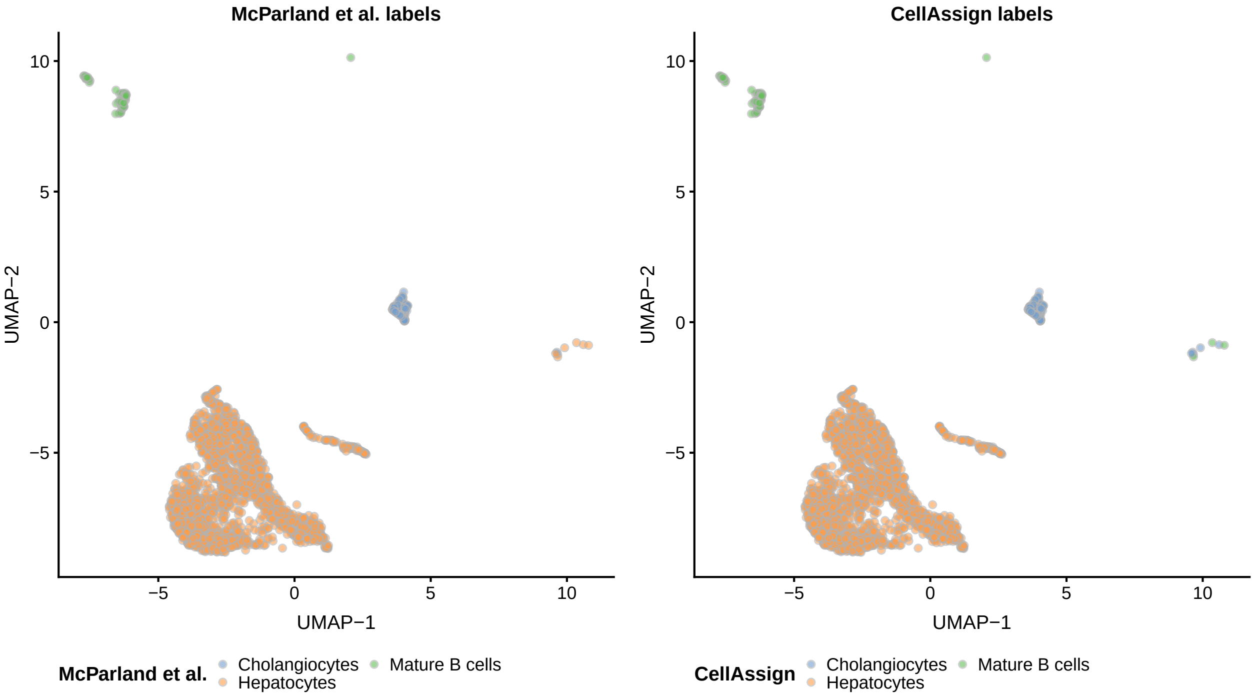
We annotate our results on the SingleCellExperiment, and assess cells that are assigned with a CellAssign probability of >= 0.99.

```r
sce_merged_subset_labeled <- sce_merged_subset

colData(sce_merged_subset_labeled) <-
  colData(sce_merged_subset_labeled) %>%
  as.data.frame %>%
  cbind(fit$mle_params$gamma) %>%
  dplyr::mutate(max_prob=rowMaxs(fit$mle_params$gamma)) %>%
  DataFrame()

sce_merged_subset_labeled$cellassign_clust <- fit$cell_type

sce_merged_subset_labeled <- sce_merged_subset_labeled %>%
  scater::filter(max_prob >= 0.99)
```

Finally, we'll plot the cells in UMAP space:

```r
## MacParland et al.
p1 <- plotUMAP(sce_merged_subset_labeled,
               colour_by = "celltype") +
  guides(fill = guide_legend(title = "McParland et al.",
                             ncol = 2)) +
  theme(legend.position = "bottom",
        legend.direction = "horizontal",
        legend.key.size = unit(0.2, "cm"),
        legend.margin = unit(0.5, "lines"),
        legend.title = element_text(face = "bold"),
        legend.box.just = "left") +
  xlab("UMAP-1") +
  ylab("UMAP-2") +
  ggtitle("McParland et al. labels")
```

```
## CellAssign
p2 <- plotUMAP(sce_merged_subset_labeled,
               colour_by = "cellassign_clust") +
  guides(fill = guide_legend(title = "CellAssign", ncol = 2)) +
  theme(legend.position = "bottom",
        legend.direction = "horizontal",
        legend.key.size = unit(0.2, "cm"),
        legend.margin = unit(0.5, "lines"),
        legend.title = element_text(face = "bold"),
        legend.box.just = "left") +
  xlab("UMAP-1") +
  ylab("UMAP-2") +
  ggtitle("CellAssign labels")

cowplot::plot_grid(p1, p2, ncol = 2, align = 'hv', axis = 'tblr')
```



Contingency table

```
with(colData(sce_merged_subset_labeled),
     table(celltype, cellassign_clust))
```

```
##                   cellassign_clust
## celltype            Cholangiocytes Hepatocytes Mature B cells
##   Cholangiocytes               106           0              0
##   Hepatocytes                    3        2330              4
##   Mature B cells                 0           0             65
```