Phytoplankton North Adriatic

Peter M.J. Herman

9 oktober 2018

Description of dataset copied from EMODNET metadata

We provide datasets that include parameters regarding phytoplankton community, primary production and chl a measurements. Chlorophyll a concentrations analysis started from 1972 till 2017 and included coastal and open waters stations. Phytoplankton community analysis consist of datasets started from 1952-2017 while primary production measurements were obtained from 1962.

Datasets were gained from projects Vir- Konavle started from 1976. This project included coastal stations along Middle and South Adraitic and was performed seasonaly. National monitoring programe Jadran started 1998 and among coastal stations, measurements were performed at open waters (three transects across Adriatic). The project was active till 2012. Operational and regulatory monitoring of transitional and coastal waters of 2014-2018 is ongoing project. Primary production has been measured monthly form 1962 on two station in coastal and open waters.

Contributors

Institute of Oceanography and Fisheries (IZOR), more, data creator Bakrač, Ana Bužančić, Mia Heliodor, Prelesnik Nincevic Gladan, Zivana Skejić, Sanda

Publications based on this dataset

Gladan, Z.N. et al. (2010). Inter-decadal variability in phytoplankton community in the Middle Adriatic (Kaštela Bay) in relation to the North Atlantic Oscillation. Est. Coast. 33(2): 376-383. https://hdl.handle.net/10.1007/s12237-009-9223-3, more Marasovic, I. et al. (2005). Long-term changes of basic biological and chemical parameters at two stations in the middle Adriatic. J. Sea Res. 54(Spec. Issue 1): 3-14. http://dx.doi.org/10.1016/j.seares.2005.02.007, more Marasovic, I. et al. (2005). Long-term changes of basic biological and chemical parameters at two stations in the middle Adriatic, in: Mills, D.K. et al. (Ed.) Contrasting approaches to understanding eutrophication effects on phytoplankton. Journal of Sea Research, 54(1): pp. 3-14

Publications describing this dataset

Skejic, S. et al. (2015). Long-term regulating mechanisms of phytoplankton biomass in a traditional shellfish aquaculture area. Fresenius Environ. Bull. 24(9a): 3001-3013, more

URLs

Dataset information: http://baltazar.izor.hr/azopub/beionet_tab

Obtaining the dataset

Two files were downloaded from the EMODNET Biology download centre, one containing the main records, and one containing the measurements and facts.

Files are read in and merged in the following code.Note that the webservices from the EMODNET Biology portal are used in this example.url1 points to the main data set, url2 finds the 'measurements and facts' file for this dataset. There is a backup using stored files.

```
url1<-paste("http://geo.vliz.be/geoserver/wfs/ows?service=WFS&version=1.1.0&",
            "request=GetFeature&typeName=Dataportal%3Aeurobis&resultType=results",
            "&viewParams=where%3Adatasetid+IN+%285718%29%3Bcontext%3A0100",
            "&propertyName=datelastmodified%2Ccatalognumber%2Cscientificname%2C",
            "aphiaid%2Cscientificname_accepted%2Caphiaidaccepted%2Ckingdom%2Cphylum",
            "%2Cclass%2Corder%2Cfamily%2Cgenus%2Csubgenus%2Cspecies%2Csubspecies%2C",
            "yearcollected%2Cmonthcollected%2Cdaycollected%2Clongitude%2Clatitude%2C",
            "coordinateprecision%2Cminimumdepth%2Cmaximumdepth%2Csex%2Clifestage%2C",
            "individualcount%2Cobservedindividualcount%2Cobservedweight%2Csamplesize%2C",
            "datasetid%2Cseason%2Ctaxonlsid%2Coccurrenceid&outputFormat=csv",sep="")

url2<-paste("http://geo.vliz.be/geoserver/Dataportal/ows?service=WFS&version=1.0.0&",
            "request=GetFeature&typeName=Dataportal:eurobis_measurementorfacts",
            "&&viewParams=where:dataproviderid=1073&outputformat=csv",sep="")

#url1<-"basicfile_phyto_NA.csv"
#url2<-"Meas_fact_phyto_NA.csv"
pna<-read.csv(file=url1,header=T,stringsAsFactors = F)
pna2<-read.csv(file=url2,header=T,stringsAsFactors = F)
pna<-merge(pna,pna2,by="occurrenceid")
save(pna,file="pna.Rdata")
```

A check on the coordinates shows that most data have been collected at the point (16.38167,43.51833) but in 2007 and 2015 spatial surveys have been made. We concentrate here on the time series at the central point. We make basic graphs of all species at this point, as time series.

In order to do so, we first add a date column that can be used as x axis in the graphs. For groups that have no proper accepted scientific name in the files, we use the field scientificname instead. Next, we make a cross-table, which is important to have all zeroes (species was looked for but not found). From the cross-table we calculate the frequency of all species. We only retain here the species that have been found at least 15 times in the time series. There are many more rare species, often with just a single finding, but we do not graph them here.

Finally, we make basic graphs of the abundance of all species (double squareroot transformed) to have a first impression of the time series.

```
require(reshape2)
```

```
## Loading required package: reshape2
```

```
## Warning: package 'reshape2' was built under R version 3.2.5
```

```
load("pna.Rdata")
##############################################################################
######## only select data from the permanent station
##############################################################################
pna<-pna[round(pna$longitude,2)==16.38 & round(pna$latitude,2)==43.52,]
##############################################################################
#######  construct dates of sampling
```

```
##############################################################################
pna$daycollected[is.na(pna$daycollected)]<-1
for(i in 1:nrow(pna)){
  pna$datechr[i]<-paste(pna$yearcollected[i],pna$monthcollected[i],
                        pna$daycollected[i],sep="/")
}
pna$date<-as.Date(pna$datechr,format="%Y/%m/%d")
##############################################################################
#### use field 'scientificName' if no 'scientificName_accepted' is available ###
##############################################################################
pna$scientificname_accepted[pna$scientificname_accepted==""]<-
                      pna$scientificname[pna$scientificname_accepted==""]
##############################################################################
# give a DisplayClass to all species
##############################################################################
pna$displayclass<-pna$class
pna$displayclass[pna$scientificname=="Coccolithophora sp."]<-"Coccolithophora"
pna$displayclass[pna$scientificname=="Nanoflagellates spp."]<-"Nanoflagellates"
pna$displayclass[pna$scientificname=="Dinoflagellata < 20Âµm"]<-"Dinophyceae"
##############################################################################
# remove Ophioaster sp.
##############################################################################
weg<-which(pna$scientificname=="Ophioaster sp.")
pna<-pna[-weg,]
##############################################################################
# make crosstab to have all zeroes present in dataset
##############################################################################
pnact<-dcast(pna,formula=date~scientificname_accepted,fun.aggregate = sum,
             value.var = "measurementvalue")
##############################################################################
# make dataframe with species names and their displayclass
##############################################################################
spsel<-as.data.frame(unique(cbind(pna$scientificname_accepted,pna$displayclass)))
names(spsel)<-c("displayName","displayClass")
##############################################################################
# make dataframe of dates, years, months
##############################################################################
dates<-data.frame(date=pnact[,1],
                  year=as.numeric(as.character(pnact[,1],format="%Y")),
                  month=as.numeric(as.character(pnact[,1],format="%m")))
##############################################################################
# remove dates as first column of pnact, so that only species data remain
##############################################################################
pnact<-pnact[,-1]
row.names(pnact)<-dates$date
##############################################################################
# sum all species abundances per displayclass
##############################################################################
classes<-unique(spsel$displayClass)
nclass<-length(classes)
sumclass<-matrix(nrow=nrow(pnact),ncol=nclass,data=0)
for(i in 1:nclass){
  clspec<-which(spsel$displayClass==classes[i])
```

```
  for(j in 1:length(clspec))
    sumclass[,i]<-sumclass[,i]+pnact[,clspec[j]]
}
sumclass<-as.data.frame(sumclass)
names(sumclass)<-classes
##############################################################################
# select the most frequent species
##############################################################################
spfr<-apply(pnact,2,FUN=function(x) length(x[x>0]))
pnact<-pnact[,which(spfr>15)]
##############################################################################
#make species list
##############################################################################
splist<-names(pnact)
nsp<-length(splist)
weg<-which(!spsel$displayName%in%splist)
spsel<-spsel[-weg,]
##############################################################################
# store the data needed by the Shiny app
##############################################################################
Phytoplankton<-cbind(dates,sumclass,pnact)
clsel<-data.frame(displayName=classes,displayClass=classes)
speciesPhyto<-c(clsel,spsel)
save(dates,speciesPhyto,Phytoplankton,file="Phytoplankton.Rdata")
##############################################################################
# make control graphs of all species in a pdf
##############################################################################
pdf("speciesgraphs.pdf")
for (i in 1:nsp){
  plot(dates$date,sqrt(sqrt(pnact[,i])),main=splist[i],xlab="Time",
       ylab="sqrt(sqrt(Abundance))",type="b")
}
dev.off()
```

```
## pdf
##   2
```

Multivariate analysis

A basic multivariate analysis of the data has been performed. This analysis shows major trends in the data: seasonal variation and long-term changes ('trend') in the community composition. In the code we use the community matrix prepared in the preceding section. We perform a simple PCA. Based on the PCA scores of the stations, we calculate mean scores for years and months. These will be used in the plots.

The plot function does a number of operations. First, it displays the mean scores on axes 1 and 2 for the years and the months. It adds information on what represents the long-term trend and the seasonal variation. It plots the species scores on top of this graph, but increasing the scale for visibility. Among the species vectors, one or a group can be selected to be highlighted in green. We can give a species or "displayClass" name to the routine, and than the selected species, or all species in the class, are highlighted. In this way we can see how the different species and species groups have contributed to the main trends in the community composition.

Sample plots for all species and species groups are stored in a pdf file for reference.

```
library(ade4)
```

```
## Warning: package 'ade4' was built under R version 3.2.3
```

```
################################################################################
# double square root transformation
################################################################################
phsp<-sqrt(sqrt(pnact))
################################################################################
# calculate PCA
################################################################################
pca_phsp<-dudi.pca(phsp,scannf=F,nf=2)
################################################################################
# calculate centroids of years and months, needed for plots
################################################################################
pca_phsp$li$years<-as.factor(dates$year)
pca_phsp$li$months<-as.factor(dates$month)
mnsax1<-aggregate(Axis1~years,pca_phsp$li,FUN=mean)$Axis1
mnsax2<-aggregate(Axis2~years,pca_phsp$li,FUN=mean)$Axis2
mmnsax1<-aggregate(Axis1~months,pca_phsp$li,FUN=mean)$Axis1
mmnsax2<-aggregate(Axis2~months,pca_phsp$li,FUN=mean)$Axis2
yrmns<-data.frame(ax1=mnsax1,ax2=mnsax2,labs=as.character(2002:2016))
momns<-data.frame(ax1=mmnsax1,ax2=mmnsax2,labs=as.character(1:12))
################################################################################
# store species scores in spseldf (also needed for plots)
################################################################################
spseldf<-data.frame(ax1=pca_phsp$co[,1],ax2=pca_phsp$co[,2],displayName=splist)
spseldf<-merge(spseldf,spsel,by="displayName")
spseldf$displayClass<-as.character(spseldf$displayClass)
################################################################################
# save results for use by Shiny app
################################################################################
save(yrmns,momns,spseldf,file="multivSpecies.Rdata")
################################################################################
# control plots of the results of the multivariate analysis, in a pdf file
################################################################################
plotmultv<-function (specname){
  par(pty="s")
  plot(yrmns$ax1,yrmns$ax2,col='red',xlim=c(-6,4),ylim=c(-6,4),type="l",asp=1,xlab="",ylab="",
       axes=FALSE,frame.plot=TRUE,main=paste("PCA biplot,",specname,"in green"))
  Axis(side=1, labels=FALSE)
  Axis(side=2, labels=FALSE)
  abline(h=0)
  abline(v=0)
  text(yrmns$ax1,yrmns$ax2,yrmns$labs,col='red',cex=.7)
  lines(momns$ax1,momns$ax2,col='blue')
  text(momns$ax1,momns$ax2,momns$labs,col='blue')
  arrows(-4,4,4,1.5,col='red',lwd=2,code=2)
  text(2,3,'Trend',col='red',cex=1)
  arrows(-.5,2,.5,-2,col='blue',lwd=2,code=3)
  text(.5,-3,'Season',col='blue',cex=1)
  colv<-rep('lightgray',80)
  lwdv<-rep(0.1,80)
```

```
   if(!specname%in%spseldf$displayName){
     colv[which(spseldf$displayClass==specname)]<-'green'
     lwdv[which(spseldf$displayClass==specname)]<-2
   }
  arrows(0,0,spseldf$ax1*5,spseldf$ax2*5,col=colv,lwd=lwdv,code=0)
  if(specname%in%spseldf$displayName){
    spec=which(spseldf$displayName==specname)
    arrows(0,0,spseldf$ax1[spec]*5,spseldf$ax2[spec]*5,col="green",lwd=3,code=0)
    points(spseldf$ax1[spec]*5,spseldf$ax2[spec]*5,col="green",cex=2,pch=16)
  }
}

pdf("multvarplots.pdf")
for(i in 1:8)plotmultv(unique(spseldf$displayClass)[i])
for(i in 1:length(splist))plotmultv(splist[i])
dev.off()
```

```
## pdf
##   2
```

```
################################################################################
```

Code of the Shiny app

The Shiny app consists of three source files, that are called ui.R, server.R and helper.R. The latter contains specific routines for the production of the plots, that are referred to in the other scripts. The ui and server scripts strictly follow rules for the Shiny app, that can be found back in the manuals of Shiny. In the helper routine, we read in the binary files that contain the data and the results of the multivariate analysis, and provide plotting routines using these data. Below we give the three source codes, one per file.

server.R

```
library(shiny)

source("helper.R")

shinyServer(function(input, output,session) {

# the data used - updated whenever a new selection
        outVar1 = reactive({
            mydata = get(input$set1)
            names(mydata)[4:(length(mydata[1,]))]
            })
        observe({
            updateSelectInput(session, "spec.set1",choices = outVar1()
            )})
    output$plot1 <- renderPlot({
        plotspecs(input$set1,input$spec.set1,input$transform)
    })
  output$plot4 <- renderPlot({
```

```
    plotseas(input$set1,input$spec.set1,input$transform)
  })
  output$plM1 <- renderPlot({
    plotmultv(input$set1,input$spec.set1)
  })
    output$about <- renderUI({
      doc
      })
})
```

ui.R

```
library(shiny)
shinyUI(fluidPage(
  titlePanel("LTER North Adriatic plankton series"),
  fluidRow(
    column(4,
           wellPanel(
             selectInput(inputId = "set1", label = "",
                         choices = c("Phytoplankton"),
                         selected = "Phytoplankton"),
             selectInput(inputId = "spec.set1", label = "Choose a species (group)",
                         choices="",selected = "Bacillariophyceae"),
             br(),
             checkboxInput(inputId = "transform",
                           label = strong("value double sqrt transformed"),
                           value = TRUE)
             )
    ),
    column(8,
           tabsetPanel(
             tabPanel("Observations", plotOutput("plot1", height ="375px", width = "100%"),
                                 plotOutput("plot4", height ="375px", width = "100%")),
             tabPanel("Multiv 1",plotOutput("plM1", height ="700px", width = "100%")),
             tabPanel("about",htmlOutput("about"))

           )

    )
  )
))
```

helper.R

```
load("Phytoplankton.Rdata")
load("multivSpecies.Rdata")
doc<-HTML(readLines('about.html'))

ldc<-3
```

```
plotspecs<-function(datset,species,transf){
  tt<-get(datset)
  if(species=="")species<-names(tt)[ldc+1]
  if(!species%in%names(tt))species<-names(tt)[ldc+1]
  ttt<-data.frame(y=tt[,'year'],sp=tt[,species])
  if(transf) ttt$sp<-sqrt(sqrt(ttt$sp))
  miy<-min(ttt$y)
  may<-max(ttt$y)
  boxplot(sp~y,data=ttt,main=species,ylab='abundance',outline=T,
          xlim=c(1,15),at=(miy-2001):(may-2001))
}
plotseas<-function(datset,species,transf){
  tt<-get(datset)
  if(species=="")species<-names(tt)[ldc+1]
  if(!species%in%names(tt))species<-names(tt)[ldc+1]
  ttt<-data.frame(m=tt[,'month'],sp=tt[,species])
  if(transf) ttt$sp<-sqrt(sqrt(ttt$sp))
  miy<-min(ttt$m)
  may<-max(ttt$m)
  boxplot(sp~m,data=ttt,main=species,ylab='abundance',outline=T,xlim=c(1,12),at=miy:may)
}


plotmultv<-function (datset,specname){
  if (datset=="Phytoplankton"){
      par(pty="s")
      plot(yrmns$ax1,yrmns$ax2,col='red',xlim=c(-6,4),ylim=c(-6,4),type="l",asp=1,xlab="",ylab="",
           axes=FALSE,frame.plot=TRUE,main=paste("PCA biplot,",specname,"in green"))
      Axis(side=1, labels=FALSE)
      Axis(side=2, labels=FALSE)
      abline(h=0)
      abline(v=0)
      text(yrmns$ax1,yrmns$ax2,yrmns$labs,col='red',cex=.7)
      lines(momns$ax1,momns$ax2,col='blue')
      text(momns$ax1,momns$ax2,momns$labs,col='blue')
      arrows(-4,4,4,1.5,col='red',lwd=2,code=2)
      text(2,3,'Trend',col='red',cex=1)
      arrows(-.5,2,.5,-2,col='blue',lwd=2,code=3)
      text(.5,-3,'Season',col='blue',cex=1)
      colv<-rep('lightgray',80)
      lwdv<-rep(0.1,80)

      if(!specname%in%spseldf$displayName){
        colv[which(spseldf$displayClass==specname)]<-'green'
        lwdv[which(spseldf$displayClass==specname)]<-2
      }
      arrows(0,0,spseldf$ax1*5,spseldf$ax2*5,col=colv,lwd=lwdv,code=0)
      if(specname%in%spseldf$displayName){
        spec=which(spseldf$displayName==specname)
        arrows(0,0,spseldf$ax1[spec]*5,spseldf$ax2[spec]*5,col="green",lwd=3,code=0)
        points(spseldf$ax1[spec]*5,spseldf$ax2[spec]*5,col="green",cex=2,pch=16)
      }
   }else{
```

```
    frame()
  }
}
```