Kukula_Elwha_Exploratory

Kaitlyn Kukula

2025-11-13

NOTES

The goal of this study is to perform a general review of coastal organisms following the removal of two dams from the Elwha River in Washington. Some interesting questions to consider could be the use of eDNA for tracking fish movements or determining food chain transformations.

USGS Report (PDF in folder): chrome-extension://efaidnbmnnnibpcajpcglclefindmkaj/https://pubs.usgs.gov/sir/2011/5120/pdf/sir20115120.pdf

A general structure of the study could include: 1. System analysis: streamflow at available USGS gage site, estuary sediment transport 2. Comparing water quality metrics before and after dam removal Literature review may be required to put these values in context 3. Mapping and comparing species distribution/abundance before and after dam removal 4. Mapping and comparing eDNA and species distribution/abundance

Available data: - eDNA shedding from live and dead fish - Estuarian fish - Estuarian riparian veg abundance - Estuarian riparian veg richness - Estuarian sediment - Estuarian WQ - eDNA post-removal

Data we need to gather or find: - USGS Gage Station #12045500 - USGS Gage Station #12044900 - Watershed spatial data

Some maps I think would be useful: Map 1. Elwha River watershed including major tributaries, dams, USGS monitoring stations, and an inset map showing the general location in Washington Map 2-n. Sample collection sites for organism studies. This could be an interactive map. Map n+1: Might be interested in showing the spatial species distributions if we find anything of note? This could be an interactive map.

Some figures I think would be useful: - Streamflow at both sites as a function of time - Species comparisons before and after dam removal

```
library(ggplot2)
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v lubridate 1.9.4     v tibble    3.3.0
## v purrr     1.1.0     v tidyr     1.3.1
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(here)
```

```
## here() starts at /Users/kaitlynkukula/KukulaHipp
```

```
library(lubridate)
library(cowplot)
```

```
##
## Attaching package: 'cowplot'
##
## The following object is masked from 'package:lubridate':
##
##     stamp
```

```
here()
```

```
## [1] "/Users/kaitlynkukula/KukulaHipp"
```

```
kak_theme <- theme_classic() +
  theme(plot.title = element_text(size = 11, hjust = 0.5, face = "bold"), #Adjust title
        axis.text.x = element_text(size = 9), # Adjust x-axis values
        axis.text.y = element_text(size = 9), # Adjust y-axis values
        axis.title.x = element_text(size = 10.5, face = "bold"), # Adjust x-axis title
        axis.title.y = element_text(size = 10.5, face = "bold"), # Adjust y-axis title
        legend.position = "bottom", # Define legend position
        legend.text = element_text(size = 9), # Define legend entry sizes
        legend.title = element_text(size = 9)) # Define legend name sizes
set_theme(kak_theme)
```

Daily discharge data 1998 - 2025

```
discharge_12045500_raw <- read.csv(here("Data/Raw/daily_discharge_ 12045500.csv"))
discharge_12045500 <- discharge_12045500_raw %>%
  mutate(time = mdy(time),
         value = as.numeric(value),
         day = day(time),
         month = month(time),
         year = year(time))

# Plotting the daily discharge data from 2011 to 2020

ggplot(data = discharge_12045500 %>% filter(year > 2011 & year < 2020)) +
  geom_line(aes(x = time,y = value))
```

```
# Calculating the discharge over a set of years

year_start = 1998
year_end = 2024
```

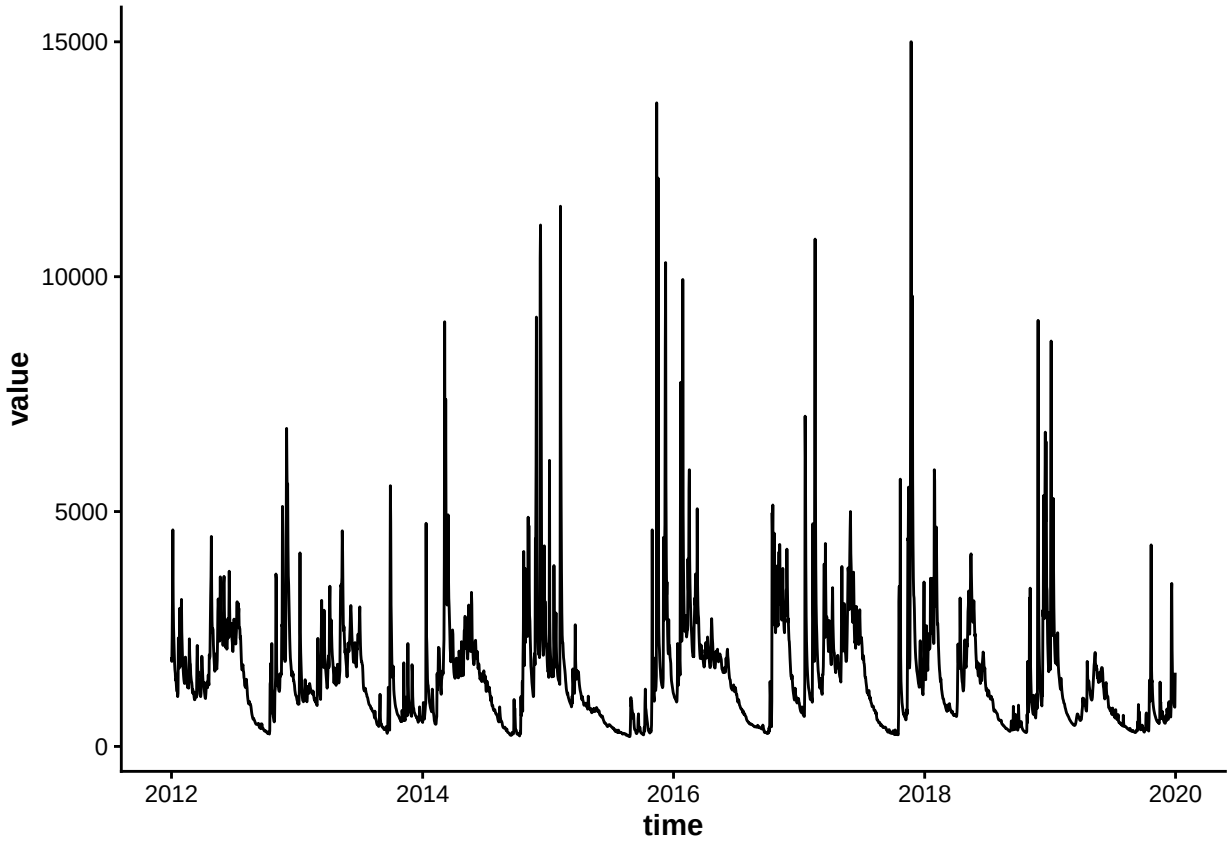

Figure 1: Daily Discharge of the Elwha River from 2011 - 2020 (gage station 12045500)

```r
discharge_12045500_mean <- discharge_12045500 %>%
  group_by(month) %>%
  filter(year > year_start & year < year_end) %>%
  summarise(mean_discharge = mean(value),
            min_discharge = min(value),
            max_discharge = max(value),
            q25_discharge = quantile(value, 0.25),
            q75_discharge = quantile(value, 0.75))

# Plotting discharge

ggplot(discharge_12045500_mean) +
  geom_point(aes(x = month, y = mean_discharge)) +
  geom_line(aes(x = month, y = mean_discharge)) +
  labs(x = "Month",
       y = "Discharge [ft^3/s]") +
  scale_x_continuous(breaks = 1:12,
                     labels = month.abb) +
  scale_y_continuous(breaks=seq(0,3000,500)) +
  geom_errorbar(aes(x = month,
                    y = mean_discharge,
                    ymin = q25_discharge,
                    ymax = q75_discharge),
                width = 0.2)
```

Daily gage data

Estuarian Invertebrates

This dataset contains observations of 20 estuarian invertebrate species at three sites for one timepoint before and one timepoint after the dam removal.

```r
inverts_raw <- read.csv(here("Data/Raw/Elwha_Estuary_Invertabrates/Elwha_estuary_aquatic_invertebrates_1
inverts <- inverts_raw %>%
  mutate(Date = mdy(Date))

colnames(inverts_raw)
```

```
##  [1] "Date"          "Dam.condition" "Site"          "Replicate"
##  [5] "Acarina"       "Amphipoda"     "Araneae"       "Coleoptera"
##  [9] "Collembola"    "Cyclopodia"    "Diptera"       "Gastropoda"
## [13] "Harpacticoida" "Hemiptera"     "Hirudinea"     "Hymenoptera"
## [17] "Megaloptera"   "Nematoda"      "Odonata"       "Oligochaeta"
## [21] "Ostracoda"     "Psocoptera"    "Thysanoptera"  "Trichoptera"
```

```r
unique(inverts_raw$Date)
```

```
## [1] "5/17/2007" "7/16/2007" "9/16/2007" "5/15/2013" "7/15/2013" "9/16/2013"
## [7] ""
```

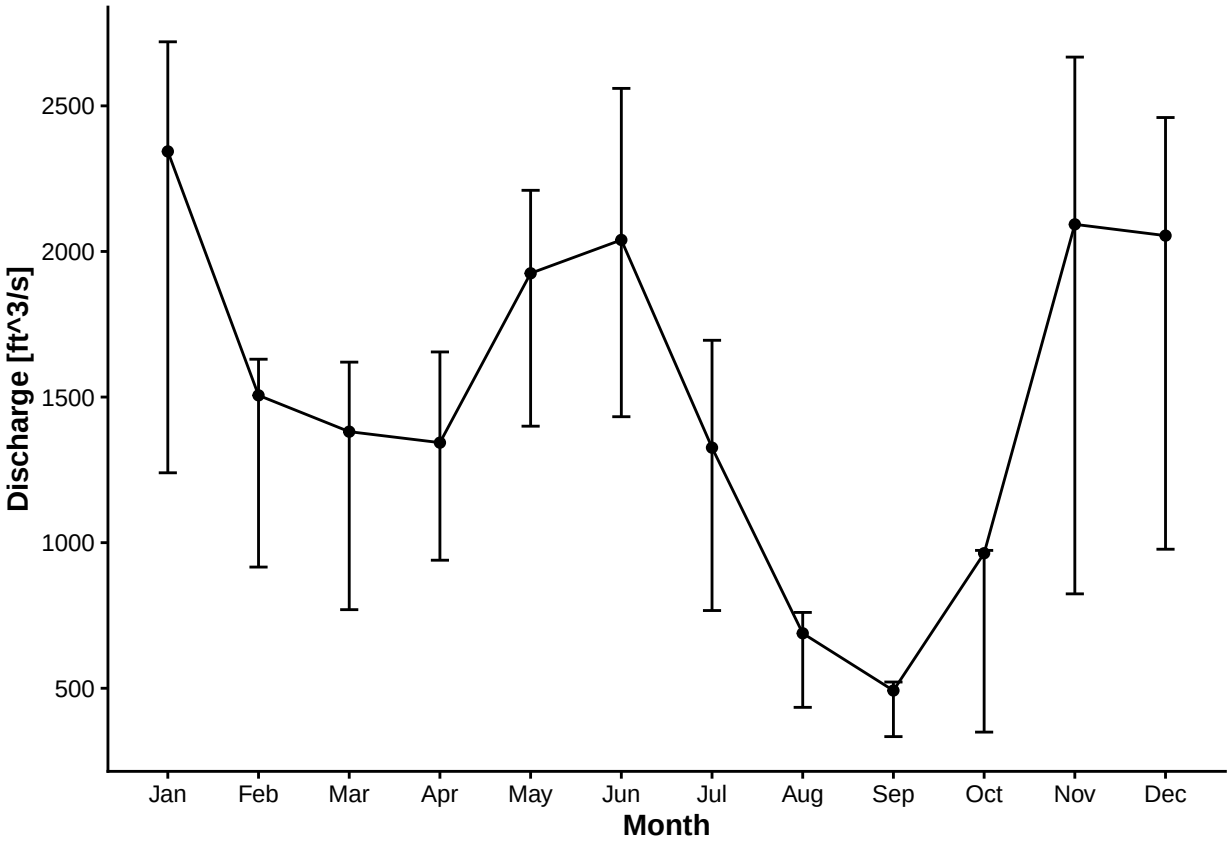

Figure 2: Mean Monthly Discharge of the Elwha River from 1998 - 2024 (gage station 12045500)

Subtidal Communities

How many species are observed at each site? Does this change over the years? What taxon (amphipods and dipterans and others - see page 187 of report) are present and how does this vary at the different sites?

```
subtidal_dive_raw <- read.csv(here("Data/Raw/Elwha_Subtidal_Communities/Elwha dive site data.csv"))
unique(subtidal_dive_raw$Site)
```

```
##  [1] "A1"  "A2"  "C1"  "C2"  "D1"  "D2"  "E1"  "E2"  "F1"  "F2"  "GP1" "GP2"
## [13] "H1"  "H2"  "J"   "K"   "L"
```

```
subtidal_temp_raw <- read.csv(here("Data/Raw/Elwha_Subtidal_Communities/Elwha temperature data.csv"))
subtidal_towed_raw <- read.csv(here("Data/Raw/Elwha_Subtidal_Communities/Elwha towed video data.csv"))
```

Estuary Fish Assemblages

Dates of observations: May - Sept of 2006, 2007, 2013, and 2014

Month vs count with each line representing a year and each plot representing a species

```
estuary_fish_raw <- read.csv("~/KukulaHipp/Data/Raw/Elwha_Estuary_Fish/Elwha estuary_fish_2006to2014.csv
view(estuary_fish_raw)

unique(estuary_fish_raw$Date.collected)
```

```
##  [1] "5/11/06" "6/9/06"  "7/20/06" "8/3/06"  "9/7/06"  "5/3/07"  "6/14/07"
##  [8] "7/26/07" "8/8/07"  "9/6/07"  "5/15/13" "6/5/13"  "7/10/13" "8/7/13"
## [15] "9/4/13"  "5/7/14"  "6/4/14"  "7/2/14"  "8/13/14" "9/10/14" "5/29/13"
## [22] "6/4/13"
```

Estuary water quality before and after dam removal - selecting variables for individual datasets

```
estuary_wq_raw <- read.csv("~/KukulaHipp/Data/Raw/Elwha_Estuary_WQ/Elwha_estuary_water_quality_2006-2014
  mutate(Date = mdy(Date.Collected))

# Nitrogen
estuary_wq_nitrate <- estuary_wq_raw %>%
  mutate(Nitrate = as.numeric(Nitrate...Nitrite.concentration)) %>%
  select(Site.Name:Longitude, Nitrate, Date)

# Phosphate
estuary_wq_phos <- estuary_wq_raw %>%
  mutate(Phosphate = as.numeric(Phosphate.concentration)) %>%
  select(Site.Name:Longitude, Phosphate, Date)

# Salinity
estuary_wq_salinity <- estuary_wq_raw %>%
  mutate(Salinity = as.numeric(Salinity)) %>%
  select(Site.Name:Longitude, Salinity, Date)
```

```
# Temperature
estuary_wq_temperature <- estuary_wq_raw %>%
  mutate(Temperature = as.numeric(Temperature)) %>%
  select(Site.Name:Longitude, Temperature, Date)

# Dissolved oxygen
estuary_wq_do <- estuary_wq_raw %>%
  mutate(Dissolved_Oxygen = as.numeric(Dissolved.oxygen)) %>%
  select(Site.Name:Longitude, Dissolved_Oxygen, Date)

# pH
estuary_wq_ph <- estuary_wq_raw %>%
  mutate(pH = as.numeric(pH)) %>%
  select(Site.Name:Longitude, pH, Date)

# Dataset legend (grabbed from nitrate subset)
estuary_wq_legend <- get_legend(
  ggplot(estuary_wq_nitrate %>% filter(Date >= "2007-08-30")) +
  geom_point(aes(x = Date, y = Nitrate, color = Site.Name)) +
    scale_color_manual(values = c("ES1" = "#1b9e77","ES2" = "#d95f02",
                                  "IEC" = "#7570b3", "WESC1" = "#e7298a",
                                  "WESC2" = "#66a61e")) +
    labs(color = "Site"))

# Dataset title
dataset_title <- ggplot() +
  labs(title = "Water Quality at the Elwha River Estuary Before and During Dam Removal",
       subtitle = "Dam removal: 2011 - 2014") +
  theme(plot.subtitle = element_text(hjust = 0.5))
```

Nitrate

```
# Create a before plot
nitrate_before_plot <- ggplot(estuary_wq_nitrate %>% filter(Date <= "2007-08-30")) +
  geom_point(aes(x = Date, y = Nitrate, color = Site.Name)) +
  geom_line(aes(x = Date, y = Nitrate, color = Site.Name), linewidth = 0.75) +
  scale_x_date(limits = c(as.Date("2006-06-01"), as.Date("2007-09-01")),
               breaks = "3 month", date_labels = "%b %Y") +
  ylim(c(0,20)) +
  theme(axis.text.x = element_text(angle = 45, hjust = 1),
        legend.position = "none") +
  labs(x = NULL, y = expression(paste("Nitrate [", mu, "M]"))) +
  scale_color_manual(values = c("ES1" = "#1b9e77", "ES2" = "#d95f02"))

# Create an after plot
nitrate_after_plot <- ggplot(estuary_wq_nitrate %>% filter(Date > "2007-08-30")) +
  geom_point(aes(x = Date, y = Nitrate, color = Site.Name)) +
  geom_line(aes(x = Date, y = Nitrate, color = Site.Name), linewidth = 0.75) +
  scale_x_date(limits = c(as.Date("2013-06-28"), as.Date("2014-10-12")),
               breaks = "3 month", date_labels = "%b %Y") +
```

```
  ylim(c(0,20)) +
  theme(axis.text.x = element_text(angle = 45, hjust = 1), legend.position = "none") +
  labs(x = NULL, y = NULL) +
  scale_color_manual(values = c("ES1" = "#1b9e77","ES2" = "#d95f02",
                                "IEC" = "#7570b3", "WESC1" = "#e7298a",
                                "WESC2" = "#66a61e"))
# Create a legend plot
nitrate_legend <- get_legend(
  ggplot(estuary_wq_nitrate %>% filter(Date >= "2007-08-30")) +
  geom_point(aes(x = Date, y = Nitrate, color = Site.Name)) +
    scale_color_manual(values = c("ES1" = "#1b9e77","ES2" = "#d95f02",
                                "IEC" = "#7570b3", "WESC1" = "#e7298a",
                                "WESC2" = "#66a61e")) +
    labs(color = "Site"))

# Create a window for the before and after plots
nitrate_plot <- plot_grid(nitrate_before_plot, nitrate_after_plot)

# Create a window for the title
nitrate_title <- ggplot() +
  labs(title = "Nitrate concentrations") +
  theme(plot.subtitle = element_text(hjust = 0.5))

# Generate the final plot with title, data, and legend
nitrate_plot_legend <- plot_grid(nitrate_title, nitrate_plot, nitrate_legend,
                                 nrow = 3, rel_heights = c(0.1, 0.8, 0.1))
# Display plot
nitrate_plot_legend
```

Phosphate

```
# Create a before plot
phos_before_plot <- ggplot(estuary_wq_phos %>% filter(Date <= "2007-08-30")) +
  geom_point(aes(x = Date, y = Phosphate, color = Site.Name)) +
  geom_line(aes(x = Date, y = Phosphate, color = Site.Name), linewidth = 0.75) +
  scale_x_date(limits = c(as.Date("2006-06-01"), as.Date("2007-09-01")),
               breaks = "3 month", date_labels = "%b %Y") +
  ylim(c(0,0.75)) +
  theme(axis.text.x = element_text(angle = 45, hjust = 1),
        legend.position = "none") +
  labs(x = NULL, y = expression(paste("Phosphate [", mu, "M]"))) +
  scale_color_manual(values = c("ES1" = "#1b9e77", "ES2" = "#d95f02"))

# Create an after plot
phos_after_plot <- ggplot(estuary_wq_phos %>% filter(Date > "2007-08-30")) +
  geom_point(aes(x = Date, y = Phosphate, color = Site.Name)) +
  geom_line(aes(x = Date, y = Phosphate, color = Site.Name), linewidth = 0.75) +
  scale_x_date(limits = c(as.Date("2013-06-28"), as.Date("2014-10-12")),
               breaks = "3 month", date_labels = "%b %Y") +
  ylim(c(0,0.75)) +
```

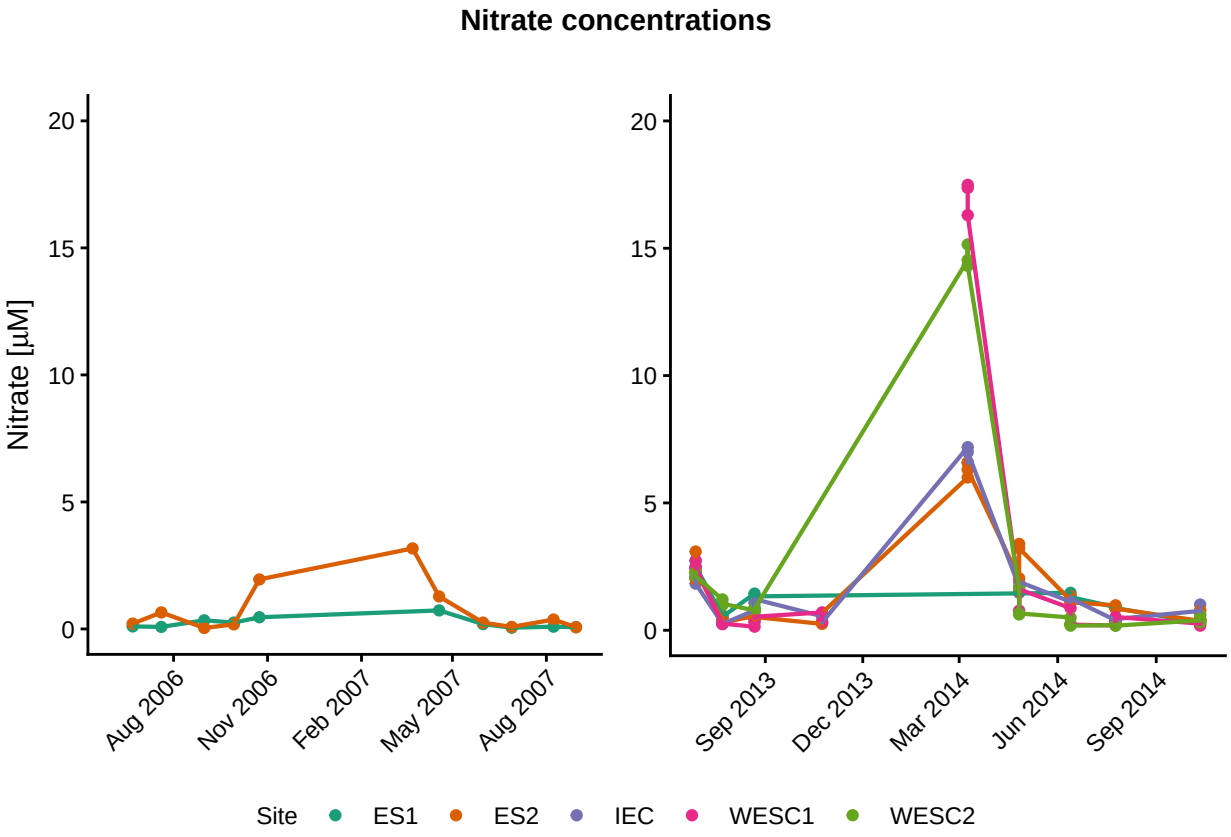

Figure 3: Measured nitrate concentration [uM] before and after dam removal

```
  theme(axis.text.x = element_text(angle = 45, hjust = 1), legend.position = "none") +
  labs(x = NULL, y = NULL) +
  scale_color_manual(values = c("ES1" = "#1b9e77","ES2" = "#d95f02",
                                "IEC" = "#7570b3", "WESC1" = "#e7298a",
                                "WESC2" = "#66a61e"))
# Create a legend plot
phos_legend <- get_legend(
  ggplot(estuary_wq_phos %>% filter(Date >= "2007-08-30")) +
  geom_point(aes(x = Date, y = Phosphate, color = Site.Name)) +
    scale_color_manual(values = c("ES1" = "#1b9e77","ES2" = "#d95f02",
                                "IEC" = "#7570b3", "WESC1" = "#e7298a",
                                "WESC2" = "#66a61e")) +
    labs(color = "Site"))
```

```
## Warning: Removed 2 rows containing missing values or values outside the scale range
## (`geom_point()`).
```

```
# Create a window for the before and after plots
phos_plot <- plot_grid(phos_before_plot, phos_after_plot)
```

```
## Warning: Removed 2 rows containing missing values or values outside the scale range
## (`geom_point()`).
```

```
# Create a window for the title
phos_title <- ggplot() +
  labs(title = "Phosphate concentrations") +
  theme(plot.subtitle = element_text(hjust = 0.5))

# Generate the final plot with title, data, and legend
phos_plot_legend <- plot_grid(phos_title, phos_plot, phos_legend,
                              nrow = 3, rel_heights = c(0.1, 0.8, 0.1))
# Display plot
phos_plot_legend
```

Salinity

```
# Create a before plot
salinity_before_plot <- ggplot(estuary_wq_salinity %>% filter(Date <= "2007-08-30")) +
  geom_point(aes(x = Date, y = Salinity, color = Site.Name)) +
  geom_line(aes(x = Date, y = Salinity, color = Site.Name), linewidth = 0.75) +
  scale_x_date(limits = c(as.Date("2006-06-01"), as.Date("2007-09-01")),
               breaks = "3 month", date_labels = "%b %Y") +
  ylim(c(0,0.75)) +
  theme(axis.text.x = element_text(angle = 45, hjust = 1),
        legend.position = "none") +
  labs(x = NULL, y = expression(paste("Salinity [units]"))) +
  scale_color_manual(values = c("ES1" = "#1b9e77", "ES2" = "#d95f02"))

# Create an after plot
```

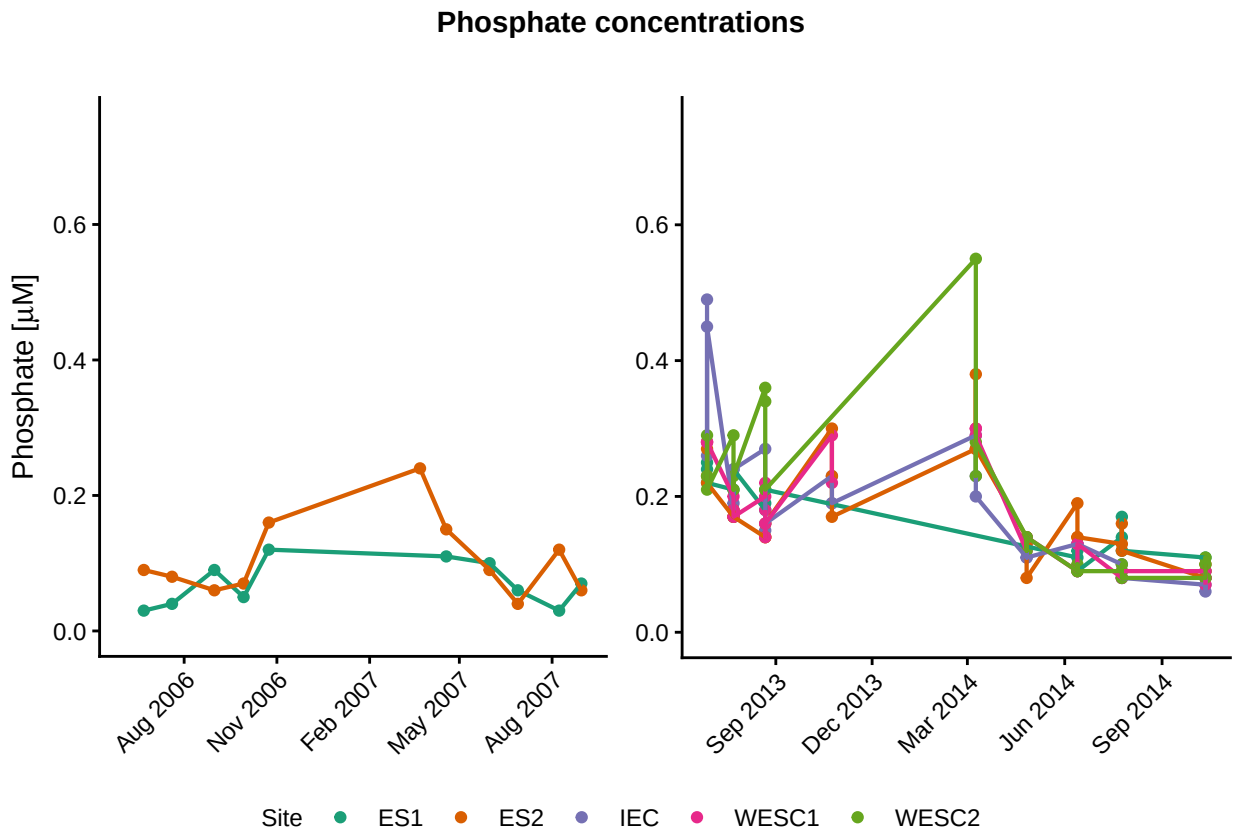

Figure 4: Measured phosphate [units] before and after dam removal

```
salinity_after_plot <- ggplot(estuary_wq_salinity %>% filter(Date > "2007-08-30")) +
  geom_point(aes(x = Date, y = Salinity, color = Site.Name)) +
  geom_line(aes(x = Date, y = Salinity, color = Site.Name), linewidth = 0.75) +
  scale_x_date(limits = c(as.Date("2013-06-28"), as.Date("2014-10-12")),
               breaks = "3 month", date_labels = "%b %Y") +
  ylim(c(0,0.75)) +
  theme(axis.text.x = element_text(angle = 45, hjust = 1), legend.position = "none") +
  labs(x = NULL, y = NULL) +
  scale_color_manual(values = c("ES1" = "#1b9e77","ES2" = "#d95f02",
                                "IEC" = "#7570b3", "WESC1" = "#e7298a",
                                "WESC2" = "#66a61e"))
# Create a legend plot
salinity_legend <- get_legend(
  ggplot(estuary_wq_salinity %>% filter(Date >= "2007-08-30")) +
  geom_point(aes(x = Date, y = Salinity, color = Site.Name)) +
    scale_color_manual(values = c("ES1" = "#1b9e77","ES2" = "#d95f02",
                                "IEC" = "#7570b3", "WESC1" = "#e7298a",
                                "WESC2" = "#66a61e")) +
    labs(color = "Site"))
```

```
## Warning: Removed 8 rows containing missing values or values outside the scale range
## (`geom_point()`).
```

```
# Create a window for the before and after plots
salinity_plot <- plot_grid(salinity_before_plot, salinity_after_plot)
```

```
## Warning: Removed 8 rows containing missing values or values outside the scale range
## (`geom_point()`).

## Warning: Removed 2 rows containing missing values or values outside the scale range
## (`geom_line()`).

## Warning: Removed 8 rows containing missing values or values outside the scale range
## (`geom_point()`).
```

```
# Create a window for the title
salinity_title <- ggplot() +
  labs(title = "Salinity") +
  theme(plot.subtitle = element_text(hjust = 0.5))

# Generate the final plot with title, data, and legend
salinity_plot_legend <- plot_grid(salinity_title, salinity_plot, salinity_legend,
                                  nrow = 3, rel_heights = c(0.1, 0.8, 0.1))
# Display plot
salinity_plot_legend
```

Temperature

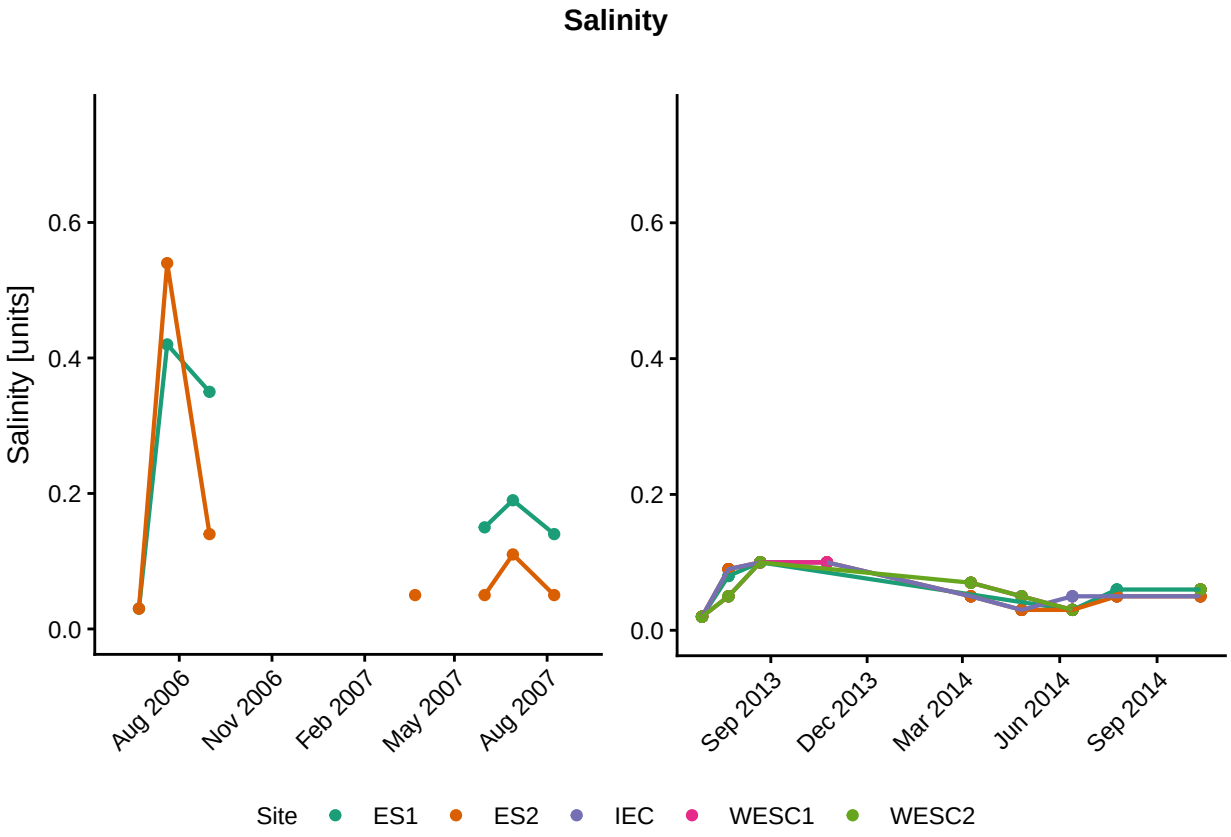

Figure 5: Measured salinity [units] before and after dam removal

```
# Create a before plot
temp_before_plot <- ggplot(estuary_wq_temperature %>% filter(Date <= "2007-08-30")) +
  geom_point(aes(x = Date, y = Temperature, color = Site.Name)) +
  geom_line(aes(x = Date, y = Temperature, color = Site.Name), linewidth = 0.75) +
  scale_x_date(limits = c(as.Date("2006-06-01"), as.Date("2007-09-01")),
               breaks = "3 month", date_labels = "%b %Y") +
  ylim(c(0,15)) +
  theme(axis.text.x = element_text(angle = 45, hjust = 1),
        legend.position = "none") +
  labs(x = NULL, y = expression(paste("Salinity [units]"))) +
  scale_color_manual(values = c("ES1" = "#1b9e77", "ES2" = "#d95f02"))

# Create an after plot
temp_after_plot <- ggplot(estuary_wq_temperature %>% filter(Date > "2007-08-30")) +
  geom_point(aes(x = Date, y = Temperature, color = Site.Name)) +
  geom_line(aes(x = Date, y = Temperature, color = Site.Name), linewidth = 0.75) +
  scale_x_date(limits = c(as.Date("2013-06-28"), as.Date("2014-10-12")),
               breaks = "3 month", date_labels = "%b %Y") +
  ylim(c(0,15)) +
  theme(axis.text.x = element_text(angle = 45, hjust = 1), legend.position = "none") +
  labs(x = NULL, y = NULL) +
  scale_color_manual(values = c("ES1" = "#1b9e77","ES2" = "#d95f02",
                                "IEC" = "#7570b3", "WESC1" = "#e7298a",
                                "WESC2" = "#66a61e"))
# Create a legend plot
temp_legend <- get_legend(
  ggplot(estuary_wq_temperature %>% filter(Date >= "2007-08-30")) +
  geom_point(aes(x = Date, y = Temperature, color = Site.Name)) +
    scale_color_manual(values = c("ES1" = "#1b9e77","ES2" = "#d95f02",
                                "IEC" = "#7570b3", "WESC1" = "#e7298a",
                                "WESC2" = "#66a61e")) +
    labs(color = "Site"))
```

```
## Warning: Removed 8 rows containing missing values or values outside the scale range
## (`geom_point()`).
```

```
# Create a window for the before and after plots
temp_plot <- plot_grid(temp_before_plot, temp_after_plot)
```

```
## Warning: Removed 2 rows containing missing values or values outside the scale range
## (`geom_point()`).
```

```
## Warning: Removed 43 rows containing missing values or values outside the scale range
## (`geom_point()`).
```

```
# Create a window for the title
temp_title <- ggplot() +
  labs(title = "Temperature") +
  theme(plot.subtitle = element_text(hjust = 0.5))

# Generate the final plot with title, data, and legend
```

```
temp_plot_legend <- plot_grid(temp_title, temp_plot, temp_legend,
                              nrow = 3, rel_heights = c(0.1, 0.8, 0.1))
# Display plot
temp_plot_legend
```

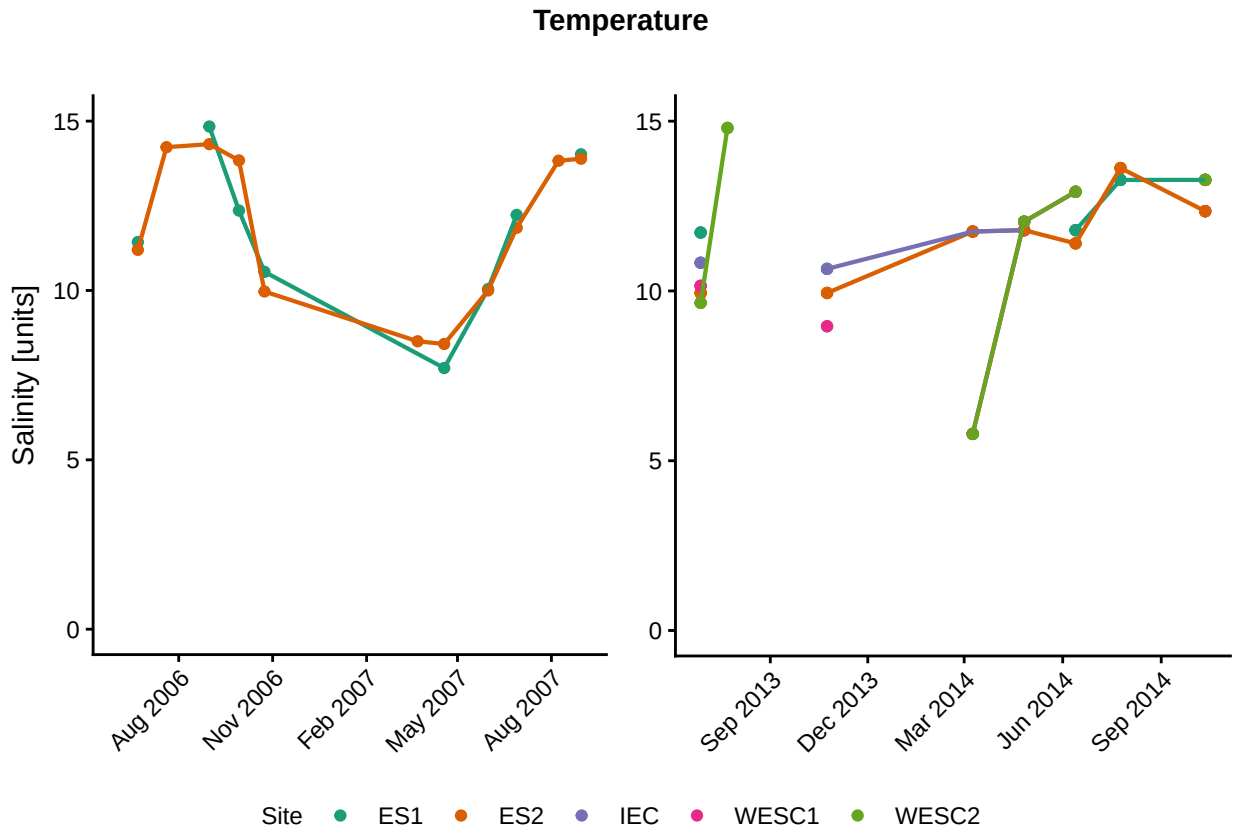

Figure 6: Measured temperature [units] before and after dam removal

Dissolved oxygen

```
# Create a before plot
do_before_plot <- ggplot(estuary_wq_do %>% filter(Date <= "2007-08-30")) +
  geom_point(aes(x = Date, y = Dissolved_Oxygen, color = Site.Name)) +
  geom_line(aes(x = Date, y = Dissolved_Oxygen, color = Site.Name), linewidth = 0.75) +
  scale_x_date(limits = c(as.Date("2006-06-01"), as.Date("2007-09-01")),
               breaks = "3 month", date_labels = "%b %Y") +
  ylim(c(0,15)) +
  theme(axis.text.x = element_text(angle = 45, hjust = 1),
        legend.position = "none") +
  labs(x = NULL, y = expression(paste("Dissolved Oxygen [units]"))) +
  scale_color_manual(values = c("ES1" = "#1b9e77", "ES2" = "#d95f02"))
```

```r
# Create an after plot
do_after_plot <- ggplot(estuary_wq_do %>% filter(Date > "2007-08-30")) +
  geom_point(aes(x = Date, y = Dissolved_Oxygen, color = Site.Name)) +
  geom_line(aes(x = Date, y = Dissolved_Oxygen, color = Site.Name), linewidth = 0.75) +
  scale_x_date(limits = c(as.Date("2013-06-28"), as.Date("2014-10-12")),
               breaks = "3 month", date_labels = "%b %Y") +
  ylim(c(0,15)) +
  theme(axis.text.x = element_text(angle = 45, hjust = 1), legend.position = "none") +
  labs(x = NULL, y = NULL) +
  scale_color_manual(values = c("ES1" = "#1b9e77","ES2" = "#d95f02",
                                "IEC" = "#7570b3", "WESC1" = "#e7298a",
                                "WESC2" = "#66a61e"))
# Create a legend plot
do_legend <- get_legend(
  ggplot(estuary_wq_do %>% filter(Date >= "2007-08-30")) +
  geom_point(aes(x = Date, y = Dissolved_Oxygen, color = Site.Name)) +
    scale_color_manual(values = c("ES1" = "#1b9e77","ES2" = "#d95f02",
                                "IEC" = "#7570b3", "WESC1" = "#e7298a",
                                "WESC2" = "#66a61e")) +
    labs(color = "Site"))
```

```
## Warning: Removed 26 rows containing missing values or values outside the scale range
## (`geom_point()`).
```

```r
# Create a window for the before and after plots
do_plot <- plot_grid(do_before_plot, do_after_plot)
```

```
## Warning: Removed 26 rows containing missing values or values outside the scale range
## (`geom_point()`).

## Warning: Removed 12 rows containing missing values or values outside the scale range
## (`geom_line()`).
```

```r
# Create a window for the title
do_title <- ggplot() +
  labs(title = "Dissolved Oxygen") +
  theme(plot.subtitle = element_text(hjust = 0.5))

# Generate the final plot with title, data, and legend
do_plot_legend <- plot_grid(do_title, do_plot, do_legend,
                                nrow = 3, rel_heights = c(0.1, 0.8, 0.1))
# Display plot
do_plot_legend
```

pH

```r
# Create a before plot
ph_before_plot <- ggplot(estuary_wq_ph %>% filter(Date <= "2007-08-30")) +
```

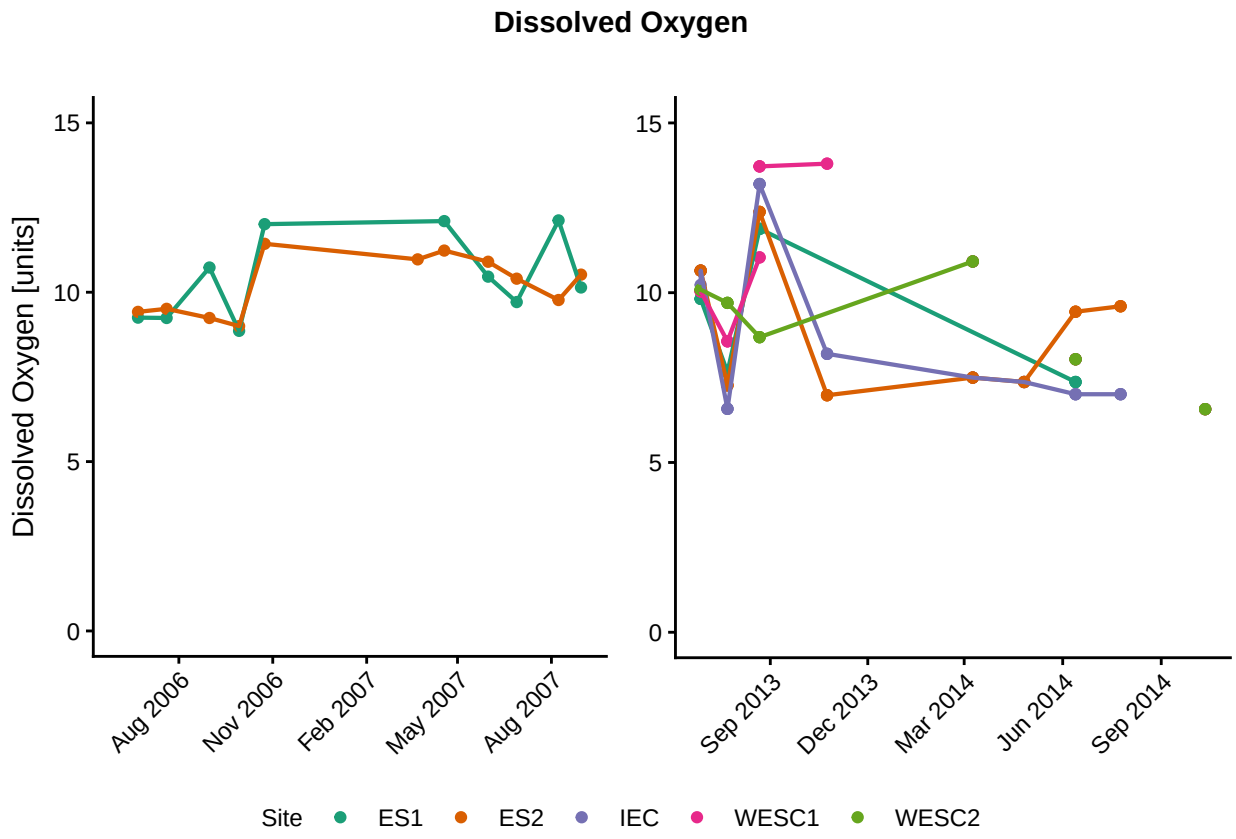

Figure 7: Measured dissolved oxygen [units] before and after dam removal

```
  geom_point(aes(x = Date, y = pH, color = Site.Name)) +
  geom_line(aes(x = Date, y = pH, color = Site.Name), linewidth = 0.75) +
  scale_x_date(limits = c(as.Date("2006-06-01"), as.Date("2007-09-01")),
               breaks = "3 month", date_labels = "%b %Y") +
  ylim(c(0,10)) +
  theme(axis.text.x = element_text(angle = 45, hjust = 1),
        legend.position = "none") +
  labs(x = NULL, y = expression(paste("Dissolved Oxygen [units]"))) +
  scale_color_manual(values = c("ES1" = "#1b9e77", "ES2" = "#d95f02"))

# Create an after plot
ph_after_plot <- ggplot(estuary_wq_ph %>% filter(Date > "2007-08-30")) +
  geom_point(aes(x = Date, y = pH, color = Site.Name)) +
  geom_line(aes(x = Date, y = pH, color = Site.Name), linewidth = 0.75) +
  scale_x_date(limits = c(as.Date("2013-06-28"), as.Date("2014-10-12")),
               breaks = "3 month", date_labels = "%b %Y") +
  ylim(c(0,10)) +
  theme(axis.text.x = element_text(angle = 45, hjust = 1), legend.position = "none") +
  labs(x = NULL, y = NULL) +
  scale_color_manual(values = c("ES1" = "#1b9e77","ES2" = "#d95f02",
                                "IEC" = "#7570b3", "WESC1" = "#e7298a",
                                "WESC2" = "#66a61e"))
# Create a legend plot
ph_legend <- get_legend(
  ggplot(estuary_wq_ph %>% filter(Date >= "2007-08-30")) +
  geom_point(aes(x = Date, y = pH, color = Site.Name)) +
    scale_color_manual(values = c("ES1" = "#1b9e77","ES2" = "#d95f02",
                                "IEC" = "#7570b3", "WESC1" = "#e7298a",
                                "WESC2" = "#66a61e")) +
    labs(color = "Site"))
```

```
## Warning: Removed 23 rows containing missing values or values outside the scale range
## (`geom_point()`).
```

```
# Create a window for the before and after plots
ph_plot <- plot_grid(ph_before_plot, ph_after_plot)
```

```
## Warning: Removed 2 rows containing missing values or values outside the scale range
## (`geom_point()`).
## Removed 23 rows containing missing values or values outside the scale range
## (`geom_point()`).

## Warning: Removed 6 rows containing missing values or values outside the scale range
## (`geom_line()`).
```

```
# Create a window for the title
ph_title <- ggplot() +
  labs(title = "pH") +
  theme(plot.subtitle = element_text(hjust = 0.5))

# Generate the final plot with title, data, and legend
ph_plot_legend <- plot_grid(ph_title, ph_plot, ph_legend,
```

```
                                nrow = 3, rel_heights = c(0.1, 0.8, 0.1))
# Display plot
ph_plot_legend
```

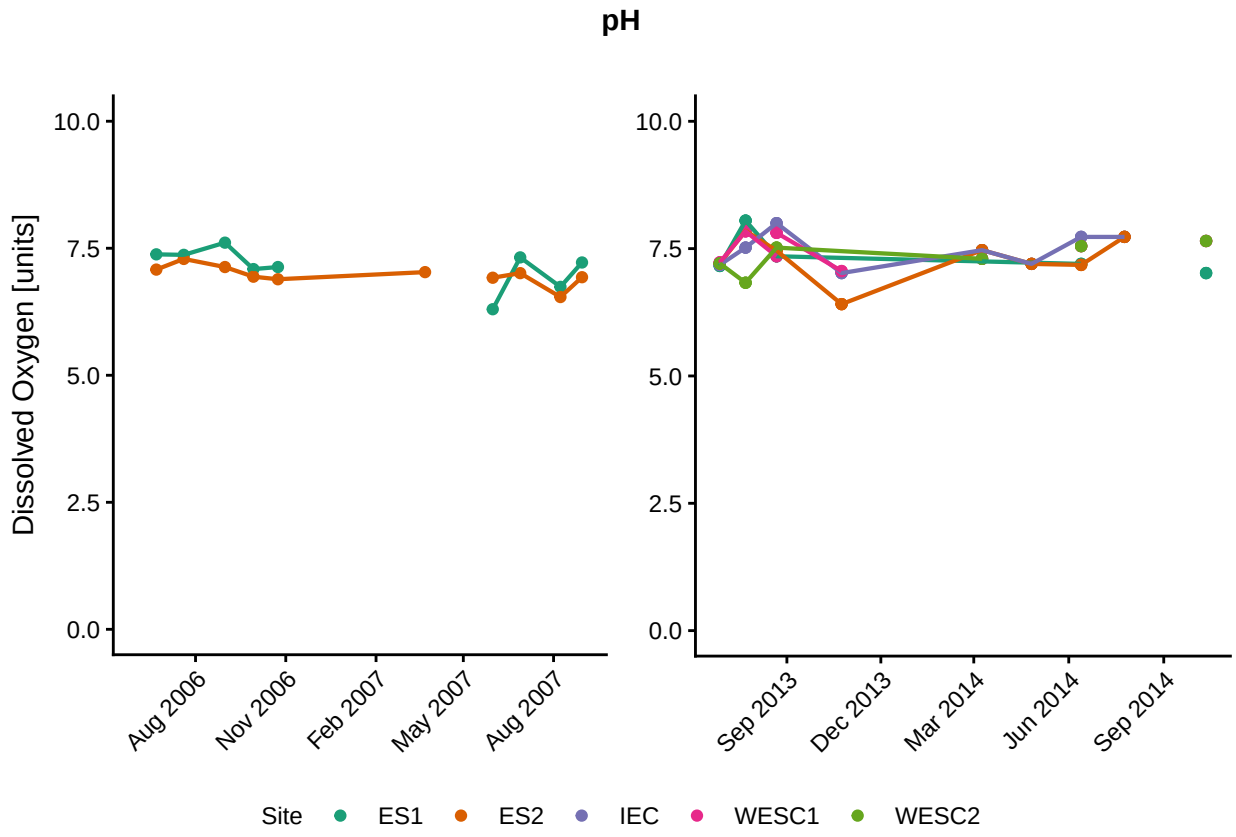

Figure 8: Measured pH [units] before and after dam removal

Stacked plots

```
# Final water quality plot
estuary_wq_plot <- plot_grid(dataset_title,
                             nitrate_title, nitrate_plot,
                             phos_title, phos_plot,
                             temp_title, temp_plot,
                             do_title, do_plot,
                             ph_title, ph_plot,
                             estuary_wq_legend,
                             nrow = 12, rel_heights = c(2,1,7,1,7,1,7,1,7,1,7,1,7,1))
estuary_wq_plot
```

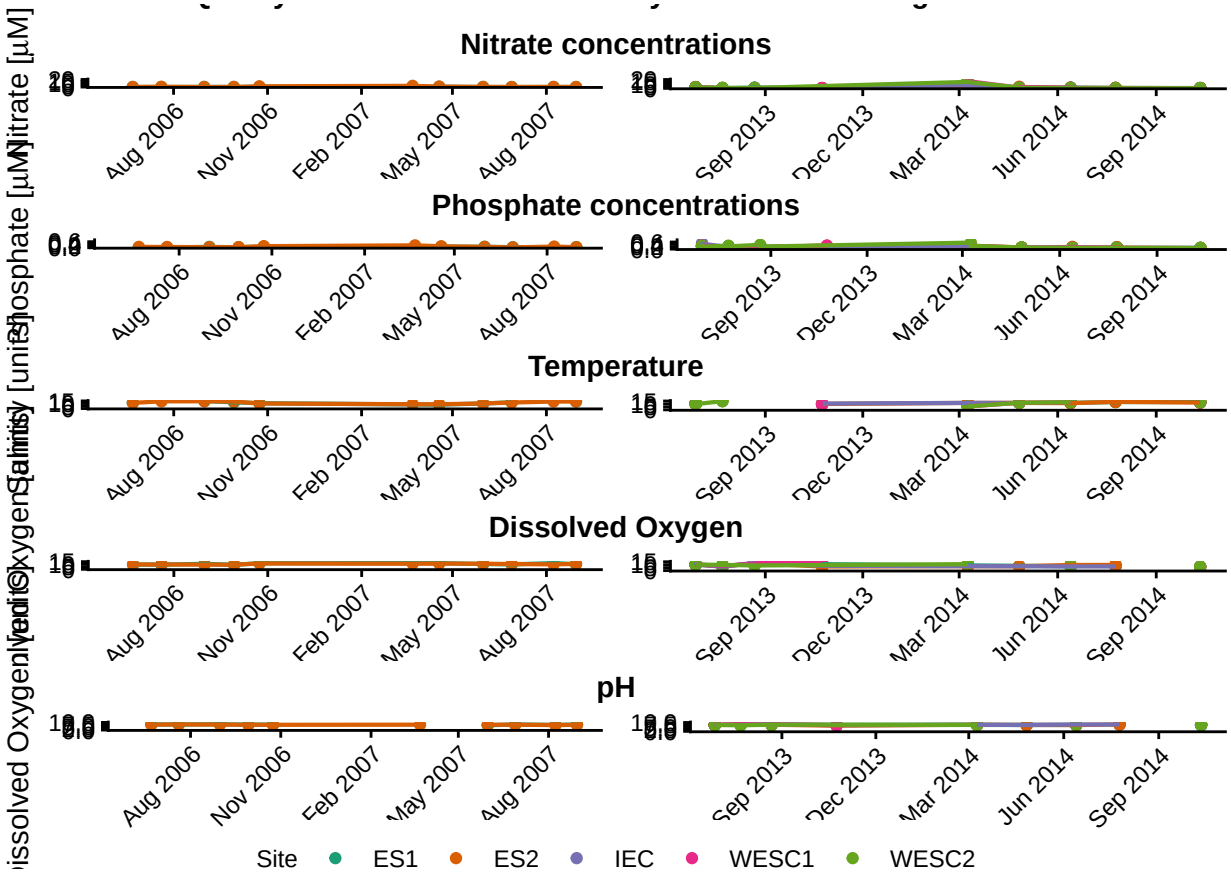

Figure 9: Water quality in the Elwha River estuary before and during dam removal

Salmonid distribution and abundance

Dates of observations: 2007, 2008, 2018, 2019

Could do some sort of bubble map with this dataset to show where more fish are getting to. Remove columns that have limited data

```
salmonid_raw <- read.csv("~/KukulaHipp/Data/Raw/Elwha_Salmon_Movement/Elwha_Riverscape_reach_data.csv")
unique(salmonid_raw$Year)
```

```
## [1] 2007 2008 2018 2019
```