#'@name chlcoef

#'@title Calculation of extracted versus fluoresced chlorophyll coefficients

#'@description Calculate extracted versus fluoresced chlorophyll coefficients

#'@param yearmon numeric date of survey

#'@param varlist character vector of variable names to use in model

#'@param remove.flags logical trim dataset based on flags?

#'@param overwrite logical overwrite previous coefficients?

#'@param streamcov numeric percentage of non-missing streaming data used to exclude variables under consideration

#'@param checkvif logical check final equation for multicollinearity using VIF?

#'@param fdir file.path to data folder

#'@param logtransform logical

#'@param corcut numeric used to screen bad variables with bad correlation coefficients

#'@param polypcut numeric used to switch to a polynomial regression fit

#'@details this function should be interactive

#'@export

#'@importFrom MASS stepAIC

#'@importFrom car vif

#'@importFrom stats complete.cases as.formula cor fitted lm pf

#'@importFrom utils read.csv write.csv

#'@importFrom graphics abline

#'@examples

#'\dontrun{

#'chlcoef(201308)

#'chlcoef(201606, varlist = c("chl.ug.l", "turb.ntu", "phycoc", "c6cdom", "c6chl", "phycoe"), corcut = 0.75)

#'}

chlcoef <- function(yearmon, varlist = NA, remove.flags = TRUE, overwrite = FALSE, fdir = getOption("fdir"), polypcut = 0.6, corcut = 0.7, streamcov = 0.5, checkvif = TRUE, logtransform = FALSE){

dt <- grabget(yearmon, remove.flags = remove.flags)

if(logtransform == TRUE){

dt[,"chla"] <- log(dt[,"chla"])

}

#if(remove.flags==TRUE){

# dt<-dt[-which(dt$flags=="s"),]

#}

#choose variables

if(all(is.na(varlist))){

varlist <- c("chla", "cdom", "chlaiv", "phycoe", "c6chl", "phycoc", "c6cdom")

}else{

varlist <- c("chla", varlist)

}

varlist <- varlist[sapply(varlist, function(x) sum(!is.na(dt[,x])) > 1)]

#VARLIST NEEDS TO BE UPDATED TO ALSO INCLUDE: "c6chla.R", change "c6chl" to "c6chla.B" (I think?), add "bga.pe.rfu"

#exclude variables with streaming data less than streamcov

streamdata <- streamget(yearmon = yearmon, qa = TRUE)

streamvarlist <- varlist

streamvarlist[which(varlist == "chlaiv")] <- "chla"

streamvarlist[which(streamvarlist == "c6chl")] <- "c6chla"

streamvarlist <- streamvarlist[-1]

streamvarlist <- streamvarlist[sapply(streamvarlist, function(x) sum(!is.na(streamdata[,x])) / nrow(streamdata)) > 0.74]

if(length(varlist) > 2){

cormat <- cor(dt[,varlist], use = "complete")[-1, 1]

varlist <- names(cormat[abs(cormat) > corcut])

}

if(!length(varlist)>0){stop("linear correlations too low")}

lmeq <- as.formula(paste("chla ~ ", paste(varlist,collapse="+")))

fit <- lm(lmeq,data = dt[complete.cases(dt[,varlist]),])

if(length(varlist)>1){

dt<-dt[apply(dt[,match(varlist,names(dt))],1,function(x) !all(is.na(x))),]

}else{

dt<-dt[!is.na(dt[,match(varlist,names(dt))]),]

#dt[,match(varlist,names(dt))]<-dt[,match(varlist,names(dt))][!is.na(dt[,match(varlist,names(dt))])]

}

message("Initial correlation matrix")

print(cormat)

message("MLR with all variables...")

print(summary(fit))

polyp<-FALSE

if((summary(fit)$adj.r.squared) < polypcut){#poly fit

lmeq<-as.formula(paste("chla ~ ", paste("poly(",varlist,",2,raw=TRUE)",collapse="+")))

fit<-lm(lmeq,data=dt[complete.cases(dt[,varlist]),c("chla",varlist)])

polyp <- TRUE

}

if(summary(fit)$r.squared<0.1){

stop("bad fit. r-squared not high enough.")

}

if(length(varlist)>1){

message("Reducing full equation by backwards AIC variable selection...")

saic<-MASS::stepAIC(fit)#pick reduced eq according to maximized AIC (remove terms with a the smallest (largest negative score))

rmlist<-as.character(saic$anova$Step)

rmlist<-gsub("-","",rmlist)

rmlist<-gsub(" ","",rmlist)

rmlist<-rmlist[nchar(rmlist)>1]

rmlist<-gsub("poly\\(","",rmlist)

rmlist<-gsub(",2,raw=TRUE\\)","",rmlist)

varlist<-varlist[is.na(match(varlist,rmlist))]

}

message("checking for redundacy in paired variables")

pairedmat<-matrix(c("chlaiv","c6chl","c6cdom","cdom"),ncol=2,byrow = TRUE)

pairedmat<-pairedmat[apply(pairedmat,1,function(x) all(!is.na(match(x,varlist)))),]

rmlist<-names(which.min(cormat[which(!is.na(match(names(cormat),pairedmat)))]))

if(length(rmlist)>0){

varlist<-varlist[-which(varlist==rmlist)]

lmeq<-as.formula(paste("chla ~ ", paste(varlist,collapse="+")))

fit<-lm(lmeq,data=dt[complete.cases(dt[,varlist]),])

}

#update variables to match varlist from above; need to add "c6chla.R", change "c6chl" to "c6chla.B", add "bga.pe.rfu"

if(polyp==TRUE){

lmeq<-as.formula(paste("chla ~ ", paste("poly(",varlist,",2,raw=TRUE)",collapse="+")))

fit<-lm(lmeq,data=dt[complete.cases(dt[,varlist]),c("chla",varlist)])

}else{

lmeq<-as.formula(paste("chla ~ ", paste(varlist,collapse="+")))

fit<-lm(lmeq,data=dt)

}

message("Final AIC reduced fit")

print(summary(fit))

if(checkvif==TRUE){

message("Checking VIF...")

#examine VIF; If GVIF > 5:10, then remove colinear terms, Helsel and Hirsh 2002

if(length(fit$coefficients)>2 & (polyp!=TRUE & length(fit$coefficients>3))){

viftest<-car::vif(fit)

if(polyp==TRUE){

viftest<-car::vif(fit)[,3]^2

}

while(max(viftest)>10 & length(varlist)>2){

rmlist<-names(which.max(viftest))

if(polyp==TRUE){

rmlist<-strsplit(rmlist,",")[[1]][1]

rmlist<-strsplit(rmlist,"\\(")[[1]][2]

}

varlist<-varlist[is.na(match(varlist,rmlist))]

lmeq<-as.formula(paste("chla ~ ", paste(varlist,collapse="+")))

if(polyp==TRUE){

lmeq<-as.formula(paste("chla ~ ", paste("poly(",varlist,",2,raw=TRUE)",collapse="+")))

}

fit<-lm(lmeq,data=dt[complete.cases(dt[,varlist]),c("chla",varlist)])

viftest<-car::vif(fit)

if(polyp==TRUE){

viftest<-car::vif(fit)[,3]^2

}

}

}

}

if(summary(fit)$adj.r.squared==1){

message("Overfit reducing to simple linear eq.")

lmeq<-as.formula(paste("chla ~ ", paste(varlist,collapse="+")))

fit<-lm(lmeq,data=dt)

polyp <- FALSE

}

message("Final statistical fit")

print(summary(fit))

if(logtransform==TRUE){

plot(exp(fitted(fit)),exp(dt$chla[complete.cases(dt[,varlist])]),ylab="Ext. chl",main="1 to 1 line")

}else{

plot(fitted(fit),dt$chla[complete.cases(dt[,varlist])],ylab="Ext. chl",main="1 to 1 line")

}

abline(0,1)

#message("Accept final fit?")

# #optimize for low values (<1)?####

# #hist(dt2$CHLa)

# dt2<-dt2[dt2$CHLa<2.5,]

# #full eq

# fit<-lm(CHLa~poly(CDOM,2,raw=T)+poly(CHLAiv,2,raw=T)+poly(Phycoerythrin,1,raw=T)+poly(C6Chla,2,raw=T)+poly(C6cdom,2,raw=T),data=dt3)

# #DF only eq

# fit<-lm(CHLa~poly(CDOM,2,raw=T)+poly(CHLA,2,raw=T),data=dt3)

# stepAIC(fit)

# fit2<-lm(CHLa~poly(CDOM,2,raw=T),data=dt3)

# plot(predict(fit2,dt2),dt2$CHLa,xlim=range(dt2$CHLa))#predict based on full dataset

# plot(predict(fit2,dt3),dt3$CHLa,xlim=range(dt3$CHLa))#predict based on reduced dataset

#

# fit2<-lm(CHLa~poly(C6Chla,2,raw=T),data=dt3)

# plot(fitted(fit2),dt3$CHLa)#inspect 1:1

# abline(a=0,b=1)

# #fit to full dataset

# plot(predict(fit2,dt2),dt2$CHLa)#inspect 1:1

# abline(a=0,b=1)

# #####

#retrieve chla coefficients####

coeflist <- read.csv(file.path(fdir, "DF\_GrabSamples", "extractChlcoef2.csv"), header = TRUE, na.strings = "NA")[,-1]

names(coeflist) <- tolower(names(coeflist))

vartemplate <- c("yearmon", "survey", "date", "cdom", "chl.ug.l", "chlaiv", "phycoe", "c6chl", "c6cdom", "phycoc", "cdom2", "chlaiv2", "phycoe2", "c6chl2", "c6cdom2", "phycoc2", "intercept", "rsquared", "pvalue", "model", "notes")

outtemp <- data.frame(matrix(NA, nrow = 1, ncol = length(vartemplate)))

names(outtemp) <- vartemplate

outcoef <- fit$coefficients

names(outcoef)[1] <- "intercept"

if(polyp == TRUE){

cname <- names(outcoef)[2:length(outcoef)]

cname <- unlist(strsplit(cname,"\\("))

cname <- unlist(strsplit(cname,"\\)"))

cname <- cname[-seq(from=1,to=length(cname)-2,3)]

cname <- matrix(cname,ncol=2,byrow=TRUE)

vname <- unlist(strsplit(cname[,1],","))

cname[,1] <- vname[seq(from=1,to=length(vname),3)]

cname[cname[,2] == 1,2] <- ""

names(outcoef)[2:length(outcoef)]<-paste(cname[,1],cname[,2],sep="")

}

outtemp[match(names(outcoef), names(outtemp))] <- outcoef

outtemp[,"yearmon"] <- yearmon

outtemp[,"date"] <- Mode(dt[,"date"])

outtemp[,"rsquared"] <- summary(fit)$r.squared

lmp <- function (modelobject) {

if (class(modelobject) != "lm") stop("Not an object of class 'lm' ")

f <- summary(modelobject)$fstatistic

p <- pf(f[1], f[2], f[3], lower.tail = FALSE)

attributes(p) <- NULL

return(p)

}

outtemp[,"pvalue"] <- lmp(fit)

model <- outtemp[4:16]

model[1,]<-round(as.numeric(model[1,]),5)

model<-data.frame(matrix(c(model,names(model)),nrow=2,byrow=TRUE))

model<-model[,!is.na(model[1,])]

model[1,]<-sapply(model[1,],as.character)

intercept<-model[1,ncol(model)]

model<-model[,-ncol(model)]

if(length(model)<3){

model<-data.frame(matrix(unlist(model),nrow=2))#new

}

outtemp[,"model"]<-paste("Chla = ",gsub(" ","+",gsub(",","",toString(apply(model,2,function(x) paste(x[1],x[2],sep="\*"))))),"+",intercept,sep="")

if(any(outtemp[,"yearmon"]==coeflist[,"yearmon"],na.rm=TRUE)&overwrite==FALSE){

warning("Fit already exists for this survey. Specify overwrite =TRUE or open file and delete in order to replace.")

}else{

coeflist<-rbind(coeflist,outtemp)

write.csv(coeflist,file.path(fdir, "DF\_GrabSamples", "extractChlcoef2.csv"))

}

fit

}

#output extracted chl raster

#list.files(paste("DF\_Surfaces/",yearmon,"/",sep=""))

# test<-raster(paste("DF\_Surfaces/",yearmon,"/","c6chla",".tif",sep=""))

# test2<-raster(paste("DF\_Surfaces/",yearmon,"/","c6cdom",".tif",sep=""))

# test<-reclassify(test,c(-Inf,0,NA))

# test2<-reclassify(test2,c(-Inf,0,NA))

# plot(test2)

#

# testfinal<-test2\*(-0.004969)+test\*(0.041821)-0.37868

# testfinal<-reclassify(testfinal,c(-Inf,0,0))

# writeRaster(testfinal,filename=paste("DF\_Surfaces/",yearmon,"/",yearmon,"chlext",".tif",sep=""),overwrite=T,format="GTiff")

#

#

# #calculate extracted chlorophyll####

# # chldt<-merge(fulldataset.over,coeflist,by="SURVEY")

# # chldt$one<-chldt$CDOM\_RFU.y\*chldt$CDOM\_RFU.x

# # chldt$two<-chldt$FLUOR\_CHLA\_RFU.y\*chldt$FLUOR\_CHLA\_RFU.x

# # chldt$three<-chldt$PHYCOE\_RFU.y\*chldt$PHYCOE\_RFU.x

# # chldt$four<-chldt$C6CHLA\_RFU.y\*chldt$C6CHLA\_RFU.x

# # chldt$five<-chldt$C6CDOM\_RFU.y\*chldt$C6CDOM\_RFU.x

# # chldt$six<-chldt$CDOM\_RFU2\*chldt$CDOM\_RFU.x\*chldt$CDOM\_RFU.x

# # chldt$seven<-chldt$FLUOR\_CHLA\_RFU2\*chldt$FLUOR\_CHLA\_RFU.x\*chldt$FLUOR\_CHLA\_RFU.x

# # chldt$eight<-chldt$PHYCOE\_RFU2\*chldt$PHYCOE\_RFU.x\*chldt$PHYCOE\_RFU.x

# # chldt$nine<-chldt$C6CHLA\_RFU2\*chldt$C6CHLA\_RFU.x\*chldt$C6CHLA\_RFU.x

# # chldt$ten<-chldt$C6CDOM\_RFU2\*chldt$C6CDOM\_RFU.x\*chldt$C6CDOM\_RFU.x

# }