

R_swim_path_plots

Catarina Pien

2024-02-08

Load example fish data

```
smelt_data_join <- readRDS(here("output/smelt_data_example.rds")) %>%
  mutate(Release = case_when(
    # Tag == "RBP" ~ "11/30 Soft Release",
    Tag == "LRA" ~ "11/30 Hard Release",
    Tag == "ROP" ~ "01/18 Hard Release",
    Tag == "ad" ~ "01/18 Trailer Release",
    Tag == "RGP" ~ "01/26 Soft Release",
    Tag == "LOA" ~ "01/25 Hard Release"),
    Release = factor(Release, levels = c("11/30 Hard Release",
                                         # "11/30 Soft Release",
                                         "01/18 Hard Release",
                                         "01/18 Trailer Release",
                                         "01/25 Hard Release",
                                         "01/26 Soft Release")))

# Read in spatial data
sp_releases <- readRDS(here("output/spatial_release_data_example.rds"))
sp_station <- readRDS(here("output/spatial_station_data_example.rds"))
raster_baydelta_tr <- readRDS(here("data/raster_baydelta_tr.rds"))
# raster_baydelta<-raster(here("data/baydelta.tif"))
```

Reorganize data

```
# Subset station to captured delta smelt
sp_station <- sp_station %>% filter(StationCode %in% unique(smelt_data_join$StationCode))

# Make a table of locations A to locations B
sf_release <- st_as_sf(sp_releases) %>%
  mutate(lon1 = sf::st_coordinates(.)[,1],
        lat1 = sf::st_coordinates(.)[,2])

sf_catch <- sp_station%>%
  ungroup() %>%
  st_as_sf() %>%
  mutate(lon2 = sf::st_coordinates(.)[,1],
```

```

lat2 = sf::st_coordinates(.)[, 2])

releaseToCatch <- smelt_data_join %>%
  left_join(sf_release) %>%
  rename(geometryR = geometry) %>%
  left_join(sf_catch) %>%
  rename(geometryC = geometry)

```

function for making shortest path lines

```

f_lines <- function(x) {
  paths <- shortestPath(raster_baydelta_tr, c(releaseToCatch$lon1[x], releaseToCatch$lat1[x]),
                        c(releaseToCatch$lon2[x], releaseToCatch$lat2[x]), output = "SpatialLines")
  line <- sf::st_as_sf(paths)
}

```

Run function to make lines - takes a while

```

# lineTest <- lapply(1:nrow(releaseToCatch), f_lines)
linesbind <- bind_rows(lineTest) %>%
  st_as_sf() %>%
  st_set_crs(st_crs(sp_station))
saveRDS(linesbind, file = here("output", "allpaths.rds"), compress = "xz")

```

Plotting - start here

Read in data and make df for plotting

```

linesbind <- readRDS(here("output", "allpaths.rds"))
linesbind_df <- cbind(linesbind,
                      smelt_data_join %>% dplyr::select(SampleDate, Tag, Release.Method, StationCode, F)
                      mutate(days = as.numeric(SampleDate - FirstDayRelease)))

sp_station_Tag <- full_join(sp_station, smelt_data_join %>% dplyr::select(SampleDate, Tag, Release.Method))

sp_release_Tag <- full_join(sp_releases, smelt_data_join %>% dplyr::select(SampleDate, Tag, Release.Method)
                           filter(Tag %in% unique(smelt_data_join$Tag)))

```

Plotting theme

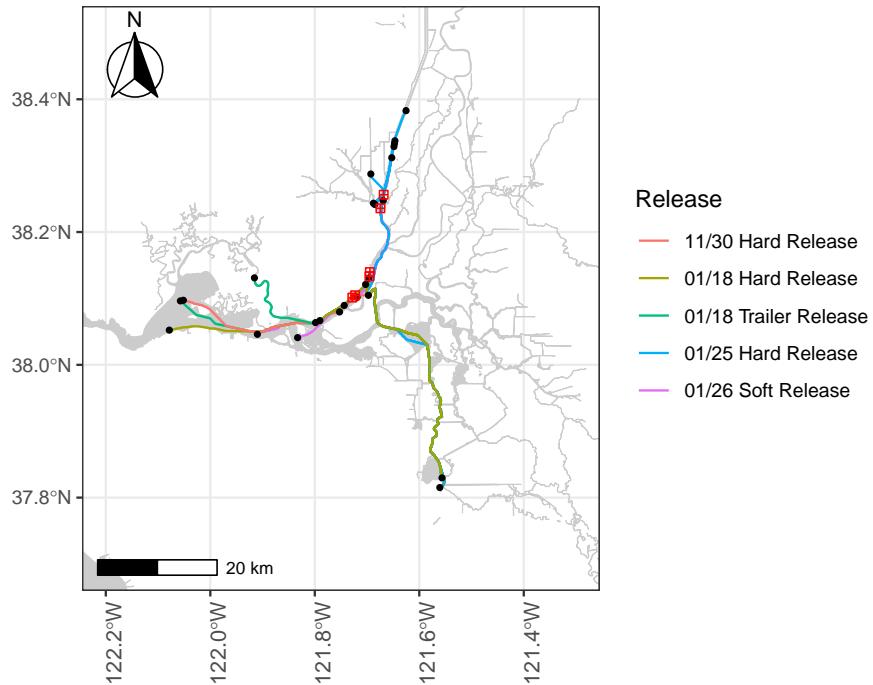
```

theme_plots <-
  theme_bw() +
  theme(axis.text.x = element_text(angle = 90, size = 10))

```

Plot all lines

```
plot_allreleases <- ggplot() +  
  geom_sf(data=WW_Delta, fill="gray80", color="gray80") +  
  geom_sf(data=linesbind_df, aes(color = Release)) +  
  geom_sf(data=sp_station,color="black", shape = 20)+  
  geom_sf(data=sp_releases,color="red", shape = 12)+  
  xlim(c(-122.2, -121.3)) +  
  ylim(c(37.7, 38.5))+  
  annotation_north_arrow(location = "tl", which_north = "true",  
    pad_x = unit(.04, "in"), pad_y = unit(0.05, "in"),  
    style = north_arrow_fancy_orienteering) +  
  annotation_scale(location = "bl", bar_cols = c("black", "white", "black", "white")) +  
  theme_plots  
  
plot_allreleases
```



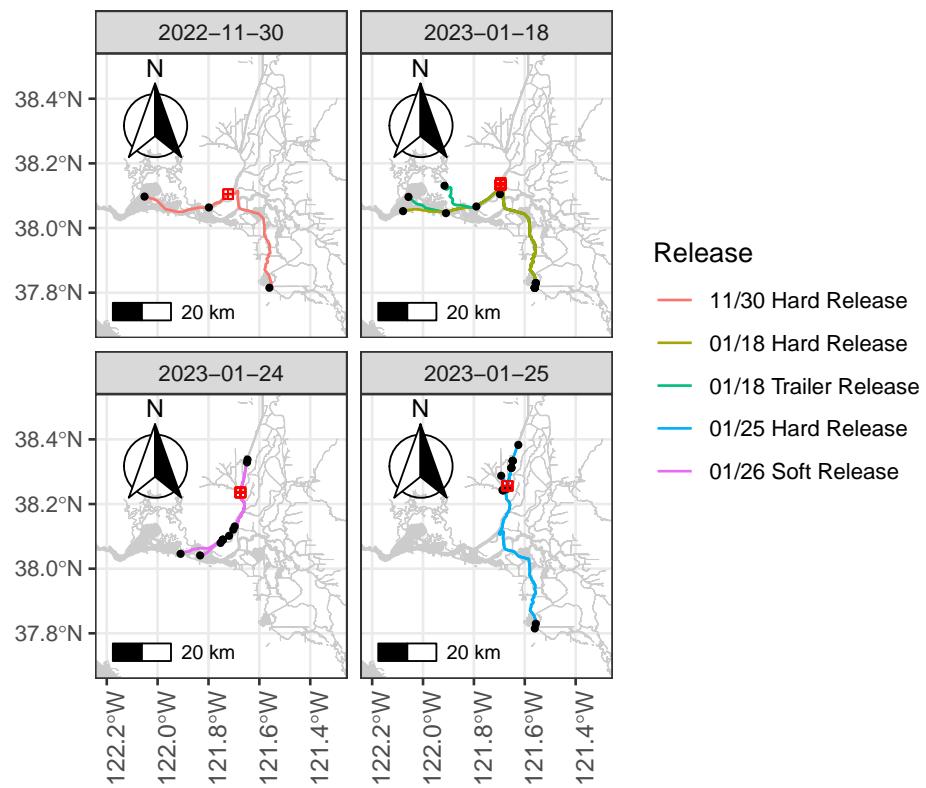
Plot by date

```
ggplot() +  
  geom_sf(data=WW_Delta, fill="gray80", color="gray80") +  
  geom_sf(data=linesbind_df, aes(color = Release)) +  
  geom_sf(data=sp_station_Tag, color="black", shape = 20)+  
  geom_sf(data=sp_release_Tag,color="red", shape = 12)+  
  xlim(c(-122.2, -121.3)) +  
  ylim(c(37.7, 38.5))+  
  facet_wrap(~FirstDayRelease) +
```

```

annotation_north_arrow(location = "tl", which_north = "true",
                       pad_x = unit(.04, "in"), pad_y = unit(0.05, "in"),
                       style = north_arrow_fancy_orienteering) +
annotation_scale(location = "bl", bar_cols = c("black", "white", "black", "white")) +
theme_plots

```

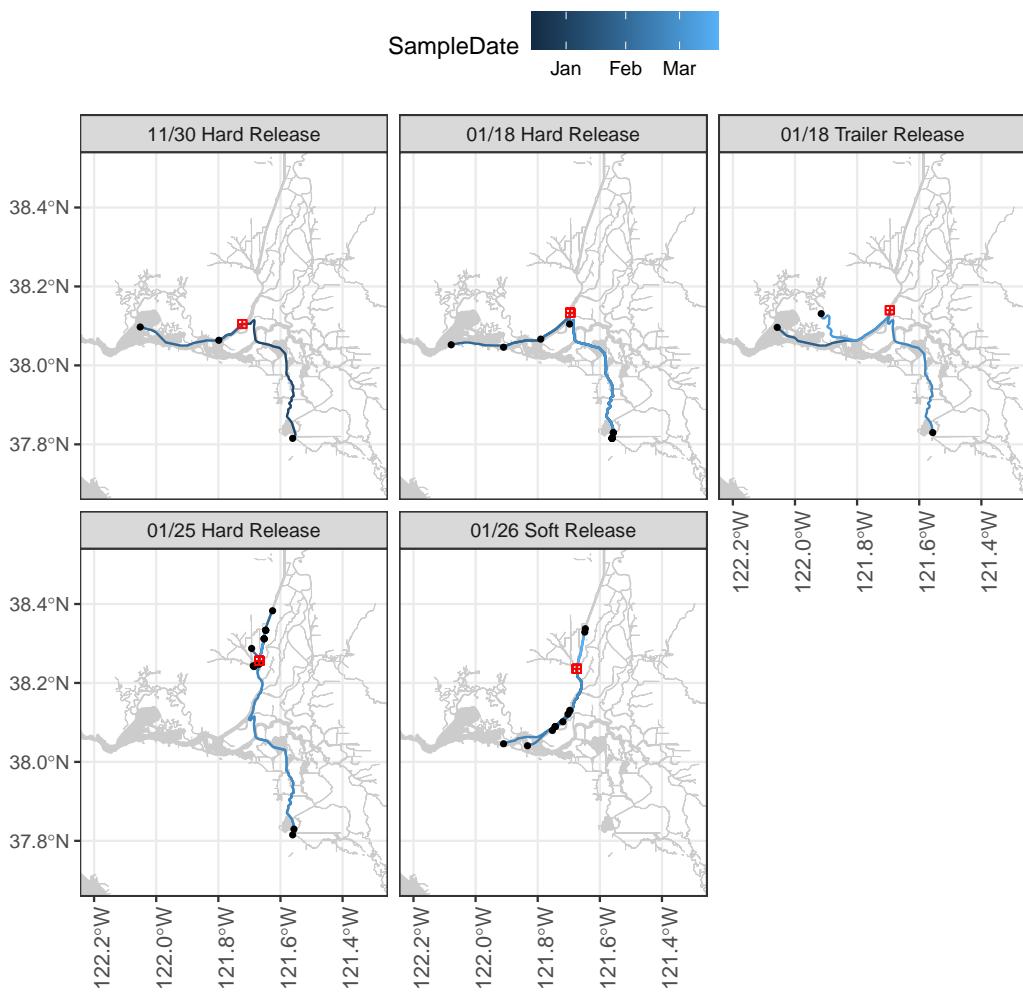


Plot by release and date caught

```

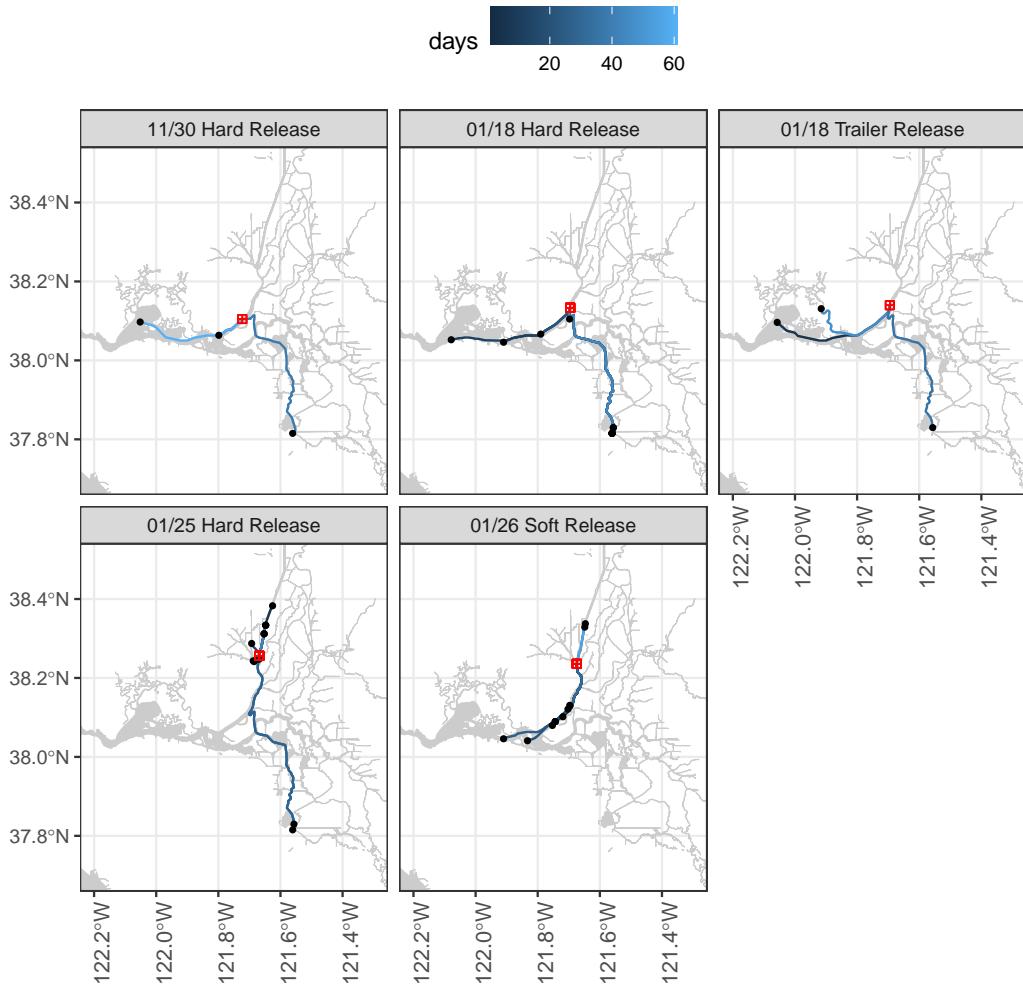
ggplot() +
  geom_sf(data=WW_Delta, fill="gray80", color="gray80") +
  geom_sf(data=linesbind_df, aes(color = SampleDate)) +
  geom_sf(data=sp_station_Tag, color="black", shape = 20)+ 
  geom_sf(data=sp_release_Tag,color="red", shape = 12)+ 
  xlim(c(-122.2, -121.3)) +
  ylim(c(37.7, 38.5))+ 
  facet_wrap(~Release) +
  theme_plots +
  theme(legend.position = "top")

```



Plot by release and days traveling (similar to above)

```
ggplot() +
  geom_sf(data=WW_Delta, fill="gray80", color="gray80") +
  geom_sf(data=linesbind_df, aes(color = days)) +
  geom_sf(data=sp_station_Tag, color="black", shape = 20) +
  geom_sf(data=sp_release_Tag,color="red", shape = 12) +
  xlim(c(-122.2, -121.3)) +
  ylim(c(37.7, 38.5)) +
  facet_wrap(~Release) +
  theme_plots +
  theme(legend.position = "top")
```



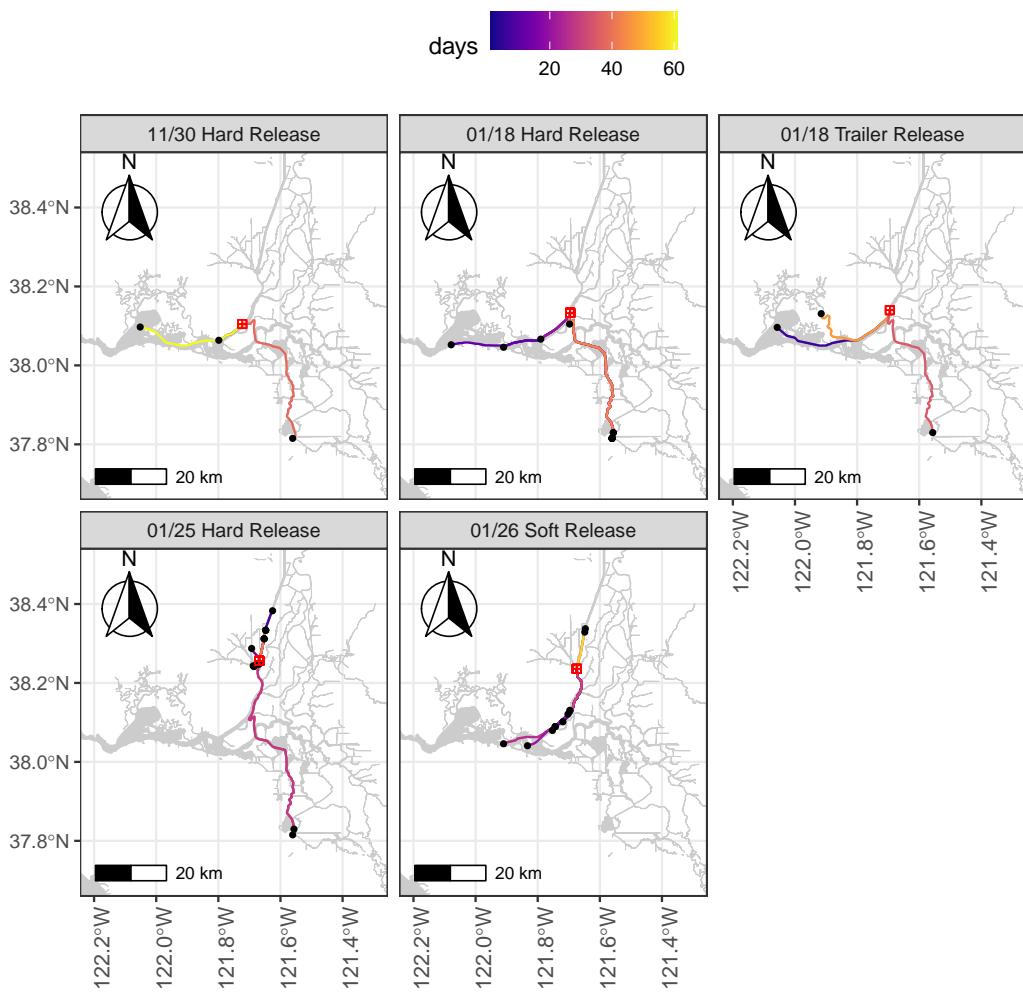
Plot by release and days traveling with viridis

```

plot_release_days <- ggplot() +
  geom_sf(data=WW_Delta, fill="gray80", color="gray80") +
  geom_sf(data=linesbind_df, aes(color = days)) +
  geom_sf(data=sp_station_Tag, color="black", shape = 20) +
  geom_sf(data=sp_release_Tag,color="red", shape = 12) +
  viridis::scale_color_viridis(option = "plasma") +
  xlim(c(-122.2, -121.3)) +
  ylim(c(37.7, 38.5)) +
  facet_wrap(~Release) +
  annotation_north_arrow(location = "tl", which_north = "true",
                         pad_x = unit(.02, "in"), pad_y = unit(0.03, "in"),
                         style = north_arrow_fancy_orienteering) +
  annotation_scale(location = "bl", bar_cols = c("black", "white", "black", "white")) +
  theme_plots +
  theme(legend.position = "top")

plot_release_days

```



```

png(here("output/Figure_swim_path_release_days.png"), width = 7, height = 6, res = 300, units = "in")
plot_release_days
dev.off()

png(here("output/Figure_swim_path_all.png"), width = 7, height = 6, res = 300, units = "in")
plot_allreleases
dev.off()

```