Summary of data processing for Patuxent River Estuary

Marcus W. Beck, Rebecca Murphy

May 22, 2015

The following describes additional processing of raw station data for the Patuxent River Estuary. Note that the raw data file ('PAX_TRIB...') was edited manually to remove station TF1.0 and to change the chlorophyll value for TF1.4 on 11/28/1988 from 2.99 to 5.98. All data for 1985 was also removed because of incomplete coverage for many stations. Brief descriptions of additional processing steps are provided below.

The following raw data files were imported:

1. `PAX_TRIB_CHLAandSALINITY_85to14.csv`: chlorophyll and salinity data for all stations in the Patuxent River from 1985 to 2014 (from R. Murphy)

2. `PAX_station_info.csv`: metadata for each station including lat/lon, salinity zone, etc. (from R. Murphy) - not used in this document

The data were first imported into R.

```
# code for processing raw data, see email from R. Murphy on 3/13/15
# created March 2015, M. Beck

## packages to use
library(dplyr)
library(gridExtra)
library(tidyr)
library(ggplot2)
library(RColorBrewer)
library(scales)
library(wesanderson)

## import
```

```r
root_path <- gsub('text', '', getwd())

# meta
pax_meta <- read.csv(paste0(root_path, '/inst/PAX_station_info.csv'),
  header = TRUE, stringsAsFactors = FALSE)

# data
pax_data <- read.csv(paste0(root_path, '/inst/PAX_TRIB_CHLAandSALINITY_85TO14.csv'),
  header = TRUE, stringsAsFactors = FALSE)

# flow data, combined with data after processing in a later chunk
pax_flow <- read.csv(paste0(root_path, '/inst/PatuxentDailyFlow_84to15.csv'),
  header = TRUE, stringsAsFactors = FALSE)
pax_flow <- mutate(pax_flow,
  date = as.Date(date, format = '%m/%d/%Y'),
  lnQ = log(flow_cfs)
  ) %>%
  select(-flow_cfs)

# remove 1985 from pax_data
pax_data <- filter(pax_data, !grepl('/1985$', date))

# reorder STATION variable along trib axis
stats <- c('TF1.3', 'TF1.4', 'TF1.5', 'TF1.6', 'TF1.7',
  'RET1.1', 'LE1.1', 'LE1.2', 'LE1.3', 'LE1.4')
pax_data$STATION <- factor(pax_data$STATION, level = stats)
```

Salinity data were vertically-integrated for each unique date. The integration function averaged all salinity values after interpolating from the surface to the maximum depth. Salinity values at the most shallow and deepest sampling depth were repeated for zero depth and maximum depths, respectively, to bound the interpolations within the range of the data.

```r
##
# get vertically integrated salinity

# function for vertical integration
int_fun <- function(TOTAL_DEPTH, DEPTH, AvgValue){

  # only interpolate if > 1 salinity value
  if(length(na.omit(AvgValue)) < 2 ) return(na.omit(AvgValue))

  # setup for interpolation
  max_depths <- mean(unique(TOTAL_DEPTH), na.rm = TRUE)
```

```
  depths <- c(0, DEPTH, max_depths)
  vals <- c(AvgValue[1], AvgValue, AvgValue[length(AvgValue)])

  # get the average of the interpolation
  out <- mean(approx(depths, vals)$y)

  return(out)

  }

# process
# note that there are no 'PROBLEM' values, lab and method do not change
sal_tmp <- filter(pax_data, PARAMETER == 'SALINITY') %>%
  mutate(date = as.Date(date, format = '%m/%d/%Y')) %>%
  group_by(date, STATION) %>%
  summarize(sal = int_fun(TOTAL_DEPTH, DEPTH, AvgValue))
```

Chlorophyll values at each station were retained only for surface samples and no 'problem' codes. Chlorophyll were also transformed with the natural-log.

```
##
# get only surface estimates for chlorophyll
# remove those w/ problem codes (only a few obs)
schl_tmp <- filter(pax_data,
  PARAMETER == 'CHLA' & LAYER == 'S'& PROBLEM == ''
  ) %>%
  mutate(lnchla = log(AvgValue)) %>%
  mutate(date = as.Date(date, format = '%m/%d/%Y')) %>%
  select(date, STATION, lnchla)

# repeat but for bottom chlorophyll
bchl_tmp <- filter(pax_data,
  PARAMETER == 'CHLA' & LAYER == 'B'& PROBLEM == ''
  ) %>%
  mutate(lnchla = log(AvgValue)) %>%
  mutate(date = as.Date(date, format = '%m/%d/%Y')) %>%
  select(date, STATION, lnchla)
```

The data were recombined by date and station. Daily flow data were also merged by matching only the days in the water quality data. Some plots of the raw data are below.

```r
##
# merge surface chl and salinity data, then plot
pax_data <- full_join(schl_tmp, sal_tmp, by = c('date', 'STATION')) %>%
  left_join(pax_flow, by = 'date')

# merge surface and bottom chlorophyll
pax_chldata <- full_join(schl_tmp, bchl_tmp, by = c('date', 'STATION')) %>%
  mutate(
    lnchla_s = lnchla.x,
    lnchla_b = lnchla.y
    ) %>%
  select(-lnchla.x, -lnchla.y) %>%
  gather('loc', 'lnval', 3, 4)

# save the data
save_path <- gsub('text$', 'data', getwd())
save(pax_data, file = paste0(save_path, '/pax_data.RData'))
save(pax_chldata, file = paste0(save_path, '/pax_chldata.RData'))
save(pax_meta, file = paste0(save_path, '/pax_meta.RData'))
save_path <- gsub('text$', 'inst', getwd())
write.csv(pax_data, paste0(save_path, '/pax_data.csv'),
  row.names = FALSE, quote = FALSE)
```

Here are some code chunks for figures. The first creates boxplot summaries of annual and seasonal trends by station.

```r
devtools::load_all()
data(pax_data)
data(pax_meta)
data(pax_clip)

# color palette
cols <- wes_palette('Zissou', 100, 'continuous') %>%
  as.character %>%
  .[1:60]

# change default ggplot theme
theme_mine <- function (base_size = 12, base_family = "") {
  theme_bw(base_size = base_size, base_family = base_family) %+replace%
  theme(
    axis.text.x = element_text(size = 8),
    plot.background = element_rect(fill='transparent',
      colour = NA),
```

```r
      panel.background = element_rect(fill='transparent',
        colour = NA),
      legend.background = element_rect(fill='transparent',
        colour = NA),
      strip.background = element_rect(fill =
          alpha(cols[length(cols)], 0.5)),
      legend.key = element_rect(fill = 'transparent',
        colour = NA)
      )
}
theme_set(theme_mine())

# upper limit on boxplots
y_max<- 7
y_lab <- expression(
  paste('ln-chlorophyll  ',italic(a),' (',italic(mu),'g',l^-1,')')
  )

# make plots for each station
stats <- unique(pax_meta$STATION)
pdf('figs/pax_chl.pdf', height = 5, width = 10, family = 'serif')
for(stat in stats){

  # add yr, month columns
  pax_stat <- filter(pax_data, STATION == stat) %>%
    mutate(
      mo = strftime(date, '%m'),
      yr = strftime(date, '%Y')
    )

  # medians for plotting
  yr_med <- group_by(pax_stat, yr) %>%
    summarize(yr_med = median(lnchla, na.rm = T))
  mo_med <- group_by(pax_stat, mo) %>%
    summarize(mo_med = median(lnchla, na.rm = T))

  # combine with original dataset
  pax_stat <- left_join(pax_stat, yr_med, by = 'yr') %>%
    left_join(mo_med, by = 'mo')

  # boxplot by year
  p1 <- ggplot(pax_stat, aes(x = yr, y = lnchla, fill = yr_med)) +
    geom_boxplot() +
    scale_x_discrete('Year') +
```

```r
    scale_y_continuous(
            y_lab,
            limits = c(0, y_max)
            ) +
  scale_fill_gradientn(colours = cols) +
  theme_classic() +
  theme(legend.position = 'none',
    axis.text.x = element_text(size = 8, angle = 90, hjust = 1)
    )

# boxplot by month
p2 <- ggplot(pax_stat, aes(x = mo, y = lnchla, fill = mo_med)) +
  geom_boxplot() +
  scale_x_discrete('Month') +
  scale_y_continuous(
            y_lab,
            limits = c(0, y_max)
            ) +
  scale_fill_gradientn(colours = cols) +
  theme_classic() +
  theme(legend.position = 'none')

# map

stat_meta <- filter(pax_meta, STATION == stat)
p3 <- ggplot(pax_meta, aes(x = LONG, y = LAT)) +
  geom_polygon(data = pax_clip, aes(x = long, y = lat, group = group),
    fill = alpha(cols[10], 0.8)) +
  geom_text(size = 7, aes(label = STATION), alpha = 0.7, colour = 'black') +
  coord_map(
    xlim = c(-76.78, -76.36),
    ylim = c(38.27, 38.85)
  ) +
  # geom_point(data = stat_meta, size = 16, colour = 'red') +
  geom_text(data = stat_meta, aes(label = STATION), col = 'red', size = 7) +
  theme(axis.title = element_blank())

grid.arrange(
  arrangeGrob(p1, p2, nrow = 2),
  p3,
  ncol = 2,
  widths = c(0.8, 0.5)
  )
}
```

```
dev.off()
```

This plot shows changes in chlorophyll concentrations by month.

```
devtools::load_all()
data(pax_data)
data(pax_meta)
data(pax_clip)

# change default ggplot theme
theme_mine <- function (base_size = 12, base_family = "") {
  theme_bw(base_size = base_size, base_family = base_family) %+replace%
  theme(
    axis.text.x = element_text(size = 8),
    plot.background = element_rect(fill='transparent',
      colour = NA),
    panel.background = element_rect(fill='transparent',
      colour = NA),
    legend.background = element_rect(fill='transparent',
      colour = NA),
    strip.background = element_rect(fill =
        alpha(cols[length(cols)], 0.5)),
    legend.key = element_rect(fill = 'transparent',
      colour = NA)
    )
}
theme_set(theme_mine())

# y axis label
y_lab <- expression(
  paste('ln-chlorophyll  ',italic(a),' (',italic(mu),'g',l^-1,')')
  )

# color for water for continuity with last fig
wt_col <- wes_palette('Zissou', 100, 'continuous')[10] %>%
  alpha(0.8)

# function for color that is not linear by rank
cRamp <- function(x, alpha_val = 0.8){

  # colors to use
  cols <- wes_palette('Zissou', 1000, 'continuous') %>%
    as.character %>%
```

```r
  .[1:600]

  # map the values to the colors
  x_rang <- (x - min(x, na.rm = T))/diff(range(x, na.rm = T))
  cols <- colorRamp(cols)(x_rang)
  apply(cols, 1, function(val){
    if(NaN %in% val) NA
    else
    rgb(val[1], val[2], val[3], alpha = alpha_val * 255, maxColorValue = 255)
    })

  }

# format pax_meta for mrg with pax_plo
pax_meta <- select(pax_meta, STATION, LONG, LAT) %>%
  mutate(STATION = as.character(STATION))

# add month to pax data
pax_data <- mutate(pax_data,
  mo = as.numeric(strftime(date, '%m')),
  yr = strftime(date, '%Y')
  ) %>%
  mutate(STATION = as.character(STATION))

# get monthly medians by station, add coords
pax_mo <- group_by(pax_data, STATION, mo) %>%
  summarize(lnchla = median(lnchla, na.rm = T)) %>%
  mutate(
    ptsz = rescale(lnchla, c(3, 12)),
    cols = cRamp(lnchla, alpha_val = 1)
    ) %>%
  ungroup %>%
  left_join(., pax_meta, by = 'STATION') %>%
  select(-lnchla) %>%
  full_join( ., pax_data, by = c('STATION', 'mo'))

# make plots for each station
mons <- sort(unique(pax_mo$mo))
pdf('figs/pax_trnds.pdf', height = 4, width = 8, family = 'serif')
for(mon in mons){

  # for map
  pax_map <- filter(pax_mo, mo %in% mon) %>%
    select(STATION, ptsz, cols, LONG, LAT) %>%
```

```r
  unique

# for boxplots
pax_box <- filter(pax_mo, mo %in% mon)

# reorder STATION variable along trib axis
stats <- c('TF1.3', 'TF1.4', 'TF1.5', 'TF1.6', 'TF1.7',
  'RET1.1', 'LE1.1', 'LE1.2', 'LE1.3', 'LE1.4')
pax_map$STATION <- factor(pax_map$STATION, level = stats)
pax_map <- pax_map[order(pax_map$STATION), ]
pax_box$STATION <- factor(pax_box$STATION, level = stats)

# medians for plotting
pbox <- ggplot(na.omit(pax_box), aes(x = STATION, y = lnchla)) +
    geom_boxplot(fill = pax_map$cols) +
    scale_y_continuous(
      y_lab,
              limits = c(-1, 7)
              ) +
    theme_classic() +
    theme(
      legend.position = 'none',
      axis.title.x = element_blank()
      )

# for month on the map
mo_labs <- c('January', 'February', 'March', 'April', 'May', 'June', 'July',
  'August', 'September', 'October', 'November', 'December')
mo_labs <- data.frame(
  LONG = -76.45,
  LAT = 38.7,
  mo = mo_labs[which(mon == mons)]
  )

# map
pmap <- ggplot(pax_meta, aes(x = LONG, y = LAT)) +
  geom_polygon(data = pax_clip, aes(x = long, y = lat, group = group),
    fill = wt_col) +
  coord_map(
    xlim = c(-76.78, -76.36),
    ylim = c(38.27, 38.85)
    ) +
  geom_text(data = mo_labs, aes(label = mo)) +
  geom_point(data = pax_map, size = pax_map$ptsz,
```

```
        pch = 21, fill = pax_map$cols, colour = 'black') +
    theme(axis.title = element_blank())

  #combine
  grid.arrange(
    pbox, pmap, ncol =2, widths = c(1, 0.5)
    )

}
dev.off()
```

This plot shows changes in chlorophyll concentrations by year.

```
devtools::load_all()
data(pax_data)
data(pax_meta)
data(pax_clip)

# change default ggplot theme
theme_mine <- function (base_size = 12, base_family = "") {
  theme_bw(base_size = base_size, base_family = base_family) %+replace%
  theme(
    axis.text.x = element_text(size = 8),
    plot.background = element_rect(fill='transparent',
      colour = NA),
    panel.background = element_rect(fill='transparent',
      colour = NA),
    legend.background = element_rect(fill='transparent',
      colour = NA),
    strip.background = element_rect(fill =
        alpha(cols[length(cols)], 0.5)),
    legend.key = element_rect(fill = 'transparent',
      colour = NA)
    )
}
theme_set(theme_mine())

# y axis label
y_lab <- expression(
  paste('ln-chlorophyll  ',italic(a),' (',italic(mu),'g',l^-1,')')
  )

# color for water for continuity with last fig
```

```r
wt_col <- wes_palette('Zissou', 100, 'continuous')[10] %>%
  alpha(0.8)

# function for color that is not linear by rank
cRamp <- function(x, alpha_val = 0.8){

  # colors to use
  cols <- wes_palette('Zissou', 1000, 'continuous') %>%
    as.character %>%
    .[1:600]

  # map the values to the colors
  x_rang <- (x - min(x, na.rm = T))/diff(range(x, na.rm = T))
  cols <- colorRamp(cols)(x_rang)
  apply(cols, 1, function(val){
    if(NaN %in% val) NA
    else
    rgb(val[1], val[2], val[3], alpha = alpha_val * 255, maxColorValue = 255)
    })

}

# format pax_meta for mrg with pax_plo
pax_meta <- select(pax_meta, STATION, LONG, LAT) %>%
  mutate(STATION = as.character(STATION))

# add month to pax data
pax_data <- mutate(pax_data,
  mo = as.numeric(strftime(date, '%m')),
  yr = strftime(date, '%Y')
  ) %>%
  mutate(STATION = as.character(STATION))

# get annual medians by station, add coords
pax_yr <- group_by(pax_data, STATION, yr) %>%
  summarize(lnchla = median(lnchla, na.rm = T)) %>%
  mutate(
    ptsz = rescale(lnchla, c(3, 12)),
    cols = cRamp(lnchla, alpha_val = 1)
    ) %>%
  ungroup %>%
  left_join(., pax_meta, by = 'STATION') %>%
  select(-lnchla) %>%
  full_join( ., pax_data, by = c('STATION', 'yr'))
```

```r
# make plots for each station
yr_vals <- sort(unique(pax_yr$yr))
pdf('figs/pax_yrtrnds.pdf', height = 4, width = 8, family = 'serif')
for(yr_val in yr_vals){

  # for map
  pax_map <- filter(pax_yr, yr %in% yr_val) %>%
    select(STATION, ptsz, cols, LONG, LAT) %>%
    unique

  # for boxplots
  pax_box <- filter(pax_yr, yr %in% yr_val)

  # reorder STATION variable along trib axis
  stats <- c('TF1.3', 'TF1.4', 'TF1.5', 'TF1.6', 'TF1.7',
    'RET1.1', 'LE1.1', 'LE1.2', 'LE1.3', 'LE1.4')
  pax_map$STATION <- factor(pax_map$STATION, level = stats)
  pax_map <- pax_map[order(pax_map$STATION), ]
  pax_box$STATION <- factor(pax_box$STATION, level = stats)

  # medians for plotting
  pbox <- ggplot(na.omit(pax_box), aes(x = STATION, y = lnchla)) +
      geom_boxplot(fill = pax_map$cols) +
      scale_y_continuous(
        y_lab,
                limits = c(-1, 7)
                ) +
      theme_classic() +
      theme(
        legend.position = 'none',
        axis.title.x = element_blank()
        )

  # for month on the map
  yr_labs <- yr_vals
  yr_labs <- data.frame(
    LONG = -76.45,
    LAT = 38.7,
    yr = yr_labs[which(yr_val == yr_vals)]
    )

  # map
  pmap <- ggplot(pax_meta, aes(x = LONG, y = LAT)) +
    geom_polygon(data = pax_clip, aes(x = long, y = lat, group = group),
```

```
      fill = wt_col) +
    coord_map(
      xlim = c(-76.78, -76.36),
      ylim = c(38.27, 38.85)
    ) +
    geom_text(data = yr_labs, aes(label = yr)) +
    geom_point(data = pax_map, size = pax_map$ptsz,
      pch = 21, fill = pax_map$cols, colour = 'black') +
    theme(axis.title = element_blank())

  #combine
  grid.arrange(
    pbox, pmap, ncol = 2, widths = c(1, 0.5)
    )

}
dev.off()
```