C3\_model\_and\_fit.R

90921160

Thu Jun 11 14:53:03 2015

Photosyn<- function (VPD = 1.5, Ca = 400, PPFD = 1500, Tleaf = 25, Patm = 101,  
 RH = NULL, gsmodel = c("BBOpti", "BBLeuning", "BallBerry"),   
 g1 = 4, g0 = 0, gk = 0.5, vpdmin = 0.5, D0 = 5, GS = NULL,  
 alpha = 0.24, theta = 0.85, Jmax = 100, Vcmax = 50, gmeso = NULL,   
 Rd0 = 0.92, Q10 = 1.92, Rd = NULL, TrefR = 25, Rdayfrac = 1,   
 EaV = 82620.87, EdVC = 0, delsC = 645.1013, EaJ = 39676.89,   
 EdVJ = 2e+05, delsJ = 641.3615, Ci = NULL, Tcorrect = TRUE,   
 returnParsOnly = FALSE, whichA = c("Ah", "Amin", "Ac", "Aj"))   
{  
 whichA <- match.arg(whichA)  
 gsmodel <- match.arg(gsmodel)  
 inputCi <- !is.null(Ci)  
 inputGS <- !is.null(GS)  
 if (inputCi & inputGS)   
 stop("Cannot provide both Ci and GS.")  
 Rgas <- .Rgas()  
 GCtoGW <- 1.57  
 Jfun <- function(PPFD, alpha, Jmax, theta) {  
 (alpha \* PPFD + Jmax - sqrt((alpha \* PPFD + Jmax)^2 -   
 4 \* alpha \* theta \* PPFD \* Jmax))/(2 \* theta)  
 }  
 g0 <- g0/GCtoGW  
 if (is.null(Rd)) {  
 Rd <- Rdayfrac \* Rd0 \* Q10^((Tleaf - TrefR)/10)  
 }  
 GammaStar <- TGammaStar(Tleaf)  
 Km <- TKm(Tleaf)  
 if (Tcorrect) {  
 Vcmax <- Vcmax \* TVcmax(Tleaf, EaV, delsC, EdVC)  
 Jmax <- Jmax \* TJmax(Tleaf, EaJ, delsJ, EdVJ)  
 }  
 if (returnParsOnly) {  
 return(list(Vcmax = Vcmax, Jmax = Jmax, Km = Km, GammaStar = GammaStar))  
 }  
 J <- Jfun(PPFD, alpha, Jmax, theta)  
 VJ <- J/4  
 if (gsmodel == "BBOpti") {  
 vpduse <- VPD  
 vpduse[vpduse < vpdmin] <- vpdmin  
 GSDIVA <- (1 + g1/(vpduse^(1 - gk)))/Ca  
 }  
 if (gsmodel == "BBLeuning") {  
 GSDIVA <- g1/Ca/(1 + VPD/D0)  
 GSDIVA <- GSDIVA/GCtoGW  
 }  
 if (gsmodel == "BallBerry") {  
 if (is.null(RH))   
 RH <- VPDtoRH(VPD, Tleaf)  
 RH <- RH/100  
 GSDIVA <- g1 \* RH/Ca  
 GSDIVA <- GSDIVA/GCtoGW  
 }  
 if (inputGS) {  
 GC <- GS/GCtoGW  
 A <- 1/GC  
 B <- (Rd - Vcmax)/GC - Ca - Km  
 C <- Vcmax \* (Ca - GammaStar) - Rd \* (Ca + Km)  
 Ac <- (-B - sqrt(B \* B - 4 \* A \* C))/(2 \* A)  
 B <- (Rd - VJ)/GC - Ca - 2 \* GammaStar  
 C <- VJ \* (Ca - GammaStar) - Rd \* (Ca + 2 \* GammaStar)  
 Aj <- (-B - sqrt(B \* B - 4 \* A \* C))/(2 \* A)  
 Ac <- Ac + Rd  
 Aj <- Aj + Rd  
 }  
 else {  
 if (!inputCi) {  
 getCI <- function(VJ, GSDIVA, PPFD, VPD, Ca, Tleaf,   
 vpdmin, g0, Rd, Vcmax, Jmax, Km, GammaStar) {  
 if (PPFD == 0) {  
 vec <- c(Ca, Ca)  
 return(vec)  
 }  
 A <- g0 + GSDIVA \* (Vcmax - Rd)  
 B <- (1 - Ca \* GSDIVA) \* (Vcmax - Rd) + g0 \*   
 (Km - Ca) - GSDIVA \* (Vcmax \* GammaStar + Km \*   
 Rd)  
 C <- -(1 - Ca \* GSDIVA) \* (Vcmax \* GammaStar +   
 Km \* Rd) - g0 \* Km \* Ca  
 CIC <- (-B + sqrt(B \* B - 4 \* A \* C))/(2 \* A)  
 A <- g0 + GSDIVA \* (VJ - Rd)  
 B <- (1 - Ca \* GSDIVA) \* (VJ - Rd) + g0 \* (2 \* GammaStar - Ca)   
 - GSDIVA \* (VJ \* GammaStar + 2 \* GammaStar \* Rd)  
 C <- -(1 - Ca \* GSDIVA) \* GammaStar \* (VJ + 2 \* Rd) - g0 \* 2 \* GammaStar \* Ca  
 if (A == 0)   
 CIJ <- -C/B  
 else CIJ <- (-B + sqrt(B \* B - 4 \* A \* C))/(2 \*   
 A)  
 return(c(CIJ, CIC))  
 }  
 x <- mapply(getCI, VJ = VJ, GSDIVA = GSDIVA, PPFD = PPFD,   
 VPD = VPD, Ca = Ca, Tleaf = Tleaf, vpdmin = vpdmin,   
 g0 = g0, Rd = Rd, Vcmax = Vcmax, Jmax = Jmax,   
 Km = Km, GammaStar = GammaStar)  
 CIJ <- x[1, ]  
 CIC <- x[2, ]  
 }  
 else {  
 if (length(Ci) == 1) {  
 Ci <- rep(Ci, length(Km))  
 }  
 CIJ <- Ci  
 CIJ[CIJ < GammaStar] <- GammaStar[CIJ < GammaStar]  
 CIC <- Ci  
 }  
 if (is.null(gmeso)) {  
 Ac <- Vcmax \* (CIC - GammaStar)/(CIC + Km)  
 Aj <- VJ \* (CIJ - GammaStar)/(CIJ + 2 \* GammaStar)  
 }  
 else {  
 A <- -1/gmeso  
 BC <- (Vcmax - Rd)/gmeso + CIC + Km  
 CC <- Rd \* (CIC + Km) - Vcmax \* (CIC - GammaStar)  
 Ac <- mapply(QUADP, A = A, B = BC, C = CC)  
 BJ <- (VJ - Rd)/gmeso + CIC + 2 \* GammaStar  
 CJ <- Rd \* (CIC + 2 \* GammaStar) - VJ \* (CIC - GammaStar)  
 Aj <- mapply(QUADP, A = A, B = BJ, C = CJ)  
 Ac <- Ac + Rd  
 Aj <- Aj + Rd  
 }  
 if (!inputCi) {  
 lesslcp <- vector("logical", length(Aj))  
 lesslcp <- Aj - Rd < 0  
 if (length(Ca) == 1)   
 Ca <- rep(Ca, length(CIJ))  
 if (length(GammaStar) == 1)   
 GammaStar <- rep(GammaStar, length(CIJ))  
 if (length(VJ) == 1)   
 VJ <- rep(VJ, length(CIJ))  
 CIJ[lesslcp] <- Ca[lesslcp]  
 Aj[lesslcp] <- VJ[lesslcp] \* (CIJ[lesslcp] - GammaStar[lesslcp])/(CIJ[lesslcp]   
 + 2 \* GammaStar[lesslcp])  
 Ci <- ifelse(Aj < Ac, CIJ, CIC)  
 }  
 }  
 hmshape <- 0.9999  
 Am <- (Ac + Aj - sqrt((Ac + Aj)^2 - 4 \* hmshape \* Ac \* Aj))/(2 \*   
 hmshape) - Rd  
 if (!inputCi && !inputGS) {  
 if (whichA == "Ah")   
 GS <- g0 + GSDIVA \* Am  
 if (whichA == "Aj")   
 GS <- g0 + GSDIVA \* (Aj - Rd)  
 if (whichA == "Ac")   
 GS <- g0 + GSDIVA \* (Ac - Rd)  
 }  
 if (inputCi) {  
 if (whichA == "Ah")   
 GS <- Am/(Ca - Ci)  
 if (whichA == "Aj")   
 GS <- (Aj - Rd)/(Ca - Ci)  
 if (whichA == "Ac")   
 GS <- (Ac - Rd)/(Ca - Ci)  
 }  
 if (!inputGS) {  
 GS <- GS \* GCtoGW  
 }  
 if (inputGS) {  
 Ci <- Ca - Am/GC  
 }  
 E <- 1000 \* GS \* VPD/Patm  
 df <- data.frame(Ci = Ci, ALEAF = Am, GS = GS, ELEAF = E,   
 Ac = Ac, Aj = Aj, Rd = Rd, VPD = VPD, Tleaf = Tleaf,   
 Ca = Ca, PPFD = PPFD, Patm = Patm)  
 return(df)  
}  
  
  
  
  
fitaci<-function (data, varnames = list(ALEAF = "Photo", Tleaf = "Tleaf",   
 Ci = "Ci", PPFD = "PARi"), Tcorrect = TRUE, citransition = NULL,   
 quiet = FALSE, startValgrid = TRUE, algorithm = "default",   
 ...)   
{  
 m <- as.list(match.call())  
 a <- as.list(formals(fitaci))  
 f <- names(formals(Photosyn))  
 extrapars <- setdiff(names(m), c(names(a), ""))  
 for (i in seq\_along(extrapars)) {  
 if (extrapars[i] %in% f) {  
 val <- eval(m[[extrapars[i]]])  
 formals(Photosyn)[extrapars[i]] <- val  
 }  
 else {  
 warning("Parameter ", extrapars[i], " not recognized.")  
 }  
 }  
 photpars <- formals(Photosyn)  
 removevars <- c("whichA")  
 photpars <- photpars[-which(names(photpars) %in% removevars)]  
 if (!varnames$PPFD %in% names(data)) {  
 data$PPFD <- 1800  
 if (!quiet)   
 warning("PARi not in dataset; assumed PARi = 1800.")  
 }  
 else data$PPFD <- data[, varnames$PPFD]  
 if (!varnames$Tleaf %in% names(data)) {  
 data$Tleaf <- 25  
 if (!quiet)   
 warning("Tleaf not in dataset; assumed Tleaf = 25.")  
 }  
 else {  
 data$Tleaf <- data[, varnames$Tleaf]  
 }  
 data$Ci <- data[, varnames[["Ci"]]]  
 data$ALEAF <- data[, varnames[["ALEAF"]]]  
 TcorrectVJ <- Tcorrect  
 acifun\_wrap <- function(Ci, ..., returnwhat = "ALEAF") {  
 r <- Photosyn(Ci = Ci, Tcorrect = TcorrectVJ, ...)  
 if (returnwhat == "ALEAF")   
 return(r$ALEAF)  
 if (returnwhat == "Ac")   
 return(r$Ac - r$Rd)  
 if (returnwhat == "Aj")   
 return(r$Aj - r$Rd)  
 }  
 Rd\_guess <- 1.5  
 maxCi <- max(data$Ci)  
 mi <- which.max(data$Ci)  
 maxPhoto <- data$ALEAF[mi]  
 Tl <- data$Tleaf[mi]  
 gammastar <- TGammaStar(Tl)  
 VJ <- (maxPhoto + Rd\_guess)/((maxCi - gammastar)/(maxCi +   
 2 \* gammastar))  
 Jmax\_guess <- VJ \* 4  
 if (Tcorrect) {  
 Teffect <- TJmax(Tl, EaJ = 39676.89, delsJ = 641.3615,   
 EdVJ = 2e+05)  
 Jmax\_guess <- Jmax\_guess/Teffect  
 }  
 dato <- data[data$Ci < 150 & data$Ci > 60 & data$ALEAF >   
 0, ]  
 if (nrow(dato) > 0) {  
 Km <- TKm(dato$Tleaf)  
 gammastar <- TGammaStar(dato$Tleaf)  
 vcmax <- with(dato, (ALEAF + Rd\_guess)/((Ci - gammastar)/(Ci +   
 Km)))  
 Vcmax\_guess <- median(vcmax)  
 }  
 else {  
 Vcmax\_guess <- Jmax\_guess/1.8  
 }  
 if (Tcorrect) {  
 Teffect <- TVcmax(Tl, EaV = 82620.87, delsC = 645.1013,   
 EdVC = 0)  
 Vcmax\_guess <- Vcmax\_guess/Teffect  
 }  
 if (startValgrid) {  
 aciSS <- function(Vcmax, Jmax, Rd) {  
 Photo\_mod <- acifun\_wrap(data$Ci, PPFD = data$PPFD,   
 Vcmax = Vcmax, Jmax = Jmax, Rd = Rd, Tleaf = data$Tleaf)  
 SS <- sum((data$ALEAF - Photo\_mod)^2)  
 return(SS)  
 }  
 d <- 0.3  
 n <- 20  
 gg <- expand.grid(Vcmax = seq(Vcmax\_guess \* (1 - d),   
 Vcmax\_guess \* (1 + d), length = n), Rd = seq(Rd\_guess \*   
 (1 - d), Rd\_guess \* (1 + d), length = n))  
 m <- with(gg, mapply(aciSS, Vcmax = Vcmax, Jmax = Jmax\_guess,   
 Rd = Rd))  
 ii <- which.min(m)  
 Vcmax\_guess <- gg$Vcmax[ii]  
 Rd\_guess <- gg$Rd[ii]  
 }  
 if (is.null(citransition)) {  
 nlsfit <- nls(ALEAF ~ acifun\_wrap(Ci, PPFD = PPFD, Vcmax = Vcmax,   
 Jmax = Jmax, Rd = Rd, Tleaf = Tleaf), algorithm = algorithm,   
 data = data, control = nls.control(maxiter = 500,   
 minFactor = 1/10000), start = list(Vcmax = Vcmax\_guess,   
 Jmax = Jmax\_guess, Rd = Rd\_guess))  
   
 p <- coef(nlsfit)  
 pars <- summary(nlsfit)$coefficients[, 1:2]  
 }  
 else {  
 dat\_vcmax <- data[data$Ci < citransition, ]  
 dat\_jmax <- data[data$Ci >= citransition, ]  
 if (nrow(dat\_vcmax) > 0) {  
 nlsfit\_vcmax <- nls(ALEAF ~ acifun\_wrap(Ci, PPFD = PPFD,   
 Vcmax = Vcmax, Jmax = 10000, Rd = Rd, Tleaf = Tleaf,   
 returnwhat = "Ac"), algorithm = algorithm, data = dat\_vcmax,   
 control = nls.control(maxiter = 500, minFactor = 1/10000),   
 start = list(Vcmax = Vcmax\_guess, Rd = Rd\_guess))  
 p1 <- coef(nlsfit\_vcmax)  
 }  
 else {  
 nlsfit\_vcmax <- NULL  
 p1 <- c(Vcmax = NA, Rd = NA)  
 }  
 Rd\_vcmaxguess <- if (!is.null(nlsfit\_vcmax))   
 coef(nlsfit\_vcmax)[["Rd"]]  
 else 1.5  
 if (nrow(dat\_jmax) > 0) {  
 nlsfit\_jmax <- nls(ALEAF ~ acifun\_wrap(Ci, PPFD = PPFD,   
 Vcmax = 10000, Jmax = Jmax, Rd = Rd\_vcmaxguess,   
 Tleaf = Tleaf, returnwhat = "Aj"), algorithm = algorithm,   
 data = dat\_jmax, control = nls.control(maxiter = 500,   
 minFactor = 1/10000), start = list(Jmax = Jmax\_guess))  
 p2 <- coef(nlsfit\_jmax)  
 }  
 else {  
 nlsfit\_jmax <- NULL  
 p2 <- c(Jmax = NA)  
 }  
 p <- c(p1[1], p2, p1[2])  
 pars1 <- if (!is.null(nlsfit\_vcmax))   
 summary(nlsfit\_vcmax)$coefficients[, 1:2]  
 else matrix(rep(NA, 4), ncol = 2)  
 pars2 <- if (!is.null(nlsfit\_jmax))   
 summary(nlsfit\_jmax)$coefficients[, 1:2]  
 else matrix(rep(NA, 2), ncol = 2)  
 pars <- rbind(pars1[1, ], pars2, pars1[2, ])  
 rownames(pars) <- c("Vcmax", "Jmax", "Rd")  
 nlsfit <- list(nlsfit\_vcmax = nlsfit\_vcmax, nlsfit\_jmax = nlsfit\_jmax)  
 }  
 acirun <- Photosyn(Ci = data$Ci, Vcmax = p[[1]], Jmax = p[[2]],   
 Rd = p[[3]], PPFD = data$PPFD, Tleaf = data$Tleaf, Tcorrect = Tcorrect)  
   
 acirun$Ameas <- data$ALEAF  
 acirun$ELEAF <- NULL  
 acirun$GS <- NULL  
 acirun$Ca <- NULL  
 names(acirun)[names(acirun) == "ALEAF"] <- "Amodel"  
 avars <- match(c("Ci", "Ameas", "Amodel"), names(acirun))  
 acirun <- acirun[, c(avars, setdiff(1:ncol(acirun), avars))]  
 l <- list()  
 l$df <- acirun[order(acirun$Ci), ]  
 l$pars <- pars  
 l$nlsfit <- nlsfit  
 l$Tcorrect <- Tcorrect  
 formals(Photosyn)$Tleaf <- mean(data$Tleaf)  
 formals(Photosyn)$PPFD <- mean(data$PPFD)  
 formals(Photosyn)$Vcmax <- l$pars[1]  
 formals(Photosyn)$Jmax <- l$pars[2]  
 formals(Photosyn)$Rd <- l$pars[3]  
 formals(Photosyn)$Tcorrect <- Tcorrect  
 l$Photosyn <- Photosyn  
 l$Ci\_transition <- findCiTransition(l$Photosyn)  
 l$Vcmax\_guess <- Vcmax\_guess  
 l$Jmax\_guess <- Jmax\_guess  
 l$Rd\_guess <- Rd\_guess  
 class(l) <- "acifit"  
 return(l)  
}