Bio-Rad Take Home Project

Todd Warczak

11/2/2021

```r
library(tidyverse)
library(data.table)
library(here)
library(gridExtra)
library(gt)
library(ggtext)
library(ggdark)
```

```r
count_tbl_filt <- data.table::fread(here("data", "count_tbl_filt.csv"))
```

Data set dimensions (rows, columns)

```r
dim(count_tbl_filt)
```

```
## [1] 128804   5580
```

First 4 rows and 2 columns of `count_tbl_filt`.

```r
count_tbl_filt[1:4, 1:2]
```

```
##                          V1 alignments.possorted.tagged_BC00001_N01
## 1: hg19_chr1:713971-714221                                       0
## 2: hg19_chr1:714480-714730                                       0
## 3: hg19_chr1:762720-762970                                       0
## 4: hg19_chr1:825991-826241                                       0
```

Create useful variables from the gene ID (`V1`) column.

```r
count_tbl_filt_2 <- count_tbl_filt %>%
  separate(V1, c("org_chr", "bp_loc"), ":", remove = FALSE) %>%
  separate(org_chr, c("ref_genome", "chromosome"), "_") %>%
  mutate(chromosome = str_remove(chromosome, 'chr')) %>%
  mutate(chromosome = factor(chromosome,
                             levels  = c('1','2','3','4','5','6','7','8','9','10',
                                         '11','12','13','14','15','16','17','18',
                                         '19','20','21','22','X'),
                             ordered = TRUE)) %>%
  separate(bp_loc, c("from_loc", "to_loc")) %>%
  mutate_at(c("from_loc", "to_loc"), as.numeric) %>%
  select(ref_genome, chromosome, from_loc, to_loc, drops_with_gene, 6:last_col())

rm(count_tbl_filt)
```

Now we can group_by, filter, … `count_tbl_filt_2` by reference genome, chromosome, or locus. The `drops_with_gene` variable was created earlier when filtering out genes present in < 1% of droplets.

First 4 rows and 5 columns of `count_tbl_filt_2`.

```
count_tbl_filt_2[1:4, 1:5]
```

```
##    ref_genome chromosome from_loc to_loc drops_with_gene
## 1:       hg19          1   713971 714221             630
## 2:       hg19          1   714480 714730              88
## 3:       hg19          1   762720 762970             189
## 4:       hg19          1   825991 826241              56
```

Find total gene read counts for both reference genomes in each droplet.

```
count_tbl_sum <- count_tbl_filt_2 %>%
  group_by(ref_genome) %>%
  summarise(across(5:last_col(), .fns = sum))
```

View total read counts for both reference genomes in first 2 droplets.

```
count_tbl_sum[,1:3]
```

```
## # A tibble: 2 x 3
##   ref_genome alignments.possorted.tagged_BC00001_N01 alignments.possorted.tagge~
##   <chr>                                        <int>                       <int>
## 1 hg19                                            51                         146
## 2 mm10                                           962                        3724
```

Reshape `count_tbl_sum` so droplets are observations (rows). Filter to retain droplets that have > 800 read counts in at least 1 reference genome.

```
count_tbl_sum_long <- count_tbl_sum %>%
  pivot_longer(cols = -ref_genome) %>%
  pivot_wider(names_from = ref_genome, values_from = value) %>%
  rename(total_hg19_counts = hg19, total_mm10_counts = mm10) %>%
  filter(total_hg19_counts > 800 | total_mm10_counts > 800)
```

View first 4 rows in `count_tbl_sum_long`.

```
count_tbl_sum_long[1:4,]
```

```
## # A tibble: 4 x 3
##   name                                    total_hg19_counts total_mm10_counts
##   <chr>                                               <int>             <int>
## 1 alignments.possorted.tagged_BC00001_N01                51               962
## 2 alignments.possorted.tagged_BC00002_N02               146              3724
## 3 alignments.possorted.tagged_BC00003_N03                78              2067
## 4 alignments.possorted.tagged_BC00004_N03             12796               149
```

Create a summary table to tally unique genes expressed in each droplet. This is different than total counts. If a gene in the `count_tbl` is >= 1, `count_tbl_tally` will count it as 1.

```
count_tbl_tally <- count_tbl_filt_2 %>%
  group_by(ref_genome) %>%
  summarise(across(5:last_col(), ~ length(which(.x != 0))))
```

Create `drops_filter` to get names of droplets with > 800 read counts in at least 1 reference genome. Then use `drops_filter` to filter `count_tbl_tally`.

```
drops_filter <- count_tbl_sum_long %>% pull(name)

count_tbl_tally_long <- count_tbl_tally %>%
  pivot_longer(cols = -ref_genome) %>%
  pivot_wider(names_from = ref_genome, values_from = value) %>%
  rename(unique_hg19_genes = hg19, unique_mm10_genes = mm10) %>%
  filter(name %in% drops_filter)
```

View first 4 rows in `count_tbl_tally_long`.

```
count_tbl_tally_long[1:4,]
```

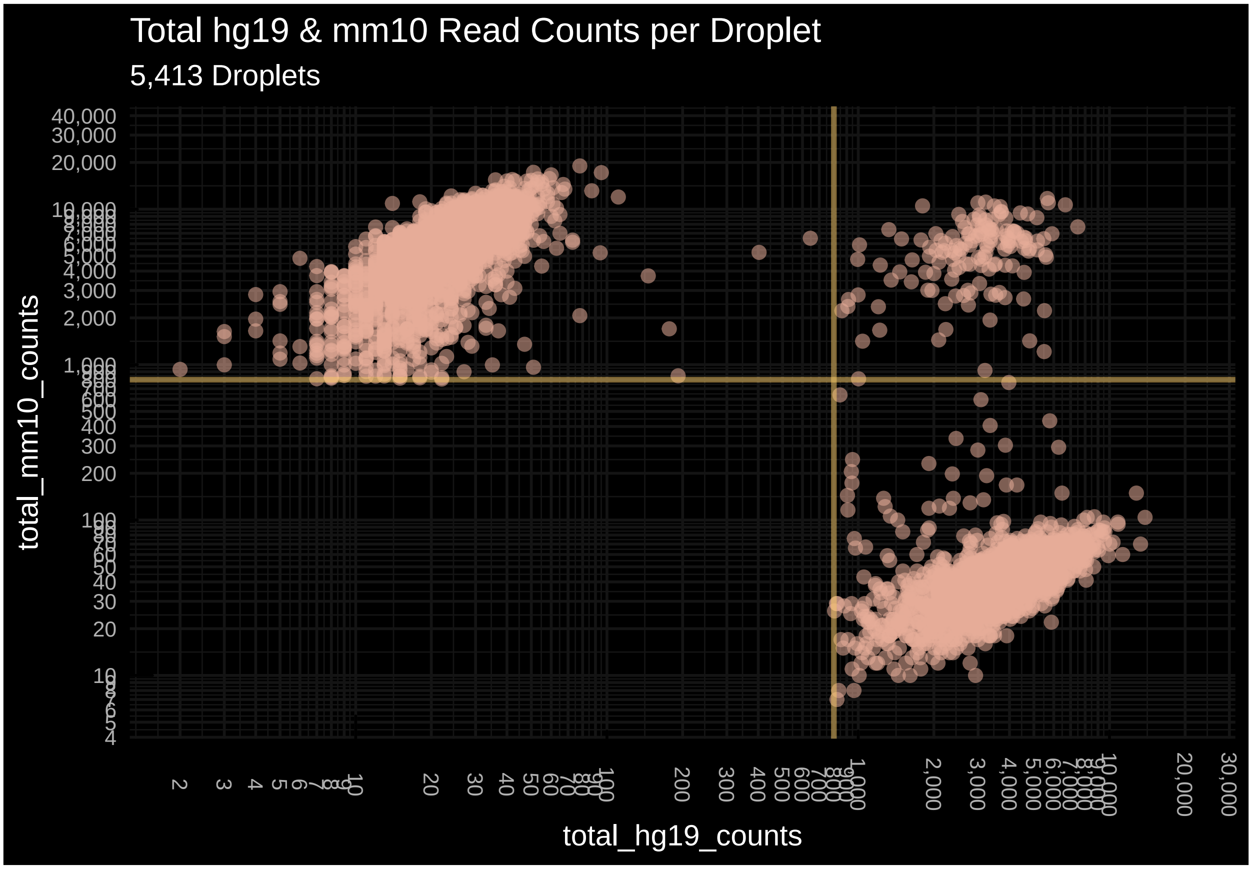
```
## # A tibble: 4 x 3
##   name                                     unique_hg19_genes unique_mm10_genes
##   <chr>                                                <int>             <int>
## 1 alignments.possorted.tagged_BC00001_N01                 51               935
## 2 alignments.possorted.tagged_BC00002_N02                144              3554
## 3 alignments.possorted.tagged_BC00003_N03                 78              1935
## 4 alignments.possorted.tagged_BC00004_N03              10853               141
```

Join `count_tbl_sum_long` and `count_tbl_tally_long` summary tables. Create 2 variables for the ratios of unique_genes/total_counts, one for each reference genome.

```
count_tbl_join <- count_tbl_sum_long %>%
  left_join(count_tbl_tally_long, by = "name") %>%
  mutate(hg19_unique_count_ratio = log10(unique_hg19_genes)/log10(total_hg19_counts),
         mm10_unique_count_ratio = log10(unique_mm10_genes)/log10(total_mm10_counts))
```

Add a plot here of log10(hg19 Counts) vs log10(mm10 Counts), no grouping. Just to show its easy to identify Mouse + Human doublets. Count those and use that number to find the number of expected Human + Human and Mouse + Doublets.

```
count_tbl_join  %>%
  ggplot(aes(x = total_hg19_counts, y = total_mm10_counts)) +
    geom_point(alpha = 0.55, size = 2, color = "#e7ad99") +
    scale_y_log10(breaks = scales::breaks_log(n = 38),
                  labels = scales::number_format(accuracy = 1, big.mark = ","),
                  limit  = c(6, 30000)) +
    scale_x_log10(breaks = scales::breaks_log(n = 40),
                  labels = scales::number_format(accuracy = 1, big.mark = ","),
                  limit  = c(2, 20000)) +
    geom_hline(yintercept = c(800),
               color      = "#FFCF6A",
               alpha      = 0.5,
               size       = 1) +
    geom_vline(xintercept = c(800),
               color      = "#FFCF6A",
               alpha      = 0.5,
               size       = 1) +
    ggdark::dark_theme_minimal() +
    theme(axis.text.x     = element_text(angle = -90, vjust = 0.5, size = rel(0.9)),
          axis.text.y     = element_text(size = rel(0.9)),
          title           = element_text(size = rel(1.0)),
          legend.position = c(0.4, 0.3),
          legend.text     = element_text(size = rel(0.8))) +
    annotation_logticks(scaled = FALSE) +
    labs(title    = "Total hg19 & mm10 Read Counts per Droplet",
         subtitle = "5,413 Droplets") +
    guides(shape = "none")
```



We now need to identify multiplets (droplets containing multiple cells) Multiplets containing a human and mouse cell are easiest to identify: Droplets with total read counts for both reference genomes > 800 or all droplets in quadrant I.

Count droplets in each quadrant and use those counts to find expected number of multiplets in each quadrant.

```
quad_1_count <- count_tbl_join %>%
  filter(total_hg19_counts > 800 & total_mm10_counts > 800) %>%
  nrow()

quad_4_count <- count_tbl_join %>%
  filter(total_hg19_counts > 800 & total_mm10_counts < 800) %>%
  nrow()

quad_2_count <- count_tbl_join %>%
  filter(total_hg19_counts < 800 & total_mm10_counts > 800) %>%
  nrow()

ratio_quad_2_4 <- quad_2_count/quad_4_count
ratio_quad_4_2 <- quad_4_count/quad_2_count

expect_quad_2_multi_count <- round(quad_1_count*ratio_quad_2_4, dig = 0)
expect_quad_4_multi_count <- round(quad_1_count*ratio_quad_4_2, dig = 0)
```

`expect_quad_2_multi_count` is the expected n doublets in quadrant II. Droplets containing 2+ human OR 2+ mouse cells can be identified by having lower ratio of unique genes to total read counts in the hg19 or mm10 reference genome, respectively. Droplets containing 1+ human AND 1+ mouse cells will have typical ratios of unique genes to total read counts. Find the value for nth lowest `unique_count_ratio.` Droplets with values <= to the nth lowest `unique_count_ratio` will be classified as multiplets. Remember to filter counts for quadrant II and IV.

```
hg19_unique_count_ratio_cutoff <- count_tbl_join %>%
                                  filter(total_hg19_counts > 800) %>%
                                  select(hg19_unique_count_ratio) %>%
                                  arrange(hg19_unique_count_ratio) %>%
                                  pluck(1, expect_quad_4_multi_count)

mm10_unique_count_ratio_cutoff <- count_tbl_join %>%
                                  filter(total_mm10_counts > 800) %>%
                                  select(mm10_unique_count_ratio) %>%
                                  arrange(mm10_unique_count_ratio) %>%
                                  pluck(1, expect_quad_2_multi_count)
```

Create new variables to label multiplets in the different quadrants:

- Logical variables to label Human + Human in quadrant IV (`hg19_multiplet`), Mouse + Mouse in quadrant II (`mm10_multiplet`), and Human + Mouse in quadrant I (`hg19_mm10_multiplet`).
- Logical variable that labels any multiplet droplet (`Multiplet`).
- Character variable that labels the type of multiplet (`Multiplet_Type`).
- Character variable that labels multiplet type and singlet type (`Droplet_Contents`).

```
count_tbl_summary <- count_tbl_join %>%
  mutate(
    hg19_multiplet = ifelse(hg19_unique_count_ratio <= hg19_unique_count_ratio_cutoff
                            & total_hg19_counts > 800, TRUE, FALSE),
    mm10_multiplet = ifelse(mm10_unique_count_ratio <= mm10_unique_count_ratio_cutoff
                            & total_mm10_counts > 800, TRUE, FALSE),
    hg19_mm10_multiplet = ifelse(total_hg19_counts > 800
                                 & total_mm10_counts > 800, TRUE, FALSE),
    Multiplet = ifelse(hg19_multiplet
                       | mm10_multiplet
                       | hg19_mm10_multiplet, TRUE, FALSE),
    Multiplet_Type = ifelse(hg19_mm10_multiplet,
                            "Human + Mouse",
                            ifelse(Multiplet == TRUE & total_hg19_counts > 800
                                   & total_mm10_counts < 800, "Human + Human",
                                 ifelse(Multiplet == TRUE & total_mm10_counts > 800
                                        & total_hg19_counts < 800, "Mouse + Mouse",
                                        "Singlet"))),
    Droplet_Contents = ifelse(Multiplet_Type == "Human + Mouse", "Human + Mouse",
                              ifelse(Multiplet_Type == "Human + Human", "Human + Human",
                                     ifelse(Multiplet_Type == "Mouse + Mouse",
                                            "Mouse + Mouse",
                                            ifelse(Multiplet_Type == "Singlet"
                                                   & total_hg19_counts < 800, "Mouse",
                                                   "Human")))))
```

View new variables in `count_tbl_summary`.

```
glimpse(count_tbl_summary)
```

```
## Rows: 5,413
## Columns: 13
## $ name                     <chr> "alignments.possorted.tagged_BC00001_N01", "al~
## $ total_hg19_counts        <int> 51, 146, 78, 12796, 8706, 49, 63, 39, 5323, 38~
## $ total_mm10_counts        <int> 962, 3724, 2067, 149, 105, 11183, 5610, 10488,~
## $ unique_hg19_genes        <int> 51, 144, 78, 10853, 7761, 48, 59, 35, 4829, 34~
## $ unique_mm10_genes        <int> 935, 3554, 1935, 141, 104, 9559, 5208, 9176, 9~
## $ hg19_unique_count_ratio <dbl> 1.0000000, 0.9972323, 1.0000000, 0.9825851, 0.~
## $ mm10_unique_count_ratio <dbl> 0.9958556, 0.9943175, 0.9913555, 0.9889714, 0.~
## $ hg19_multiplet           <lgl> FALSE, FALSE, FALSE, TRUE, TRUE, FALSE, FALSE,~
## $ mm10_multiplet           <lgl> FALSE, FALSE, FALSE, FALSE, FALSE, FALSE, FALS~
## $ hg19_mm10_multiplet      <lgl> FALSE, FALSE, FALSE, FALSE, FALSE, FALSE, FALS~
## $ Multiplet                <lgl> FALSE, FALSE, FALSE, TRUE, TRUE, FALSE, FALSE,~
## $ Multiplet_Type           <chr> "Singlet", "Singlet", "Singlet", "Human + Huma~
## $ Droplet_Contents         <chr> "Mouse", "Mouse", "Mouse", "Human + Human", "H~
```

Make sure the number of expected multiplets we calculated in `expect_quad_4_multi_count` and `expect_quad_2_multi_count` match the number of TRUE in `count_tbl_summary$hg19_multiplet` and `count_tbl_summary$mm10_multiplet`.

```
expect_quad_4_multi_count == count_tbl_summary %>%
                            filter(hg19_multiplet == TRUE) %>%
                            nrow()
```

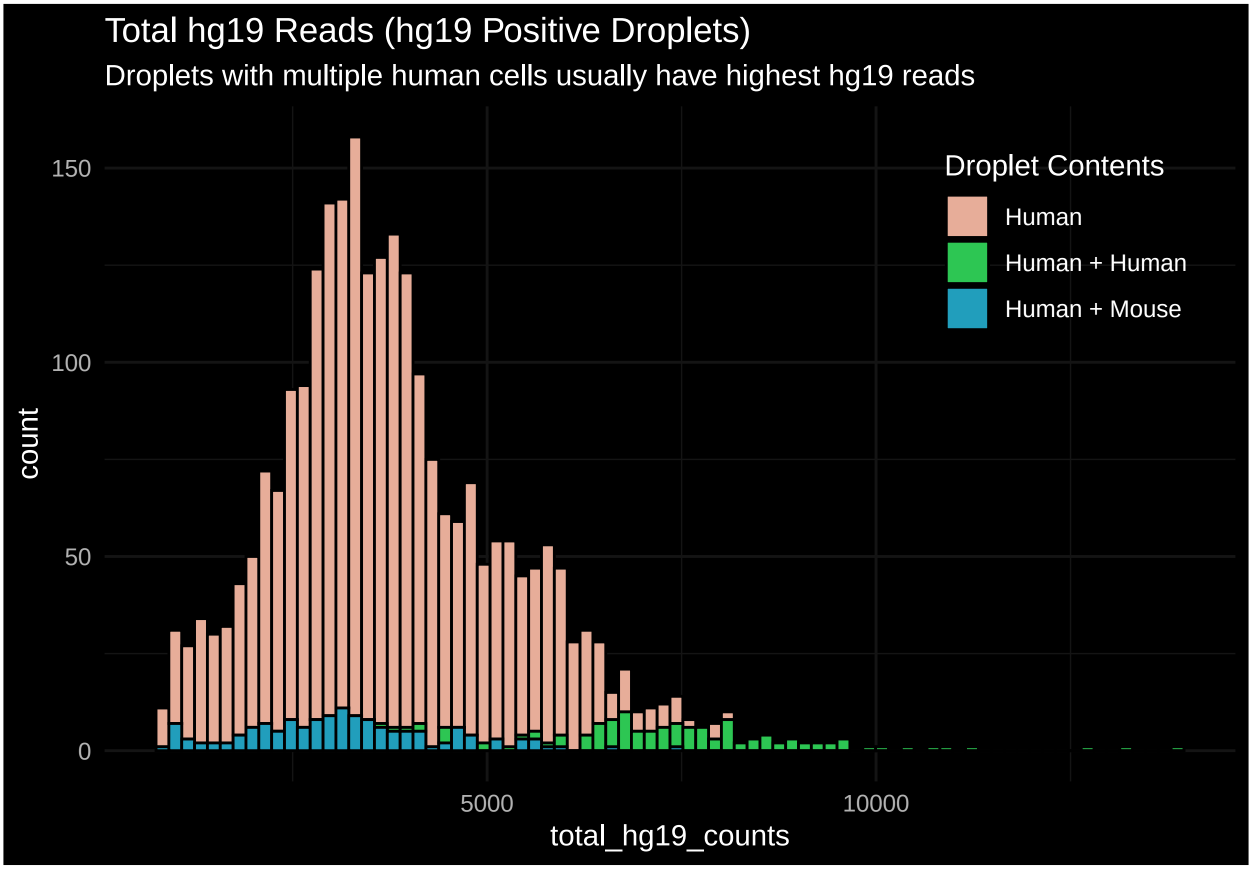
```
## [1] TRUE
```

```
expect_quad_2_multi_count == count_tbl_summary %>%
                            filter(mm10_multiplet == TRUE) %>%
                            nrow()
```

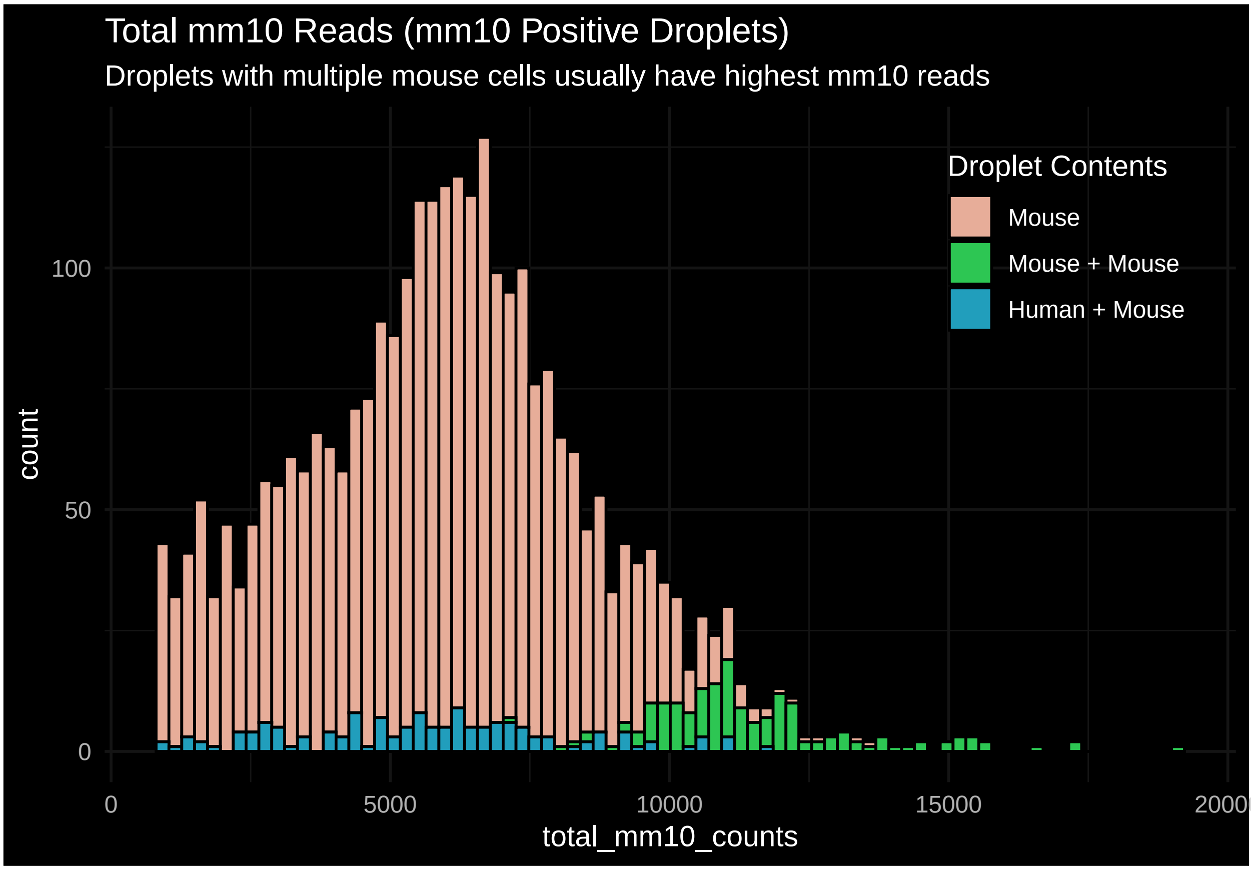
```
## [1] TRUE
```

Droplets with 2+ Humand OR 2+ Mouse cells typically have higher read counts per droplet for that reference genome. Droplets with 1+ Humand AND 1+ Mouse cells have typical read counts for each reference genome. View histogram of hg19 and mm10 reads on separate plots.

```
count_tbl_summary %>%
  filter(total_hg19_counts > 800) %>%
  ggplot(aes(x = total_hg19_counts, fill = Droplet_Contents)) +
  geom_histogram(bins = 80, color = "black") +
  scale_fill_manual(values = c("#e7ad99", "#2dc653", "#219ebc")) +
  ggdark::dark_theme_minimal() +
  labs(title    = "Total hg19 Reads (hg19 Positive Droplets)",
       subtitle = "Droplets with multiple human cells usually have highest hg19 reads",
       fill     = "Droplet Contents") +
  theme(legend.position = c(0.85, 0.8),
        legend.text     = element_text(size = rel(0.8)))
```



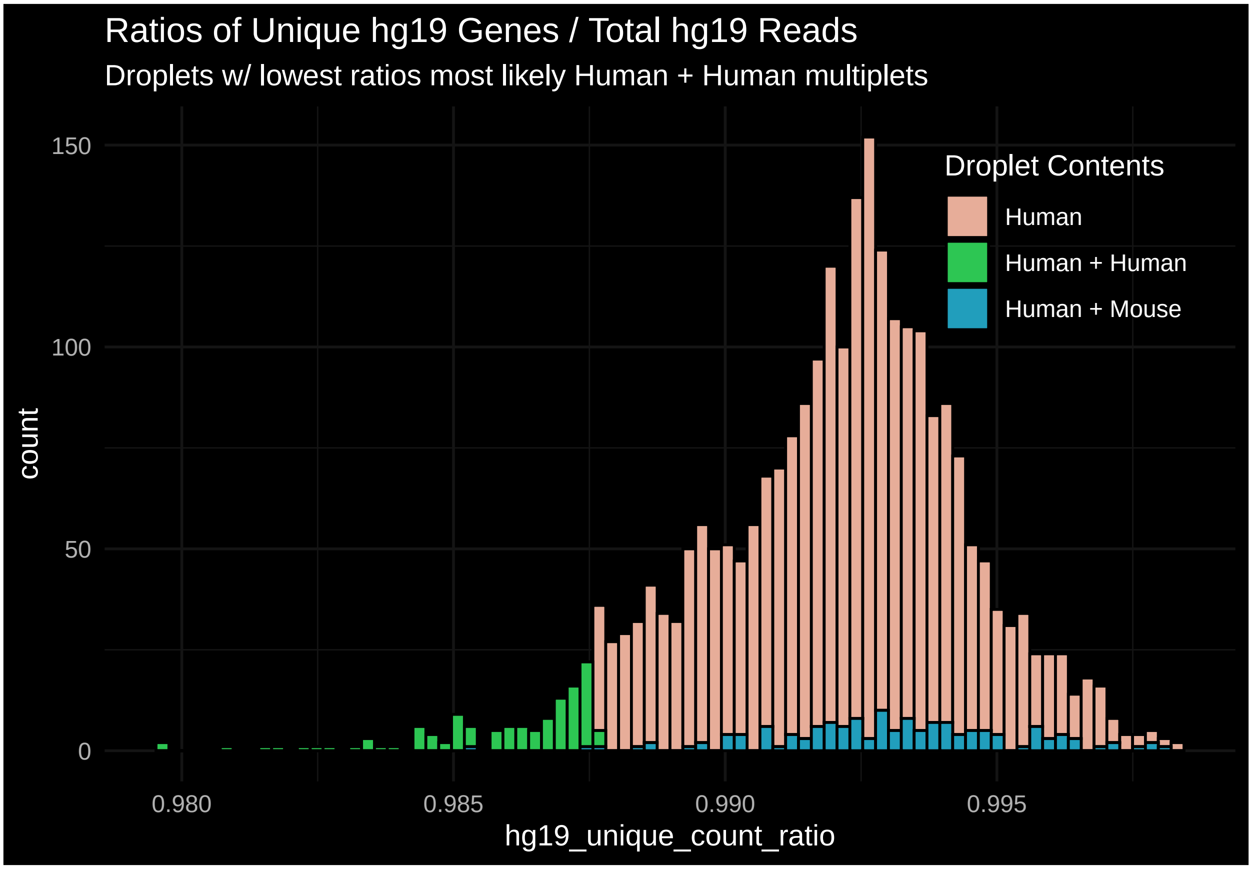
```
count_tbl_summary %>%
  filter(total_mm10_counts > 800) %>%
  ggplot(aes(x    = total_mm10_counts,
             fill = factor(Droplet_Contents, levels = c("Mouse",
                                                         "Mouse + Mouse",
                                                         "Human + Mouse")))) +
  geom_histogram(bins = 80, color = "black") +
  scale_fill_manual(values = c("#e7ad99", "#2dc653", "#219ebc")) +
  ggdark::dark_theme_minimal() +
  labs(title    = "Total mm10 Reads (mm10 Positive Droplets)",
       subtitle = "Droplets with multiple mouse cells usually have highest mm10 reads",
       fill     = "Droplet Contents") +
  theme(legend.position = c(0.85, 0.8),
        legend.text     = element_text(size = rel(0.8)))
```



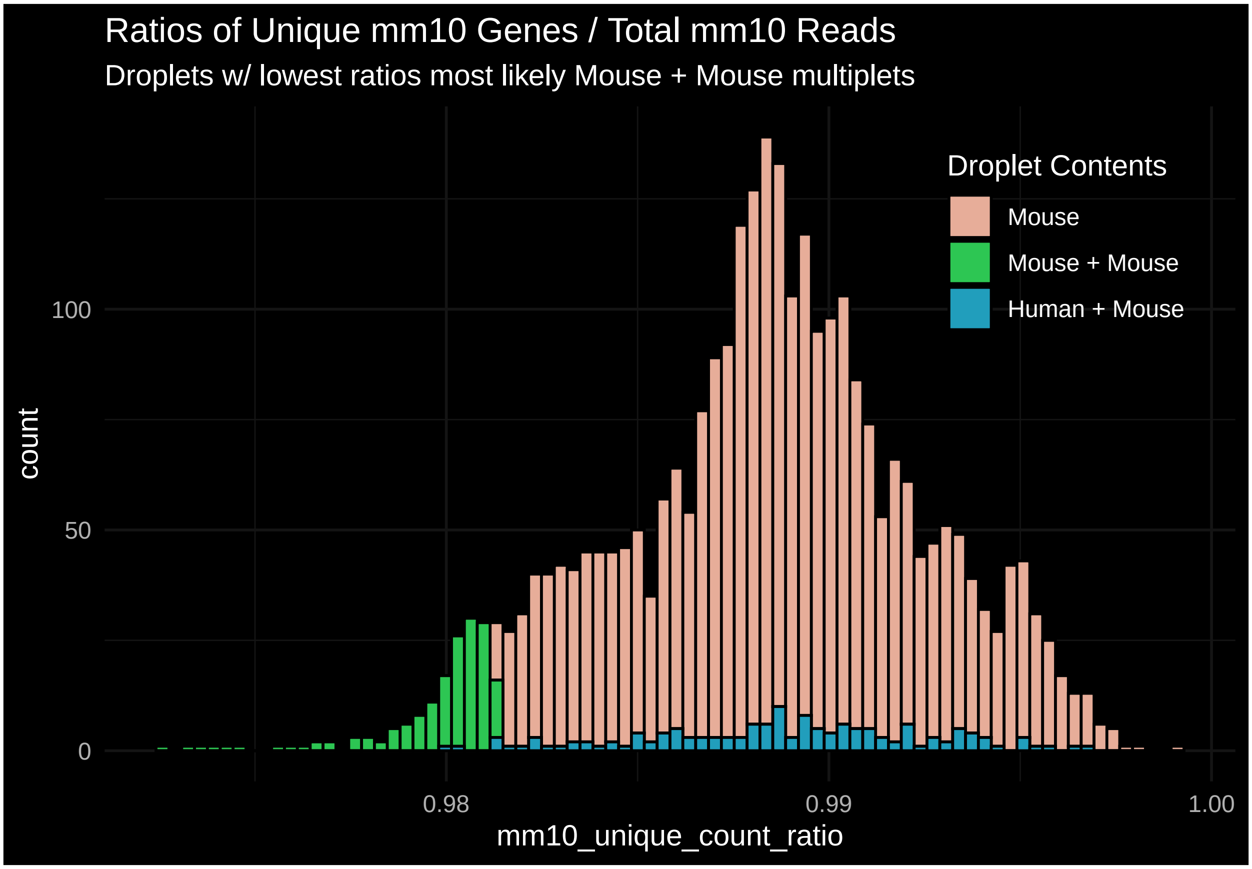
Droplets with 2+ Humand OR 2+ Mouse cells typically have the lowest unique gene count to total read count ratios for that reference genome. Droplets with 1+ Humand AND 1+ Mouse cells have typical ratios for each reference genome.

View histogram of hg19 and mm10 ratios on separate plots.

```
count_tbl_summary %>%
  filter(total_hg19_counts > 800) %>%
  ggplot(aes(x = hg19_unique_count_ratio, fill = Droplet_Contents)) +
  geom_histogram(bins = 80, color = "black") +
  scale_fill_manual(values = c("#e7ad99", "#2dc653", "#219ebc")) +
  ggdark::dark_theme_minimal() +
  labs(title    = "Ratios of Unique hg19 Genes / Total hg19 Reads",
       subtitle = "Droplets w/ lowest ratios most likely Human + Human multiplets",
       fill     = "Droplet Contents") +
  theme(legend.position = c(0.85, 0.8),
        legend.text     = element_text(size = rel(0.8)))
```

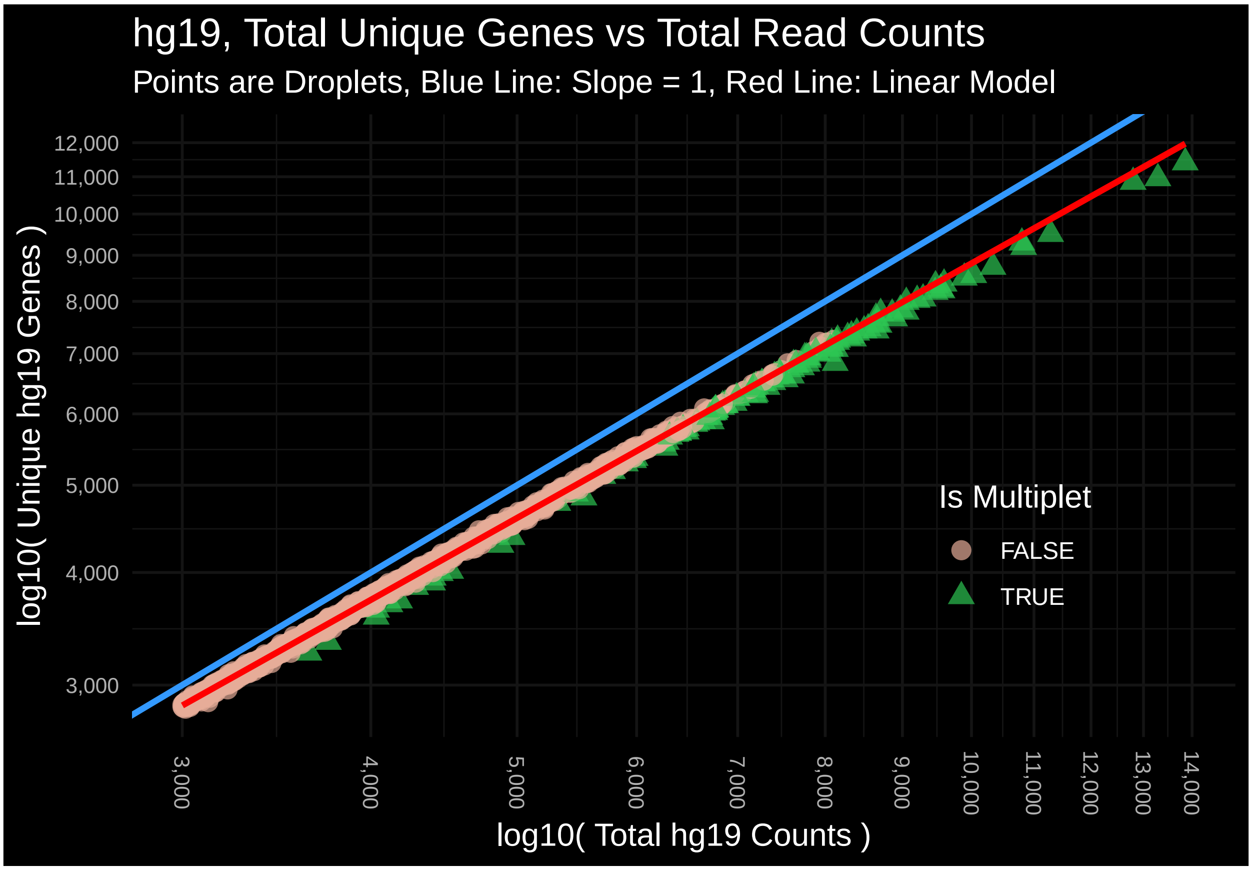


```
count_tbl_summary %>%
  filter(total_mm10_counts > 800) %>%
  ggplot(aes(x    = mm10_unique_count_ratio,
             fill = factor(Droplet_Contents, levels = c("Mouse",
                                                        "Mouse + Mouse",
                                                        "Human + Mouse")))) +
  geom_histogram(bins = 80, color = "black") +
  scale_fill_manual(values = c("#e7ad99", "#2dc653", "#219ebc")) +
  ggdark::dark_theme_minimal() +
  labs(title    = "Ratios of Unique mm10 Genes / Total mm10 Reads",
       subtitle = "Droplets w/ lowest ratios most likely Mouse + Mouse multiplets",
       fill     = "Droplet Contents") +
  theme(legend.position = c(0.85, 0.8),
        legend.text     = element_text(size = rel(0.8)))
```



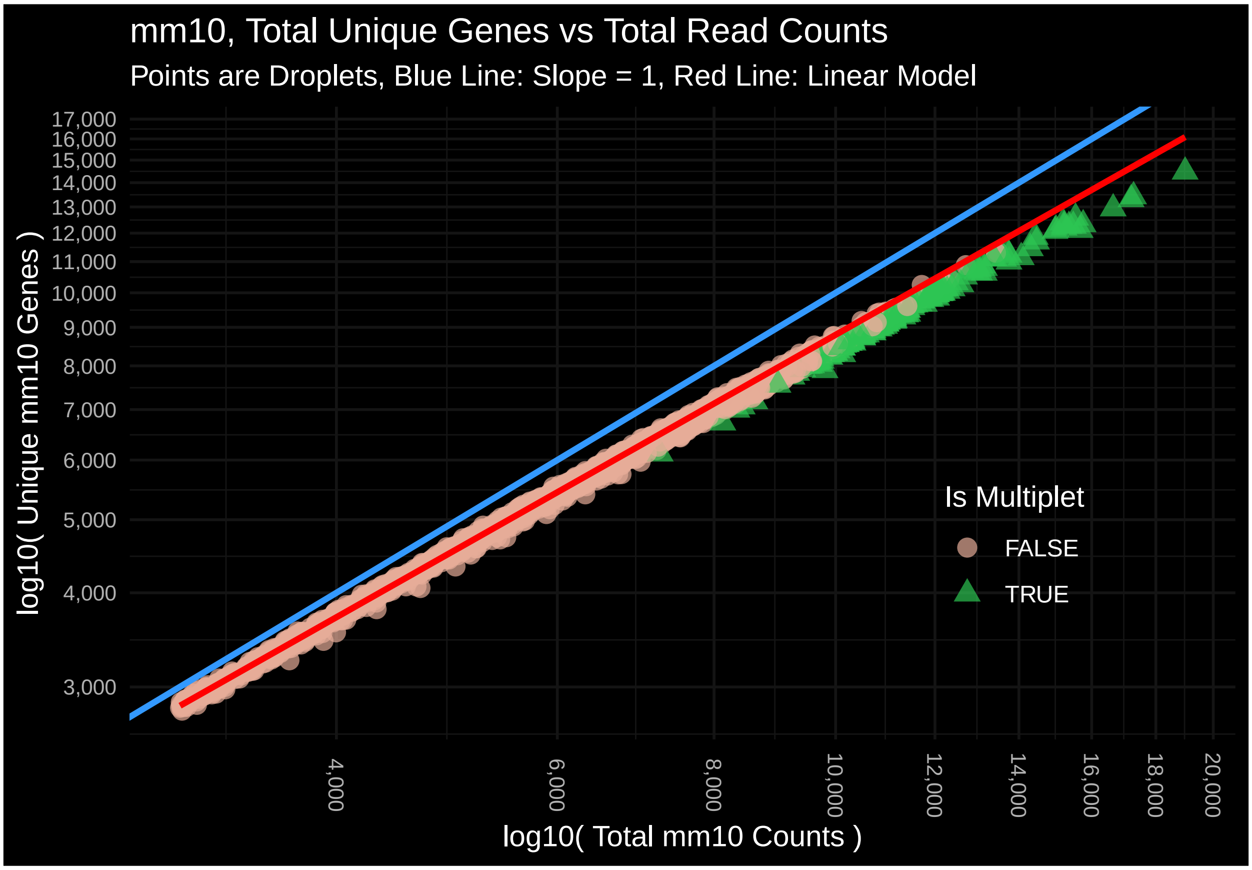
Another view of how unique gene count to total gene count ratios and total read counts are good indicators of multiplets. Fit a linear model on `total_hg19_counts` for low count droplets, and see how high count droplets deviate from that model.

```
count_tbl_summary %>%
  filter(total_hg19_counts > 3000) %>%
  ggplot(aes(x = total_hg19_counts, y = unique_hg19_genes, color = hg19_multiplet)) +
    geom_point(aes(shape = hg19_multiplet), alpha = 0.7, size = 3.2) +
    geom_smooth(data      = count_tbl_summary %>% filter(total_hg19_counts < 6000
                                                         & total_hg19_counts > 3000),
                method    = "lm",
                fullrange = TRUE,
                color     = "red",
                size      = 1.1) +
    scale_color_manual(values = c("#e7ad99", "#2dc653")) +
    scale_y_log10(breaks = scales::breaks_log(n = 12),
                  labels = scales::number_format(accuracy = 1, big.mark = ",")) +
    scale_x_log10(breaks = scales::breaks_log(n = 12),
                  labels = scales::number_format(accuracy = 1, big.mark = ",")) +
    geom_abline(slope = 1,
                color = "#3399FF",
                size  = 1.1) +
    ggdark::dark_theme_minimal() +
    theme(axis.text.x      = element_text(angle = -90, vjust = 0.5, size = rel(0.9)),
          axis.text.y      = element_text(size = rel(0.9)),
          title            = element_text(size = rel(1.1)),
          legend.position = c(0.8, 0.3)) +
    labs(title    = "hg19, Total Unique Genes vs Total Read Counts",
         subtitle = "Points are Droplets, Blue Line: Slope = 1, Red Line: Linear Model",
         x        = "log10( Total hg19 Counts )",
         y        = "log10( Unique hg19 Genes )",
         color    = "Is Multiplet",
         shape    = "Is Multiplet")
```



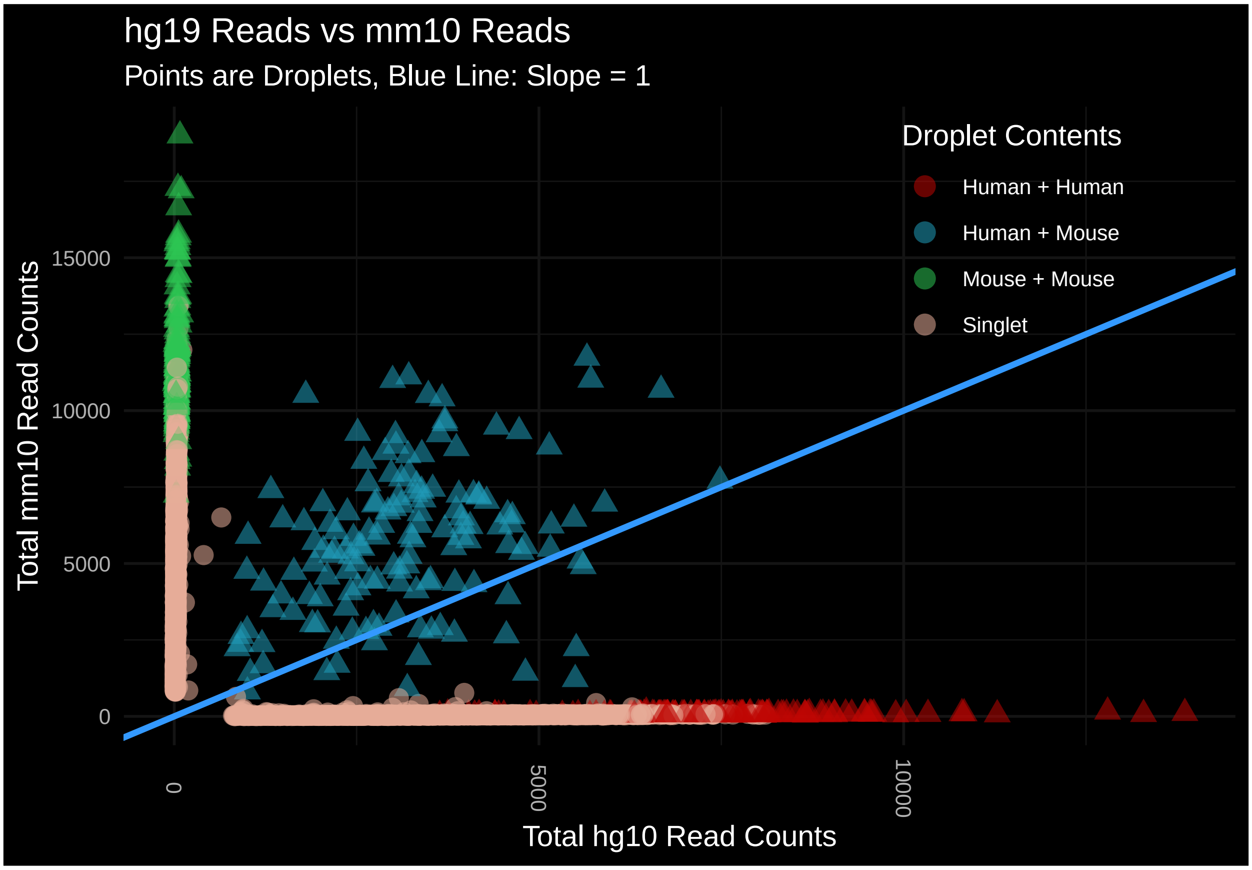
Same for `total_mm10_counts`.

```
count_tbl_summary %>%
  filter(total_mm10_counts > 3000) %>%
  ggplot(aes(x = total_mm10_counts, y = unique_mm10_genes, color = mm10_multiplet)) +
    geom_point(aes(shape = mm10_multiplet), alpha = 0.7, size = 3.2) +
    geom_smooth(data      = count_tbl_summary %>% filter(total_mm10_counts < 6000
                                                         & total_mm10_counts > 3000),
                method    = "lm",
                fullrange = TRUE,
                color     = "red",
                size      = 1.1) +
    scale_color_manual(values = c("#e7ad99", "#2dc653")) +
    scale_y_log10(breaks = scales::breaks_log(n = 12),
                  labels = scales::number_format(accuracy = 1, big.mark = ",")) +
    scale_x_log10(breaks = scales::breaks_log(n = 12),
                  labels = scales::number_format(accuracy = 1, big.mark = ",")) +
    geom_abline(slope = 1,
                color = "#3399FF",
                size  = 1.1) +
    ggdark::dark_theme_minimal() +
    theme(axis.text.x     = element_text(angle = -90, vjust = 0.5, size = rel(0.9)),
          axis.text.y     = element_text(size = rel(0.9)),
          title           = element_text(size = rel(1.0)),
          legend.position = c(0.8, 0.3)) +
    labs(title    = "mm10, Total Unique Genes vs Total Read Counts",
         subtitle = "Points are Droplets, Blue Line: Slope = 1, Red Line: Linear Model",
         x        = "log10( Total mm10 Counts )",
         y        = "log10( Unique mm10 Genes )",
         color    = "Is Multiplet",
         shape    = "Is Multiplet")
```



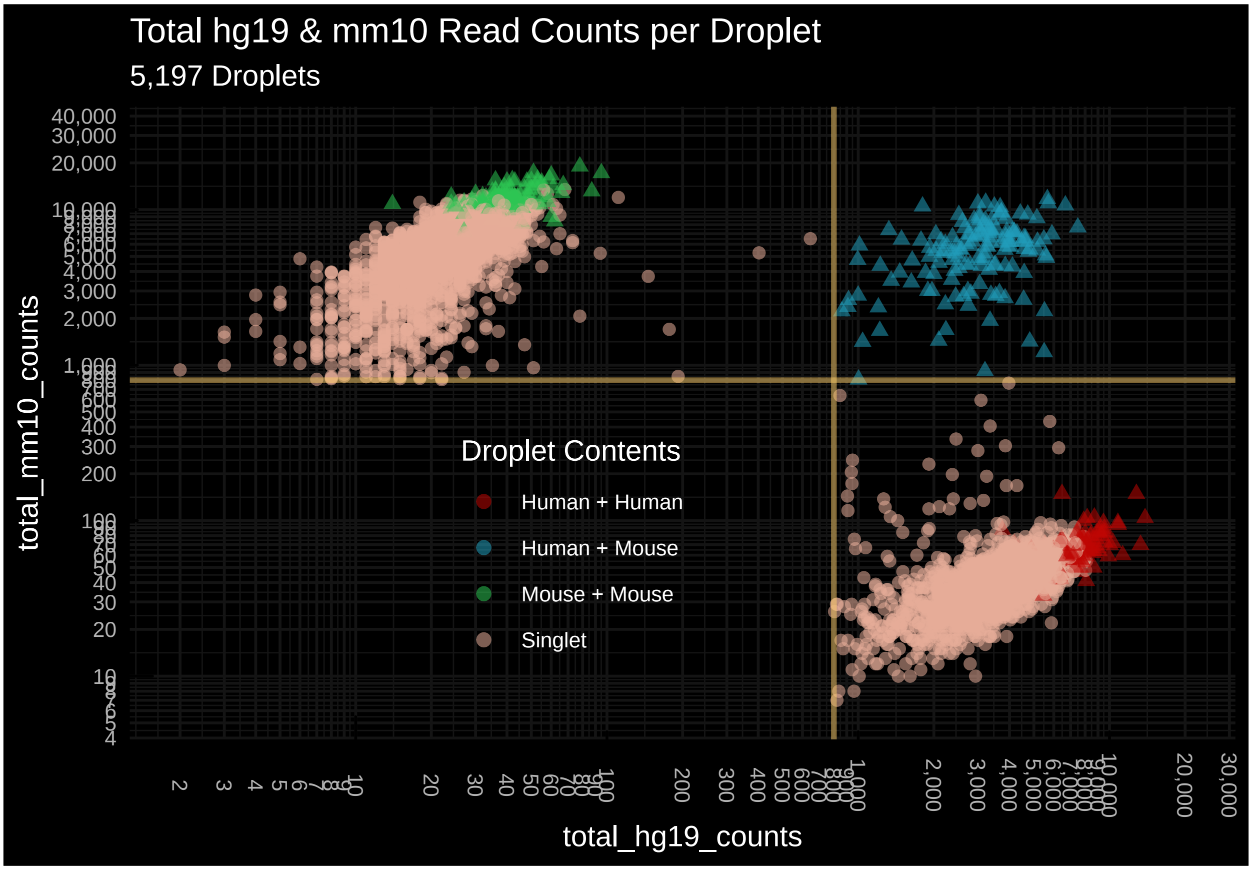
View `total_hg19_counts` vs `total_mm10_counts` without log scaled values.

```
count_tbl_summary %>%
ggplot(aes(x = total_hg19_counts, y = total_mm10_counts, color = Multiplet_Type)) +
  geom_point(aes(shape = Multiplet), alpha = 0.55, size = 3.2) +
  scale_color_manual(values = c("#bf0603", "#219ebc", "#2dc653", "#e7ad99")) +
  geom_abline(slope = 1,
              color = "#3399FF",
              size  = 1.1) +
  ggdark::dark_theme_minimal() +
  theme(axis.text.x     = element_text(angle = -90, vjust = 0.5, size = rel(0.9)),
        axis.text.y     = element_text(size = rel(0.9)),
        title           = element_text(size = rel(1.0)),
        legend.position = c(0.80, 0.80),
        legend.text     = element_text(size = rel(0.75))) +
  labs(title    = "hg19 Reads vs mm10 Reads",
       subtitle = "Points are Droplets, Blue Line: Slope = 1",
       x        = "Total hg10 Read Counts",
       y        = "Total mm10 Read Counts",
       color    = "Droplet Contents") +
  guides(shape = "none")
```



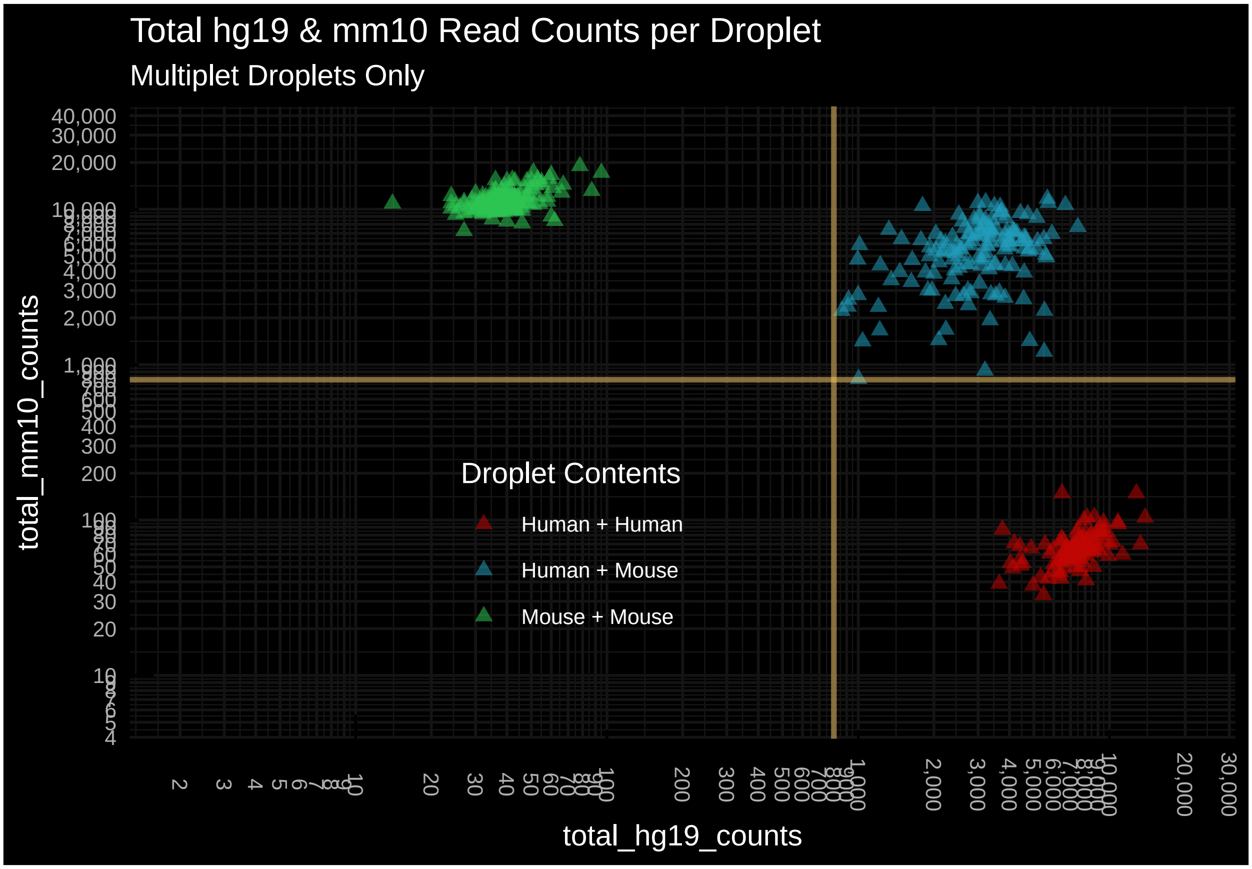
View `total_hg19_counts` vs `total_mm10_counts` with log scaled values.

```
count_tbl_summary  %>%
  ggplot(aes(x     = total_hg19_counts,
             y     = total_mm10_counts,
             color = Multiplet_Type,
             shape = Multiplet)) +
    geom_point(alpha = 0.55, size = 2) +
    scale_y_log10(breaks = scales::breaks_log(n = 38),
                  labels = scales::number_format(accuracy = 1, big.mark = ","),
                  limit  = c(6, 30000)) +
    scale_x_log10(breaks = scales::breaks_log(n = 40),
                  labels = scales::number_format(accuracy = 1, big.mark = ","),
                  limit  = c(2, 20000)) +
    scale_color_manual(values = c("#bf0603", "#219ebc", "#2dc653", "#e7ad99")) +
    geom_hline(yintercept = c(800),
               color       = "#FFCF6A",
               alpha       = 0.5,
               size        = 1) +
    geom_vline(xintercept = c(800),
               color       = "#FFCF6A",
               alpha       = 0.5,
               size        = 1) +
    ggdark::dark_theme_minimal() +
    theme(axis.text.x     = element_text(angle = -90, vjust = 0.5, size = rel(0.9)),
          axis.text.y     = element_text(size = rel(0.9)),
          title           = element_text(size = rel(1.0)),
          legend.position = c(0.4, 0.3),
          legend.text     = element_text(size = rel(0.75))) +
    annotation_logticks(scaled = FALSE) +
    labs(title    = "Total hg19 & mm10 Read Counts per Droplet",
         subtitle = "5,197 Droplets",
         color    = "Droplet Contents") +
    guides(shape = "none")
```



View `total_hg19_counts` vs `total_mm10_counts` with log scaled values, but just the multiplets.

```
count_tbl_summary %>% filter(Multiplet == TRUE)  %>%
  ggplot(aes(x = total_hg19_counts, y = total_mm10_counts)) +
    geom_point(aes(color = Multiplet_Type),
               alpha = 0.55,
               size  = 2,
               shape = 17) +
    scale_y_log10(breaks = scales::breaks_log(n = 38),
                  labels = scales::number_format(accuracy = 1, big.mark = ","),
                  limit  = c(6, 30000)) +
    scale_x_log10(breaks = scales::breaks_log(n = 40),
                  labels = scales::number_format(accuracy = 1, big.mark = ","),
                  limit  = c(2, 20000)) +
    scale_color_manual(values = c("#bf0603", "#219ebc", "#2dc653")) +
    geom_hline(yintercept = c(800),
               color      = "#FFCF6A",
               alpha      = 0.5,
               size       = 1) +
    geom_vline(xintercept = c(800),
               color      = "#FFCF6A",
               alpha      = 0.5,
               size       = 1) +
    ggdark::dark_theme_minimal() +
    theme(axis.text.x     = element_text(angle = -90, vjust = 0.5, size = rel(0.9)),
          axis.text.y     = element_text(size = rel(0.9)),
          title           = element_text(size = rel(1.0)),
          legend.position = c(0.4, 0.3),
          legend.text     = element_text(size = rel(0.75))) +
    annotation_logticks(scaled = FALSE) +
    labs(title    = "Total hg19 & mm10 Read Counts per Droplet",
         subtitle = "Multiplet Droplets Only",
         color    = "Droplet Contents") +
    guides(shape = "none")
```



Summary table of singlet and multiplet droplets identified.

```
count_tbl_summary %>%
  group_by(Droplet_Contents) %>%
  summarise(n = n()) %>%
  gt(rowname_col = "Droplet_Contents") %>%
  tab_header(title = "Droplet Contents",
             subtitle = "5413 Total Droplets") %>%
  tab_options(heading.subtitle.font.size = 12,
              heading.align = "left",
              table.border.top.color = "black",
              column_labels.border.bottom.color = "black",
              column_labels.border.bottom.width = px(3),
              column_labels.border.top.width = px(4),
              table.border.top.width = px(3))
```

Droplet Contents

5413 Total Droplets

	n
Human	2328
Human + Human	124
Human + Mouse	145
Mouse	2652
Mouse + Mouse	164

Save `count_tbl_summary` for further exploratory data analysis.

```r
data.table::fwrite(count_tbl_summary, here("data", "count_tbl_summary.csv"))
```