

T-cell survival: DLI

Load data

```
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)
```

```
data %>%
  filter(donor) %>%
  select(sample.id, DLI) %>%
  unique
```

```
## # A tibble: 19 x 2
##   sample.id DLI
##   <chr>     <lgl>
## 1 D10      FALSE
## 2 D11      FALSE
## 3 D12      FALSE
## 4 D13      FALSE
## 5 D14      FALSE
## 6 D15      FALSE
## 7 D6       FALSE
## 8 D7       FALSE
## 9 D8       FALSE
## 10 D9      FALSE
## 11 D1*     TRUE
## 12 D16*    TRUE
## 13 D17*    TRUE
## 14 D18*    TRUE
## 15 D19*    TRUE
## 16 D2*     TRUE
## 17 D3*     TRUE
## 18 D4*     TRUE
## 19 D5*     TRUE
```

Merge data

```
data.m <- full_join(data %>%
  filter(donor) %>%
  mutate(count.don = count) %>%
  select(-donor, -count),
  data %>%
```

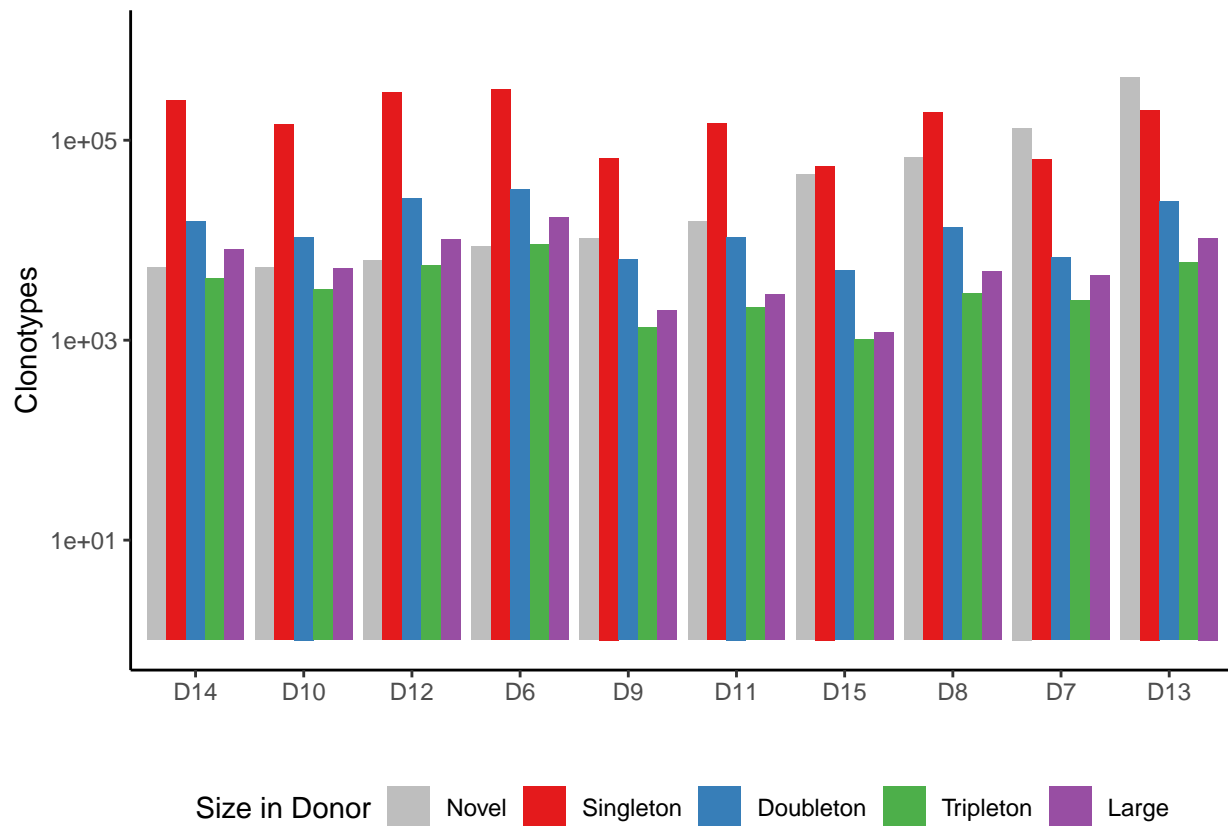
```

        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_manual("Size in Donor",
                    values = c("grey", "#e41a1c", "#377eb8",
                              "#4daf4a", "#984ea3")) +
  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_manual("Size in Receptient",
                    values = c("grey", "#e41a1c", "#377eb8",
                              "#4daf4a", "#984ea3")) +
  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)

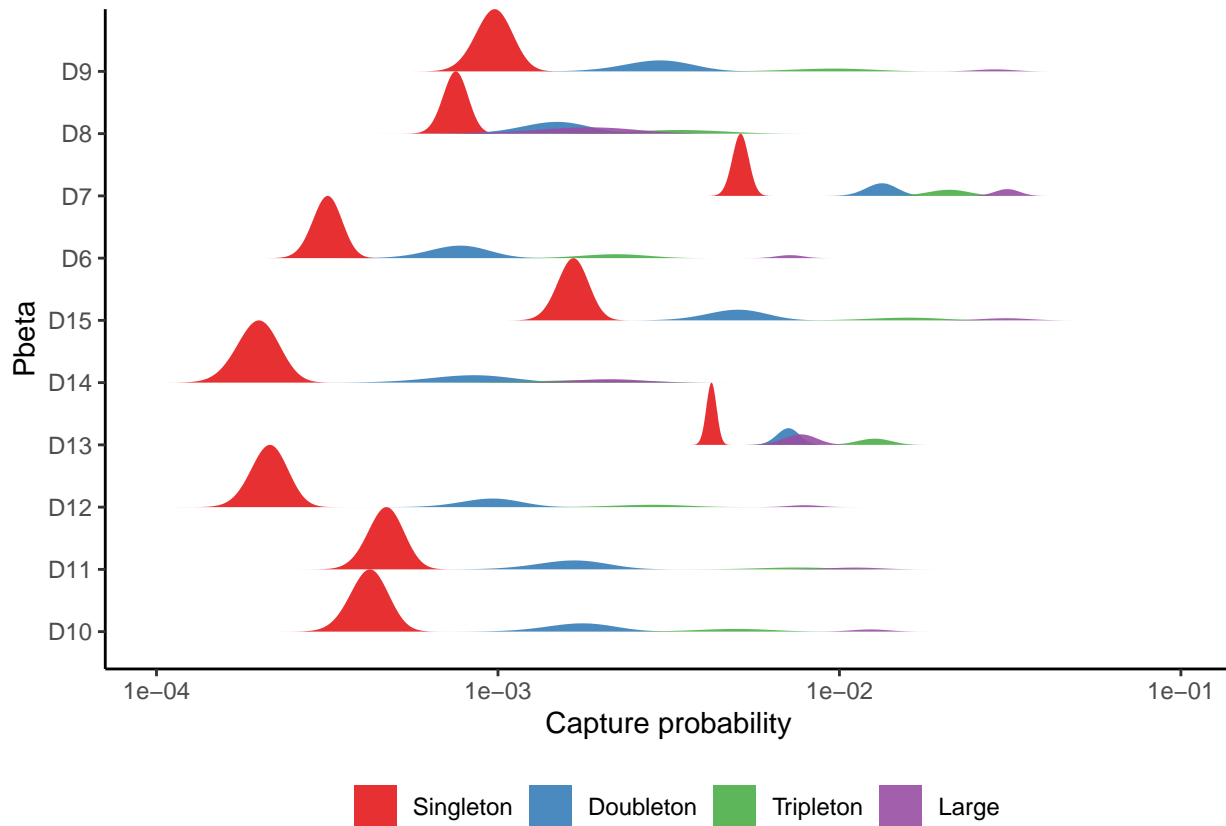
## `summarise()` has grouped output by 'DLI', 'sample.id', 'quantile.don',
## 'total.don'. You can override using the `.groups` argument.

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 = "Set1") +
theme_classic() +
theme(legend.position = "bottom")
p3
```

Warning: Transformation introduced infinite values in continuous x-axis

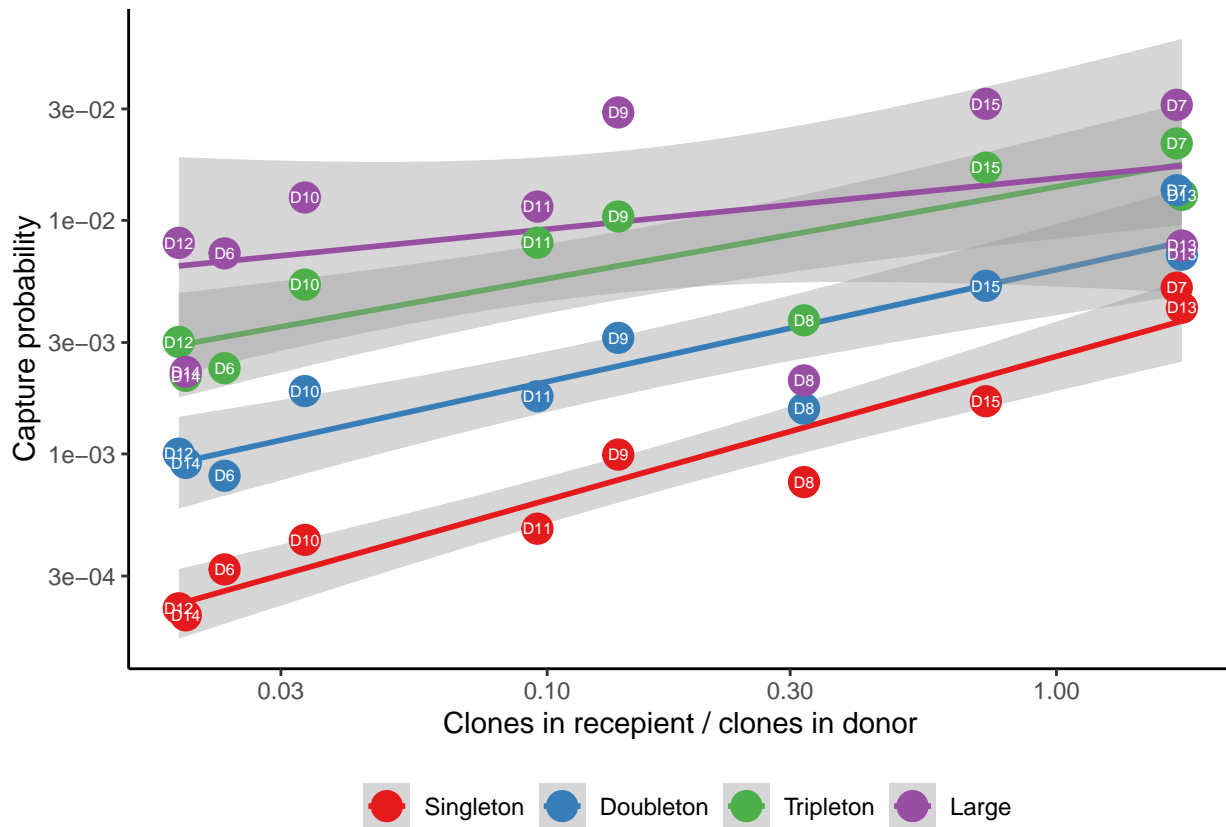


```
p4 <- data.s %>%
  filter(quantile.don != "Novel", !DLI) %>%
  ggplot(aes(x = total.rec / total.don,
             y = alpha / (alpha + beta),
             group = quantile.don,
             color = quantile.don)) +
  geom_smooth(method = "lm", aes(), size = 1) +
  geom_point(size = 5) +
  geom_text(aes(label = sample.id), size = 2, color = "white") +
  scale_x_log10("Clones in receipient / clones in donor") +
  scale_y_log10("Capture probability") +
  scale_color_brewer("", palette = "Set1") +
  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 ***
## QuantileTriplet    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, 2),
         std.error = signif(std.error, 1),
         p.value = format(signif(p.value, digits = 1), scientific = T)) %>%
  select(term, estimate, std.error, p.value)

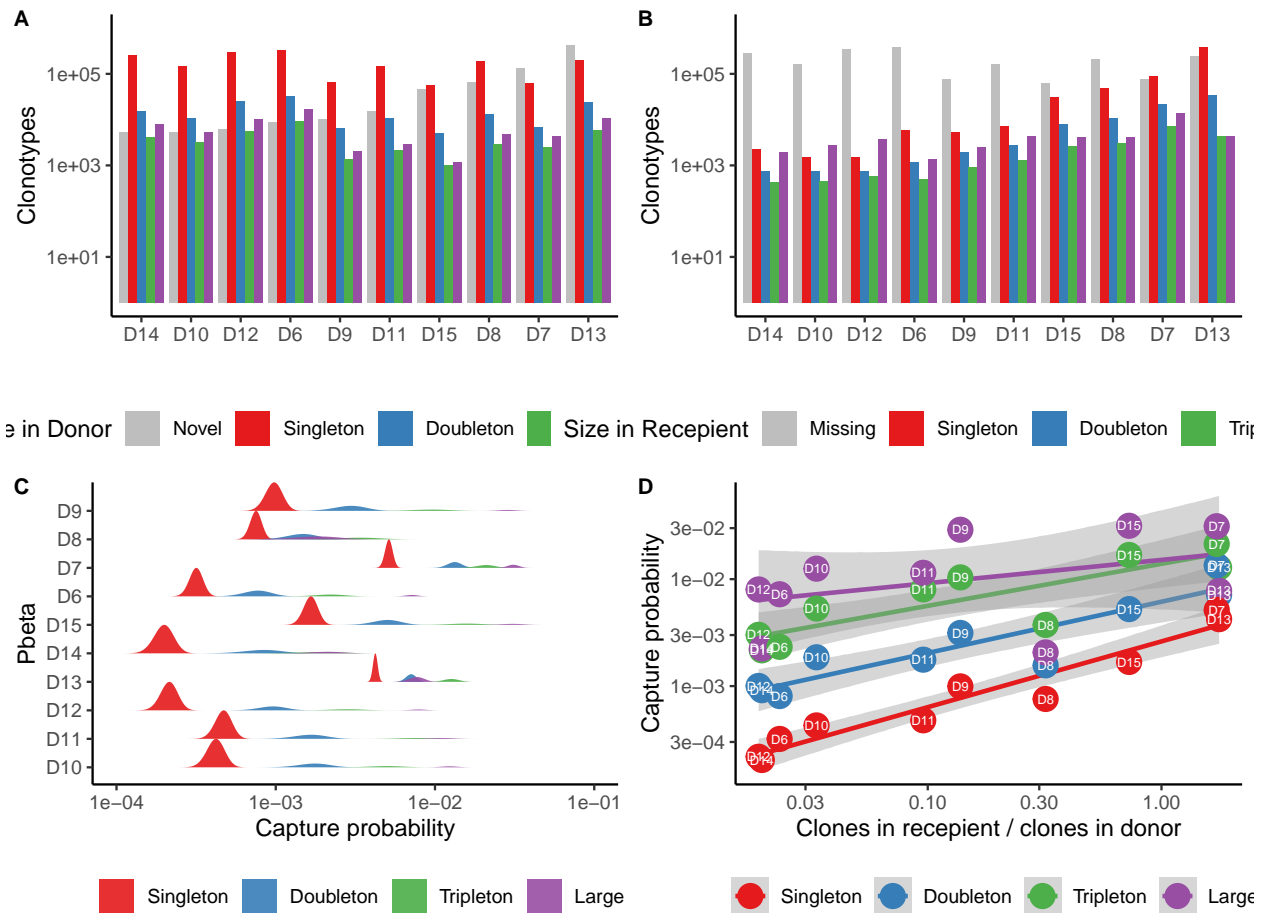
knitr::kable(tbl)
```

term	estimate	std.error	p.value
(Intercept)	-2.70	0.10	2e-25
QuantileDoubleton	0.49	0.10	3e-04
QuantileTripletton	0.92	0.10	7e-09
QuantileLarge	1.10	0.10	9e-11
LogDiversityRatio	0.43	0.06	1e-08

```
# p4 +
#   annotate(geom = "text", x = 0.1, y = 0.5,
#             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
```

```
plot_grid(p1, p2, p3, p4,
          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'
fig2
```



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

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

```
Check for difference between DLI and non-DLI using our model
```

```
mdl2 <- lm(LogRecaptureProb ~ Quantile + LogDiversityRatio + DLI,
  data.s %>%
    filter(quantile.don != "Novel") %>%
    mutate(LogRecaptureProb = log10(alpha / (alpha + beta)),
           LogDiversityRatio = log10(total.rec / total.don),
           Quantile = quantile.don))
summary(mdl2)
```

```
##
```

```
## Call:
```

```
## lm(formula = LogRecaptureProb ~ Quantile + LogDiversityRatio +
```

```
## DLI, data = data.s %>% filter(quantile.don != "Novel") %>%
```

```
## mutate(LogRecaptureProb = log10(alpha/(alpha + beta)), LogDiversityRatio = log10(total.rec/total
```

```
## Quantile = quantile.don))
```

```
##
```

```
## Residuals:
```

```
##      Min       1Q   Median       3Q      Max
```

```
## -0.78600 -0.16310 -0.01385  0.20876  0.74752
```

```
##
```

```
## Coefficients:
```



```
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)    -2.70921    0.09005 -30.085 < 2e-16 ***
## QuantileDoubleton  0.41628    0.09676   4.302 5.38e-05 ***
## QuantileTripleton  0.74876    0.09676   7.738 5.54e-11 ***
## QuantileLarge     0.99093    0.09676  10.241 1.47e-15 ***
## LogDiversityRatio  0.36482    0.05684   6.419 1.41e-08 ***
## DLITRUE          0.27749    0.07264   3.820 0.000286 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2982 on 70 degrees of freedom
## Multiple R-squared:  0.7401, Adjusted R-squared:  0.7215
## F-statistic: 39.87 on 5 and 70 DF,  p-value: < 2.2e-16
```

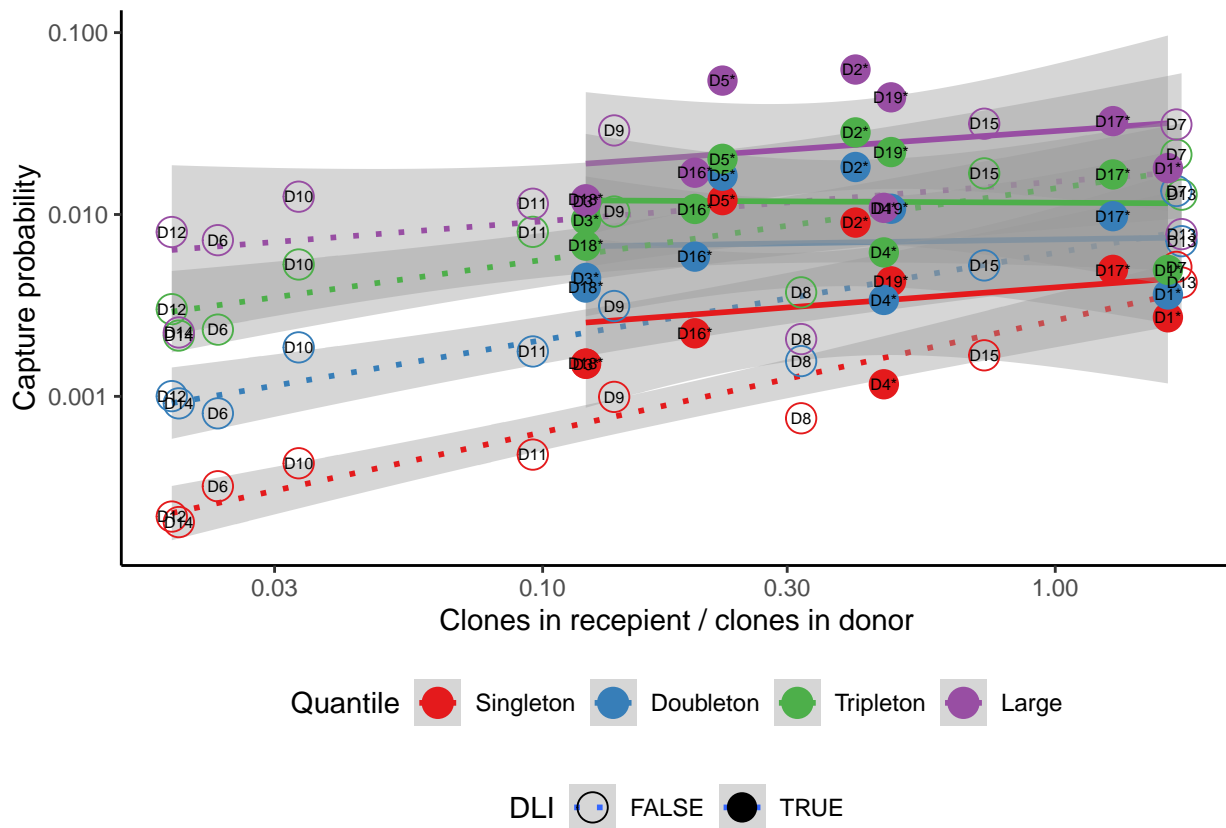
```
tbl2 <- mdl2 %>%
  tidy %>%
  mutate(estimate = signif(estimate, 1),
         std.error = signif(std.error, 1),
         p.value = signif(p.value, 1)) %>%
  select(term, estimate, std.error, p.value)
```

```
knitr::kable(tbl2)
```

term	estimate	std.error	p.value
(Intercept)	-3.0	0.09	0e+00
QuantileDoubleton	0.4	0.10	5e-05
QuantileTripleton	0.7	0.10	0e+00
QuantileLarge	1.0	0.10	0e+00
LogDiversityRatio	0.4	0.06	0e+00
DLITRUE	0.3	0.07	3e-04

```
p5 <- data.s %>%
  filter(quantile.don != "Novel") %>%
  ggplot(aes(x = total.rec / total.don,
             y = alpha / (alpha + beta),
             group = quantile.don,
             color = quantile.don,
             shape = DLI,
             linetype = DLI)) +
  geom_smooth(method = "lm",
             aes(group = paste(quantile.don, DLI)), size = 1) +
  geom_point(size = 5) +
  geom_text(aes(label = sample.id), size = 2, color = "black") +
  scale_x_log10("Clones in receipient / clones in donor") +
  scale_y_log10("Capture probability") +
  scale_linetype_manual("DLI", values = c("dotted", "solid")) +
  scale_color_brewer("Quantile", palette = "Set1") +
  scale_shape_manual("DLI", values = c(1, 16)) +
  theme_classic() +
  theme(legend.position = "bottom", legend.box="vertical")
p5
```

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



Conventional way - correlation of log counts

```
data.m %>%
  filter(!is.na(count.don), !is.na(count.rec)) %>%
  group_by(DLI, sample.id) %>%
  group_modify(~cor.test(log10(.x$count.don),
                          log10(.x$count.rec)) %>% tidy) -> data.cor

knitr::kable(data.cor)
```

DLI	sample.id	estimate	statistic	p.value	parameter	onf.low	conf.high	method	alternative
FALSE	D10	0.1645500	2.0969459	0.0375910	158	0.0096373	0.3117494	Pearson's product-moment correlation	two.sided
FALSE	D11	0.1210482	1.4063373	0.1619561	133	-0.0489092	0.2841930	Pearson's product-moment correlation	two.sided
FALSE	D12	0.0895622	1.2230936	0.2228501	185	-0.0546331	0.2300984	Pearson's product-moment correlation	two.sided
FALSE	D13	0.0812194	2.7944784	0.0052828	1176	0.0242160	0.1376965	Pearson's product-moment correlation	two.sided
FALSE	D14	0.2418720	2.3116664	0.0231869	86	0.0341603	0.4295538	Pearson's product-moment correlation	two.sided

DLI	sample.id	estimate	statistic	p.value	parameter	onf.low	onf.high	method	alternative
FALSED15		0.3437868	4.7452182	0.0000044	168	0.2038191	0.4699819	Pearson's product-moment correlation	two.sided
FALSED6		0.2558083	4.3157016	0.0000225	266	0.1402871	0.3644584	Pearson's product-moment correlation	two.sided
FALSED7		0.0697681	1.7202607	0.0858964	605	- 0.0098678	0.1485246	Pearson's product-moment correlation	two.sided
FALSED8		0.1663787	2.2574605	0.0251875	179	0.0210314	0.3048388	Pearson's product-moment correlation	two.sided
FALSED9		0.0620527	0.7665139	0.4445594	152	- 0.0970606	0.2180731	Pearson's product-moment correlation	two.sided
TRUED1*		0.3238989	12.1427352	0.0000000	1258	0.2735669	0.3724618	Pearson's product-moment correlation	two.sided
TRUED16*		0.0911390	1.0712129	0.2859572	137	- 0.0765230	0.2537886	Pearson's product-moment correlation	two.sided
TRUED17*		- 0.1002565	- 1.1879919	0.2368625	139	- 0.2612389	0.0661520	Pearson's product-moment correlation	two.sided
TRUED18*		0.0512639	0.4533463	0.6515581	78	- 0.1703720	0.2679626	Pearson's product-moment correlation	two.sided
TRUED19*		0.3211900	5.0874072	0.0000008	225	0.1993138	0.4332811	Pearson's product-moment correlation	two.sided
TRUED2*		0.0276139	0.5654588	0.5720644	419	- 0.0681383	0.1228621	Pearson's product-moment correlation	two.sided
TRUED3*		0.2700681	2.4772273	0.0154036	78	0.0535274	0.4623498	Pearson's product-moment correlation	two.sided
TRUED4*		0.1208444	1.5156079	0.1316552	155	- 0.0364843	0.2723277	Pearson's product-moment correlation	two.sided
TRUED5*		0.0938781	4.5172909	0.0000066	2295	0.0531837	0.1342613	Pearson's product-moment correlation	two.sided

```
p6 <- data.m %>%
  filter(!is.na(count.don), !is.na(count.rec)) %>%
  ggplot(aes(x = count.don, y = count.rec)) +
  #geom_point(alpha = 0.5, aes(color = quantile.don)) +
  #geom_abline(slope = 1, intercept = 0, linetype = "dashed") +
  geom_point(aes(color = DLI)) +
  geom_text(data = data.cor %>%
    mutate(text = paste0("R2=", round(estimate^2, 3),
```

```

                                "\nP=", signif(p.value,1))) %>%
  select(sample.id, text),
  aes(label = text, x = 1000, y = 100000)) +
  geom_smooth(method = "lm", color = "black", linetype = "dotted") +
  scale_x_log10("Count in donor", limits = c(1, 1e6)) +
  scale_y_log10("Count in recipient", limits = c(1, 1e6),
               breaks = c(10, 1000, 100000)) +
  scale_color_brewer(palette = "Set2", guide = F) +
  facet_wrap(. ~ sample.id) +
  theme_classic()

```

```
## Adding missing grouping variables: `DLI`
```

```
p6
```

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

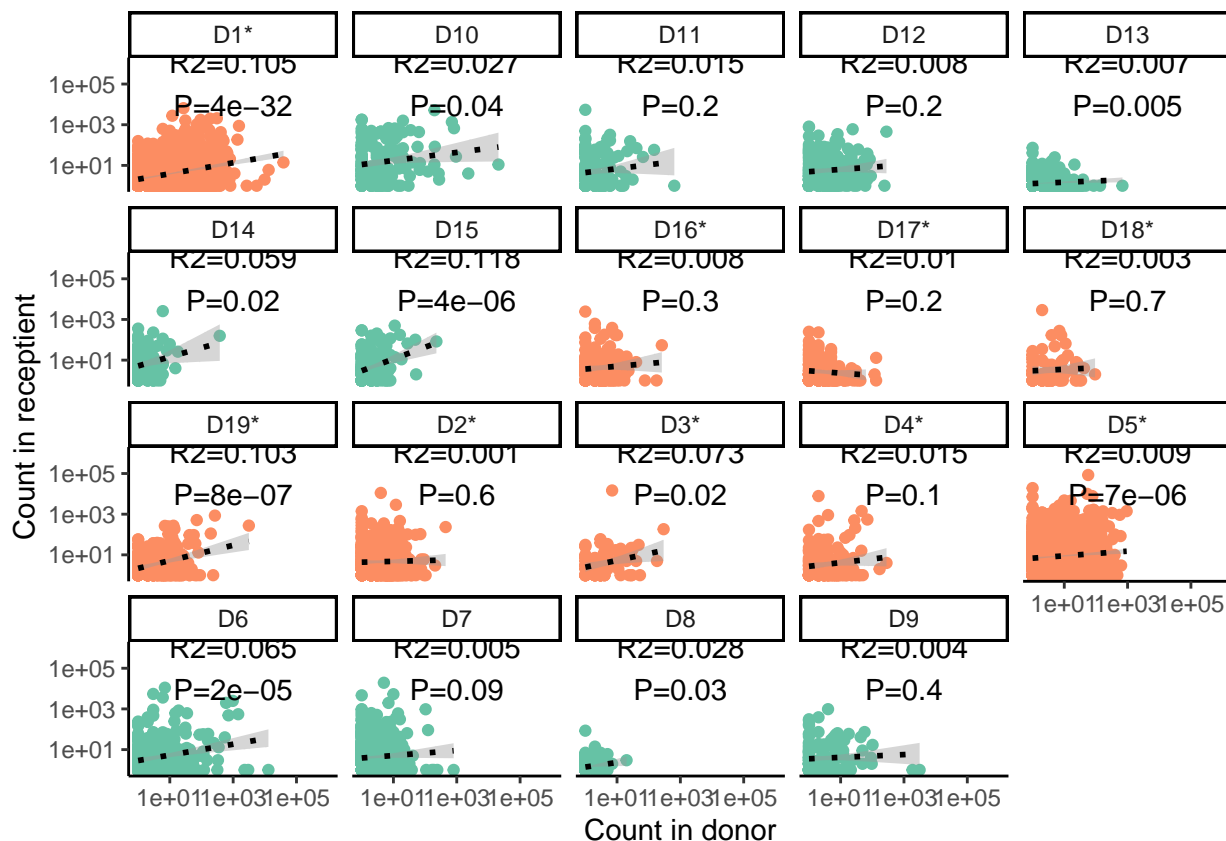
```
## Warning: The `guide` argument in `scale_*()` cannot be `FALSE`. This was deprecated in
## ggplot2 3.3.4.
```

```
## i Please use "none" instead.
```

```
## This warning is displayed once every 8 hours.
```

```
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
```

```
## generated.
```



```

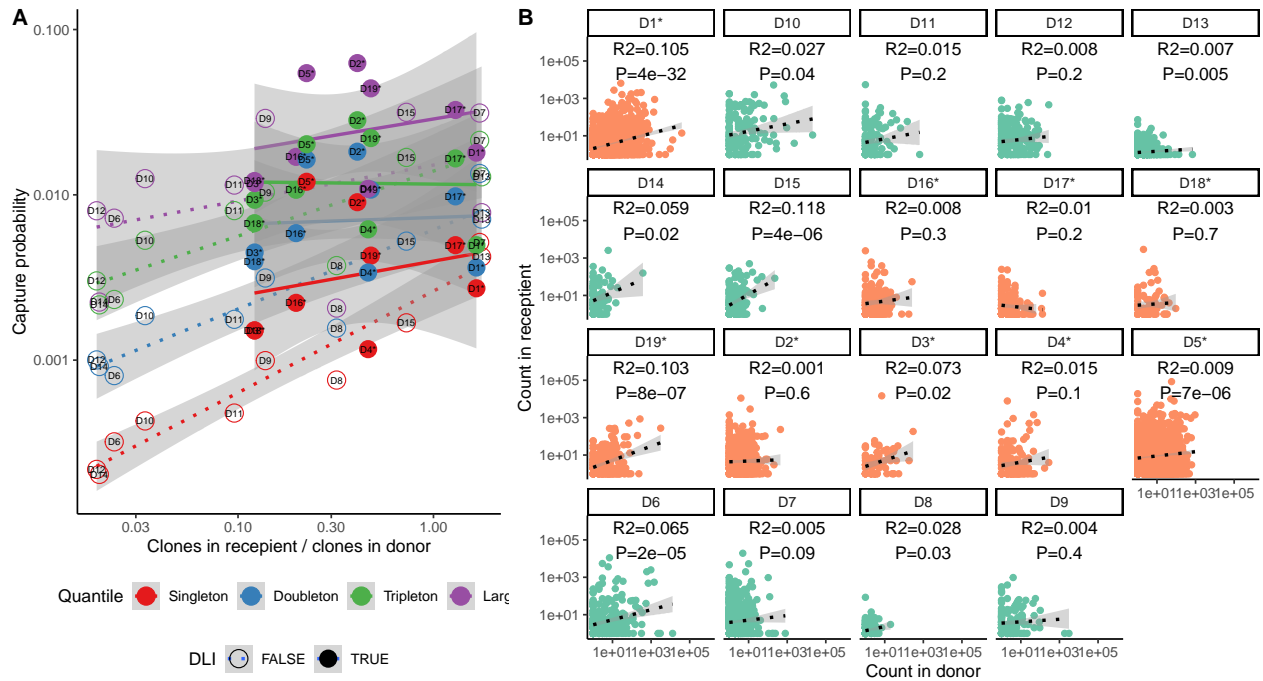
plot_grid(p5, p6,
  labels = c("A", "B"),
  rel_widths = c(1, 1.5)) -> fig5

```

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

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

```
fig5
```



```
ggsave("../figures/fig5.pdf", fig5)
```

```
## Saving 11 x 6 in image
```