Assessing source-sink dynamics in invaded habitat using metabarcoding

Script by Anna J Holmquist

2023-01-28

Read in data frames

```r
# Total samples from collection
collection <- read.csv(file = "collection_data.csv")

# Dataframe including all sequences, without filtering
all_pagio <- read.csv("all_reads.csv")

# Dataframe including all prey sequences following filtering
prey_final <- read.csv("prey_reads.csv")

# Native versus non-native prey status
native_status <- read.csv("native_status.csv")

# Parasites
parasites <- read.csv("parasites.csv")
```

Summaries of results

Collection

```r
# By block, transect and site
collection %>%
  group_by(site_status) %>%
  summarise(n_distinct(id))
```

```
## # A tibble: 2 x 2
##   site_status `n_distinct(id)`
##   <chr>                  <int>
## 1 Ginger                    82
## 2 Native                    86
```

```r
collection %>%
  group_by(site_status, transect, block) %>%
  summarise(n = n_distinct(id)) %>%
  ungroup() %>%
  summarise(mean(n), plotrix::std.error(n))
```

```
## # A tibble: 1 x 2
##   `mean(n)` `plotrix::std.error(n)`
##       <dbl>                   <dbl>
## 1      5.79                   0.547
```

```
collection %>%
  group_by(transect) %>%
  summarise(n = n_distinct(id)) %>%
  ungroup() %>%
  summarise(mean(n), plotrix::std.error(n))
```

```
## # A tibble: 1 x 2
##   `mean(n)` `plotrix::std.error(n)`
##       <dbl>                   <dbl>
## 1      16.8                    2.38
```

Molecular

```
# Reads removed
total_reads <- sum(all_pagio$count)
removed_reads <- sum(all_pagio$count) - sum(prey_final$count)

# From the spiders
all_pagio %>%
  filter(family == "Philodromidae") %>%
  summarize(sum(count) / removed_reads,
            sum(count))
```

```
##   sum(count)/removed_reads sum(count)
## 1                0.6412852     679534
```

```
# From fungi
all_pagio %>%
  filter(kingdom == "Fungi") %>%
  summarize(sum(count),
            sum(count) / removed_reads)
```

```
##   sum(count) sum(count)/removed_reads
## 1     234630                0.2214234
```

```
# Hymenopterans - parisitoids
all_pagio %>%
  filter(order == "Hymenoptera") %>%
  summarise(sum(count),
            sum(count) / removed_reads)
```

```
##   sum(count) sum(count)/removed_reads
## 1      60060              0.05667941
```

```
# Summary of filtered data set

  # Reads and ASVs per spider
  prey_final %>%
    group_by(site_status, id) %>%
    summarise(n = sum(count),
              nasv = n_distinct(asv)) %>%
    ungroup() %>%
    group_by(site_status) %>%
    summarise(reads_mean = mean(n), reads_stderror = plotrix::std.error(n),
              asv_mean = mean(nasv),
              asv_stderror = plotrix::std.error(nasv))
```

```
## # A tibble: 2 x 5
##   site_status reads_mean reads_stderror asv_mean asv_stderror
##   <chr>            <dbl>          <dbl>    <dbl>        <dbl>
## 1 ginger           3469.           449.     2.88        0.175
## 2 native            701.           133.     2.10        0.164
```

```
  # ASVs and specimens by marker
  prey_final %>%
    group_by(marker) %>%
    summarise(asv = n_distinct(asv),
              sample = n_distinct(id))
```

```
## # A tibble: 3 x 3
##   marker   asv sample
##   <chr>  <int>  <int>
## 1 16s      110    112
## 2 18s       10     17
## 3 28s       44     76
```

```
  rm(removed_reads)
```

Taxonomy

```
# Total taxonomy breakdown
prey_final %>%
  summarise(n_distinct(order), n_distinct(family),
            n_distinct(genus), n_distinct(species))
```

```
##   n_distinct(order) n_distinct(family) n_distinct(genus) n_distinct(species)
## 1                 9                 31                32                  23
```

```
prey_final %>%
  summarise(n_distinct(order), n_distinct(family),
            n_distinct(genus), n_distinct(species))
```

```
##   n_distinct(order) n_distinct(family) n_distinct(genus) n_distinct(species)
## 1                 9                 31                32                  23
```

```
# Percent of ASVs with matches

    # Number of ASVs retained with matches
    length(unique(prey_final$asv[!is.na(prey_final$species)])) /
      length(unique(prey_final$asv))
```

```
## [1] 0.2317073
```

```
    length(unique(prey_final$asv[!is.na(prey_final$genus)])) /
      length(unique(prey_final$asv))
```

```
## [1] 0.3780488
```

```
    length(unique(prey_final$asv[!is.na(prey_final$family)])) /
      length(unique(prey_final$asv))
```

```
## [1] 0.6158537
```

```
# Percent matches by marker
  prey_final %>%
    group_by(marker) %>%
    summarise(mean(percent_match))
```

```
## # A tibble: 3 x 2
##   marker `mean(percent_match)`
##   <chr>                  <dbl>
## 1 16s                     91.6
## 2 18s                     98.3
## 3 28s                     97.7
```

```
# Figure
  p1 <- prey_final %>%
    group_by(site_status, order) %>%
    summarise(n = n_distinct(asv)) %>%
    ungroup() %>%
    rename(Order = order) %>%
    mutate(site_status = ifelse(site_status == "ginger", "Ginger-invaded", "Native forest")) %>%
    ggplot() +
    geom_bar(aes(site_status, n, fill = Order), alpha = 0.6,
             stat = "identity", color = "black") +
    xlab("Site status") +
    ylab("Number of ASVs") +
    scale_fill_brewer(palette = "Spectral") +
    theme_minimal()
```

```
## `summarise()` has grouped output by 'site_status'. You can override using the
## `.groups` argument.
```

```
  p2 <- prey_final %>%
    filter(!is.na(family)) %>%
    group_by(site_status, family) %>%
    summarise(n = n_distinct(asv)) %>%
    ungroup() %>%
    rename(Family = family) %>%
    mutate(site_status = ifelse(site_status == "ginger", "Ginger-invaded", "Native forest")) %>%
    ggplot() +
    geom_bar(aes(site_status, n, fill = Family), alpha = 0.6,
             stat = "identity", color = "black") +
    xlab("Site status") +
    ylab("Number of ASVs") +
    scale_fill_brewer(palette = "Spectral") +
    theme_minimal()
```

```
## `summarise()` has grouped output by 'site_status'. You can override using the
## `.groups` argument.
```

Prey diversity and abundance

Summarizing composition

```
# All ASVs
length(unique(prey_final$asv))
```

```
## [1] 164
```

```
# Shared ASVs
length(intersect(prey_final$asv[prey_final$site_status == "ginger"],
         prey_final$asv[prey_final$site_status == "native"]))
```

```
## [1] 29
```

```
# Shared taxonomy
length(intersect(prey_final$order[prey_final$site_status == "ginger"],
         prey_final$order[prey_final$site_status == "native"]))
```

```
## [1] 6
```

```
length(unique(prey_final$order))
```

```
## [1] 9
```

```
length(intersect(prey_final$family[prey_final$site_status == "ginger"],
         prey_final$family[prey_final$site_status == "native"]))
```

```
## [1] 11
```

```r
length(unique(prey_final$family))
```

```
## [1] 31
```

```r
length(intersect(prey_final$genus[prey_final$site_status == "ginger"],
         prey_final$genus[prey_final$site_status == "native"]))
```

```
## [1] 10
```

```r
length(unique(prey_final$genus))
```

```
## [1] 32
```

```r
length(intersect(prey_final$species[prey_final$site_status == "ginger"],
         prey_final$species[prey_final$site_status == "native"]))
```

```
## [1] 5
```

```r
length(unique(prey_final$species))
```

```
## [1] 23
```

Hill numbers

Create community matrices

```r
# Create community matrices

  # By ASVs
    asv_comm <-
      prey_final %>%
      acast(id ~ asv,
            value.var = "count", # Count as values
            fun.aggregate = sum) %>% # Sum counts for site
      as.matrix()

  # Transform
    asv_hellinger <- decostand(asv_comm, method = "hellinger")

  # Incidence
  asv_incidence <- asv_hellinger
  asv_incidence[asv_incidence > 0] <- 1

  # By taxonomy - order
  order_comm <-
    prey_final %>%
    acast(id ~ order,
          value.var = "count",
          fun.aggregate = sum,
          fill = 0) %>%
    as.matrix()

  order_hellinger <- decostand(order_comm, method = "hellinger")

  order_incidence <- order_hellinger
  order_incidence[order_incidence > 0] <- 1
```

Calculate Hill numbers and perform Welch t-test

```
# ASVs

  asv_hill <-
  renyi(asv_hellinger,
        scales = c(0, 1, 2),
        hill = T) %>%
    cbind(collection[collection$id %in% rownames(asv_hellinger),])


    # Welch t-test - ASV
    asv_q0 <- t.test(asv_hill$`0`[asv_hill$site_status == "Ginger"],
                     asv_hill$`0`[asv_hill$site_status == "Native"])

    asv_q1 <- t.test(asv_hill$`1`[asv_hill$site_status == "Ginger"],
                     asv_hill$`1`[asv_hill$site_status == "Native"])

    asv_q2 <- t.test(asv_hill$`2`[asv_hill$site_status == "Ginger"],
                     asv_hill$`2`[asv_hill$site_status == "Native"])

    hill_asv_sum <-
      data.frame(cbind(

      rbind(round(asv_q0$estimate, digits = 3),
        round(asv_q1$estimate, digits = 3),
        round(asv_q2$estimate, digits = 3)),

      rbind(round(asv_q0$statistic, digits = 3),
        round(asv_q1$statistic, digits = 3),
        round(asv_q2$statistic, digits = 3)),

      rbind(paste0(round(asv_q0$p.value, digits = 4), "**"),
        paste0(round(asv_q1$p.value, digits = 4), "**"),
        paste0(round(asv_q2$p.value, digits = 4), "**"))
      )
    )

    colnames(hill_asv_sum) <- c("ginger", "native", "t", "p")
    hill_asv_sum$level <- "ASV"
    hill_asv_sum$q <- c("q = 0", "q = 1", "q = 2")
```

```r
# Order

     order_hill <- renyi(order_hellinger, scales = c(0, 1, 2), hill = T) %>%
         cbind(collection[collection$id %in% rownames(order_hellinger),])

     # Welch t-test - order
     order_q0 <- t.test(order_hill$`0`[order_hill$site_status == "Ginger"],
                        order_hill$`0`[order_hill$site_status == "Native"])

     order_q1 <- t.test(order_hill$`1`[order_hill$site_status == "Ginger"],
                        order_hill$`1`[order_hill$site_status == "Native"])

     order_q2 <- t.test(order_hill$`2`[order_hill$site_status == "Ginger"],
                        order_hill$`2`[order_hill$site_status == "Native"])

    hill_order_sum <- data.frame(
      cbind(
         rbind(round(order_q0$estimate, digits = 3),
           round(order_q1$estimate, digits = 3),
           round(order_q2$estimate, digits = 3)),

         rbind(round(order_q0$statistic, digits = 3),
           round(order_q1$statistic, digits = 3),
           round(order_q2$statistic, digits = 3)),

         rbind(paste0(round(order_q0$p.value, digits = 4), "***"),
           paste0(round(order_q1$p.value, digits = 4), "***"),
           paste0(round(order_q2$p.value, digits = 4), "**"))
         )
       )

     colnames(hill_order_sum) <- c("ginger", "native", "t", "p")
     hill_order_sum$q <- c("q = 0", "q = 1", "q = 2")
     hill_order_sum$level <- "Order"

# Combined table
  hill_summary <- rbind(hill_asv_sum, hill_order_sum)  %>%
    mutate(`Mean Difference` = as.numeric(ginger) - as.numeric(native)) %>%
    select(level, q, `Mean Difference`, ginger, native, t, p)

  hill_table <-  gt(hill_summary, groupname_col = "level", rowname_col = "q") %>%
    cols_label(ginger = "Ginger-invaded",
               native = "Native forest",
               t = "t-statistic",
               p = "p-value",
               `Mean Difference` = "Mean difference") %>%
    cols_width(`Mean Difference` ~ px(110),
               ginger ~ px(110),
               native ~ px(110),
               t ~ px(110),
               p ~ px(110)) %>%
    cols_align(align = "center") %>%
    tab_options(table.font.names = "Times New Roman",
                row_group.font.weight = "bold",
                row_group.padding = 5,
                column_labels.font.size = 14, table.font.size = 12,
                row_group.font.size = 14)
```

Differences in read abundances

```
reads <-
  prey_final %>%
  group_by(site_status, id) %>%
  summarise(n = sum(count)) %>%
  ungroup()

reads %>%
  group_by(site_status) %>%
  summarise(mean(n), plotrix::std.error(n))
```

```
## # A tibble: 2 x 3
##   site_status `mean(n)` `plotrix::std.error(n)`
##   <chr>           <dbl>                   <dbl>
## 1 ginger          3469.                    449.
## 2 native           701.                    133.
```

```
t.test(reads$n[reads$site_status == "native"],
       reads$n[reads$site_status == "ginger"])
```

```
##
##  Welch Two Sample t-test
##
## data:  reads$n[reads$site_status == "native"] and reads$n[reads$site_status == "ginger"]
## t = -5.9106, df = 90.205, p-value = 5.98e-08
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -3698.444 -1837.695
## sample estimates:
## mean of x mean of y
##  700.8793 3468.9487
```

Dietary composition

Calculate beta diversity using individuals

```
beta_transform_asv <- beta(asv_hellinger, func = "s", abund = T)
beta_inc_asv <- beta(asv_comm,
                     func = "jaccard", abund = F)

beta_transform_order <- beta(order_hellinger, func = "s", abund = T)
beta_inc_order <- beta(order_comm,
                       func = "jaccard", abund = F)
```

Melt beta diversity

```r
b1 <- as.matrix(beta_transform_asv$Btotal)
b1[upper.tri(b1)] <- NA

asv_b1 <- melt(b1) %>%
  filter(Var1 != Var2 & !is.na(value)) %>%
  left_join(collection, by = c("Var1" = "id")) %>%
  left_join(collection, by = c("Var2" = "id")) %>%
  mutate(comparison = case_when(
    site_status.x == site_status.y & site_status.x == "Native" ~ "Native forest",
    site_status.x == site_status.y & site_status.x == "Ginger" ~ "Ginger-invaded",
    site_status.x != site_status.y ~ "Comparison between habitat",
  ))

b2 <- as.matrix(beta_transform_order$Btotal)
b2[upper.tri(b2)] <- NA

order_b2 <- melt(b2) %>%
  filter(Var1 != Var2 & !is.na(value)) %>%
  left_join(collection, by = c("Var1" = "id")) %>%
  left_join(collection, by = c("Var2" = "id")) %>%
  mutate(comparison = case_when(
    site_status.x == site_status.y & site_status.x == "Native" ~ "Native forest",
    site_status.x == site_status.y & site_status.x == "Ginger" ~ "Ginger-invaded",
    site_status.x != site_status.y ~ "Comparison between habitat",
  ))

b1_p <- asv_b1 %>%
  ggplot(aes(comparison, value)) +
  geom_boxplot() +
  geom_jitter(alpha = 0.1) +
  xlab("") +
  theme_minimal()

b2_p <- order_b2 %>%
  ggplot(aes(comparison, value)) +
  geom_boxplot() +
  geom_jitter(alpha = 0.1) +
  xlab("") +
  theme_minimal()

beta_indv_plot <- ggarrange(b1_p, b2_p, labels = c("ASV", "Order"))
ggsave("SupplementalFigure2.pdf", height = 5, width = 12, units = "in", dpi = 600)
```

Constructing community matrices, by site

```r
# Data for sites
sites_data <-
  prey_final %>%
  group_by(site) %>%
  filter(row_number() == 1) %>%
  ungroup() %>%
  select(site, site_status)

# By ASV - all
site_community_asv <-
  prey_final %>%
  acast(site ~ asv, fill = 0,
        value.var = "count", # Count as values
        fun.aggregate = sum)

site_asv_hell <- decostand(site_community_asv, method = "hellinger")
site_asv_inc <- site_community_asv
site_asv_inc[site_asv_inc > 0] <- 1

# By order
site_community_order <-
  prey_final %>%
  acast(site ~ order, fill = 0,
        value.var = "count", # Count as values
        fun.aggregate = sum)

site_order_hell <- decostand(site_community_order, method = "hellinger")
site_order_inc <- site_community_order
site_order_inc[site_order_inc > 0] <- 1
```

Calculating distances

```r
dist_asv_hell <- vegdist(site_asv_hell, method = "euclidean")
dist_order_hell <- vegdist(site_order_hell, method = "euclidean")

dist_asv_inc <- vegdist(site_asv_inc, method = "jaccard")
dist_order_inc <- vegdist(site_order_inc, method = "jaccard")
```

PERMANOVA

```r
# PERMANOVA
perm_asv_hell <- adonis2(site_asv_hell ~ site_status,
        data = sites_data, method = "euclidean")

perm_order_hell <- adonis2(site_order_hell ~ site_status,
        data = sites_data, method = "euclidean")

perm_asv_inc <- adonis2(site_asv_inc ~ site_status,
        data = sites_data, method = "jaccard")

perm_order_inc <- adonis2(site_order_inc ~ site_status,
        data = sites_data, method = "jaccard")


# Summary
test <- as.data.frame(cbind(

rbind(perm_asv_hell$F[1],
perm_order_hell$F[1],
perm_asv_inc$F[1],
perm_order_inc$F[1]),

rbind(perm_asv_hell$R2[1],
perm_order_hell$R2[1],
perm_asv_inc$R2[1],
perm_order_inc$R2[1]

  ),

rbind(perm_asv_hell$`Pr(>F)`[1],
perm_order_hell$`Pr(>F)`[1],
paste0(perm_asv_inc$`Pr(>F)`[1], "*"),
paste0(perm_order_inc$`Pr(>F)`[1], "*"))))

colnames(test) <- c("F", "R2", "Pr(>F)")
test$`Abundance Type` <- c("Hellinger-transformed",
                           "Hellinger-transformed",
                           "Incidence",
                           "Incidence")
test$Level <- c("ASV", "Order", "ASV", "Order")

perm_table <- test %>%
  mutate(R2 = round(as.numeric(R2), digits = 4),
         F = round(as.numeric(F), digits = 4))


perm_table <- gt(perm_table, groupname_col = "Level", rowname_col = "Abundance Type") %>%
     cols_align(align = "center",
               columns = c("F", "Pr(>F)", "R2")) %>%
  tab_options(table.font.names = "Times New Roman",
              row_group.font.size = 14,
              #row_group.border.top.color = "white",
              #row_group.border.bottom.color = "white",
              column_labels.font.size = 14,
              column_labels.font.weight = "lighter",
              table.font.size = 12,
              table.width = 500
              ) %>%
  cols_width(`Abundance Type` ~ 120) %>%
  cols_label(R2 = md("R<sup>2</sup>"))
```

PERMDISP

```
mod1 <- anova(betadisper(dist_asv_hell, sites_data$site_status))
mod2 <- anova(betadisper(dist_asv_inc, sites_data$site_status))
mod3 <- anova(betadisper(dist_order_hell, sites_data$site_status))
mod4 <- anova(betadisper(dist_order_inc, sites_data$site_status))
```

```
## Warning in betadisper(dist_order_inc, sites_data$site_status): some squared
## distances are negative and changed to zero
```

```
stargazer(mod1, mod2, mod3, mod4,
          title = c("PERMDISP - ASV Hellinger",
                    "PERMDISP - ASV Incidence",
                    "PERMDISP - Order Hellinger",
                    "PERMDISP - Order Incidence"),
          type="html", out="permdisp_anova.doc")
```

```
##
## <table style="text-align:center"><caption><strong>PERMDISP - ASV Hellinger</strong></caption>
## <tr><td colspan="6" style="border-bottom: 1px solid black"></td></tr><tr><td style="text-align:left"
## <tr><td colspan="6" style="border-bottom: 1px solid black"></td></tr><tr><td style="text-align:left"
## <tr><td style="text-align:left">Sum Sq</td><td>2</td><td>0.087</td><td>0.120</td><td>0.002</td><td>0
## <tr><td style="text-align:left">Mean Sq</td><td>2</td><td>0.004</td><td>0.003</td><td>0.002</td><td>
## <tr><td style="text-align:left">F value</td><td>1</td><td>0.297</td><td></td><td>0.297</td><td>0.297
## <tr><td style="text-align:left">Pr(> F)</td><td>1</td><td>0.590</td><td></td><td>0.590</td><td>0.590
## <tr><td colspan="6" style="border-bottom: 1px solid black"></td></tr></table>
##
## <table style="text-align:center"><caption><strong>PERMDISP - ASV Incidence</strong></caption>
## <tr><td colspan="6" style="border-bottom: 1px solid black"></td></tr><tr><td style="text-align:left"
## <tr><td colspan="6" style="border-bottom: 1px solid black"></td></tr><tr><td style="text-align:left"
## <tr><td style="text-align:left">Sum Sq</td><td>2</td><td>0.024</td><td>0.034</td><td>0.00005</td><td
## <tr><td style="text-align:left">Mean Sq</td><td>2</td><td>0.001</td><td>0.001</td><td>0.00005</td><t
## <tr><td style="text-align:left">F value</td><td>1</td><td>0.026</td><td></td><td>0.026</td><td>0.026
## <tr><td style="text-align:left">Pr(> F)</td><td>1</td><td>0.873</td><td></td><td>0.873</td><td>0.873
## <tr><td colspan="6" style="border-bottom: 1px solid black"></td></tr></table>
##
## <table style="text-align:center"><caption><strong>PERMDISP - Order Hellinger</strong></caption>
## <tr><td colspan="6" style="border-bottom: 1px solid black"></td></tr><tr><td style="text-align:left"
## <tr><td colspan="6" style="border-bottom: 1px solid black"></td></tr><tr><td style="text-align:left"
## <tr><td style="text-align:left">Sum Sq</td><td>2</td><td>0.970</td><td>1.199</td><td>0.123</td><td>1
## <tr><td style="text-align:left">Mean Sq</td><td>2</td><td>0.095</td><td>0.039</td><td>0.067</td><td>
## <tr><td style="text-align:left">F value</td><td>1</td><td>1.821</td><td></td><td>1.821</td><td>1.821
## <tr><td style="text-align:left">Pr(> F)</td><td>1</td><td>0.188</td><td></td><td>0.188</td><td>0.188
## <tr><td colspan="6" style="border-bottom: 1px solid black"></td></tr></table>
##
## <table style="text-align:center"><caption><strong>PERMDISP - Order Incidence</strong></caption>
## <tr><td colspan="6" style="border-bottom: 1px solid black"></td></tr><tr><td style="text-align:left"
## <tr><td colspan="6" style="border-bottom: 1px solid black"></td></tr><tr><td style="text-align:left"
## <tr><td style="text-align:left">Sum Sq</td><td>2</td><td>0.476</td><td>0.649</td><td>0.017</td><td>0
## <tr><td style="text-align:left">Mean Sq</td><td>2</td><td>0.026</td><td>0.012</td><td>0.017</td><td>
## <tr><td style="text-align:left">F value</td><td>1</td><td>0.494</td><td></td><td>0.494</td><td>0.494
## <tr><td style="text-align:left">Pr(> F)</td><td>1</td><td>0.488</td><td></td><td>0.488</td><td>0.488
## <tr><td colspan="6" style="border-bottom: 1px solid black"></td></tr></table>
```

Perform NMDS

```r
set.seed(1)

# ASV NMDS
site_asv_nmds1 <- metaMDS(dist_asv_hell, trymax = 1000, k = 3)
site_asv_nmds2 <- metaMDS(dist_asv_inc, trymax = 1000, k = 3)

# Order NMDS
site_order_nmds1 <- metaMDS(dist_order_hell, trymax = 1000)
site_order_nmds2 <- metaMDS(dist_order_inc, trymax = 1000)
```

Plot NMDS

```r
PlotNMDS <- function(nmds, data){
 # Transformed
  scores <- as.data.frame(scores(nmds))
  scores$site <- rownames(nmds$points)
  scores <- data %>%
    right_join(scores, by = "site") %>%
    mutate(site_status = ifelse(site_status == "ginger", "Ginger-invaded", "Native forest"))

  # Polygon
    hull_native <- scores[scores$site_status == "Native forest", ][chull(scores[scores$site_status == "
    hull_ginger <- scores[scores$site_status == "Ginger-invaded", ][chull(scores[scores$site_status ==
    hull_asv <- rbind(hull_native, hull_ginger)

  # Plot
    site_plot <-
      ggplot() +
      geom_polygon(data = hull_asv, aes(x = NMDS1, y = NMDS2,
                                         group = site_status, fill = site_status),
                   alpha = 0.2) +
      geom_point(data = scores,
                 aes(NMDS1, NMDS2, color = site_status, shape = site_status),
                 alpha = 0.9) +
      annotate("text",
               label = paste0("Stress = ", round(nmds$stress, 3)),
               x = 0.25, y = 0.3, size = 3.5) +
      scale_color_manual(values = c("#9757bd", "#1d7835")) +
      scale_fill_manual(values = c( "#9757bd", "#1d7835")) +
      theme_minimal() +
      theme(legend.title = element_blank())

    return(site_plot)
}

# Adjusted x/y of annotate for all
p1 <- PlotNMDS(site_asv_nmds1, sites_data)
p2 <- PlotNMDS(site_asv_nmds2, sites_data)
p3 <- PlotNMDS(site_order_nmds1, sites_data)
```

```
p4 <- PlotNMDS(site_order_nmds2, sites_data)

nmds_plot <- ggpubr::ggarrange(p1, p2, p3, p4, ncol = 2, nrow = 2,
                               common.legend = T, labels = "AUTO")
ggsave("Figure3.pdf", dpi = 600, height = 9, width = 10, unit = "in")
```

Calculate beta diversity using communities

```
beta_transform_asv <- beta(site_asv_inc, func = "s", abund = T)
beta_inc_asv <- beta(site_asv_inc,
                     func = "jaccard", abund = F)

beta_transform_order <- beta(site_order_hell, func = "s", abund = T)
beta_inc_order <- beta(site_order_inc,
                       func = "jaccard", abund = F)
```

Melt beta diversity

```
b1 <- as.matrix(beta_transform_asv$Btotal)
b1[upper.tri(b1)] <- NA

asv_b1 <- melt(b1) %>%
  filter(Var1 != Var2 & !is.na(value)) %>%
  left_join(sites_data, by = c("Var1" = "site")) %>%
  left_join(sites_data, by = c("Var2" = "site")) %>%
  mutate(Comparison = case_when(
    site_status.x == site_status.y & site_status.x == "Native" ~ "Native forest",
    site_status.x == site_status.y & site_status.x == "Ginger" ~ "Ginger-invaded",
    site_status.x != site_status.y ~ "Between habitat"
  ))

asv_b1 %>%
  summarise(mean(value))

b2 <- as.matrix(beta_transform_order$Btotal)
b2[upper.tri(b2)] <- NA

order_b2 <- melt(b2) %>%
  filter(Var1 != Var2 & !is.na(value)) %>%
  left_join(sites_data, by = c("Var1" = "site")) %>%
  left_join(sites_data, by = c("Var2" = "site")) %>%
  mutate(Comparison = case_when(
    site_status.x == site_status.y & site_status.x == "Native" ~ "Native forest",
    site_status.x == site_status.y & site_status.x == "Ginger" ~ "Ginger-invaded",
    site_status.x != site_status.y ~ "Between habitat"
  ))

b3 <- as.matrix(beta_inc_order$Btotal)
```

```
b3[upper.tri(b3)] <- NA

order_b3 <- melt(b3) %>%
  filter(Var1 != Var2 & !is.na(value)) %>%
  left_join(sites_data, by = c("Var1" = "site")) %>%
  left_join(sites_data, by = c("Var2" = "site")) %>%
  mutate(Comparison = case_when(
    site_status.x == site_status.y & site_status.x == "Native" ~ "Native forest",
    site_status.x == site_status.y & site_status.x == "Ginger" ~ "Ginger-invaded",
    site_status.x != site_status.y ~ "Between habitat"))

b4 <- as.matrix(beta_inc_asv$Btotal)
b4[upper.tri(b4)] <- NA

asv_b4 <- melt(b4) %>%
  filter(Var1 != Var2 & !is.na(value)) %>%
  left_join(sites_data, by = c("Var1" = "site")) %>%
  left_join(sites_data, by = c("Var2" = "site")) %>%
  mutate(Comparison = case_when(
    site_status.x == site_status.y & site_status.x == "Native" ~ "Native forest",
    site_status.x == site_status.y & site_status.x == "Ginger" ~ "Ginger-invaded",
    site_status.x != site_status.y ~ "Between habitat"
  ))

asv_b4 %>%
  summarise(mean(value))

p1 <- asv_b1 %>%
  ggplot(aes(Comparison, value)) +
  geom_boxplot(aes(color = Comparison)) +
  geom_jitter(aes(color = Comparison), alpha = 0.3) +
  #scale_fill_manual(values = c("#cf9700", "#9757bd", "#1d7835")) +
  scale_color_manual(values = c("#cf9700", "#9757bd", "#1d7835")) +
  scale_x_discrete(labels = function(x) str_wrap(x, width = 15)) +
  ylab("Beta Diversity Score") +
  xlab("") +
  theme_minimal()

p2 <- asv_b4 %>%
  ggplot(aes(Comparison, value)) +
  geom_boxplot(aes(color = Comparison)) +
  geom_jitter(aes(color = Comparison), alpha = 0.3) +
  #scale_fill_manual(values = c("#cf9700", "#9757bd", "#1d7835")) +
  scale_color_manual(values = c("#cf9700", "#9757bd", "#1d7835")) +
  scale_x_discrete(labels = function(x) str_wrap(x, width = 15)) +
  ylab("Beta Diversity Score") +
  xlab("") +
  theme_minimal()

p3 <- order_b2 %>%
  ggplot(aes(Comparison, value)) +
  geom_boxplot(aes(color = Comparison)) +
  geom_jitter(aes(color = Comparison), alpha = 0.3) +
```

```r
  #scale_fill_manual(values = c("#cf9700", "#9757bd", "#1d7835")) +
  scale_color_manual(values = c("#cf9700", "#9757bd", "#1d7835")) +
  scale_x_discrete(labels = function(x) str_wrap(x, width = 15)) +
  ylab("Beta Diversity Score") +
  xlab("") +
  theme_minimal()

p4 <- order_b3 %>%
  ggplot(aes(Comparison, value)) +
  geom_boxplot(aes(color = Comparison)) +
  geom_jitter(aes(color = Comparison), alpha = 0.3) +
  #scale_fill_manual(values = c("#cf9700", "#9757bd", "#1d7835")) +
  scale_color_manual(values = c("#cf9700", "#9757bd", "#1d7835")) +
  scale_x_discrete(labels = function(x) str_wrap(x, width = 15)) +
  ylab("Beta Diversity Score") +
  xlab("") +
  theme_minimal()

comm_beta <- ggarrange(p1,p2,p3,p4,
                       common.legend = TRUE)
ggsave("comm_beta_asv.pdf", dpi = 600, height = 6, width = 10, unit = "in")

#ggsave("nmds.pdf", width = 10, heigh = 8, dpi = 600)

asv_hell_b <- summary(aov(value ~ Comparison, data = asv_b1))
TukeyHSD(aov(value ~ Comparison, data = asv_b1))

asv_inc_b <- summary(aov(value ~ Comparison, data = asv_b4))
TukeyHSD(aov(value ~ Comparison, data = asv_b4))

order_hell_b <- summary(aov(value ~ Comparison, data = order_b2))
TukeyHSD(aov(value ~ Comparison, data = order_b2))

order_inc_b <- summary(aov(value ~ Comparison, data = order_b3))
TukeyHSD(order_inc_b)


f <- c(13.53, 14.09, 154.9, 166.7)
mean_sq <- c(0.13020, 0.04772, 17.467, 15.663)
sum_sq <- c(0.260, 0.0954, 34.9, 31.3)
pr <- c("2.06e-06***", "1.22e-06***", "<2e-16***", "<2e-16***")
comp <- c("ASV", "ASV", "Order", "Order")
type <- c("Hellinger", "Incidence", "Hellinger", "Incidence")

beta_tests<-as.data.frame(matrix(nrow = 4))
beta_tests$f <- c(13.53, 14.09, 154.9, 166.7)
beta_tests$MSE <- c(0.13020, 0.04772, 17.467, 15.663)
beta_tests$SSE <- c(0.260, 0.0954, 34.9, 31.3)
beta_tests$Pr <- c("2.06e-06***", "1.22e-06***", "<2e-16***", "<2e-16***")
beta_tests$type <- c("Hellinger", "Incidence", "Hellinger", "Incidence")
beta_tests$comp <- comp <- c("ASV", "ASV", "Order", "Order")
beta_tests <- select(beta_tests, -V1)
```

```
beta <- gt::gt(beta_tests, groupname_col = "comp", rowname_col = "type") %>%
     cols_align(align = "center",
                columns = c("f", "MSE", "SSE", "Pr")) %>%
  tab_options(table.font.names = "Times New Roman",
              row_group.font.size = 14,
              #row_group.border.top.color = "white",
              #row_group.border.bottom.color = "white",
              column_labels.font.size = 14,
              column_labels.font.weight = "lighter",
              table.font.size = 12,
              table.width = 500
              ) %>%
  #cols_width(`Abundance Type` ~ 120) %>%
  cols_label(Pr = "Pr(>F)",
             f = "F")
```

Order-level diversity

```
prey_final %>%
  group_by(site_status, order) %>%
  # Number of spiders in each site eating each order
  summarise(n = n_distinct(id)) %>%
  ungroup() %>%
  # Add column for total number of spiders in each habitat
  mutate(total = ifelse(site_status == "Ginger",
                        78, 58),
         # Calculate proportion of spiders eating each order
         prop = round(n / total, 3))
```

```
## `summarise()` has grouped output by 'site_status'. You can override using the
## `.groups` argument.
```

```
## # A tibble: 15 x 5
##    site_status order                n total  prop
##    <chr>       <chr>            <int> <dbl> <dbl>
##  1 ginger      Araneae              1    58 0.017
##  2 ginger      Coleoptera          15    58 0.259
##  3 ginger      Diptera             29    58 0.5
##  4 ginger      Entomobryomorpha    27    58 0.466
##  5 ginger      Hemiptera           34    58 0.586
##  6 ginger      Lepidoptera         33    58 0.569
##  7 ginger      Neuroptera           1    58 0.017
##  8 ginger      Thysanoptera         4    58 0.069
##  9 native      Coleoptera           2    58 0.034
## 10 native      Diptera             13    58 0.224
## 11 native      Entomobryomorpha     1    58 0.017
## 12 native      Hemiptera           42    58 0.724
## 13 native      Lepidoptera         21    58 0.362
## 14 native      Psocoptera           1    58 0.017
## 15 native      Thysanoptera         2    58 0.034
```

```r
library(igraph)
```

```
##
## Attaching package: 'igraph'

## The following objects are masked from 'package:ape':
##
##     degree, edges, mst, ring

## The following object is masked from 'package:vegan':
##
##     diversity

## The following object is masked from 'package:permute':
##
##     permute

## The following objects are masked from 'package:dplyr':
##
##     as_data_frame, groups, union

## The following objects are masked from 'package:purrr':
##
##     compose, simplify

## The following object is masked from 'package:tidyr':
##
##     crossing

## The following object is masked from 'package:tibble':
##
##     as_data_frame

## The following objects are masked from 'package:stats':
##
##     decompose, spectrum

## The following object is masked from 'package:base':
##
##     union
```

```r
library(bipartite)
```

```
## Loading required package: sna

## Loading required package: statnet.common

##
## Attaching package: 'statnet.common'
## The following objects are masked from 'package:base':
##
##     attr, order

## Loading required package: network

##
## 'network' 1.18.0 (2022-10-05), part of the Statnet Project
## * 'news(package="network")' for changes since last version
## * 'citation("network")' for citation information
## * 'https://statnet.org' for help, support, and other information

##
## Attaching package: 'network'

## The following objects are masked from 'package:igraph':
##
##     %c%, %s%, add.edges, add.vertices, delete.edges, delete.vertices,
##     get.edge.attribute, get.edges, get.vertex.attribute, is.bipartite,
##     is.directed, list.edge.attributes, list.vertex.attributes,
##     set.edge.attribute, set.vertex.attribute

## The following object is masked from 'package:ComplexHeatmap':
##
##     %v%

## sna: Tools for Social Network Analysis
## Version 2.7 created on 2022-05-09.
## copyright (c) 2005, Carter T. Butts, University of California-Irvine
##  For citation information, type citation("sna").
##  Type help(package="sna") to get started.

##
## Attaching package: 'sna'

## The following objects are masked from 'package:igraph':
##
##     betweenness, bonpow, closeness, components, degree, dyad.census,
##     evcent, hierarchy, is.connected, neighborhood, triad.census

## The following object is masked from 'package:gt':
##
##     gt

## The following objects are masked from 'package:ape':
##
##     consensus, degree

##  This is bipartite 2.18.
##  For latest changes see versionlog in ?"bipartite-package". For citation see: citation("bipartite").
##  Have a nice time plotting and analysing two-mode networks.
```

```
##
## Attaching package: 'bipartite'

## The following object is masked from 'package:igraph':
##
##     strength

## The following object is masked from 'package:vegan':
##
##     nullmodel
```

```
order_div <-
  prey_final %>%
  group_by(site_status, order) %>%
  # Number of spiders in each site eating each order
  summarise(n = n_distinct(id)) %>%
  ungroup() %>%
  # Add column for total number of spiders in each habitat
  mutate(total = ifelse(site_status == "ginger",
                        78, 58),
         # Calculate proportion of spiders eating each order
         prop = round(n / total, 3)) %>%
  mutate(site_status = ifelse(site_status == "Ginger", "Ginger-invaded", "Native forest"))
```

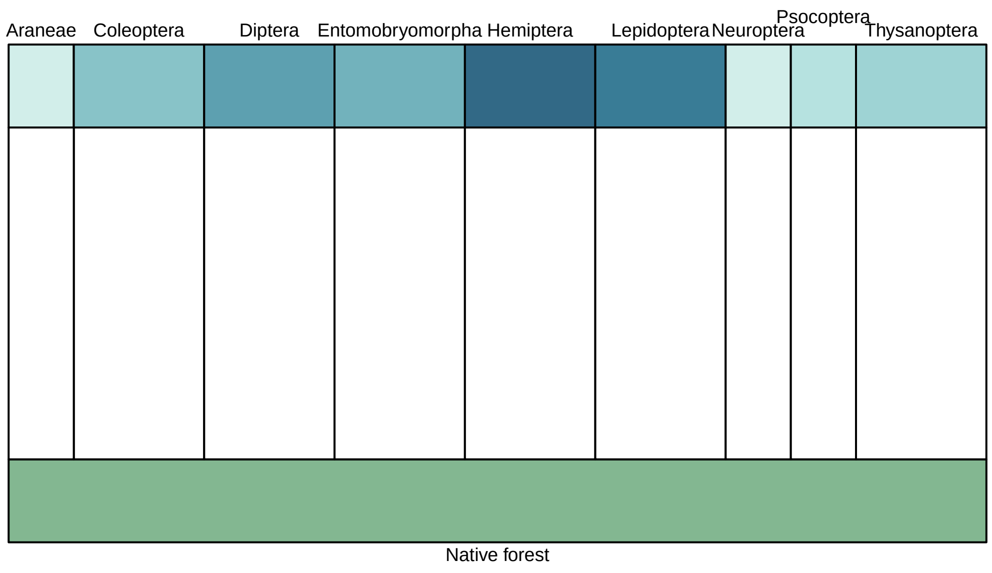
```
## `summarise()` has grouped output by 'site_status'. You can override using the
## `.groups` argument.
```

```
order_mat <-
  order_div %>%
  acast(site_status ~ order,
        fill = 0, value.var = "prop") %>%
  as.matrix()
```

```
## Aggregation function missing: defaulting to length
```

```
# Greens - 7 #216b35
green <- colorRampPalette(c("#ebf7ee", "#1d7835"))
green_col <- green(7)
# Purples - 8
purp <- colorRampPalette(c("#efe6f5", "#764694"))
purp_col <- purp(8)

int <- c("#D2EEEA", "#88C3C8", "#5DA0B0", "#72B2BC", "#326986",
  "#397C96", "#D2EEEA", "#B6E2E0", "#9ED3D4")

#Aran: "#F1F1F1"
#Neur: "#F1F1F1"
#Psc: "#D5E9EE"
#Thys: "#BDD9E7"
#Col: "#A7C6DD"
#Ent: "#8796C2"
#Dip:"#7C7BB2"
```

```
#Lep: "#755E9F"
#Hemi :"#6F3C8B"


# Bipartite
combo_color <- c("white", purp_col[1], green_col[1],
                 green_col[2], purp_col[2], purp_col[3],
                 green_col[3], purp_col[4], green_col[4],
                 purp_col[5], purp_col[6], purp_col[8], green_col[5])

plotweb(order_mat, col.low = c(green_col[4],purp_col[4]),
        col.interaction = "white", col.high = int)
```



Native versus non-native prey

```
# Count identified
length(unique(native_status$asv[native_status$status != "unknown"]))
```

```
## [1] 53
```

```
length(unique(prey_final$asv)) # 164
```

```
## [1] 164
```

```
# Determining prey composition for each individual
composition <-
  prey_final %>%
  mutate(status = case_when(
    asv %in% native_status$asv[native_status$status == "Native"] ~ "Native",
    asv %in% native_status$asv[native_status$status == "Non-native"] ~ "Non-native",
    TRUE ~ "unknown"
    )
  ) %>%
  group_by(id) %>%
  mutate(prey_comp = case_when(
    any(status == "Non-native") & any(status == "Native") ~ "Both",
    any(status == "Non-native") & !any(status == "Native") ~ "Non-native",
    !any(status == "Non-native") & any(status == "Native") ~ "Native",
    !any(status == "Non-native") & !any(status == "Native") ~ "Unknown"
  )) %>%
  #filter(row_number() == 1) %>%
  ungroup()

# Breakdown
composition %>%
  group_by(site_status, status) %>%
  summarise(n_distinct(asv))
```

```
## `summarise()` has grouped output by 'site_status'. You can override using the
## `.groups` argument.
```

```
## # A tibble: 6 x 3
## # Groups:   site_status [2]
##   site_status status     `n_distinct(asv)`
##   <chr>       <chr>                  <int>
## 1 ginger      Native                    22
## 2 ginger      Non-native                20
## 3 ginger      unknown                   85
## 4 native      Native                    15
## 5 native      Non-native                 4
## 6 native      unknown                   47
```

```
composition %>%
  group_by(id) %>%
  filter(row_number() == 1) %>%
  ungroup() %>%
  group_by(site_status, prey_comp) %>%
  summarise(n_distinct(id))
```

```
## `summarise()` has grouped output by 'site_status'. You can override using the
## `.groups` argument.
## # A tibble: 8 x 3
## # Groups:   site_status [2]
##   site_status prey_comp  `n_distinct(id)`
##   <chr>       <chr>                 <int>
## 1 ginger      Both                      9
## 2 ginger      Native                   21
## 3 ginger      Non-native               27
## 4 ginger      Unknown                  21
## 5 native      Both                      1
## 6 native      Native                   21
## 7 native      Non-native                3
## 8 native      Unknown                  33
```

Waffle plot

```
# Create waffle plot
waffling <-
  composition %>%
  group_by(id) %>%
  filter(row_number() == 1) %>%
  ungroup()

composition %>%
  group_by(site_status, prey_comp) %>%
  summarise(n_distinct(id))
```

```
## `summarise()` has grouped output by 'site_status'. You can override using the
## `.groups` argument.
```

```
## # A tibble: 8 x 3
## # Groups:   site_status [2]
##   site_status prey_comp  `n_distinct(id)`
##   <chr>       <chr>                 <int>
## 1 ginger      Both                      9
## 2 ginger      Native                   21
## 3 ginger      Non-native               27
## 4 ginger      Unknown                  21
## 5 native      Both                      1
## 6 native      Native                   21
## 7 native      Non-native                3
## 8 native      Unknown                  33
```

```
colors <- c("#7cbbc4", "white", "#12492f", "#f56038", "#dee0e6")

length(unique(waffling$id[waffling$site_status == "ginger"])) # 78
```

```
## [1] 78
```

```
ginger_grid <- expand.grid(y = 1:9, x = 1:9)
ginger_grid$cat <- c(sort(waffling$prey_comp[waffling$site_status == "ginger"]),
                     "empty", "empty", "empty")

waffle_1 <-
  ggplot(ginger_grid, aes(x = x, y = y, fill = cat)) +
  geom_tile(color = "white", linewidth = 1, alpha = 0.7) +
  scale_x_continuous(expand = c(0, 0)) +
  scale_y_continuous(expand = c(0, 0), trans = 'reverse') +
  scale_fill_manual(values = colors) +
  theme_transparent() +
  labs(fill = "Prey in diet")

length(unique(waffling$id[waffling$site_status == "native"])) # 58
```

```
## [1] 58
```

```
native_grid <- expand.grid(y = 1:9, x = 1:7)
native_grid$cat <- c(sort(waffling$prey_comp[waffling$site_status == "native"]),
                     rep("empty", 5))

waffle_2 <-
  ggplot(native_grid, aes( x = x, y = y, fill = cat)) +
  geom_tile(color = "white", linewidth = 1, alpha = 0.7) +
  scale_x_continuous(expand = c(0, 0)) +
  scale_y_continuous(expand = c(0, 0), trans = 'reverse') +
  scale_fill_manual(values = colors) +
  theme_transparent() +
  labs(fill = "Prey in diet")

waffle_combo <-
  ggarrange(waffle_1, waffle_2, common.legend = T,
          labels = c("Diet of spiders in ginger-invaded habitat",
                     "Diet of spiders in native forest"),
          vjust = -0.5, legend = "bottom")
```

Parasites

```
parasites <- parasites %>%
  mutate(type = ifelse(order == "Hymenoptera", "wasp", "fungi"))

# First, identify number of parasite ASVs and relative reads

parasite_summary <-
    parasites %>%
    acast(id ~ asv, fill = 0, value.var = "count") %>%
    as.matrix() %>%
    decostand(method = "hellinger") %>%
    melt() %>%
    filter(value != 0) %>%
    rename(id = Var1,
           asv = Var2,
          rel_read = value) %>%
    left_join(parasites, by = c("id", "asv")) %>%
    group_by(id, site_status, type) %>%
    summarise(n_asv = n_distinct(asv),
              n_read = sum(rel_read))

# Add in spiders that had no parasitism
parasite_summary <-
  prey_final %>%
  filter(!id %in% parasites$id) %>%
  group_by(id) %>%
  filter(row_number() == 1) %>%
  mutate(n_asv = 0,
         n_read = 0,
         type = NA) %>%
  select(id, site_status, type, n_asv, n_read) %>%
  rbind(parasite_summary) %>%
  mutate(site_status = tolower(site_status))

t.test(parasite_summary$n_asv[parasite_summary$site_status == "native"],
       parasite_summary$n_asv[parasite_summary$site_status == "ginger"])
```

```
##
##  Welch Two Sample t-test
##
## data:  parasite_summary$n_asv[parasite_summary$site_status == "native"] and parasite_summary$n_asv[pa
## t = -2.0666, df = 144.69, p-value = 0.04055
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.45063214 -0.01004607
## sample estimates:
## mean of x mean of y
## 0.3492063 0.5795455
```

```
parasite_summary %>%
  group_by(site_status) %>%
  summarise(mean(n_asv), plotrix::std.error(n_asv))
```

```
## # A tibble: 2 x 3
##   site_status `mean(n_asv)` `plotrix::std.error(n_asv)`
##   <chr>               <dbl>                       <dbl>
## 1 ginger              0.580                      0.0787
## 2 native              0.349                      0.0789
```

```
parasites %>%
  group_by(site_status, id) %>%
  filter(n_distinct(asv) > 1) %>%
  ungroup() %>%
  group_by(site_status) %>%
  summarise(n_distinct(id))
```

```
## # A tibble: 2 x 2
##   site_status `n_distinct(id)`
##   <chr>                  <int>
## 1 Ginger                    11
## 2 Native                     4
```

```
# Wasps
wasps <-
  parasite_summary %>%
  filter(type == "wasp" | is.na(type))

wasps %>%
  filter(!is.na(type)) %>%
  group_by(site_status) %>%
  summarise(n_distinct(id))
```

```
## # A tibble: 2 x 2
##   site_status `n_distinct(id)`
##   <chr>                  <int>
## 1 ginger                    17
## 2 native                    17
```

```
t.test(wasps$n_asv[wasps$site_status == "native"],
       wasps$n_asv[wasps$site_status == "ginger"])
```

```
##
##  Welch Two Sample t-test
##
## data:  wasps$n_asv[wasps$site_status == "native"] and wasps$n_asv[wasps$site_status == "ginger"]
## t = -0.49462, df = 122.13, p-value = 0.6218
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.3065861  0.1840055
## sample estimates:
## mean of x mean of y
## 0.3387097 0.4000000
```

```
# Fungi
fungi <-
  parasite_summary %>%
  filter(type == "fungi" | is.na(type))

t.test(parasite_summary$n_asv[parasite_summary$site_status == "native"],
       parasite_summary$n_asv[parasite_summary$site_status == "ginger"])
```

```
##
##  Welch Two Sample t-test
##
## data:  parasite_summary$n_asv[parasite_summary$site_status == "native"] and parasite_summary$n_asv[pa
## t = -2.0666, df = 144.69, p-value = 0.04055
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.45063214 -0.01004607
## sample estimates:
## mean of x mean of y
## 0.3492063 0.5795455
```

Figure

```r
# Figure
parasite_heatmap_wasp <-
  parasites %>%
  filter(order == "Hymenoptera" & percent_match >= 85) %>%
  mutate(site_status = ifelse(site_status == "Native", "Native forest", "Ginger-invaded")) %>%
  group_by(site_status, order, family) %>%
  summarise(n = n_distinct(id)) %>%
  acast(site_status ~ family, fill = 0) %>%
  as.matrix()
```

```
## `summarise()` has grouped output by 'site_status', 'order'. You can override
## using the `.groups` argument.
## Using n as value column: use value.var to override.
```

```r
parasite_heatmap_fungi <-
  parasites %>%
  filter(order != "Hymenoptera" & percent_match >= 85) %>%
  mutate(site_status = ifelse(site_status == "Native", "Native forest", "Ginger-invaded")) %>%
  group_by(site_status, order, family) %>%
  summarise(n = n_distinct(id)) %>%
  acast(site_status ~ family, fill = 0) %>%
  as.matrix()
```

```
## `summarise()` has grouped output by 'site_status', 'order'. You can override
## using the `.groups` argument.
## Using n as value column: use value.var to override.
```

```r
g <- colorRampPalette(c("#e8e3e3", "#420A25"))

parasite_wasp <-
  Heatmap(parasite_heatmap_wasp, column_names_side = "top",
        rect_gp = gpar(col = "black", lwd = 1),
        show_row_dend = F, show_column_dend = F,
        column_names_gp = gpar(fontsize = 8),
        col = c("white", g(12)),
        row_names_side = "left",
        cell_fun = function(j, i, x, y, w, h, col) { # add text to each grid
          grid.text(parasite_heatmap_wasp[i, j], x, y,
                    gp=gpar(fontsize=10, col="white", fontface="bold"))
        }, show_heatmap_legend = F)

parasite_fungi <-
  Heatmap(parasite_heatmap_fungi, column_names_side = "top",
        rect_gp = gpar(col = "black", lwd = 1),
        show_row_dend = F, show_column_dend = F,
        column_names_gp = gpar(fontsize = 8),
        col = c("white", g(3)),
        row_names_side = "left",
        cell_fun = function(j, i, x, y, w, h, col) { # add text to each grid
          grid.text(parasite_heatmap_fungi[i, j], x, y,
                    gp=gpar(fontsize=10, col="white", fontface="bold"))
        }, show_heatmap_legend = F)
```