March 2022 host and holobiont protein analysis

HM Putnam, AS Huffmyer

20220808

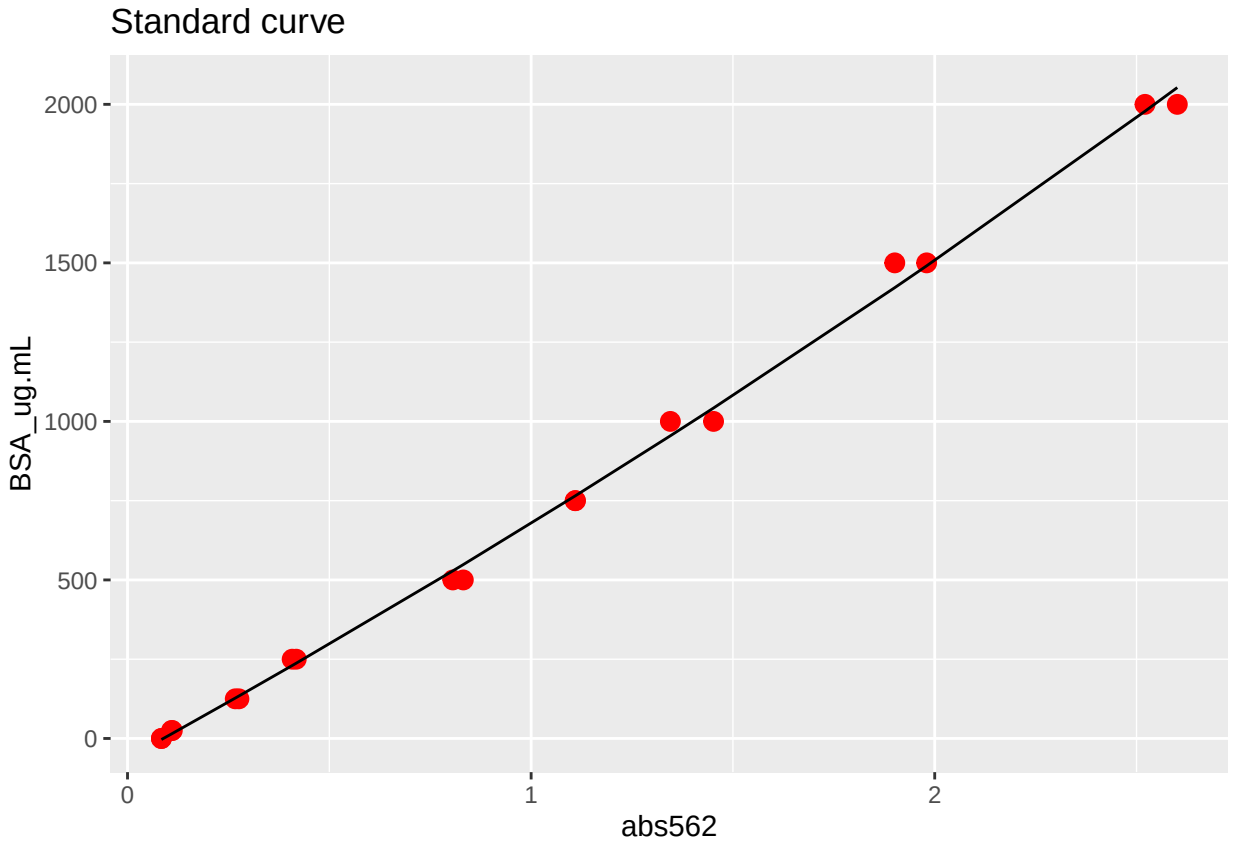
Import data for host protein from

```
# List protein data files
prot_path = "baseline_TPC_physio/data/TPC_total_protein/host/"
all_prot_files <- list.files(path = prot_path, pattern = "*.csv")          # List all files in director
prot_platemaps <- list.files(path = prot_path, pattern = "platemap")       # List platemap files
prot_data_files <- setdiff(all_prot_files, prot_platemaps)                 # List data files

# Read in all files into tibble
df_host <- tibble(file = prot_data_files) %>%
  separate(file, into = c("trip", "date", "plate"), remove = FALSE) %>%
  unite(plate, trip, date, plate) %>%
  dplyr::mutate(platemap = map(plate, ~read_csv(paste0(prot_path, ., "_platemap.csv"))),
         prot_data = map(file, ~read_csv(paste0(prot_path, .)) %>% rename(well = Well)))

# Merge platemap and data for each plate
df_host <- df_host %>%
  mutate(merged = map2(platemap, prot_data, ~ right_join(.x, .y)))
```

Plot standard curve

```
# Create standard curve following kit instructions
standards_host <- tribble(
  ~std, ~BSA_ug.mL,
  "A",        2000,
  "B",        1500,
  "C",        1000,
  "D",         750,
  "E",         500,
  "F",         250,
  "G",         125,
  "H",          25,
  "I",           0
)

std_curve_host <- df_host %>%
  unnest(merged) %>%
  filter(grepl("Standard", fragment_ID)) %>%
  select(plate, well, fragment_ID, abs562 = `562:562`) %>%
```

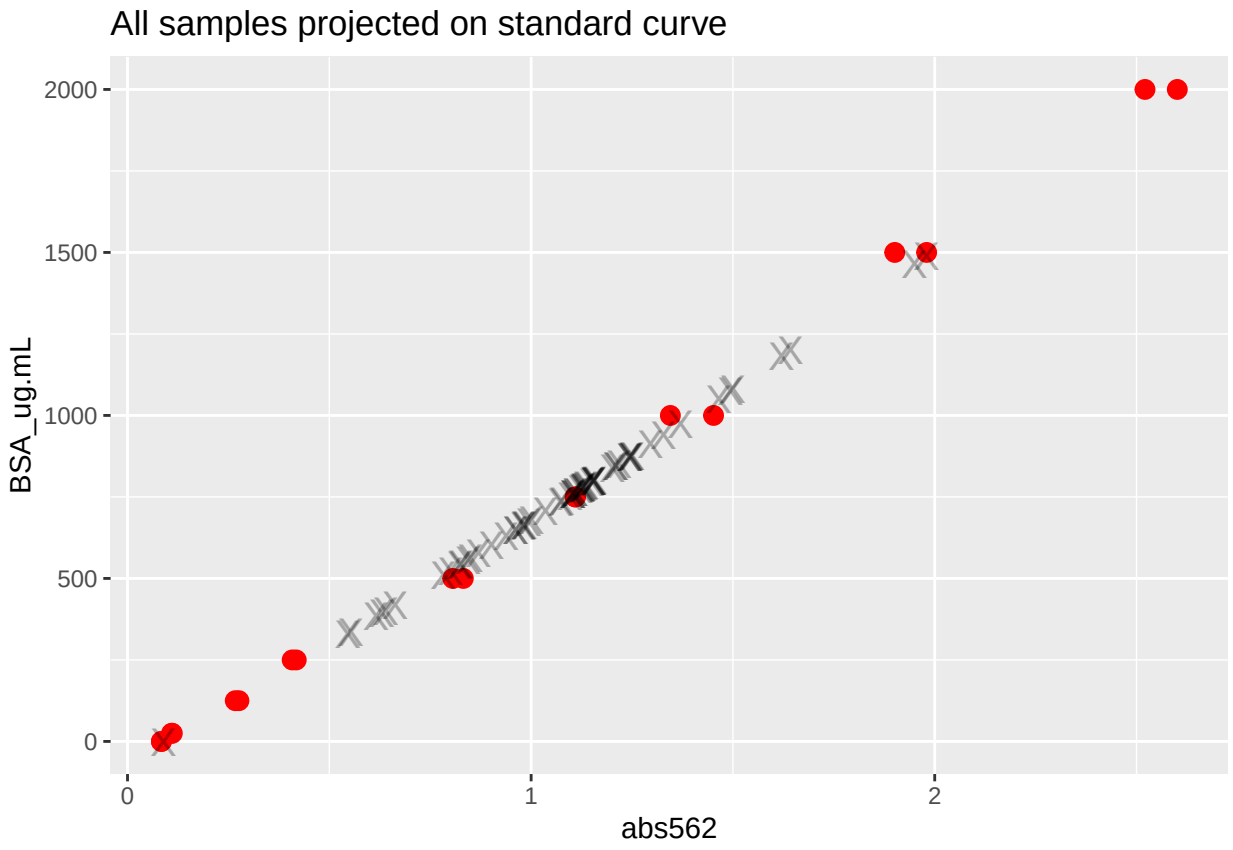
```
  rename(std = fragment_ID) %>%
  mutate(std = str_sub(std, 9, 9)) %>%
  #group_by(std) %>%
  #summarise(abs562 = mean(abs562)) %>%                    # calculate mean of standard duplicates
  #mutate(abs562.adj = abs562 - abs562[std == "I"]) %>%    # subtract blank absorbace value from all
  left_join(standards_host)


## Fit nonlinear model for standard curve
mod_host <- nls(formula = BSA_ug.mL ~ z + a * exp(b * abs562), start = list(z = 0, a = 1, b = 1), data =
fitted_host <- mod_host %>% broom::augment()

# Plot standard curve
std_curve_plot_host <- std_curve_host %>%
  ggplot(aes(x = abs562, y = BSA_ug.mL)) +
  geom_point(color = "red", size = 3)

std_curve_plot_host +
  geom_line(data = fitted_host, aes(x = abs562, y = .fitted)) +
  labs(title = "Standard curve")
```



Calculate protein concentrations

```
# Calculate protein concentration for all samples using standard curve
prot_host <- df_host %>%
```

```
  unnest(merged) %>%
  filter(!grepl("Standard", fragment_ID)) %>%                    # Get just samples (not standards)
  select(plate, well, fragment_ID, abs562 = `562:562`) %>%       # Select only needed columns
  filter(!is.na(fragment_ID)) %>%                                # Filter out empty wells
  filter(fragment_ID != "BK") %>%                                # Filter out blank wells
  mutate(prot_ug.mL = map_dbl(abs562, ~ predict(mod_host, newdata = data.frame(abs562 = .))))    # Use

std_curve_plot_host +
  geom_point(data = prot_host, aes(x = abs562, y = prot_ug.mL), pch = "X", cex = 5, alpha = 0.3) +
  labs(title = "All samples projected on standard curve")
```



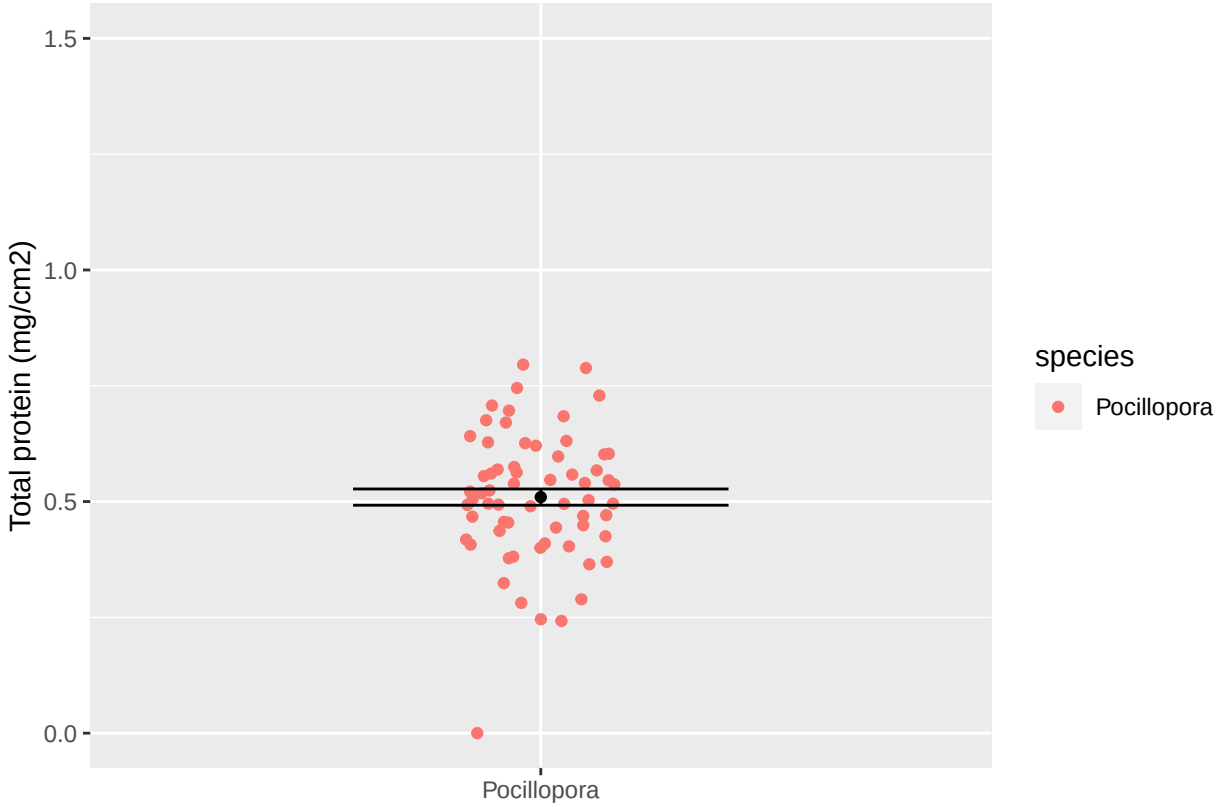
Normalize to surface area

```
# Surface area data
sa <- read.csv("baseline_TPC_physio/output/surface.area.calc.csv")

# Tissue homogenate volume data
homog_vols <- read_csv("baseline_TPC_physio/data/TPC_homogenate_vols/homogenate_vols.csv") %>% select(1

# Coral sample metadata
metadata <- read_csv("baseline_TPC_physio/metadata_POC_TPC.csv") %>% select(1:4)

# Join homogenate volumes and surface area with sample metadata
metadata <- full_join(metadata, homog_vols) %>%
```

```
  full_join(sa)

# Join prot data with metadata
prot_host <- left_join(prot_host, metadata) %>%
  mutate(prot_ug = prot_ug.mL * homog_vol_ml,
         prot_ug.cm2 = prot_ug / surface.area.cm2,
         prot_mg.cm2 = prot_ug.cm2 / 1000)
```

Plot results by species and site

```
# Plot all data points with mean ± se
prot_host %>%
  filter(!is.na(species)) %>%
  ggplot(aes(x = species, y = prot_mg.cm2, color = species)) +
  #facet_wrap(~species)  +
  coord_cartesian(ylim = c(0, 1.5))+
  labs(x = "", y = "Total protein (mg/cm2)") +
  geom_jitter(width = 0.1) +                                          # Plot all points
  stat_summary(fun.data = mean_cl_normal, fun.args = list(mult = 1),  # Plot standard error
               geom = "errorbar", color = "black", width = 0.5) +
  stat_summary(fun.y = mean, geom = "point", color = "black")         # Plot mean
```




```
# Remove outlier data points where absorbance was super low
prot_host <- prot_host %>%
  filter(abs562 > 0.1)
```

Write data to output file

```
# Write protein data to output file
prot_host %>%
  filter(!is.na(species)) %>%
  group_by(fragment_ID) %>%
  summarise(prot_ug = mean(prot_ug, na.rm = T),
            prot_ug.cm2 = mean(prot_ug.cm2, na.rm = T)) %>%
  select(fragment_ID, prot_ug, prot_ug.cm2) %>%
  write_csv(., path = "baseline_TPC_physio/output/host_protein.csv")
```

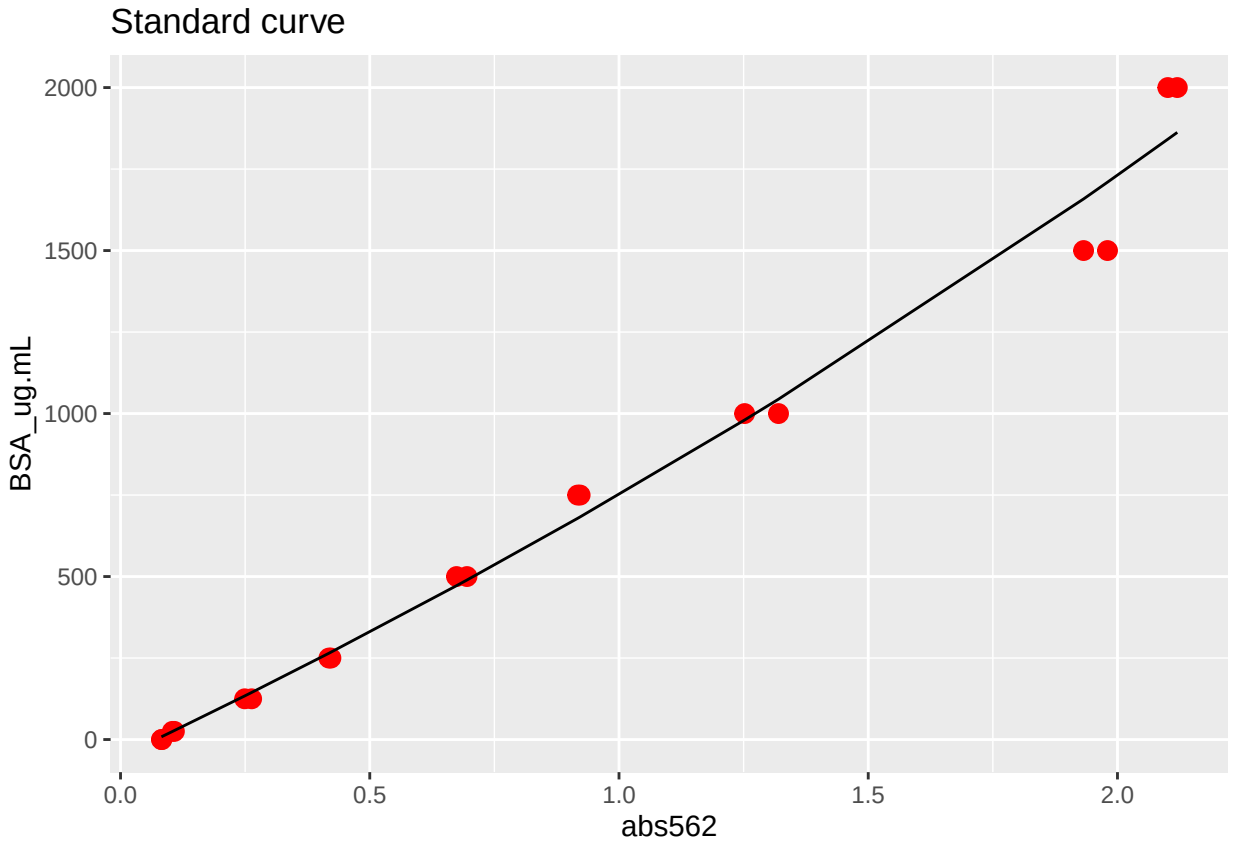
Import data for holobiont protein

```
# List protein data files
prot_path = "baseline_TPC_physio/data/TPC_total_protein/holobiont/"
all_prot_files <- list.files(path = prot_path, pattern = "*.csv")          # List all files in director
prot_platemaps <- list.files(path = prot_path, pattern = "platemap")       # List platemap files
prot_data_files <- setdiff(all_prot_files, prot_platemaps)                 # List data files

# Read in all files into tibble
df_holobiont <- tibble(file = prot_data_files) %>%
  separate(file, into = c("trip", "date", "plate"), remove = FALSE) %>%
  unite(plate, trip, date, plate) %>%
  dplyr::mutate(platemap = map(plate, ~read_csv(paste0(prot_path, ., "_platemap.csv"))),
         prot_data = map(file, ~read_csv(paste0(prot_path, .)) %>% rename(well = Well)))

# Merge platemap and data for each plate
df_holobiont <- df_holobiont %>%
  mutate(merged = map2(platemap, prot_data, ~ right_join(.x, .y)))
```

Plot standard curve

```
# Create standard curve following kit instructions
standards_holobiont <- tribble(
  ~std, ~BSA_ug.mL,
  "A",        2000,
  "B",        1500,
  "C",        1000,
  "D",         750,
  "E",         500,
  "F",         250,
  "G",         125,
  "H",          25,
  "I",           0
)

std_curve_holobiont <- df_holobiont %>%
  unnest(merged) %>%
  filter(grepl("Standard", fragment_ID)) %>%
```

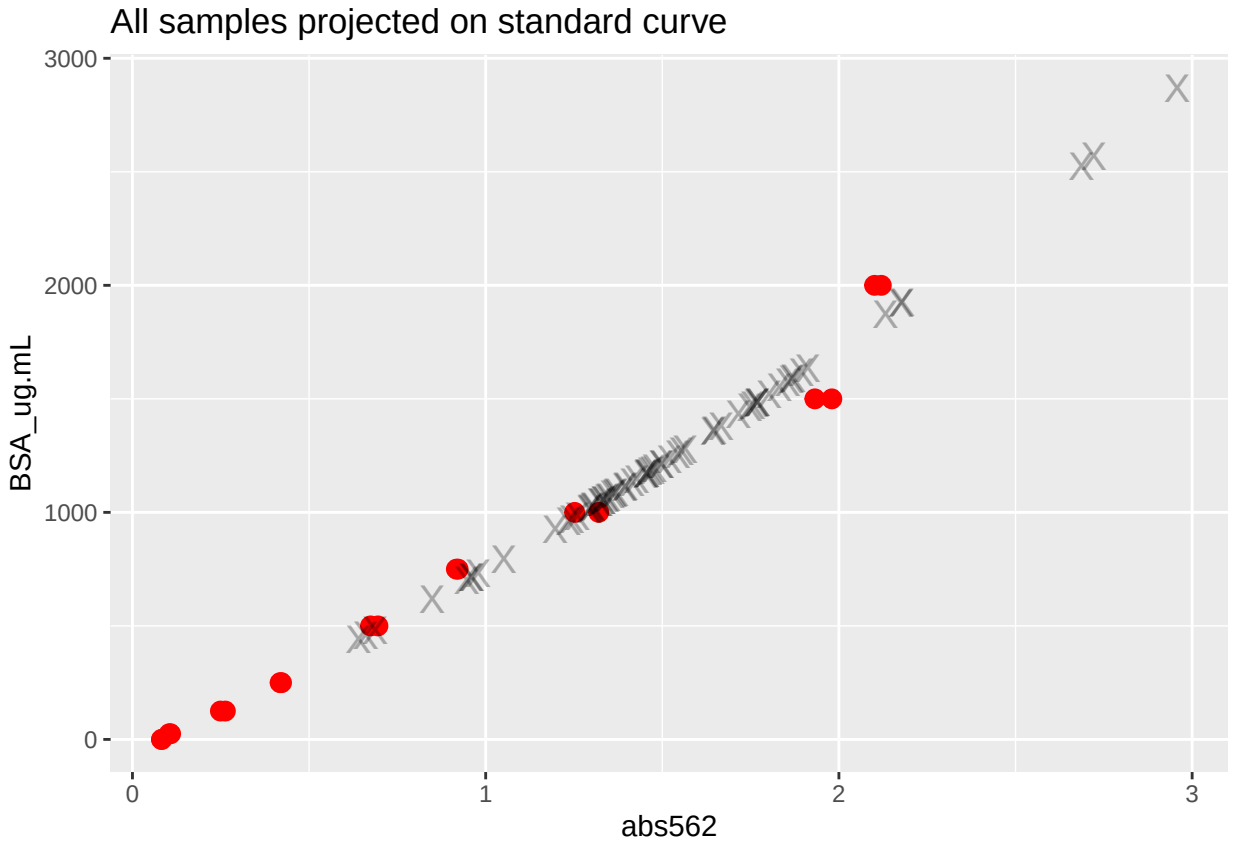
```
  select(plate, well, fragment_ID, abs562 = `562:562`) %>%
  rename(std = fragment_ID) %>%
  mutate(std = str_sub(std, 9, 9)) %>%
  #group_by(std) %>%
  #summarise(abs562 = mean(abs562)) %>%                    # calculate mean of standard duplicates
  #mutate(abs562.adj = abs562 - abs562[std == "I"]) %>%    # subtract blank absorbace value from all
  left_join(standards_holobiont)

## Fit nonlinear model for standard curve
mod_holobiont <- nls(formula = BSA_ug.mL ~ z + a * exp(b * abs562), start = list(z = 0, a = 1, b = 1), c
fitted_holobiont <- mod_holobiont %>% broom::augment()

# Plot standard curve
std_curve_plot_holobiont <- std_curve_holobiont %>%
  ggplot(aes(x = abs562, y = BSA_ug.mL)) +
  geom_point(color = "red", size = 3)

std_curve_plot_holobiont +
  geom_line(data = fitted_holobiont, aes(x = abs562, y = .fitted)) +
  labs(title = "Standard curve")
```



Calculate protein concentrations

```
# Calculate protein concentration for all samples using standard curve
prot_holobiont <- df_holobiont %>%
```

```
  unnest(merged) %>%
  filter(!grepl("Standard", fragment_ID)) %>%                    # Get just samples (not standards)
  select(plate, well, fragment_ID, abs562 = `562:562`) %>%       # Select only needed columns
  filter(!is.na(fragment_ID)) %>%                                # Filter out empty wells
  filter(fragment_ID != "BK") %>%                                # Filter out blank wells
  mutate(prot_ug.mL = map_dbl(abs562, ~ predict(mod_holobiont, newdata = data.frame(abs562 = .))))    #

std_curve_plot_holobiont +
  geom_point(data = prot_holobiont, aes(x = abs562, y = prot_ug.mL), pch = "X", cex = 5, alpha = 0.3) +
  labs(title = "All samples projected on standard curve")
```



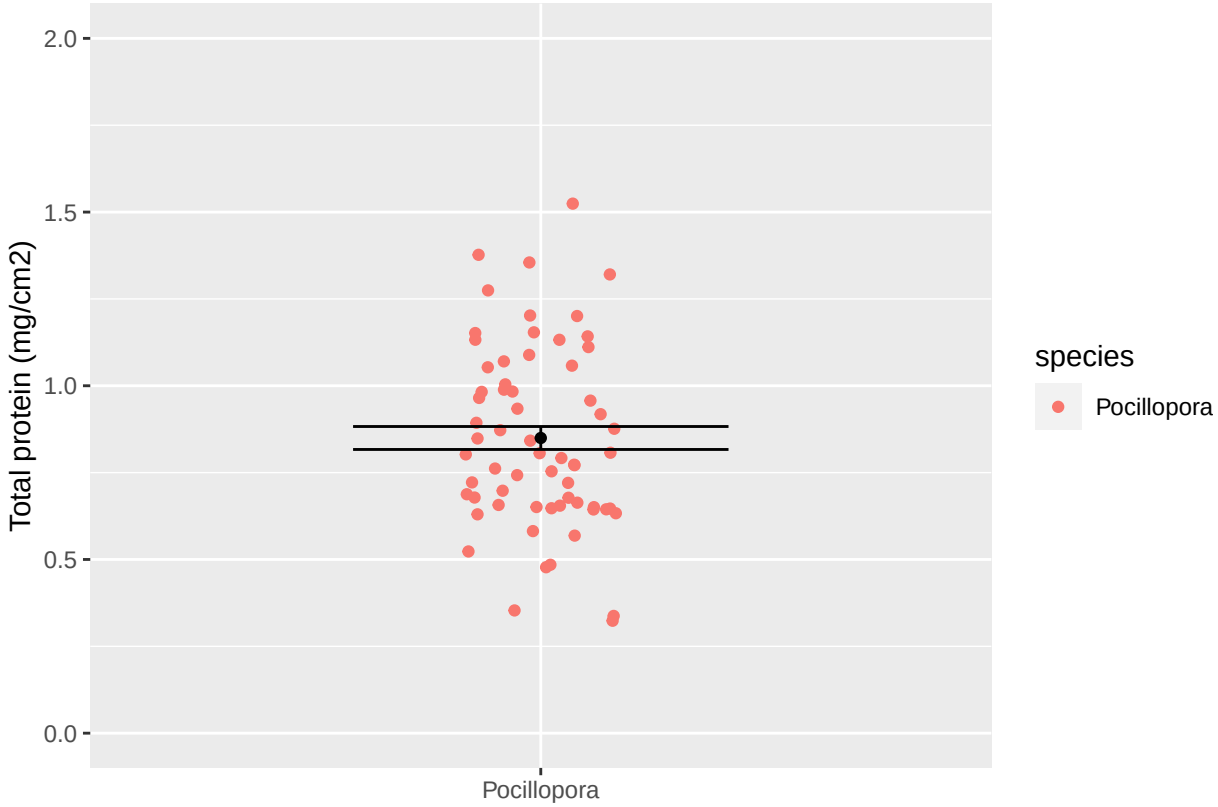
Normalize to surface area

```
# Surface area data
sa <- read.csv("baseline_TPC_physio/output/surface.area.calc.csv")

# Tissue homogenate volume data
homog_vols <- read_csv("baseline_TPC_physio/data/TPC_homogenate_vols/homogenate_vols.csv") %>% select(1

# Coral sample metadata
  metadata <- read_csv("baseline_TPC_physio/metadata_POC_TPC.csv") %>% select(1:4)

# Join homogenate volumes and surface area with sample metadata
metadata <- full_join(metadata, homog_vols) %>%
```

```
  full_join(sa)

# Join prot data with metadata
prot_holobiont <- left_join(prot_holobiont, metadata) %>%
  mutate(prot_ug = prot_ug.mL * homog_vol_ml,
         prot_ug.cm2 = prot_ug / surface.area.cm2,
         prot_mg.cm2 = prot_ug.cm2 / 1000)
```

Plot results by species and site

```
# Plot all data points with mean ± se
prot_holobiont %>%
  filter(!is.na(species)) %>%
  ggplot(aes(x = species, y = prot_mg.cm2, color = species)) +
  #facet_wrap(~species)  +
  coord_cartesian(ylim = c(0, 2))+
  labs(x = "", y = "Total protein (mg/cm2)") +
  geom_jitter(width = 0.1) +                                          # Plot all points
  stat_summary(fun.data = mean_cl_normal, fun.args = list(mult = 1),  # Plot standard error
               geom = "errorbar", color = "black", width = 0.5) +
  stat_summary(fun.y = mean, geom = "point", color = "black")         # Plot mean
```




```
# Remove outlier data points where absorbance was super low
prot_holobiont <- prot_holobiont %>%
  filter(abs562 > 0.1)
```

Write data to output file

```
# Write protein data to output file
prot_holobiont %>%
  filter(!is.na(species)) %>%
  group_by(fragment_ID) %>%
  summarise(prot_ug = mean(prot_ug, na.rm = T),
            prot_ug.cm2 = mean(prot_ug.cm2, na.rm = T)) %>%
  select(fragment_ID, prot_ug, prot_ug.cm2) %>%
  mutate(timepoint="MAY")%>%
  write_csv(., path = "baseline_TPC_physio/output/holobiont_protein.csv")
```