Compare reference genotypes for vireo

Ariel Hippen

2022-06-27

Contents

```
suppressPackageStartupMessages({
  library(data.table)
  library(scater)
  library(ggplot2)
  library(dplyr)
  library(caret)
  library(yaml)
})
```

Load data

```
# We have two sets of pooled samples, named after the date they were run (12162021 and 01132022).
sample_id <- "12162021"

params <- read_yaml("../../config.yml")
data_path <- params$data_path
local_data_path <- params$local_data_path
samples <- params$samples
```

```
# Load hash and vireo (genetic) demultiplexing assignments
# Note: variable data_path is loaded from config.R
hashing <- fread(paste(data_path, "pooled_tumors", sample_id,
                       "Cellranger/outs/multi/multiplexing_analysis/assignment_confidence_table.csv", se

vireo_path <- paste(data_path, "pooled_tumors", sample_id, "vireo", sep = "/")
chunk_ribo <- fread(paste(vireo_path, "chunk_ribo/donor_ids.tsv", sep = "/"))
dissociated_ribo <- fread(paste(vireo_path, "dissociated_ribo/donor_ids.tsv", sep = "/"))
dissociated_polyA <- fread(paste(vireo_path, "dissociated_polyA/donor_ids.tsv", sep = "/"))
```

```
# Load SingleCellExperiment object for plotting
sce <- readRDS(paste("../../data/sce_objects/", sample_id, ".rds", sep = ""))
```

```
# Filter hashing and vireo to cells in sce object, all others failed miQC filtering
hashing <- subset(hashing, hashing$Barcodes %in% sce$Barcode)
chunk_ribo <- subset(chunk_ribo, chunk_ribo$cell %in% sce$Barcode)
dissociated_ribo <- subset(dissociated_ribo, dissociated_ribo$cell %in% sce$Barcode)
dissociated_polyA <- subset(dissociated_polyA, dissociated_polyA$cell %in% sce$Barcode)
```

```
# Subset down and join into one matrix
setnames(chunk_ribo, "donor_id", "chunk_ribo_assignment")
setnames(dissociated_ribo, "donor_id", "dissociated_ribo_assignment")
setnames(dissociated_polyA, "donor_id", "dissociated_polyA_assignment")

assignments <- full_join(subset(chunk_ribo, select = c("cell", "chunk_ribo_assignment")),
                         subset(dissociated_ribo, select = c("cell", "dissociated_ribo_assignment"))) %
               full_join(., subset(dissociated_polyA, select = c("cell", "dissociated_polyA_assignment"
```

```
## Joining, by = "cell"
## Joining, by = "cell"
```

```
assignments$chunk_ribo_assignment <- as.factor(assignments$chunk_ribo_assignment)
assignments$dissociated_ribo_assignment <- as.factor(assignments$dissociated_ribo_assignment)
assignments$dissociated_polyA_assignment <- as.factor(assignments$dissociated_polyA_assignment)
```

Compare vireo assignments

```
# Run confusion matrix
confusionMatrix(assignments$chunk_ribo_assignment, assignments$dissociated_ribo_assignment)$table
```

```
##             Reference
## Prediction   donor0 donor1 donor2 donor3 doublet unassigned
##   donor0          0   2854      0      0       3          5
##   donor1          0      0   1718      1       1          0
##   donor2          0      0      0   1105       1          0
##   donor3       1008      0      0      0       1          1
##   doublet         6      0      7      6     571         22
##   unassigned      4      2      4      1       9         28
```

```
confusionMatrix(assignments$chunk_ribo_assignment, assignments$dissociated_polyA_assignment)$table
```

```
##             Reference
## Prediction   donor0 donor1 donor2 donor3 doublet unassigned
##   donor0          0   2855      0      0       3          4
##   donor1          1      0   1716      0       2          1
##   donor2       1104      0      0      0       2          0
##   donor3          0      0      0   1009       0          1
##   doublet         1      2      4      2     592         11
##   unassigned      1      2      3      2       4         36
```

```
confusionMatrix(assignments$dissociated_ribo_assignment, assignments$dissociated_polyA_assignment)$table
```

```
##             Reference
## Prediction   donor0 donor1 donor2 donor3 doublet unassigned
##   donor0          0      0      0   1010       5          3
##   donor1          0   2850      0      0       1          5
##   donor2          0      0   1719      0       8          2
##   donor3       1107      0      0      0       5          1
##   doublet         0      4      4      2     566         10
##   unassigned      0      5      0      1      18         32
```

There seems to be a lot of concordance across the sets, which is encouraging. I want to check if there are patterns in the off-diagonal cells.

Since donor labels are assigned randomly, all "donor2" cells in chunk_ribo might be called "donor0" in dissociated_ribo. I'll have to manually reassign them to a new name. I'll use A, B, C, and D.

```
assignments$CR <- assignments$chunk_ribo_assignment
assignments$CR <- recode(assignments$CR,
                         "donor0" = "A",
                         "donor1" = "B",
                         "donor2" = "C",
                         "donor3" = "D")

assignments$DR <- assignments$dissociated_ribo_assignment
assignments$DR <- recode(assignments$DR,
                         "donor0" = "D",
                         "donor1" = "A",
                         "donor2" = "B",
                         "donor3" = "C")

assignments$DP <- assignments$dissociated_polyA_assignment
assignments$DP <- recode(assignments$DP,
                         "donor0" = "C",
                         "donor1" = "A",
                         "donor2" = "B",
                         "donor3" = "D")
```

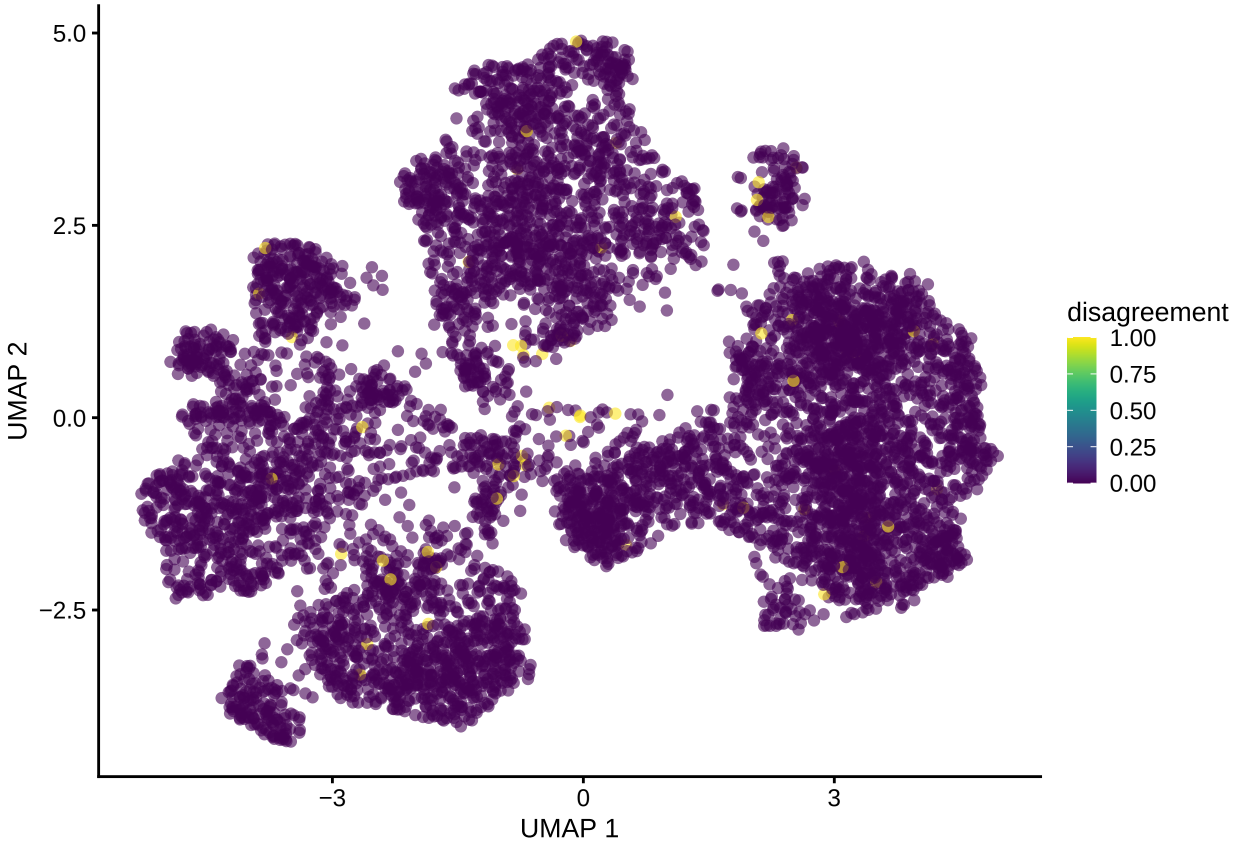
```
# Combine "doublet" and "unassigned" into one category, the distinction doesn't seem important
assignments[assignments$CR == "doublet", ]$CR <- "unassigned"
assignments[assignments$DR == "doublet", ]$DR <- "unassigned"
assignments[assignments$DP == "doublet", ]$DP <- "unassigned"

# Redo factor levels
assignments$CR <- as.factor(as.character(assignments$CR))
assignments$DR <- as.factor(as.character(assignments$DR))
assignments$DP <- as.factor(as.character(assignments$DP))
```

```
# Look for patterns in cells where there are disagreements
assignments$disagreement <- ifelse(assignments$CR != assignments$DR |
                                     assignments$CR != assignments$DP,
                                   1, 0)
sce$disagreement <- assignments$disagreement
table(sce$disagreement)
```

```
##
##    0    1
## 7298   60
```

```r
plotUMAP(sce, colour_by = "disagreement")
```



The cells with "disagreements" seem to be more or less randomly distributed. I'm happy with this.

Compare to hash demultiplexing

One last question: these are extremely concordant, but does one bulk type give a marginal increase in concordance with hash demultiplexing?

```r
#Add hashing results to assignments table
hashing <- subset(hashing, select = c("Barcodes", "Assignment"))
setnames(hashing, c("cell", "hash_assignment"))
assignments <- full_join(hashing, assignments)
```

```
## Joining, by = "cell"
```

```r
# Switch hash results to A/B/C/D notation
assignments$H <- assignments$hash_assignment
assignments[assignments$H == "Blanks" | assignments$H == "Multiplet" | assignments$H == "Unassigned", ]$
assignments$H <- as.factor(assignments$H)
```

```
assignments$H <- recode(assignments$H,
                         "anti-human_Hashtag1" = "D",
                         "anti-human_Hashtag2" = "C",
                         "anti-human_Hashtag3" = "A",
                         "anti-human_Hashtag4" = "B")
```

```
# Count number of disagreements between hashing and each run of vireo
length(which(assignments$H != assignments$CR))
```

```
## [1] 3304
```

```
length(which(assignments$H != assignments$DR))
```

```
## [1] 3299
```

```
length(which(assignments$H != assignments$DP))
```

```
## [1] 3295
```

```
confusionMatrix(assignments$H, assignments$CR)$table
```

```
## Warning in confusionMatrix.default(assignments$H, assignments$CR): Levels are
## not in the same order for reference and data. Refactoring data to match.
```

```
##             Reference
## Prediction      A    B    C    D unassigned
##   A           564    3    1    3          7
##   B           124 1405    1    7         92
##   C            51    1  954    3        110
##   D            23    0    1  788        108
##   unassigned 2100  311  149  209        343
```

```
confusionMatrix(assignments$H, assignments$DR)$table
```

```
## Warning in confusionMatrix.default(assignments$H, assignments$DR): Levels are
## not in the same order for reference and data. Refactoring data to match.
```

```
##             Reference
## Prediction      A    B    C    D unassigned
##   A           564    3    1    3          7
##   B           123 1410    1    7         88
##   C            50    0  959    3        107
##   D            23    0    1  791        105
##   unassigned 2096  316  151  214        335
```

```
confusionMatrix(assignments$H, assignments$DP)$table
```

```
## Warning in confusionMatrix.default(assignments$H, assignments$DP): Levels are
## not in the same order for reference and data. Refactoring data to match.
##             Reference
## Prediction      A    B    C    D unassigned
##   A           564    3    1    3          7
##   B           124 1406    1    7         91
##   C            51    0  956    3        109
##   D            23    0    1  792        104
##   unassigned 2097  314  148  208        345
```

Conclusion: there is only the tiniest of differences in the number of cells that are picked up by genetic demultiplexing but not hash demultiplexing, and literally no difference in the number of cells that get confidently assigned by both. Looks like it doesn't matter.