

Summary of data processing for Patuxent River Estuary

Marcus W. Beck, Rebecca Murphy

March 27, 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
# this is just to load dplyr, ggplot2
devtools::load_all('M:/docs/tidal_comp/TidalComp')

## import

# meta
pax_meta <- system.file('PAX_station_info.csv', package = 'TidalComp')
pax_meta <- read.csv(pax_meta, header = TRUE,
  stringsAsFactors = FALSE)
```

```

# data
pax_data <- system.file('PAX_TRIB_CHLAandSALINITY_85T014.csv',
  package = 'TidalComp')
pax_data <- read.csv(pax_data, header = TRUE,
  stringsAsFactors = FALSE)

# 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.

```

##
# 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)
chl_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)
```

The data were recombined by date and station. Some plots of the raw data are below.

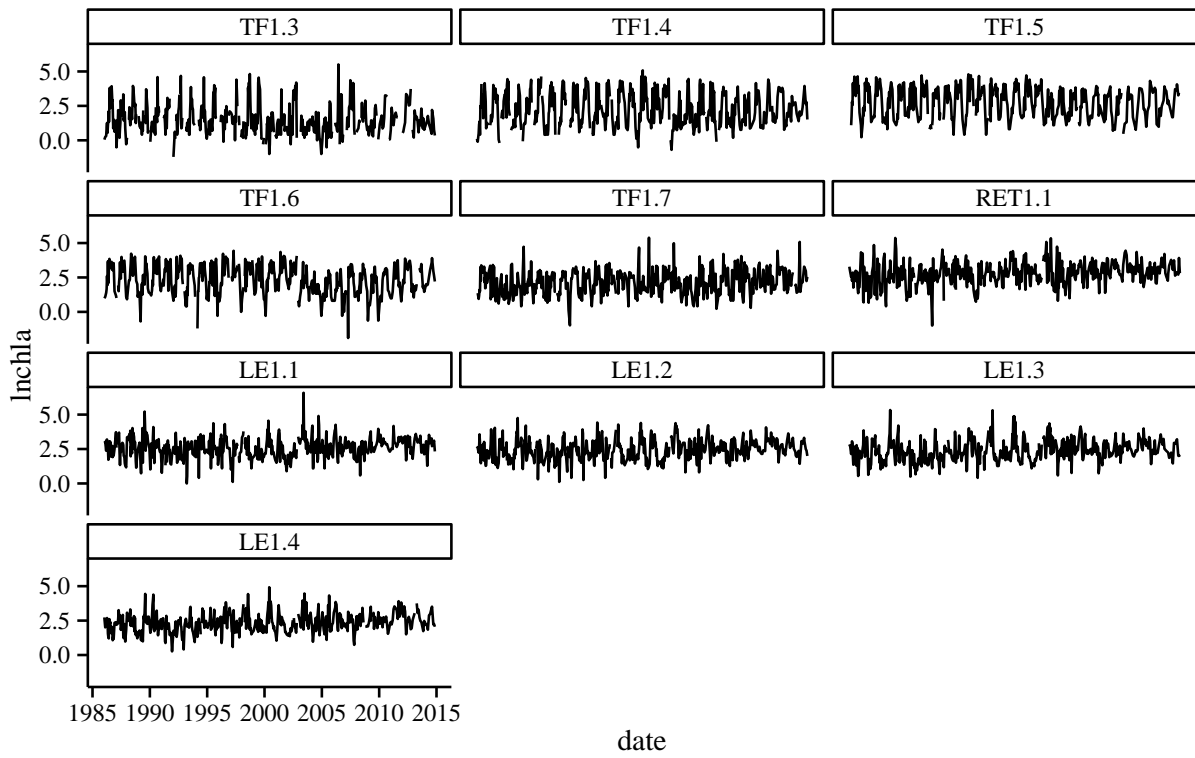
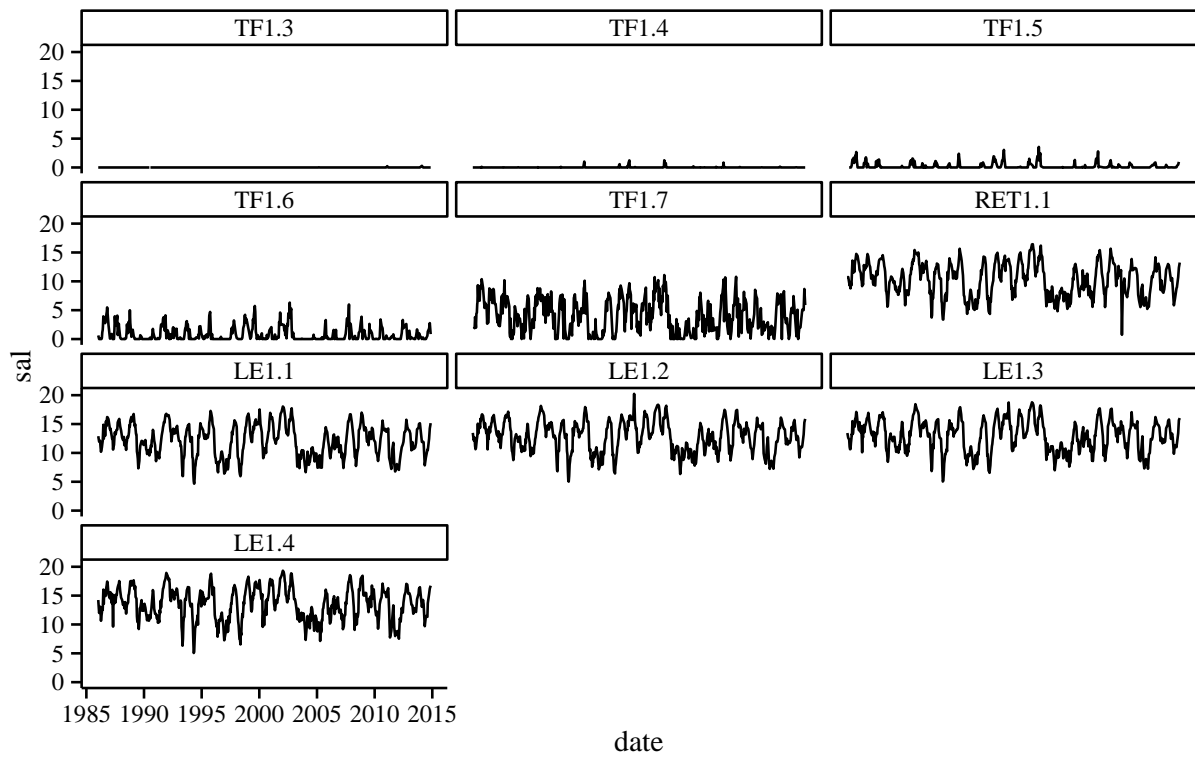
```
##
# merge chl and salinity data, then plot

pax_data <- full_join(chl_tmp, sal_tmp, by = c('date', 'STATION'))

ggplot(pax_data, aes(x = date, y = sal, group = STATION)) +
  geom_line() +
  theme_classic() +
  facet_wrap(~ STATION, ncol = 3)

ggplot(pax_data, aes(x = date, y = lnchla, group = STATION)) +
  geom_line() +
  theme_classic() +
  facet_wrap(~ STATION, ncol = 3)

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



Here are some more figures.

```

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),
    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 = strptime(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') +
    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()

```