T-cell survival: DLI

Load data

```r
data <- tibble(fname = list.files("../data/hsct/")) %>%
  mutate(donor = startsWith(fname, "don"),
         DLI = !(startsWith(fname, "don.sh.p") | startsWith(fname, "rec.sh.p")),
         sample.id = substr(fname, 5, nchar(fname))) %>%
  mutate(sample.id = paste0("D", sample.id %>% as.factor %>% as.integer,
                            ifelse(DLI, "*", ""))) %>%
  group_by(donor, DLI, sample.id, fname) %>%
  group_modify(~read_tsv(paste0("../data/hsct/", .y$fname))) %>%
  ungroup %>%
  select(-fname)
```

Merge data

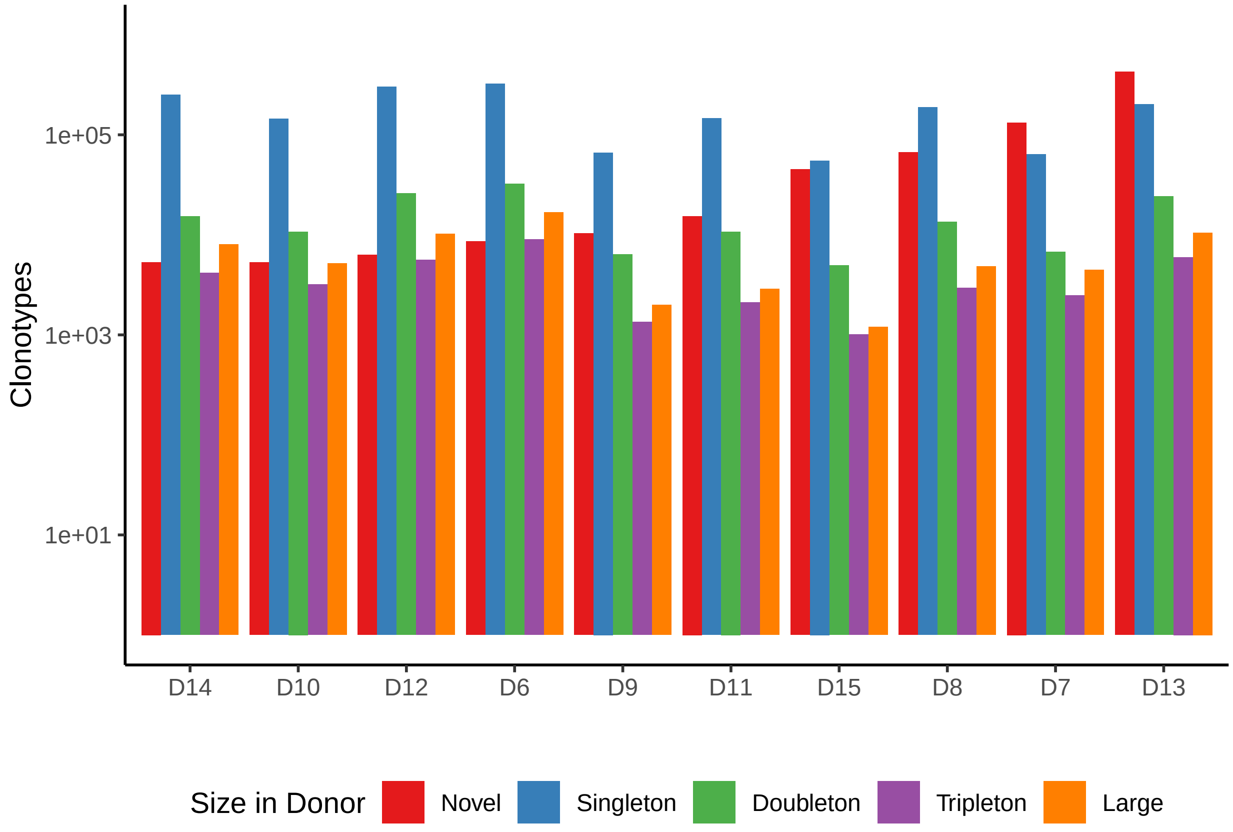
```r
data.m <- full_join(data %>%
                      filter(donor) %>%
                      mutate(count.don = count) %>%
                      select(-donor, -count),
                    data %>%
                      filter(!donor) %>%
                      mutate(count.rec = count) %>%
                      select(-donor, -count)) %>%
  mutate(quantile.don = case_when(
    is.na(count.don) ~ "Novel",
    count.don == 1 ~ "Singleton",
    count.don == 2 ~ "Doubleton",
    count.don == 3 ~ "Tripleton",
    T ~ "Large"
  ),
  quantile.rec = case_when(
    is.na(count.rec) ~ "Missing",
    count.rec == 1 ~ "Singleton",
    count.rec == 2 ~ "Doubleton",
    count.rec == 3 ~ "Tripleton",
    T ~ "Large"
  )) %>%
  mutate(quantile.don = factor(quantile.don,
                               c("Novel", "Singleton",
                                 "Doubleton", "Tripleton",
                                 "Large")),
         quantile.rec = factor(quantile.rec,
                               c("Missing", "Singleton",
                                 "Doubleton", "Tripleton",
                                 "Large")))
```

```
## Joining with `by = join_by(DLI, sample.id, cdr3nt, v)`
```

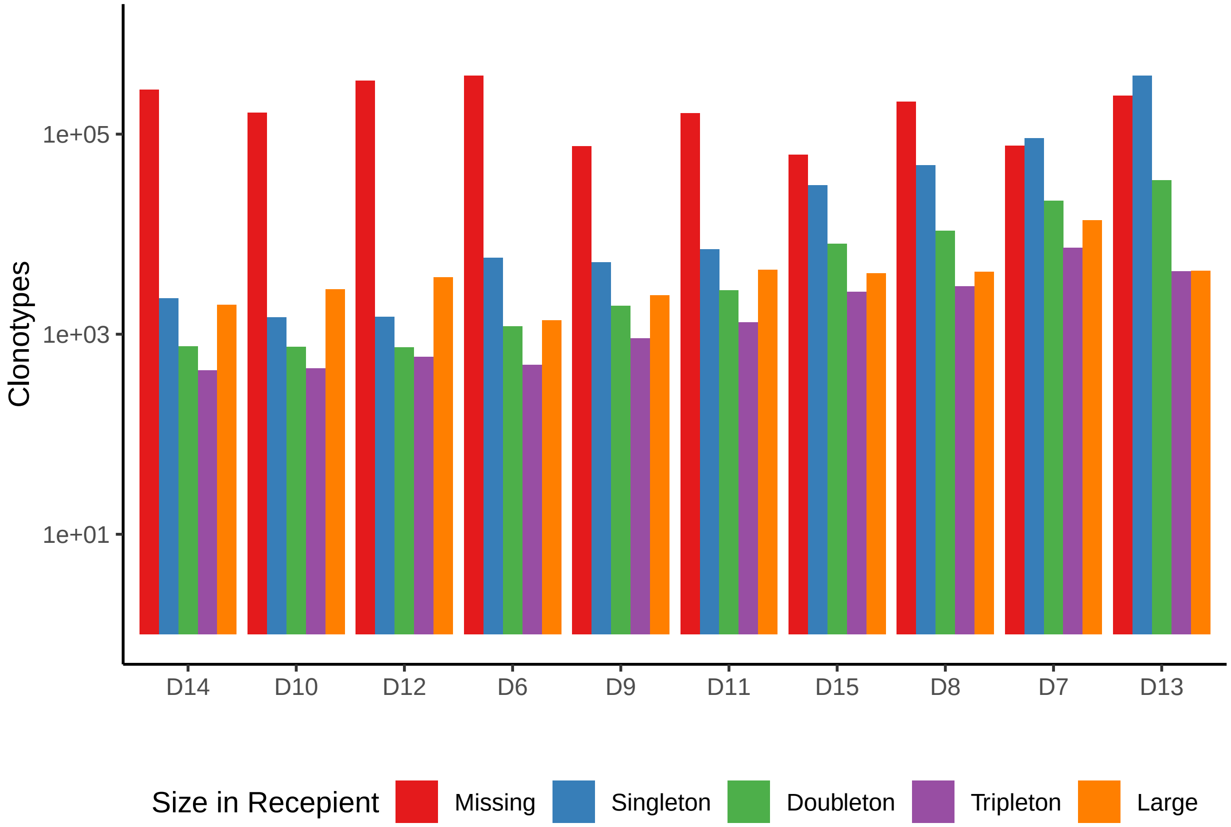
```
data.m <- data.m %>%
  left_join(data %>%
              group_by(sample.id, donor) %>%
              summarise(total = length(unique((paste(v, cdr3nt))))) %>%
              group_by(sample.id) %>%
              summarise(total.don = total[donor],
                        total.rec = total[!donor]))
```

```
## `summarise()` has grouped output by 'sample.id'. You can override using the
## `.groups` argument.
## Joining with `by = join_by(sample.id)`
```

```
p1 <- data.m %>%
  filter(!DLI) %>%
  ggplot(aes(x = sample.id %>% fct_reorder(total.rec),
             group = quantile.don,
             fill = quantile.don)) +
  geom_bar(position = "dodge") +
  scale_y_log10("Clonotypes", limits = c(1, 1e6)) +
  xlab("") +
  scale_fill_brewer("Size in Donor",
                    palette = "Set1") +
  theme_classic() +
  theme(legend.position = "bottom")
p1
```



```
p2 <- data.m %>%
  filter(!DLI) %>%
  ggplot(aes(x = sample.id %>% fct_reorder(total.rec),
             group = quantile.rec,
             fill = quantile.rec)) +
  geom_bar(position = "dodge") +
  scale_y_log10("Clonotypes", limits = c(1, 1e6)) +
  xlab("") +
  scale_fill_brewer("Size in Recepient",
                    palette = "Set1") +
  theme_classic() +
  theme(legend.position = "bottom")
p2
```

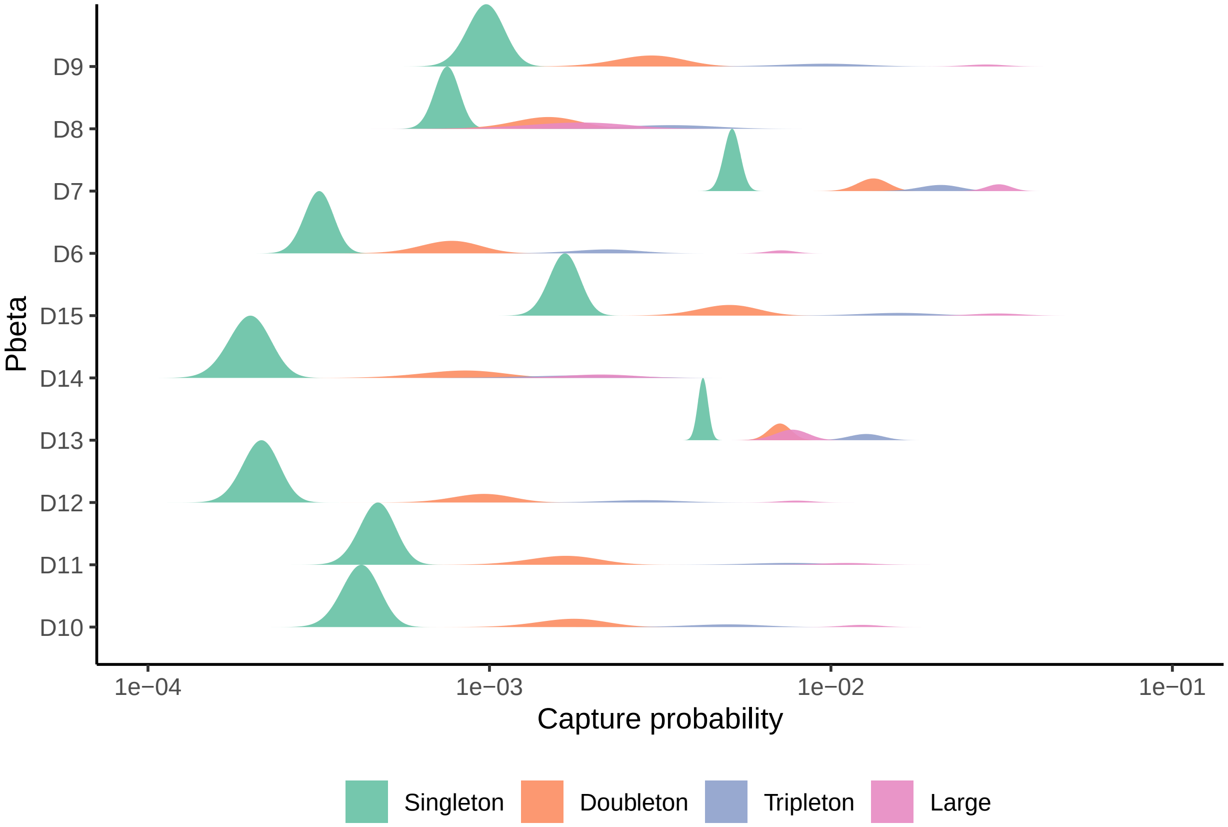


```
alpha.prior <- 1
beta.prior <- 1

data.s <- data.m %>%
  group_by(DLI,
           sample.id, quantile.don,
           total.don, total.rec) %>%
  summarise(alpha = sum(quantile.rec != "Missing") + alpha.prior,
            beta = sum(quantile.rec == "Missing") + beta.prior)
```

```
## `summarise()` has grouped output by 'DLI', 'sample.id', 'quantile.don',
## 'total.don'. You can override using the `.groups` argument.
```

```
data.b <- data.s %>%
  merge(tibble(p = c(0:1000/1000, 10^(-4000:-1000/1000)))) %>%
  group_by(sample.id, quantile.don) %>%
  mutate(Pbeta = dbeta(p, alpha, beta)) %>%
  ungroup

p3 <- data.b %>%
  filter(quantile.don != "Novel", !DLI) %>%
  group_by(sample.id) %>%
  mutate(height = Pbeta / max(Pbeta)) %>%
  ggplot(aes(x = p, y = sample.id, height = height,
             fill = quantile.don)) +
  geom_ridgeline(color = NA, alpha = 0.9) +
  scale_x_log10("Capture probability",
                limits = c(1e-4, 1e-1)) + ylab("Pbeta") +
  scale_fill_brewer("", palette = "Set2") +
  theme_classic() +
  theme(legend.position = "bottom")
p3
```

```
## Warning: Transformation introduced infinite values in continuous x-axis
```

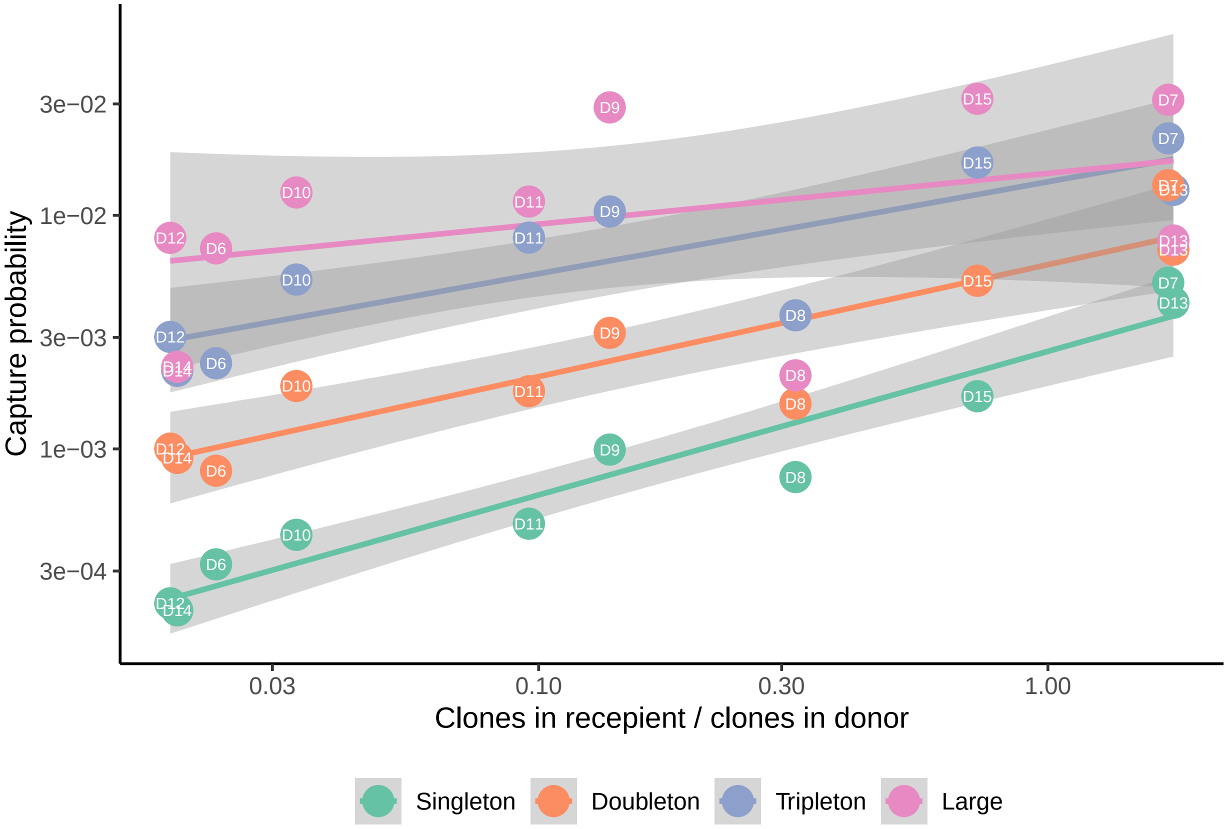


```
p4 <- data.s %>%
  filter(quantile.don != "Novel", !DLI) %>%
  ggplot(aes(x = total.rec / total.don,
             y = alpha / (alpha + beta),
             group = quantile.don,
             color = quantile.don)) +
  geom_smooth(method = "lm", aes(), size = 1) +
  geom_point(size = 5) +
  geom_text(aes(label = sample.id), size = 2, color = "white") +
  scale_x_log10("Clones in recepient / clones in donor") +
  scale_y_log10("Capture probability") +
  scale_color_brewer("", palette = "Set2") +
  theme_classic() +
  theme(legend.position = "bottom")
```

```
## Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use `linewidth` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

```
p4
```

```
## `geom_smooth()` using formula = 'y ~ x'
```

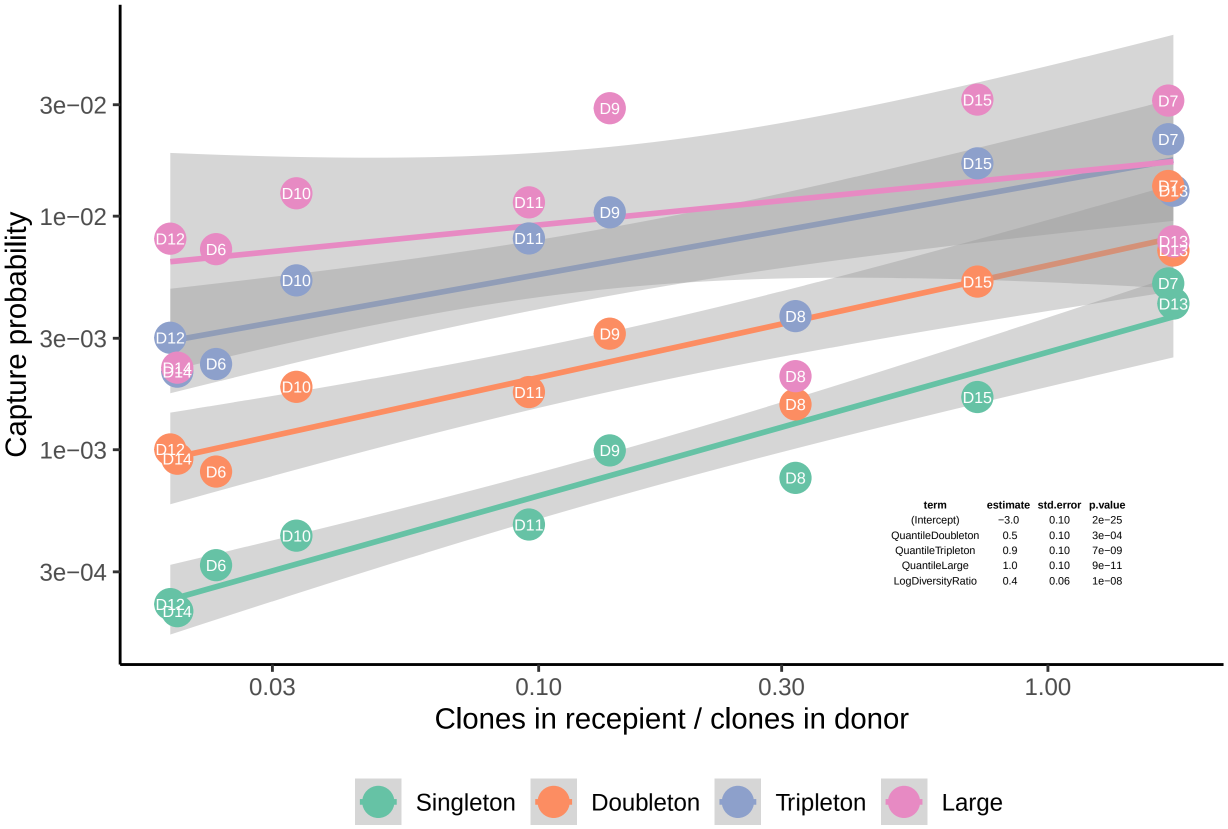


```
mdl <- lm(LogRecaptureProb ~ Quantile + LogDiversityRatio,
          data.s %>%
            filter(quantile.don != "Novel", !DLI) %>%
            mutate(LogRecaptureProb = log10(alpha / (alpha + beta)),
                   LogDiversityRatio = log10(total.rec / total.don),
                   Quantile = quantile.don))
summary(mdl)
```

```
##
## Call:
## lm(formula = LogRecaptureProb ~ Quantile + LogDiversityRatio,
##     data = data.s %>% filter(quantile.don != "Novel", !DLI) %>%
##         mutate(LogRecaptureProb = log10(alpha/(alpha + beta)),
##             LogDiversityRatio = log10(total.rec/total.don), Quantile = quantile.don))
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.83273 -0.14073  0.03197  0.17100  0.47008
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)        -2.74396    0.09865 -27.815  < 2e-16 ***
## QuantileDoubleton  0.48735    0.12092   4.030 0.000286 ***
## QuantileTripleton  0.91584    0.12092   7.574 7.05e-09 ***
## QuantileLarge      1.10293    0.12092   9.121 8.89e-11 ***
## LogDiversityRatio  0.42638    0.05739   7.430 1.07e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2704 on 35 degrees of freedom
## Multiple R-squared:  0.8148, Adjusted R-squared:  0.7937
## F-statistic: 38.51 on 4 and 35 DF,  p-value: 2.319e-12
```

```
tbl <- mdl %>%
  tidy %>%
  mutate(estimate = signif(estimate, 1),
         std.error = signif(std.error, 1),
         p.value = signif(p.value, 1)) %>%
  select(term, estimate, std.error, p.value)

p4 +
  annotate(geom = "text", label = summary(mdl)$r.squared) +
  annotation_custom(tableGrob(tbl,
                              rows = NULL,
                              theme = ttheme_minimal(base_size=4,
                                                     padding = unit(c(1, 1),
                                                                    "mm"))),
                    xmin=-0.5, ymin = -5.7) -> p5
p5
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



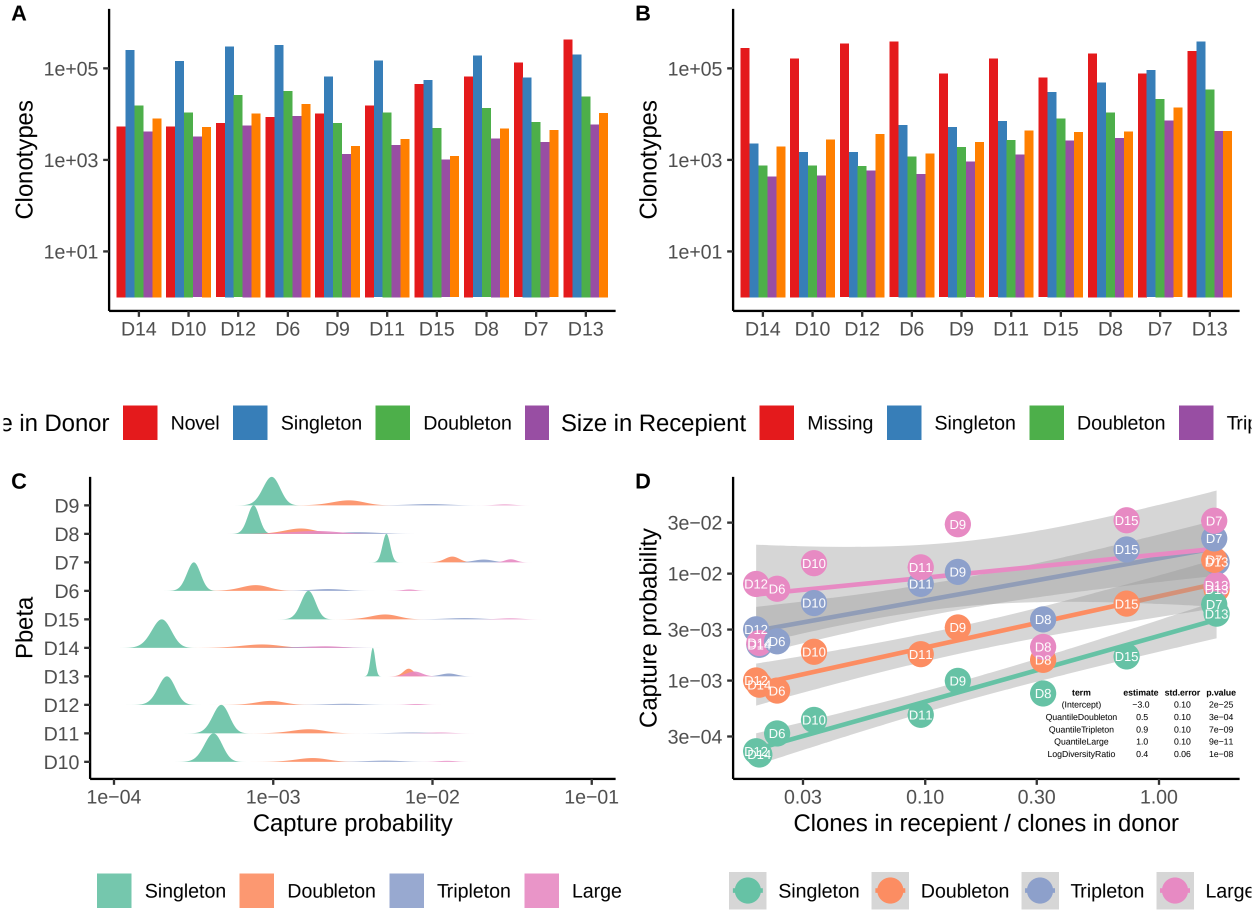
term	estimate	std.error	p.value
(Intercept)	−3.0	0.10	2e−25
QuantileDoubleton	0.5	0.10	3e−04
QuantileTripleton	0.9	0.10	7e−09
QuantileLarge	1.0	0.10	9e−11
LogDiversityRatio	0.4	0.06	1e−08

```r
plot_grid(p1, p2, p3, p5,
          labels = c("A", "B", "C", "D"),
          label_size = 10) -> fig2
```

```
## Warning: Transformation introduced infinite values in continuous x-axis
```

```
## `geom_smooth()` using formula = 'y ~ x'
```

```r
fig2
```



term	estimate	std.error	p.value
(Intercept)	−3.0	0.10	2e−25
QuantileDoubleton	0.5	0.10	3e−04
QuantileTripleton	0.9	0.10	7e−09
QuantileLarge	1.0	0.10	9e−11
LogDiversityRatio	0.4	0.06	1e−08

```
ggsave("fig2.pdf", fig2)
```

```
## Saving 8 x 6 in image
```