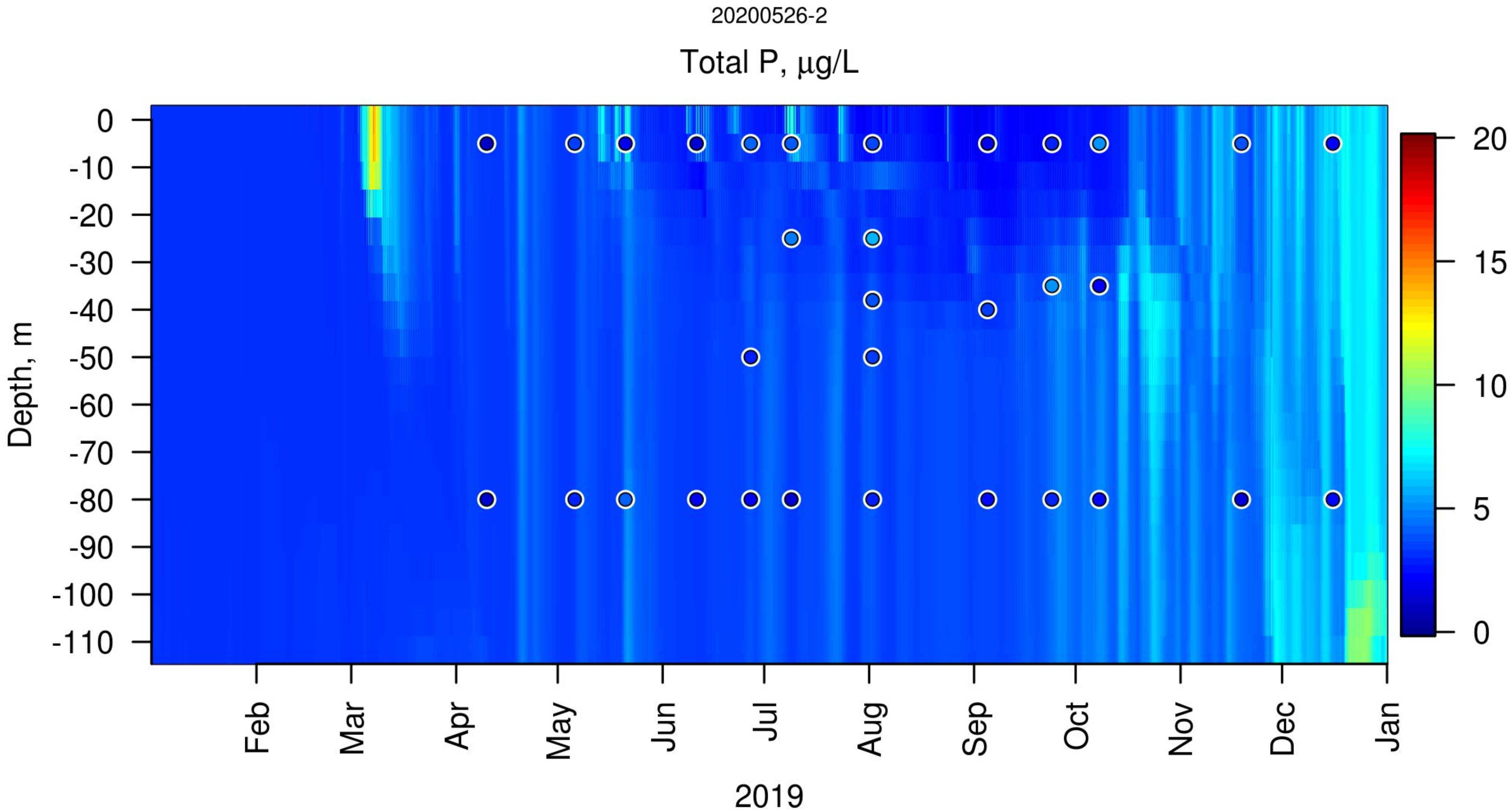
Review of model validation scripts for Lake Michigan FVCOM-GEM

10/5/2021

Peter Alsip

Depth-Time rasters



time.R

Depth-Time raster: plotting loop

- Loop through nodes (ms)
  - Read in data for given node
  - Define plot attributes for TP
  - Read in variable data and apply necessary conversions

```
for(mi in 1:length(ms)){
  detlist <- mlist[[mi]]$detlist
  nutNames <- mlist[[mi]]$nutNames
  depths <- mlist[[mi]]$depths
  m <- mlist[[mi]]$node

  nuti <- 7# chl 7, tp 6
  #nuts2plot <- 1:length(nutNames) #which( !(nutNames %in% c("GPP","BIO_M","MRATION")) )
  #nuts2plot <- which( !(nutNames1 %in% c("GPP","BIO_M","MRATION","aice")) )
  nuts2plot<-which(nutNames1 %in% c("Chl","TP"))
  #nuts2plot <-which( (nutNames1 %in% c("kdPAR")))
  for(nuti in nuts2plot){
    nutName <- nutNames1[nuti]
    #nutDim <- nutDims[nuti]
    nutDim <- nutList[[which(names(nutList)==nutName)]]$dim
    unit <- nutList[[which(names(nutList)==nutName)]]$unit
    scale <- nutList[[which(names(nutList)==nutName)]]$scale
    mx <- nutList[[which(names(nutList)==nutName)]]$mx
    nutLab <- nutList[[which(names(nutList)==nutName)]]$nutLab

    addObs2plot<-FALSE # defaults to false, TRUE for TP and CHL as of 7/22/20 -PA

    }else if(nutName == "TP"){
      detm <- detlist[[which(nutNames=="PO4")]]*scale
      TPVars <- nutList[[which(names(nutList)=="TP")]]$var
      TPVars <- TPVars[!(TPVars %in% "PO4")]
      for(i in 1:length(TPVars)){
        detm <- detm + detlist[which(nutNames==TPVars[i])][[1]]*scale*ptoc[1] # assumes ptoc is same for all
      }
      zlim <- c(0, mx)
      addObs2plot<-TRUE
```

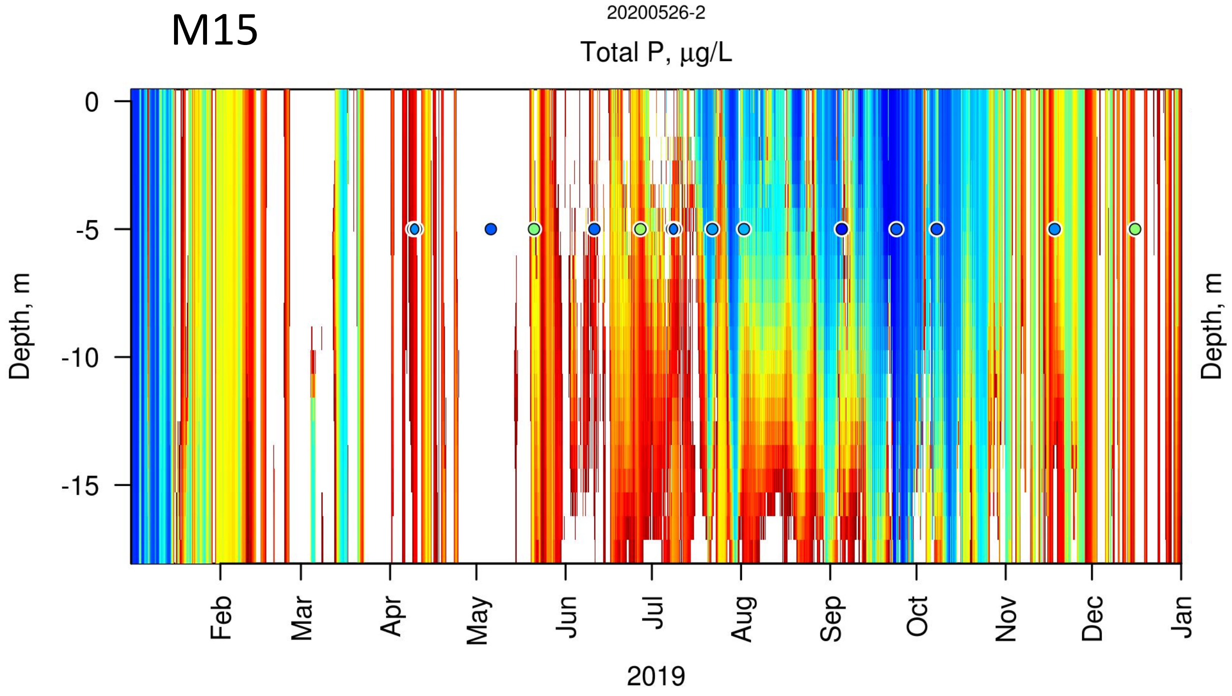
Depth-Time plots

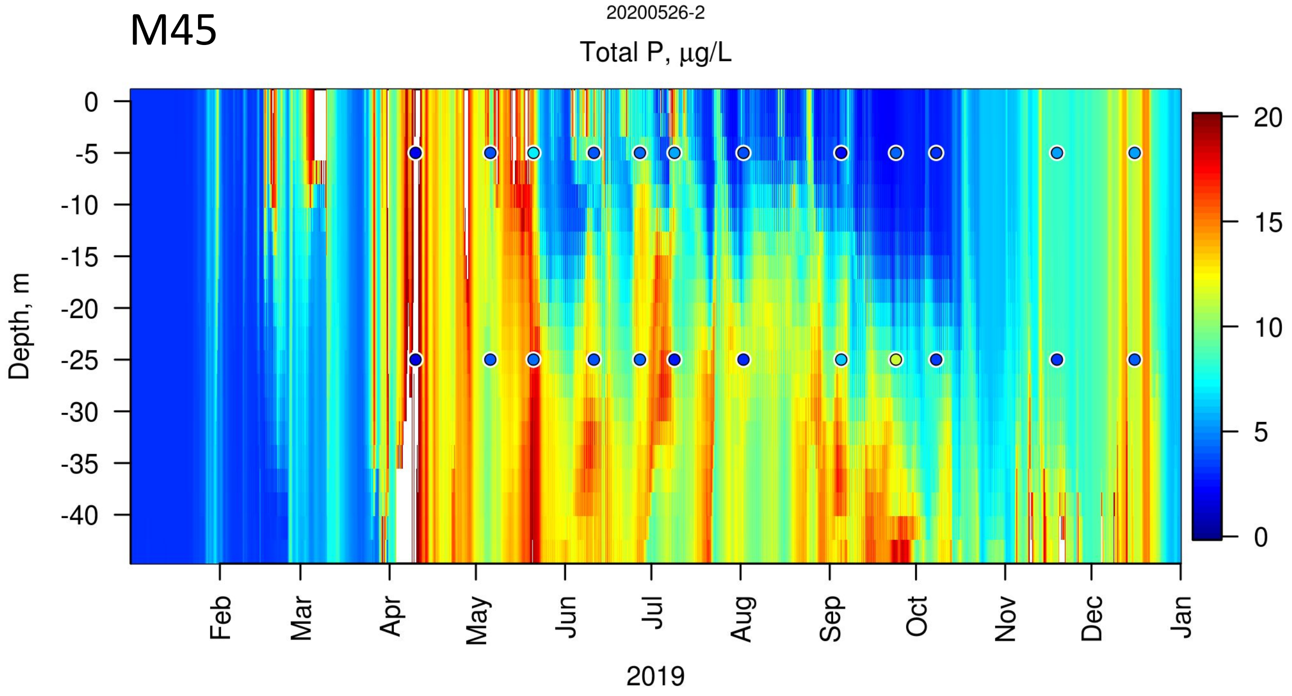
- Set up raster

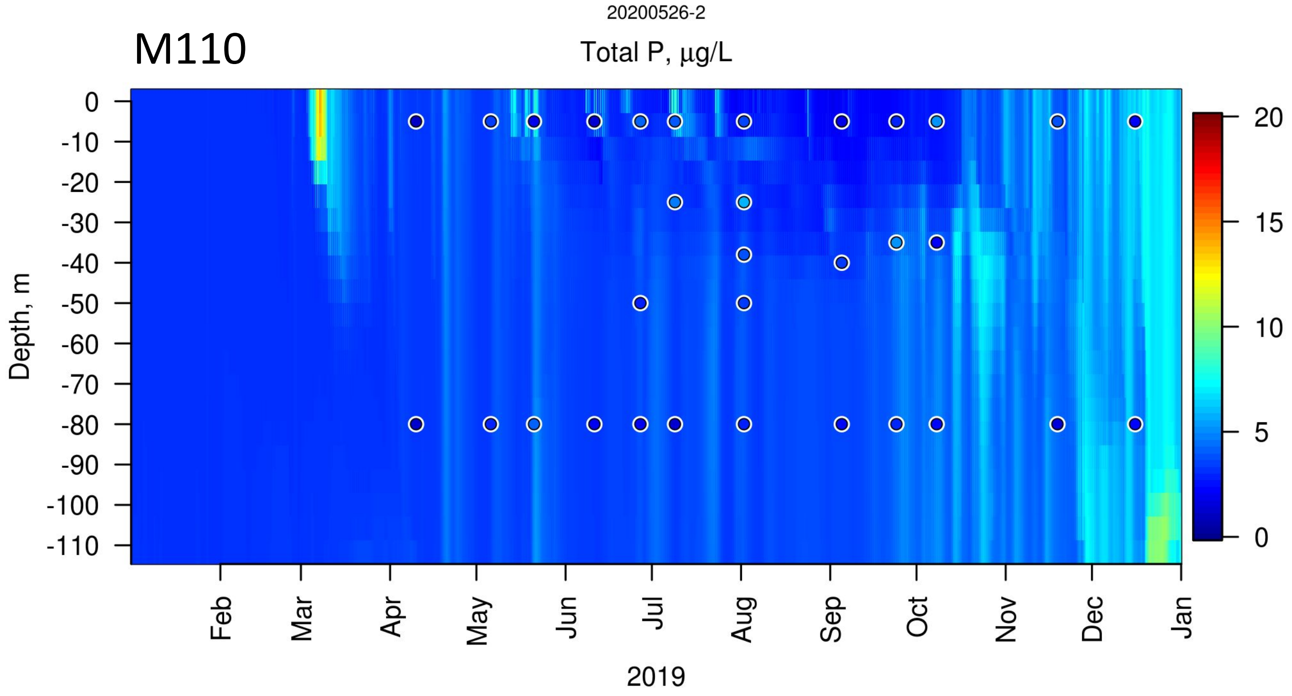
```
nd <- length(siglay)
timi <- length(time1)
ds <- depths
dmax <- depths[length(depths)]+(depths[2]-depths[1])/2 # depth at bottom edge of image
detm <- t(detm)
mat2 <- detm[1:timi,nd:1]
#mat2 <- detm[1:timi,nd:1]

#   zlim <- c(0, max(mat2, na.rm=TRUE))
#    zlim <- c(min(mat2, na.rm=TRUE), max(mat2, na.rm=TRUE))
# zlim <- c(0,10000)
cols <- tim.colors(256)

par(mar = c(6, 4, 3, 5) + 0.1) #c(bottom, left, top, right)
image(mat2, zlim=zlim, col=cols, xaxt="n", yaxt="n"
      ,xlab=" "
      ,ylab="Depth, m"
      #   ,main = nutLab
      #   ,cex.main=0.8
)
```

- Overlay obs

```
# Add observations for TP or Chl
if(addObs2plot){
  obsData<-muskegon[which(muskegon$Year == year & muskegon$Station == stas1d$station[mi]),]
  # add date column
  obsData$date2<-as.POSIXct(paste0(obsData$Year,"-",obsData$DOY),format = "%Y-%j")
  obsData$date2<-as.POSIXct(strftime(obsData$date2, format = "%Y-%m-%d %H:%M:%S"))
  odf<-obsData[,c("Station","date2","Depth",nutName)]
  odf<-odf[which(!is.na(odf[,c(nutName)])),]

  # index observations by the model time step they fall on. This is necessary for correctly
  odf$yhour<-NA
  for(i in 1:nrow(odf)){
    odf$yhour[i]<-which(time1 %in% odf$date2[i])  # index obervations by model time steps
  }
  # raster plot x axis is c(0,1) so the x position of observations and model values are dete
  odf$yhour<-odf$yhour/length(time1)

  # depth index (y position of observations)
  # subtract depth from 1
  odf$depth1<- 1-as.numeric(odf$Depth)*-1/dmax

  cexObs<-1.0

  cols2<-fields::color.scale(as.numeric(odf[,c(nutName)]), col=cols, zlim = zlim)
  points(odf$yhour, odf$depth1, col=cols2, pch=19, cex=cexObs)
  points(odf$yhour, odf$depth1, col="black", cex=cexObs)
  points(odf$yhour, odf$depth1, col="white", cex=cexObs+0.2)
}
```

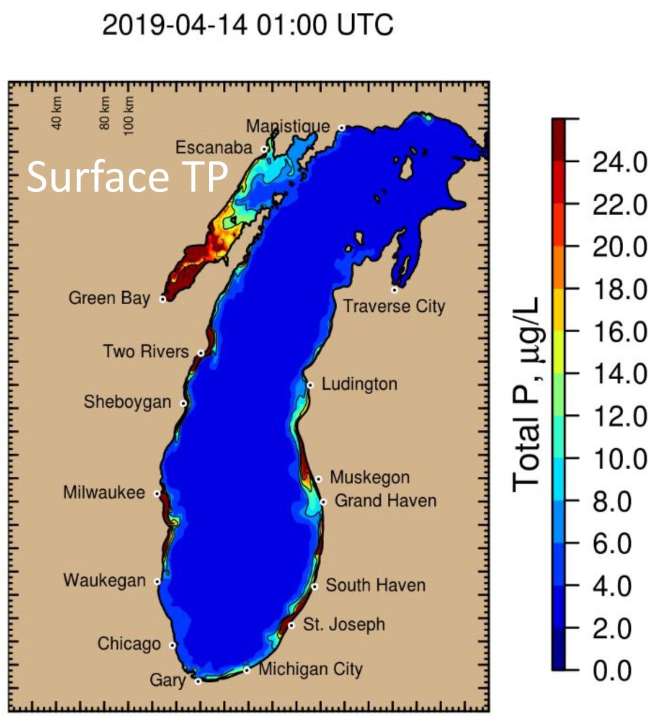

M15
20200526-2
Total P, µg/L
Depth, m
0
-5
-10
-15
Feb
Mar
Apr
May
Jun
Jul
Aug
Sep
Oct
Nov
Dec
Jan
2019

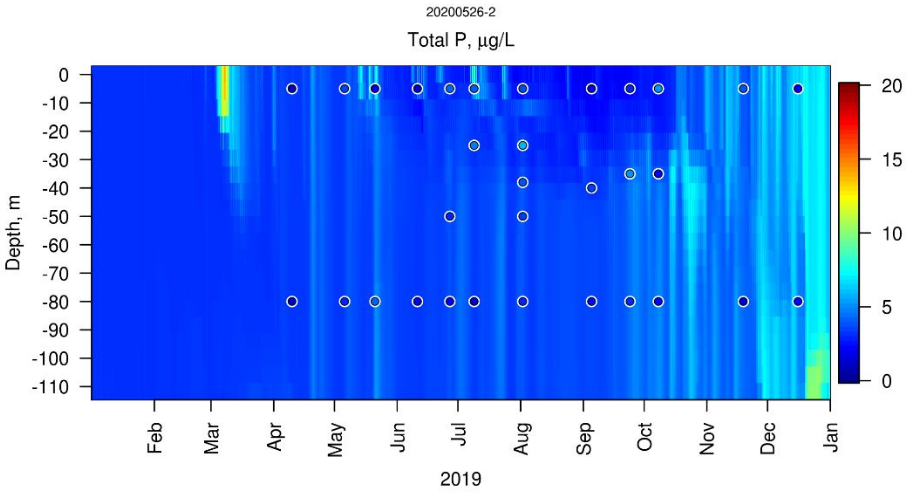

M45
20200526-2
Total P, µg/L
Depth, m
0
-5
-10
-15
-20
-25
-30
-35
-40
Feb
Mar
Apr
May
Jun
Jul
Aug
Sep
Oct
Nov
Dec
Jan
2019
20
15
10
5
0


M110
20200526-2
Total P, µg/L
Depth, m
0
-10
-20
-30
-40
-50
-60
-70
-80
-90
-100
-110
Feb
Mar
Apr
May
Jun
Jul
Aug
Sep
Oct
Nov
Dec
Jan
2019
20
15
10
5
0

lmh_hindcast_summary_plots.R

- Purpose: Summarize FVCOM-GEM outputs and provide quick comparison to observations digitized from publications and GLERL's Muskegon dataset
- Pre-reqs:
  - Matrix **[node, time]** of surface TP extracted from output netcdfs
  - depthTimeList data at Muskegon nodes
  - Muskegon observations





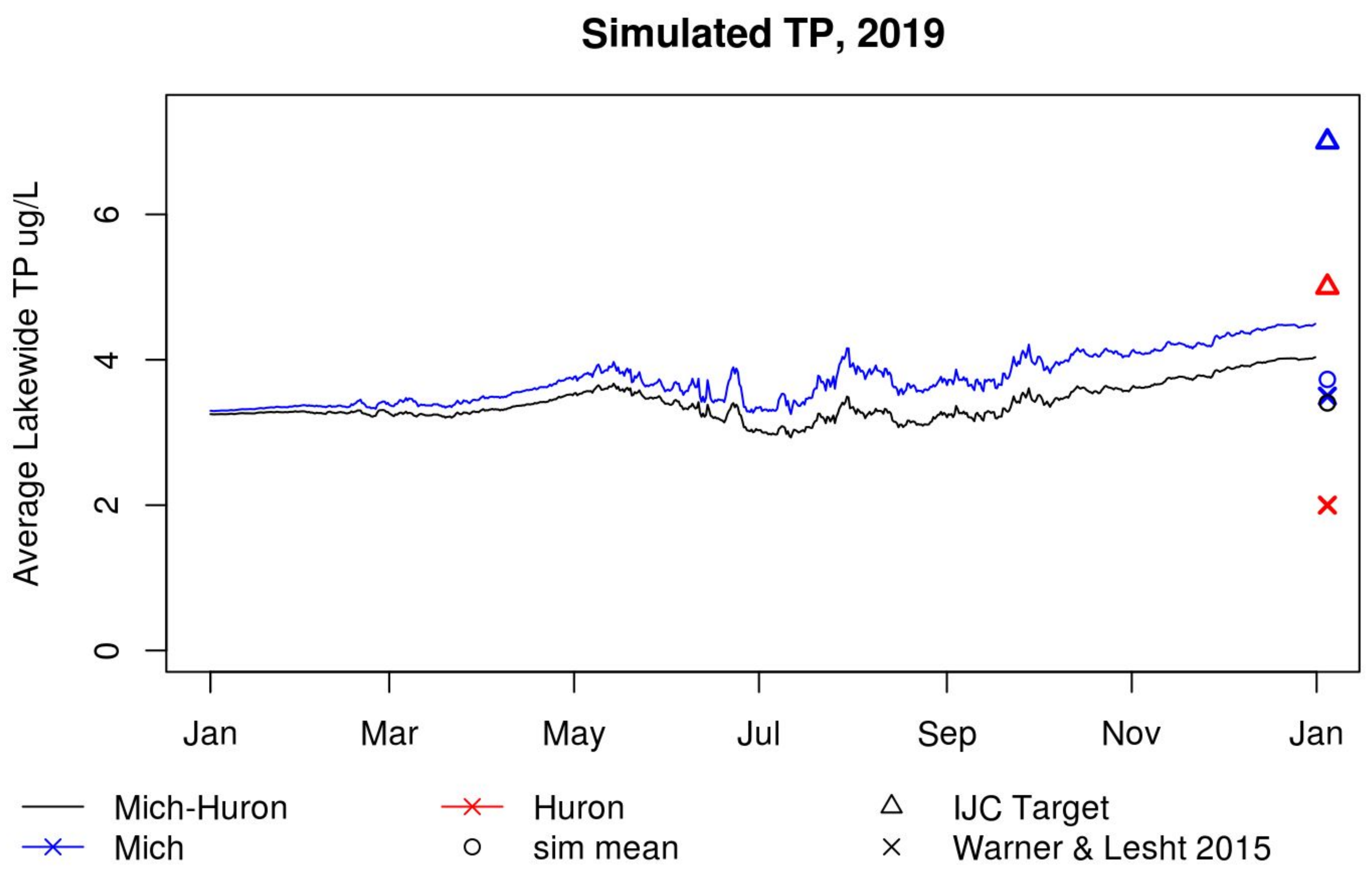
Lakewide time series

- Extract modeled surface level TP data at each time step and save to an .Rdata file (done in a separate script) and then read in data for plotting

```
po4<- list2[[which(list2Names=="PO4")]]
po4<-po4 * nutList$PO4$scale


tp<-po4
TPVars <- nutList[[which(names(nutList)=="TP")]]$var
TPVars <- TPVars[!(TPVars %in% "PO4")]
for(i in 1:length(TPVars)){
  tp <- tp + list2[which(list2Names==TPVars[i])][[1]]*ptoc[1]
}
tp <- p2*nutList$TP$scale

tp<-tp[,,] # compress to two dimensional matrix since there i
```

- Calculate volume-weighted, lake wide mean TP for each time step
    - To calculate 'Vols':
        - Read in art1 (node areas) from output netcdf and h (node depths) and then calculate node volumes:
            - Vols = art1 * h

```
# define function for calculating volume weighted mean used in apply()
vol.weighted.mean<-function(x){
  # print(length(x))
  if(length(x) < 90806){ # Just Lake Michigan with Green bay
    nodes = ro1
  }
  if(length(x) == 40313){ # Just Lake Michigan without Green bay
    nodes = ro2
  }
  if(length(x) == 90806){# Entire domain (Lake Mich-Huron)
    nodes = coordsn$node
  }

  # vols_mat<-matrix(vols,nrow=length(vols),ncol=ncol(x),byrow=FALSE)
  weighted_average<-sum(x*vols[nodes])/sum(vols[nodes])
  return(weighted_average)
}


# time series of lakewide average chl
tp_ts<- apply(tp,2,vol.weighted.mean)
mi_tp_ts<- apply(tp[nodes,],2,vol.weighted.mean)
mi_tp_ts_nogb<- apply(tp[ro2,],2,vol.weighted.mean)
```


Simulated TP, 2019
Average Lakewide TP ug/L
0
2
4
6
Jan
Mar
May
Jul
Sep
Nov
Jan
Mich-Huron
Mich
Huron
sim mean
IJC Target
Warner & Lesht 2015

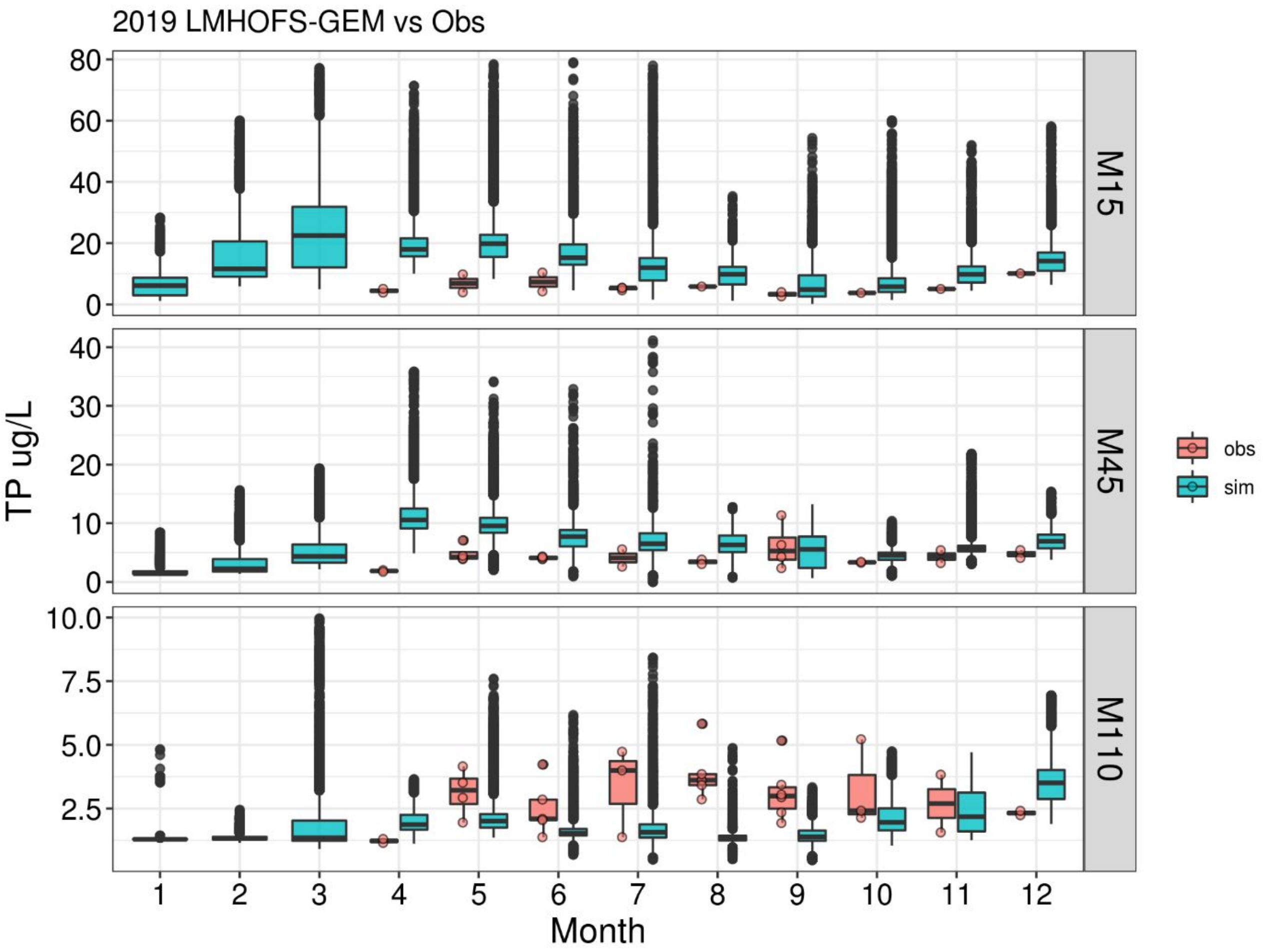
Compare model values to observations at Muskegon transect

```
# Read in Muskegon data from Steve Pothoven and compare model values to in situ values
library(readxl)

muskegon<-read_excel("/mnt/projects/hpc/alsip/lmhofs_gem/LakeData/LMICH_WQ_Rowe.xlsx",sheet = "Readin2R")

muskegon<-as.data.frame(muskegon)

muskegon$PP<-as.numeric(muskegon$PP)

# conform station naming to stas1d
# Alpha/alpha = M15
# Beta/beta = M45
# Omega/omega = M110
muskegon$Station[which(muskegon$Station == "Omega")]<-"M110"
muskegon$Station[which(muskegon$Station %in% c("alpha","Alpha") )]<-"M15"
muskegon$Station[which(muskegon$Station %in% c("beta","Beta") )]<-"M45"

# change colname so it matches stas1d
colnames(muskegon)[which(colnames(muskegon) == "Station")]<-"station"
#stas1d
# station     lon      lat         X          Y      node
# 1      M15 -86.344 43.18817 -226.5529 214.4337   15732
# 2      M45 -86.432 43.18817 -233.6943 214.8711   17441
# 3     M110 -86.536 43.18817 -242.1334 215.3975   18690

# read in Muskegon depth time data
list.files(archive_path, ".Rdata")

load(paste0(archive_path,"depthTimeList.Rdata"))
time1<-depthTimeList$time1

stas1d$node<-depthTimeList$nodes # i verified that the nodes are in order of station from shallowest to deepest (M15 -> M45 -> M110)

# add node number to muskegon data frame
muskegon<-merge(muskegon, stas1d[,c("station","node")], by ="station")
```

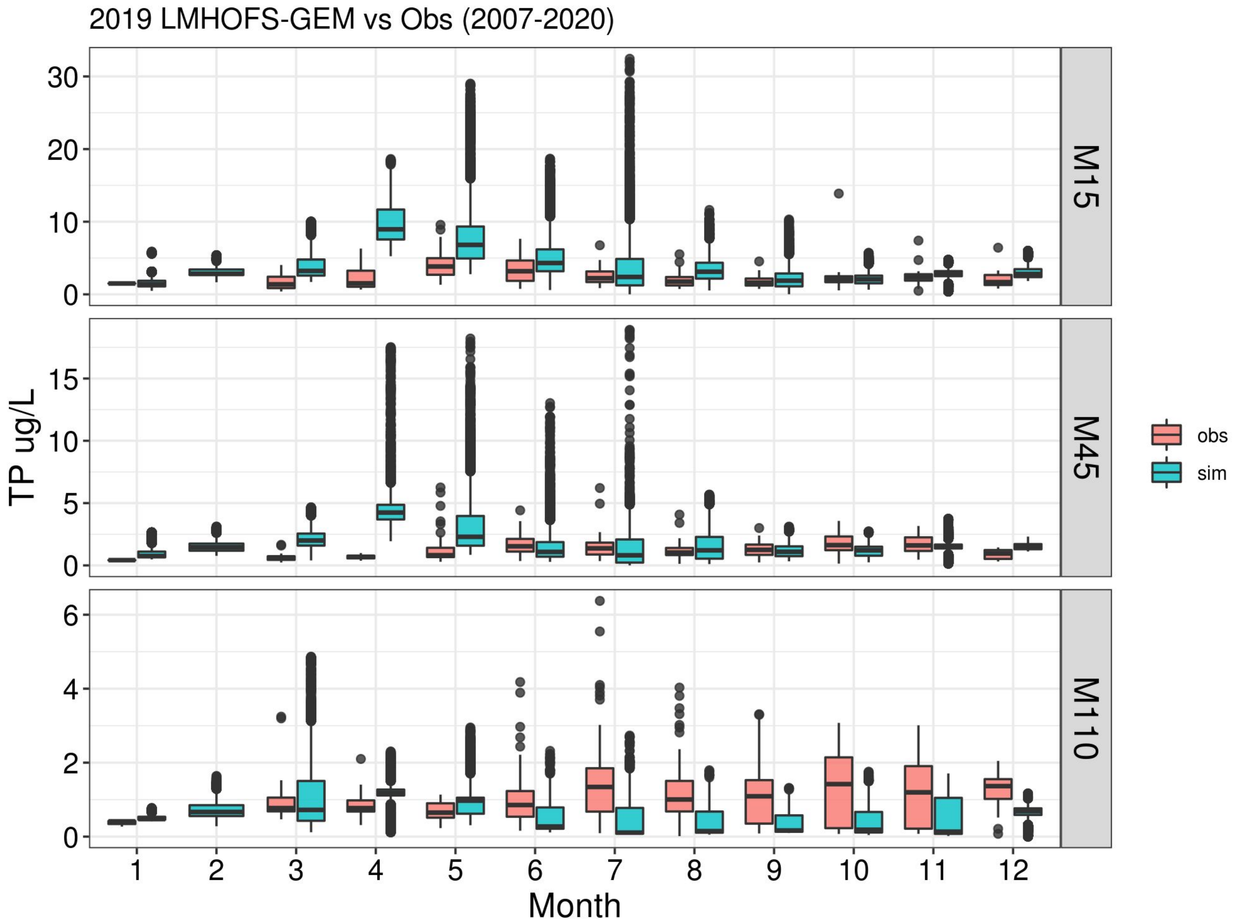
Year	Month	Day	DOY	Station	Depth	Surface Te	secchi	CHL	TP	PP	TDP
1994	10	12	285	B	5		5	2.54	4.60	3.85	0.75
1994	10	12	285	B	10		5	1.85	5.15	3.75	1.40
1994	10	12	285	B	20		5	1.74	5.05	4.00	1.05
1994	10	12	285	B	30		5	1.07	3.35	4.20	
1994	10	12	285	B	50		5	0.25	4.45	2.70	1.75
1994	10	12	285	B	90		5	0.14	6.35	3.40	2.95
1994	11	10	314	B	5			2.53	5.90	3.50	2.40
1994	11	10	314	B	10			2.68	7.55	8.40	
1994	11	10	314	B	20			2.24	6.75	3.70	3.05
1994	11	10	314	B	30			1.20	5.80	3.15	2.65
1994	11	10	314	B	50			0.35	6.45	2.35	4.10

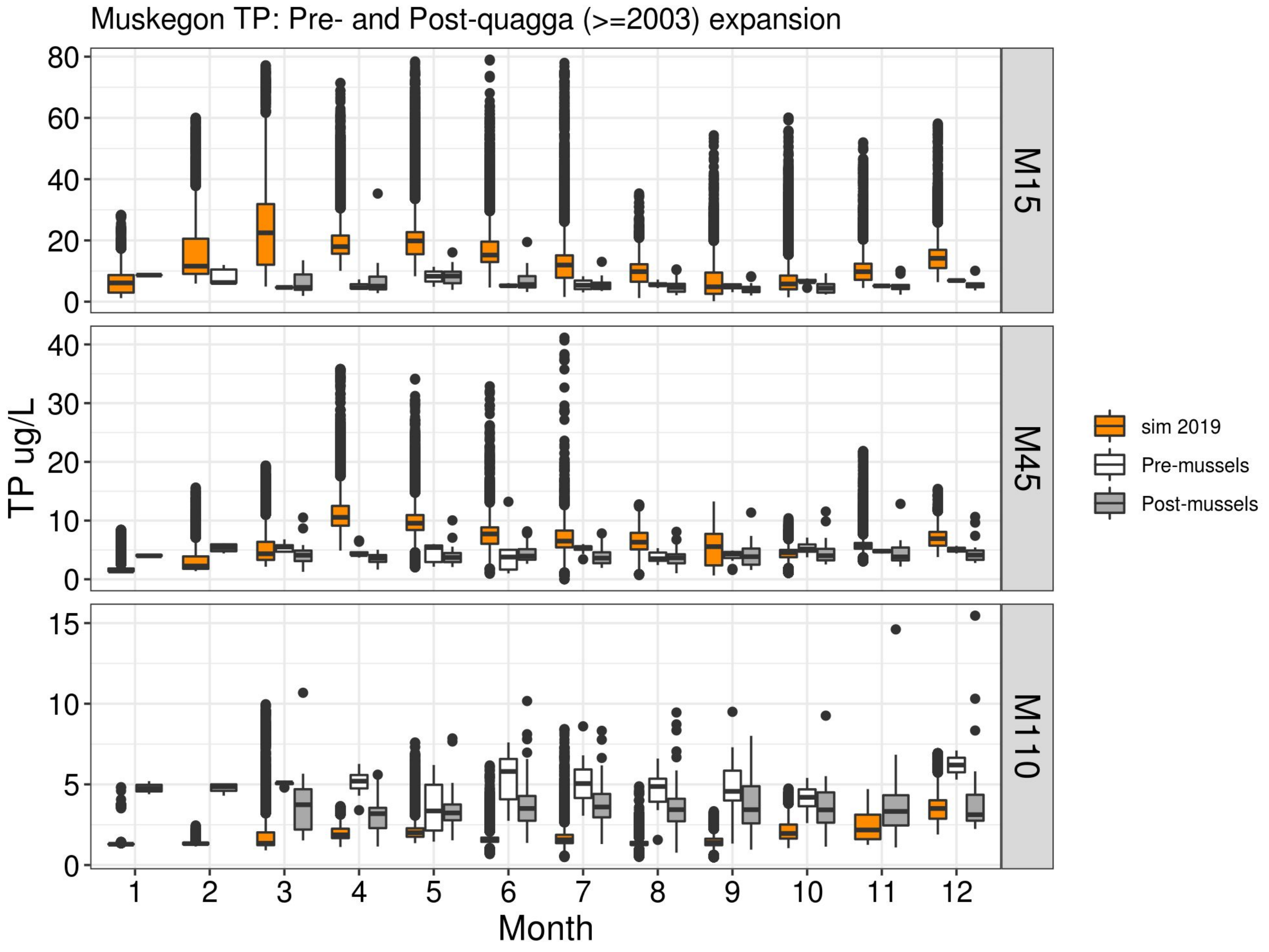
Box plot comparisons

Comparison to same year observations



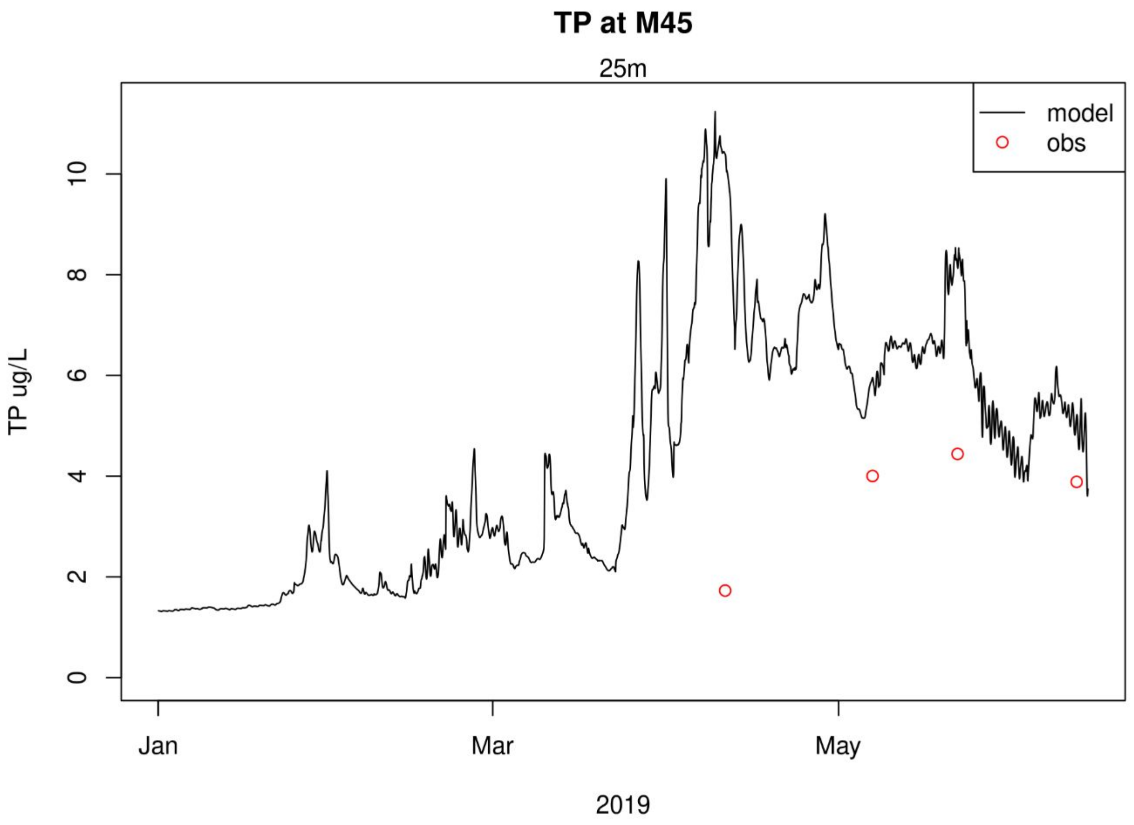
Comparison to historical distribution of observations

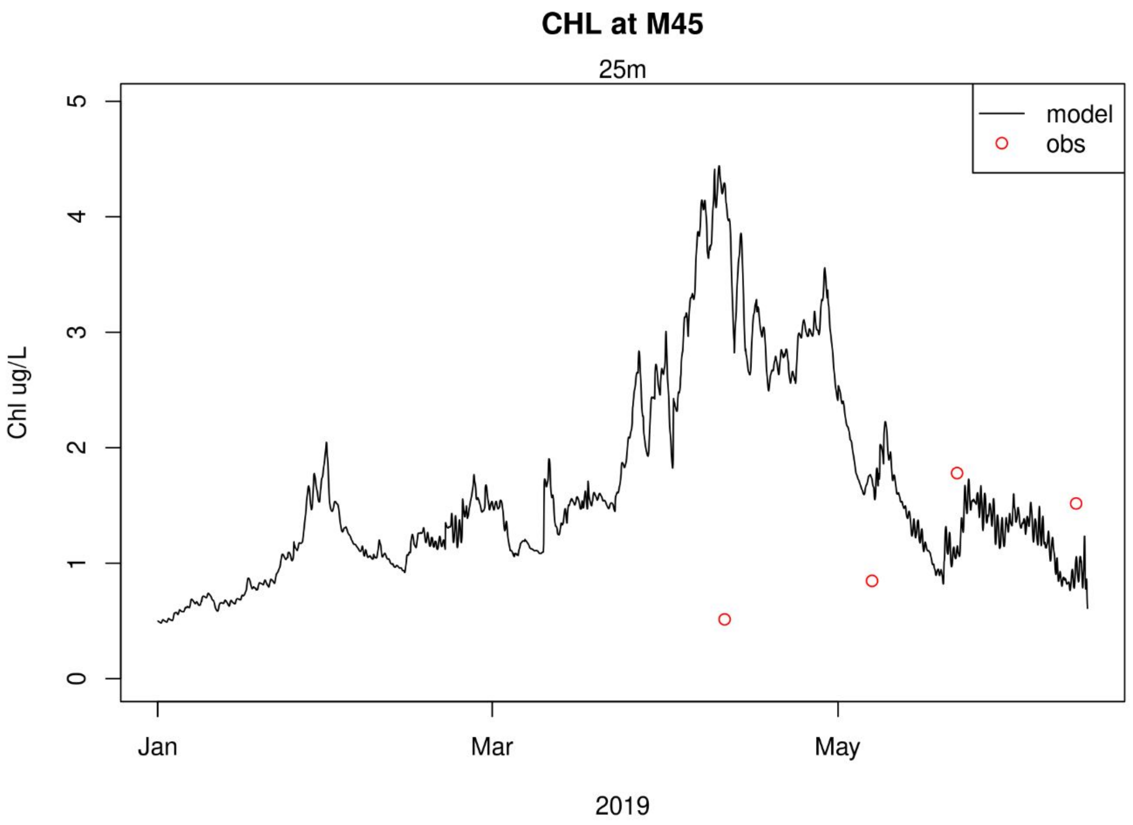




Muskegon TP: Pre- and Post-quagga (>=2003) expansion
TP ug/L
Month
M15
M45
M110
sim 2019
Pre-mussels
Post-mussels
0
20
40
60
80
0
10
20
30
40
0
5
10
15
1
2
3
4
5
6
7
8
9
10
11
12

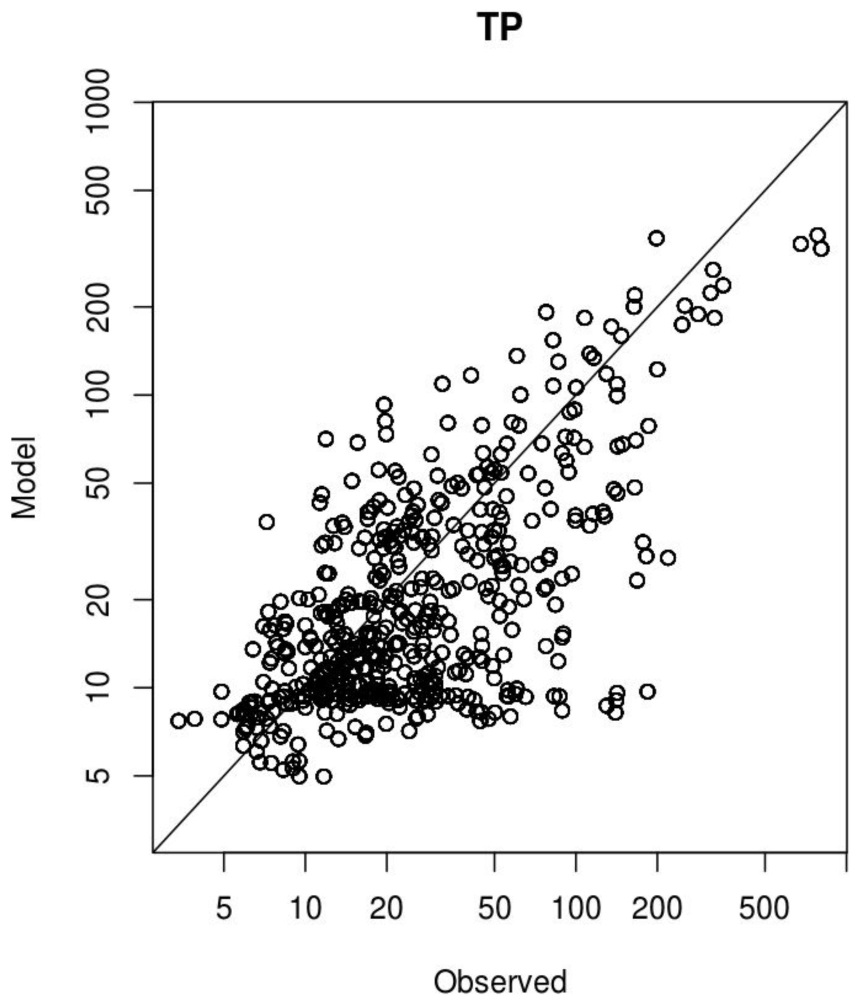
Time series at specified depth

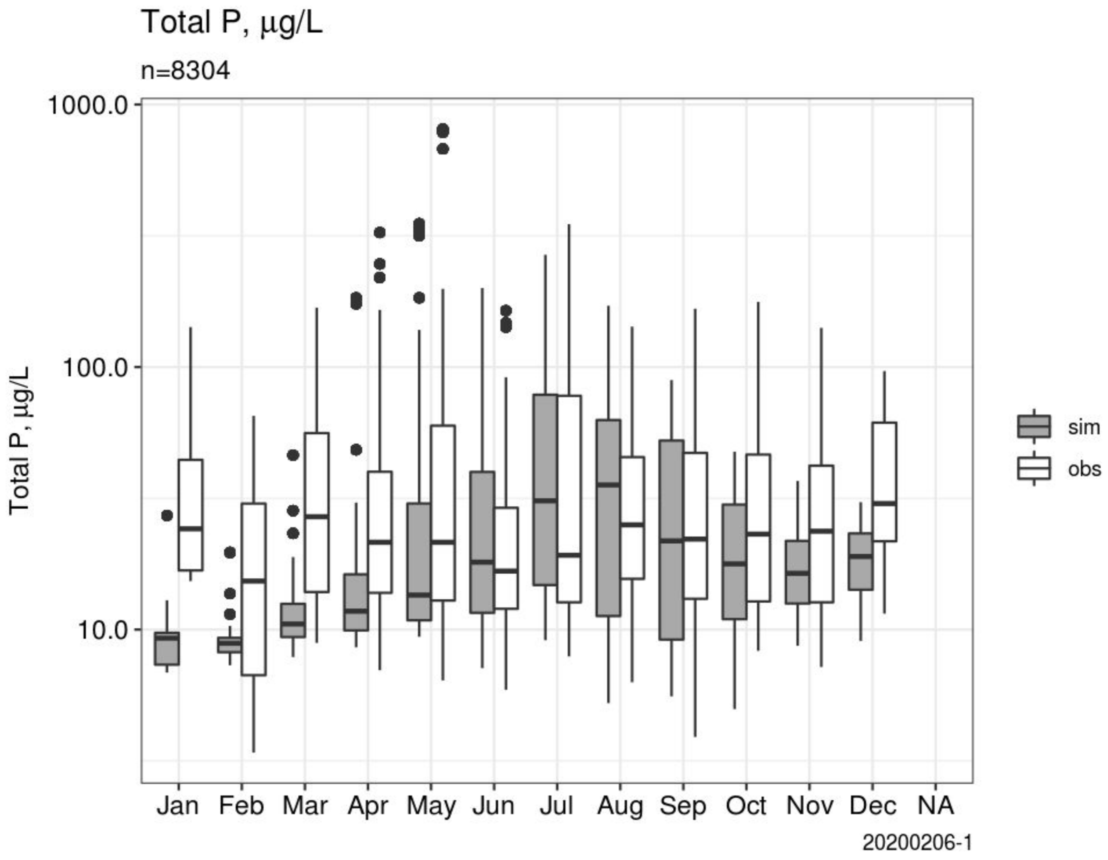




Other model validation scripts done for Lake Erie that could be adapted for Lake Michigan

```
##################################################
# Store observed values and their corresponding simulated values to a data frame for skill metric calculation
##################################################
obsOut1<-dAves1

# find the sim values corresponding to obs
obsOut1$sim <- NA
obsOut1$nutName<-nutName
for(ri in 1:nrow(obsOut1)){
  dist <- spDistsN1(pts=as.matrix(coordsn[,c("lon","lat")]), pt=c(obsOut1$lon[ri],obsOut1$lat[ri]),longlat=FALSE)
  xx <- order(dist)[1:4] # use the four closest nodes to calculate best IDW average at station
  nodes <- coordsn$node[xx]
  weights <- 1/dist[xx]
  obsOut1$sim[ri] <- sum(values[nodes]*weights/sum(weights))
}
```





BIAS	RMSE	COR	OBSMEAN	OBSSD	SIMSD	COUNT
-12.32	49.96	0.79	45.2	76.26	47.34	8301