Comment on 'Legacy nitrogen may prevent achievement of water quallity goals in the Gulf of Mexico'

Tristan Ballard (tballard@stanford.edu)

7/12/2018

First read in packages used later for plotting. If they are not installed, run install.packages('packageName').

```
library(ggplot2); library(lattice); library(dplyr)
```

Now read in the datasets and convert to consistent units if necessary.

```
## 1955-2014 loads (tons) from https://cfpub.epa.gov/roe/indicator.cfm?i=33
mrb=c(473993.6096,332897.8374,475095.9203,406752.6557,
  367069.4698,356046.3626,302033.1373,304237.7587,192904.376,
  262349.9514,390217.9949,224871.3869,251326.8442,296521.5837,
  385808.752,408957.2771,392422.6163,607373.2067,865313.9152,
  833346.9043,873030.0902,545643.8064,616191.6925,855393.1187,
  1223564.899,872020,693720,1125900,1512600,1317520,971200,
  1095510,991670,616750,704730,1108400,1221200,991400,1792500,
  995500,931600,911290,1132300,1143040,1214720,591390,963100,
  878610,727310,937150,940650,662700,964400,1330090,1062860,
  1222200,1201630,622620,997320,801490)

## Convert from tons to kg/ha
## MRB area is 291451362.1ha, and .00110231 converts tons to kg
mrb=mrb / 291451362.1  / .00110231
mrb.dat=data.frame(cbind(1955:2014, mrb))
names(mrb.dat)=c('year', 'mrb')

## Read in predicted Science fluxes
vm.pred=read.csv('VM_predicted.csv', head=T) #kilotons
vm.pred$load=vm.pred$load / 291451362.1  / .00110231 * 1000 #kg/ha
mrb2=left_join(mrb.dat, vm.pred)
```

```
## Joining, by = "year"
```

Calculate R2 values of Van Meter et al. model, 1955-2014

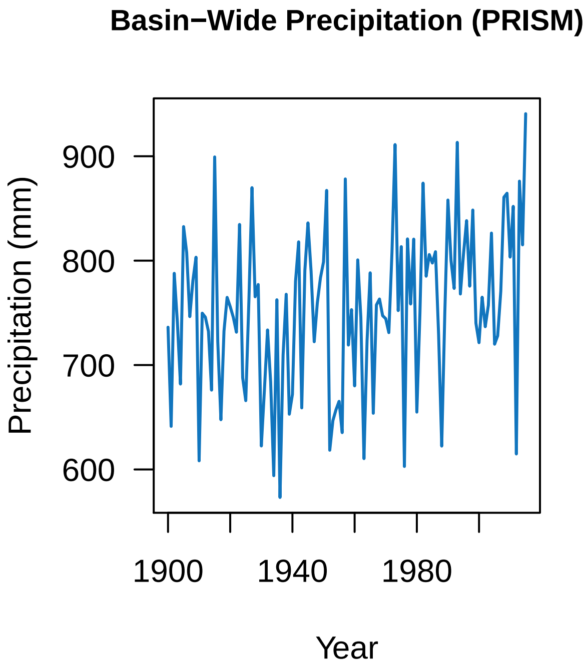
```
cor(mrb2$mrb, mrb2$load)^2 #R2=0.6722
```

```
## [1] 0.6722025
```

```
## Read in land use predictors from GBC
gbc.dat=read.csv('n_surplus.csv', head=T)

## convert from kg/ha of cropland to kg/ha of MRB by multiplying by fraction of MRB covered in cropland
crop.surplus=gbc.dat$crop_n * gbc.dat$frac_crop
surplus.df=data.frame(year=gbc.dat$year, crop.surplus)

## Note that if you do not scale the crop N term, the resulting R2 values are still very similar and con

## Read in basin-wide annual precipitation from PRISM dataset
precip.df=readRDS('precip.df.RDS') # [mm]
```

Plot the precipitation and Crop N surplus term

```
par(mfrow=c(1,2))
plot(precip.df$year, precip.df$precip, type='l', ylab='Precipitation (mm)', xlab='Year', las=1, col='#1
plot(surplus.df$year, surplus.df$crop.surplus, type='l', ylab='N Inputs (kg/ha)', xlab='Year', las=1, c
```



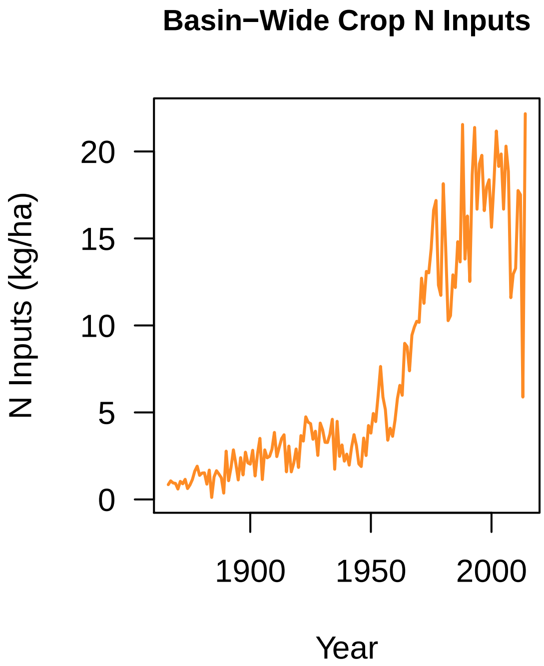


Figure 1

```r
### Two-variable models with cropN and Precip estimating observed loads
n.lag.crop=50
n.lag.precip=10
two.way.R2=matrix(rep(NA,n.lag.crop*n.lag.precip), nrow=n.lag.crop)
n.obs=length(mrb.dat$mrb)
for (i in 1:n.lag.crop){
  ## Loop through n surplus variables
  surplus.loop=rep(NA, n.obs)
    for (u in 1:n.obs){
      surplus.loop[u]=sum(surplus.df$crop.surplus[c(89+u-i+1):c(89+u)])
    }
  for (j in 1:n.lag.precip){
    ## Loop through precipitaion variables
    precip.loop=rep(NA, n.obs)
    for (v in 1:n.obs){
      precip.loop[v]=sum(precip.df$precip[c(55+v-j+1):c(55+v)])
    }
    ## Fit the regression and extract model R2
    fit=lm(mrb.dat$mrb ~ surplus.loop + precip.loop)
    two.way.R2[i,j] = cor(fit$fitted.values, mrb.dat$mrb)^2
  }
}
max(two.way.R2) #highest R2 value from 2-variable models
```

```
## [1] 0.7221609
```

```r
## Plot the 2-way R2 values
two.way.plot=two.way.R2
min.plot=.5
n.col=10
two.way.plot[two.way.plot<min.plot]=min.plot

levelplot(two.way.plot, col.regions=c('gray98',rev(heat.colors(n.col-1))), xlab=list("Legacy N (years)"
    at=c(seq(min.plot,.75,length.out=n.col+1)),xlim=c(0,50), ylim=c(0,10.5),
    colorkey=list(space='bottom', tck=.8, width=.8, height=.8),
    scales=list(x=list(at=c(0,10,20,30,40,50), labels=c(0,10,20,30,40,50)), y=list(at=c(0,5,10), labels
```

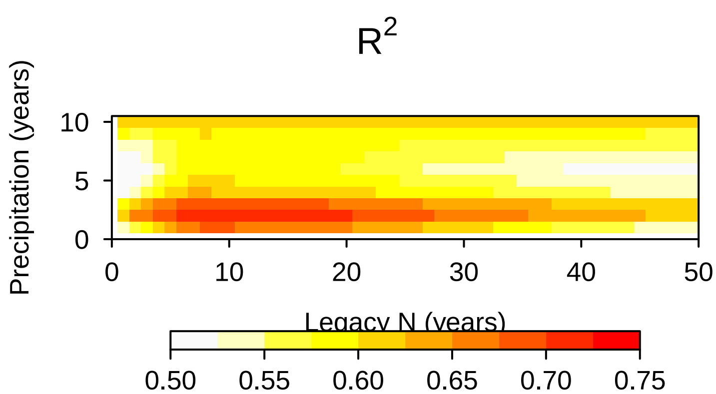


Figure 2A

```
i=8 #Cumulative crop N # of years
j=2 #Cumulative precipitaiton # of years

## Set up desired predictors for 2-variable model
surplus.loop=rep(NA, n.obs)
  for (u in 1:n.obs){
    surplus.loop[u]=sum(surplus.df$crop.surplus[c(89+u-i+1):c(89+u)])
  }
precip.loop=rep(NA, n.obs)
  for (v in 1:n.obs){
    precip.loop[v]=sum(precip.df$precip[c(55+v-j+1):c(55+v)])
  }

## Fit the highest R2 model
fit=lm(mrb.dat$mrb ~ surplus.loop + precip.loop)
summary(fit)
```

```
##
## Call:
## lm(formula = mrb.dat$mrb ~ surplus.loop + precip.loop)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.0320 -0.4808 -0.0094  0.3122  1.8736
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -6.8258575  1.2006284  -5.685 4.69e-07 ***
## surplus.loop  0.0159182  0.0023142   6.878 5.07e-09 ***
## precip.loop   0.0049918  0.0008371   5.963 1.65e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6058 on 57 degrees of freedom
## Multiple R-squared:  0.7222, Adjusted R-squared:  0.7124
## F-statistic: 74.08 on 2 and 57 DF,  p-value: < 2.2e-16
```

```
## minimum deviate precipitation for visualization
precip.md=precip.loop-min(precip.loop)
fit=lm(mrb.dat$mrb ~ surplus.loop + precip.md)
#summary(fit) #minimum deviating only changes intercept

## Make stacked time series
bottom=coef(fit)[1] + coef(fit)[2]*surplus.loop
top=coef(fit)[3]*precip.md

## Set up dataframes for plotting in ggplot
dat.plot=data.frame(pred=c(bottom,top), sector=c(rep('z', length(bottom)), rep('a', length(top))), Year=
dat.point=data.frame(load=mrb.dat$mrb, Year=1955:2014)

ggplot(data=dat.plot, aes(x=Year, y=pred, fill=sector)) +
    geom_area() +
```

```
  geom_point(data=dat.plot, aes(x=Year, y=load), size=.9, stroke=.4,shape=1) +
  guides(fill=F) +
  labs(y='Load (kg/ha)') +
  scale_fill_manual(values=c('#1175BE','#FD8B25')) +
  scale_x_continuous(breaks=seq(1955,2015,by=10)) +
  scale_y_continuous(limits=c(0,6.3), breaks=seq(0,6,by=1)) +
  theme(axis.title=element_text(size=9),axis.text.x=element_text(size=7.5), axis.text.y=element_text(s
    panel.background = element_blank(), axis.line = element_line(colour = "black"))
```

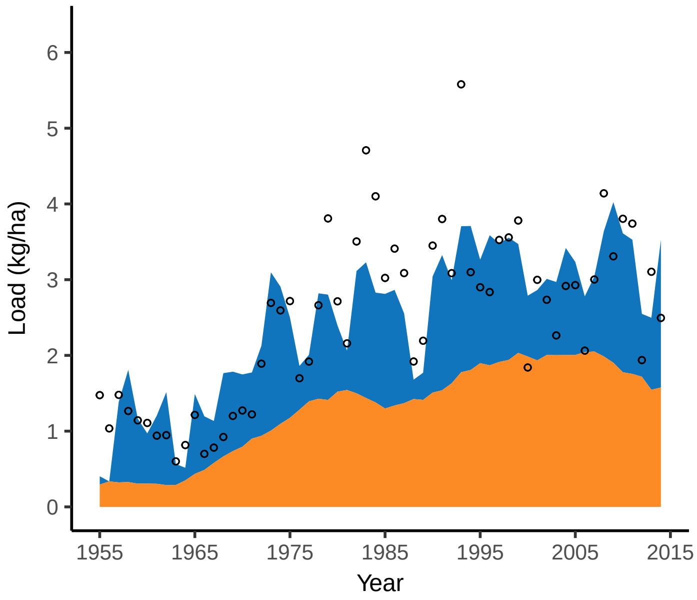


Figure 2B

```
## Read in pigment data, scraped using WebPlotDigitizer
pigment=read.csv('pigment_60_95.csv', head=T)

## Set up dataframe for plotting with units consistent with Van Meter et al (2018)
fig2b.pred=read.csv('VM_predicted.csv', head=T) #ktons
fig2b.df=left_join(fig2b.pred, surplus.df)
```

```
## Joining, by = "year"
```

```
n=nrow(fig2b.df)
## Add in the cumulative N terms
surplus.loop=rep(NA, n)
  i=8
  for (u in i:n){
    surplus.loop[u]=sum(fig2b.df$crop.surplus[c(u-i+1):c(u)])
  }
fig2b.df=data.frame(fig2b.df, cropN.cum8=surplus.loop)
fig2b.df$crop.surplus=fig2b.df$crop.surplus * 291451362.1  * .00110231 / 1000 #convert to ktons
fig2b.df$cropN.cum8=fig2b.df$cropN.cum8 * 291451362.1  * .00110231 / 1000 #convert to ktons

## Add in the 2yr precip term
fig2b.df=left_join(fig2b.df, precip.df)
```

```
## Joining, by = "year"
```

```
precip.loop=rep(NA, n)
  j=2
  for (u in i:n){
    precip.loop[u]=sum(fig2b.df$precip[c(u-j+1):c(u)])
  }
fig2b.df=data.frame(fig2b.df, precip.cum2=precip.loop)

## Add in our model predictions
fit=lm(c(mrb.dat$mrb * 291451362.1 * .00110231 / 1000) ~ cropN.cum8 + precip.cum2, data=fig2b.df[156:21
fig2b.df=data.frame(fig2b.df, preds=predict(fit, fig2b.df))

## Compare model with pigment concentrations
yr.start=1960; yr.end=1995;
idx.start=which(fig2b.df$year==yr.start)
idx.end=which(fig2b.df$year==yr.end)
cor(fig2b.df$preds[idx.start:idx.end], pigment$bcaro)^2 #0.57
```

```
## [1] 0.566728
```

```
cor(fig2b.df$preds[idx.start:idx.end], pigment$zea)^2 #0.54
```

```
## [1] 0.5377376
```

```
## Get p-values for pigment association as in Van Meter et al. (2018)
coef(summary((lm(fig2b.df$preds[idx.start:idx.end] ~ pigment$bcaro))))[2,4]
```

```
## [1] 1.181542e-07
```

```
coef(summary((lm(fig2b.df$preds[idx.start:idx.end] ~ pigment$zea))))[2,4]
```

```
## [1] 3.639026e-07
```

```
## Plotting time
lwd.plot=3.5
plot(yr.start:yr.end, fig2b.df$load[idx.start:idx.end], ylim=c(0,1400), las=1, ylab='N Load (ktons)',xla
  lines(yr.start:yr.end, fig2b.df$preds[idx.start:idx.end], col='gold', lwd=lwd.plot)
  axis(side=1, at=seq(yr.start, yr.end, 5), cex.axis=.9)
  axis(side=2, at=seq(0, 1400, 200), las=1, cex.axis=.9)
  abline(h=seq(0, 1400, 200), v=seq(yr.start, yr.end, 5), lty=3, col='grey')
  points(1960:1995, pigment$bcaro, pch=16, col='black', cex=1)
  points(1960:1995, pigment$bcaro, pch=16, col='#EA7F51', cex=.9)
  points(1960:1995, pigment$zea, pch=16, col='black', cex=1)
  points(1960:1995, pigment$zea, pch=16, col='#9B61C4', cex=.9)
par(new=T)
  plot(x=0,y=0, pch=NA, ylim=c(0,1.4), axes=F, xlab='',ylab='')
  axis(side=4, at=seq(0,1.4,.2),las=1, cex.axis=.9)
legend('topleft', bg='white',c('N Load (2-Variable Model)','N Load (Van Meter et al. (1))','Zeaxanthin'
  cex=.85,pch=c(NA,NA,16,16),lty=c(1,1,NA,NA),lwd=c(2,2,NA,NA), col=c('gold','#87CF5C','#9B61C4','#EA7F5
```