

GND analyses

June 5, 2015

The analyses for GND have focused on trends over time in selected WQ/nutrient variables at each of the sites, keeping in mind the a priori periods of interest in relation to the two spill events.

```
#####
# first event - heavy rain - April 2005
# second event - hurricane Isaac - August 2012
# for nuts: PO4H, NO23, NH4, CHLA
# timeframes: E1A Event 1 Acute, E1C Event 1 Chronic, NI Non-Input Years,
#   E2A Event 2 Acute, E2C Event 2 Chronic

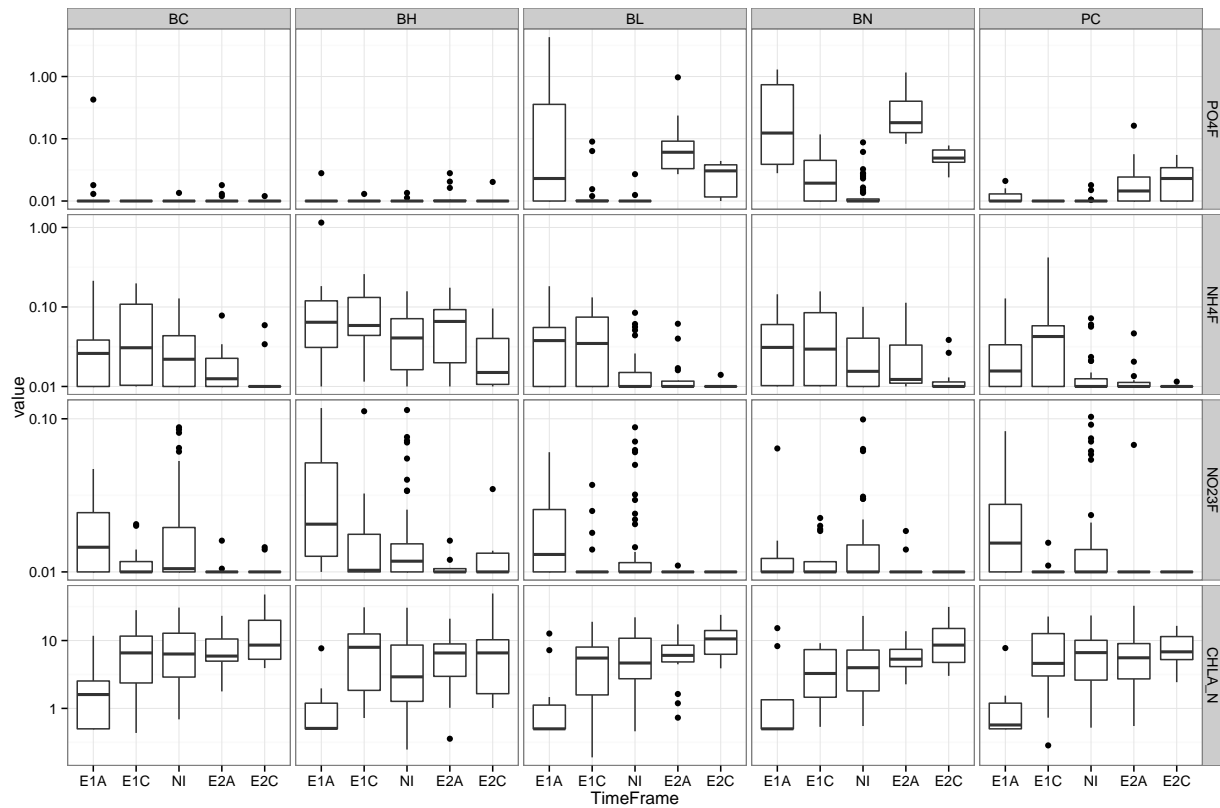
# load libraries
# install.packages(c('dplyr', 'tidyr', 'ggplot2', 'agricolae'))
library(dplyr)
library(tidyr)
library(ggplot2)
library(agricolae)

# import
raw_data <- read.csv('P04modified2005-2014grabs.csv', stringsAsFactors = F)
timeframes <- c('E1A', 'E1C', 'NI', 'E2A', 'E2C')

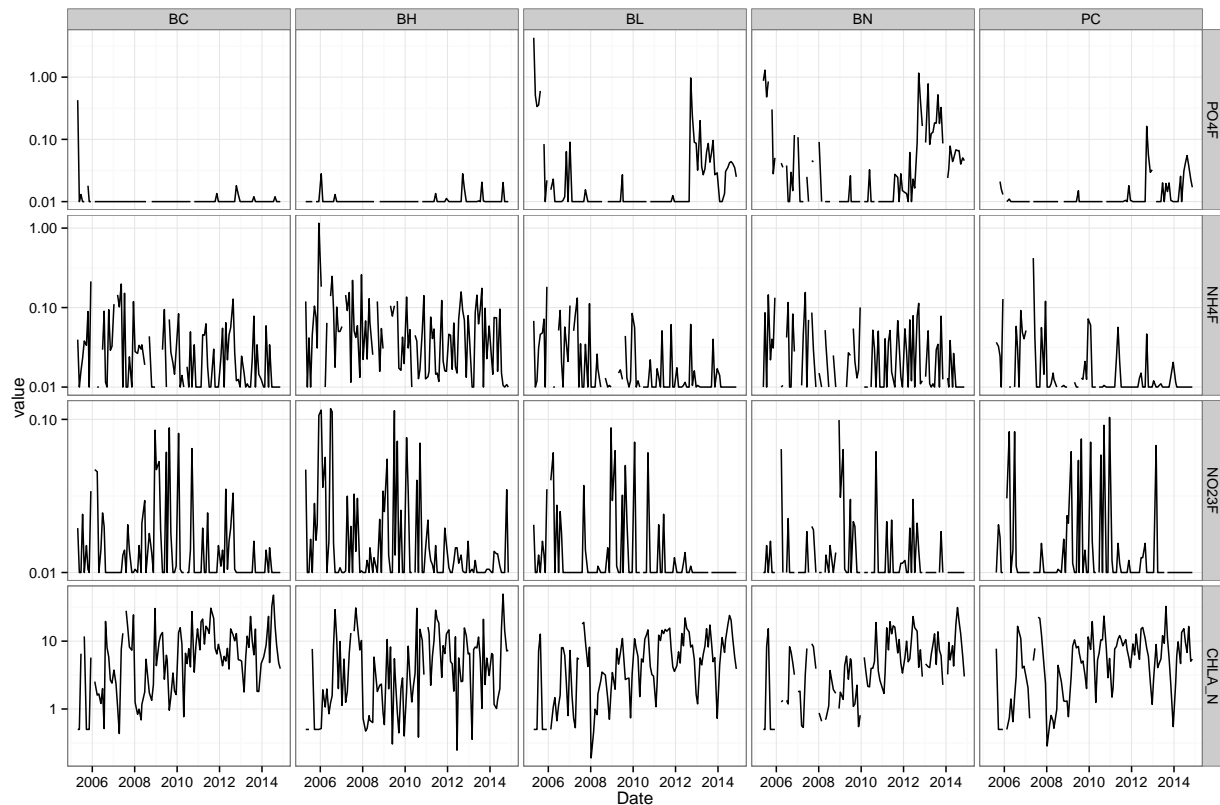
# format the data with dplyr...
# select columns of interest
# format date columns, create year column, floor by det limit for PO4, NO23, NH4
# cleanup station code, set factor levels for TimeFrame for correct plotting order,
# remove CR station
# make the nutrient variables long format with tidyr
# combine station, nutrient column for later analysis
# combine duplicate days, create logvalue
nut_data <- select(raw_data, StationCode, Date, TimeFrame, PO4F, NH4F, NO23F, CHLA_N) %>%
  mutate(
    Date = as.Date(Date, format= "%m/%d/%Y"),
    Year = as.numeric(format(Date, "%Y")),
    PO4F = pmax(0.01, PO4F),
    NH4F = pmax(0.01, NH4F),
    NO23F = pmax(0.01, NO23F),
    StationCode = toupper(gsub('gnd|nut| *', '', StationCode)),
    TimeFrame = factor(TimeFrame, levels = timeframes, labels = timeframes)
  ) %>%
  filter(StationCode != 'CR' & !is.na(TimeFrame)) %>%
  gather('nutrient', 'value', PO4F:CHLA_N) %>%
  unite(stat_nut, StationCode, nutrient, sep = ' ', remove = F) %>%
  group_by(stat_nut, StationCode, TimeFrame, Year, Date, nutrient) %>%
  summarize(value = mean(value, na.rm = TRUE)) %>%
  mutate(logvalue = log(value, 10))
```

Here are some basic plots of the data for each site, boxplots by each time frame and simple lineplots.

```
##
# boxplots
p1 <- ggplot(nut_data, aes(x = TimeFrame, y = value)) +
  geom_boxplot() +
  facet_grid(nutrient ~ StationCode, scales = 'free_y') +
  scale_y_log10() +
  theme_bw()
p1
```



```
##
# line plots
p2 <- ggplot(nut_data, aes(x = Date, y = value)) +
  geom_line() +
  facet_grid(nutrient ~ StationCode, scales = 'free_y') +
  scale_y_log10() +
  theme_bw()
p2
```



A multiple comparisons analysis is done looking at differences between each time period, separate analyses for each site and nutrient variable.

```
#####
# some analyses

# split data by unique station/nutrient variable
sep_data <- split(nut_data, nut_data$stat_nut)

# run a Tukey multiple comparison for each station, nutrient variable combo
res <- lapply(sep_data, function(x){

  mod <- aov(logvalue ~ TimeFrame, data = x)
  tuk_mod <- HSD.test(mod, 'TimeFrame', group = T)
  grps <- tuk_mod$groups
  grps$trt <- gsub('[:space:]].*$', '', grps$trt)
  grps

})

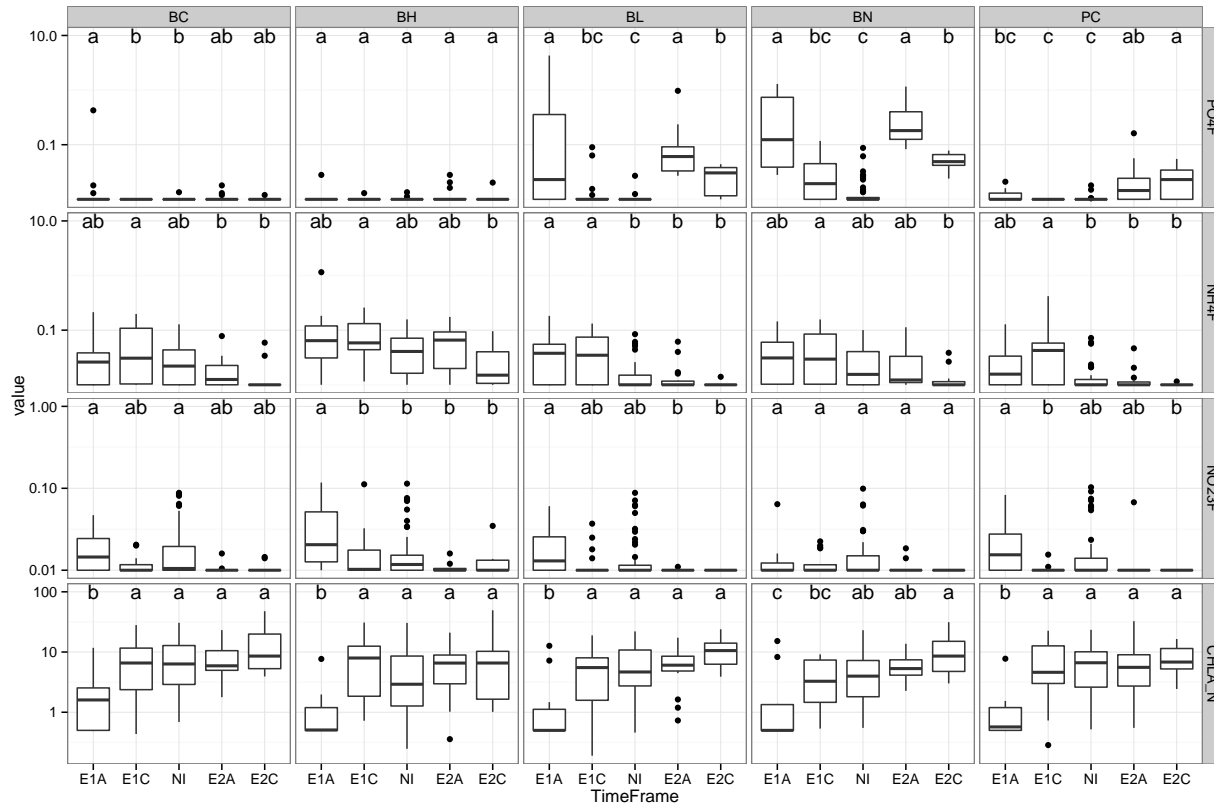
# combine results for plotting
res <- do.call('rbind', res) %>%
  mutate(
    stat_nut = gsub('\\.[0-9]', '', row.names(.)),
    TimeFrame = trt
  ) %>%
  separate(stat_nut, c('StationCode', 'nutrient'), sep = ' ')
```

```

ylocs <- data.frame(nutrient = c('P04F', 'NH4F', 'NO23F', 'CHLA_N'), ylims = c(10, 10, 1, 100))
res <- left_join(res, ylocs, by = 'nutrient')
res$TimeFrame <- factor(res$trt, levels = timeframes, labels = timeframes)

# the boxplots but with letters above each box indicating different groups
p1 + geom_text(data = res, aes(x = TimeFrame, y = ylims, label = M))

```



Survival regression was used to look at direction of trend over time *within* each period. This is the same thing as linear regression but corrections are made to account for lower detection limits.

```

#####
# survival regression for censored data by each timeframe
library(survival)

to_eval <- filter(nut_data, nutrient == 'P04F' & StationCode == 'BN')

to_eval <- split(to_eval, to_eval$TimeFrame)

res <- lapply(to_eval, function(x){

  # setup detection limit column
  x <- na.omit(x)
  x$notcens <- TRUE
  x$notcens[x$logvalue <= -2] <- FALSE

  mod <- survreg(Surv(logvalue, notcens, type = "left") ~ Date,
    data = x, dist="gaus", control = list(iter.max = 1000))

```

```

preddat <- seq.Date(min(x$Date), max(x$Date), by = 'day')
preddat <- data.frame(Date = preddat)

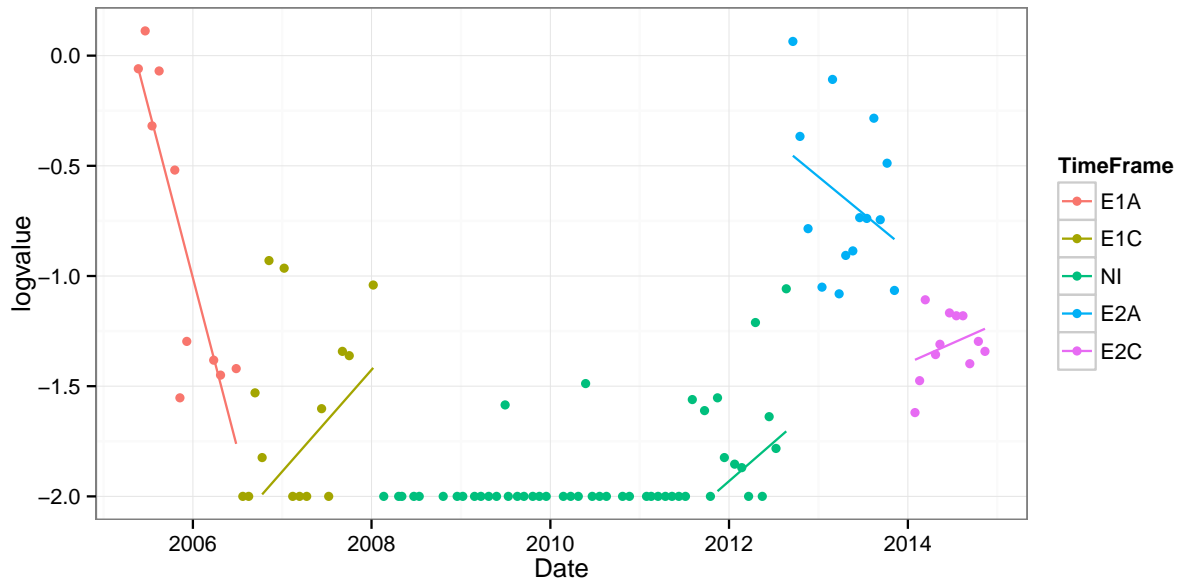
out <- predict(mod, newdata = preddat)
out <- data.frame(Date = preddat, pred = out)

out

})
res <- do.call('rbind', res)
res <- left_join(do.call('rbind', to_eval), res, by = 'Date')

ggplot(res, aes(x = Date, y = logvalue, group = TimeFrame, colour = TimeFrame)) +
  geom_point() +
  geom_line(aes(y = pred)) +
  theme_bw() +
  scale_y_continuous(limits = c(-2, max(res$logvalue)))

```



The changepoint analysis tries to locate breakpoints in the data where the resulting groups have statistically different properties. Changepoints can be located that separate data based on differences in the mean, variance, or both means/variance.

```

##
# changepoint analysis or phosphate
library(changepoint)

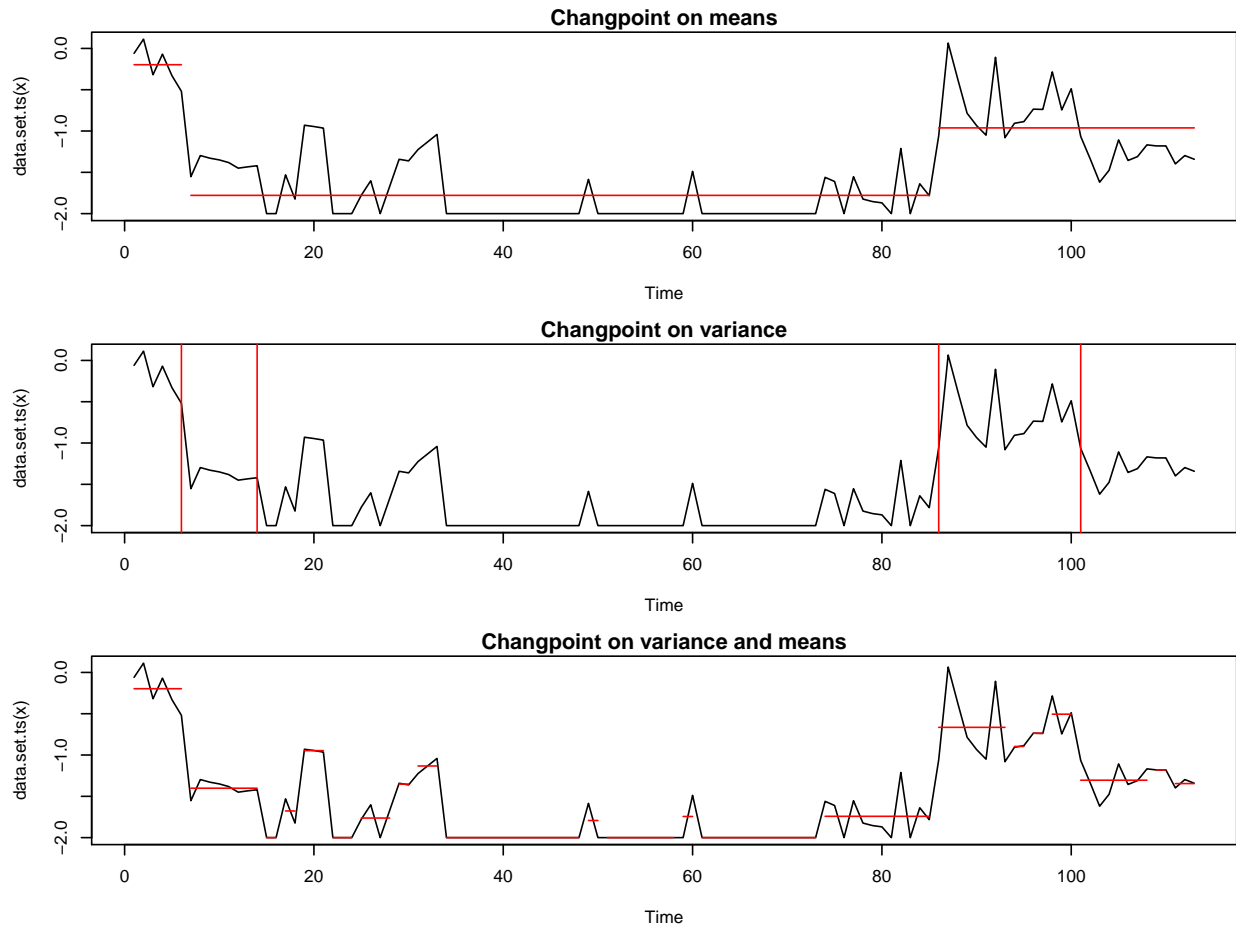
to_eval <- filter(nut_data, nutrient == 'P04F' & StationCode == 'BN') %>%
  select(Date, logvalue)

# fill missing values
interped <- approx(y = to_eval$logvalue, x = to_eval$Date, xout = to_eval$Date)
to_eval$interp <- interped$y

```

```
# changepoint analyses and plots
res <- cpt.mean(to_eval$interp, method = 'PELT')
res2 <- cpt.var(to_eval$interp, method = 'PELT')
res3 <- cpt.meanvar(to_eval$interp, method = 'PELT', Q = 5)

par(mar = c(4.5, 4.5, 1.5, 0.5), mfrow = c(3, 1))
plot(res, main = 'Changepoint on means')
plot(res2, main = 'Changepoint on variance')
plot(res3, main = 'Changepoint on variance and means')
```



Looking at the change points in relation to our predefined groups...

```
# looking at the results in relation to our groups
chngs <- to_eval$Date[cpts(res)]
chngs2 <- to_eval$Date[cpts(res2)]
chngs3 <- to_eval$Date[cpts(res3)]

p <- ggplot(to_eval, aes(x = Date, y = logvalue, group = TimeFrame, colour = TimeFrame)) +
  geom_point() +
  theme_bw()

p1 <- p + geom_vline(xintercept = as.numeric(chngs)) + ggtitle('Mean')
p2 <- p + geom_vline(xintercept = as.numeric(chngs2)) + ggtitle('Variance')
```

```
p3 <- p + geom_vline(xintercept = as.numeric(chngs3)) + ggtitle('Means and variance')
```

```
library(gridExtra)
grid.arrange(p1, p2, p3, ncol = 1)
```

