Estimating integrated production by merging FRRF or Labstaf data with CTD vertical data

Karline Soetaert

first version: 20-December-2023; current version: 23 January 2024

Abstract

Integrated photosynthesis for one station in a marine water column is estimated. Photosynthesis-irradiance curves were measured with FRRf and LabSTAF, at three water depths. These data are first extracted, standardized and fitted with a PI function. The fitted parameters are then combined with CTD-derived Chlorophyll vertical profiles to obtain parameter profiles. Combined with vertical light exinction profiles, and a time series of irradiance at the water surface, vertical photosynthesis profiles are then calculated, from which integrated primary production is estimated. The analysis of FRRF and LabSTAF data, the fitting of PI curves, and the vertical integration of production is done using functions of the R-package dtPP (digital twin of primary production) of the LTER-Life project.

Phytoplankton Primary Productivity (PhytoPP) forms the base of the marine food chain and is therefore an important measure of ocean productivity. The procedure for estimating depth integrated photosynthesis is exemplified based on two sets of data from the same station, obtained at 1, 7, and 40 m depth.

Estimates of the PI curves were done with the FRRf (Fast Repetitive Rate fluorometer) and with the Labstaf (Single Turnover Active Fluorometry).

Both instruments generate Fluorescence Light Curves (FLCs), where, for the FRRf, light is administered as short "flashlets" of 2 μs followed by a 1 or 2 μs pitch. By default a single turnover (ST) pulse in the FRRf comprises 100 flashlets over 200 μs. At the end of the ST pulse, most PSII reaction centers are closed (only once, hence the name single turnover). The reopening of the reaction centers is then followed over time by administering flashlets that are more widely spaced,

In the LabSTAF, light is administere as a series of two solid pulses of 100 μs, separated by increasingly large gaps (ranging from 200 to ~6400 μs in duration). It is assumed that approximately 12 to 27% of reaction centers are closed twice.

In general, the LabSTAF is claimed to be more accurate and faster.

Apart from the data, we also require a depth profile of Chlorophyll and a timeseries with photosynthetically active radiation (light intensity) data.

The extinction coefficient of light with water depth is also necessary.

Chlorophyll and light data

The chlorophyll data, measured with a CTD will be used s follows:

- Chlorophyll measured at the sampling depths (1, 7, 40m) are used to standardize the PI paramters alpha and pmax per unit chlorophyll; these estimates are averaged over the three depths to obtain station-specific values.
- the chlorophyll-depth profile is then used to estimate depth-dependent PI parameters.

```
CTDchl <- read.csv(file="../raw_data/CTDchl.csv")
head(CTDchl)
```

```
##   depth      Chl
## 1     1 5.000000
## 2     2 5.147211
## 3     3 5.546439
## 4     4 6.875418
## 5     5 7.065788
## 6     6 8.561156
```

Photosynthetic active radiation (PAR) data

PAR data have been estimated from shipboard ligth data, expressed in uEinst/m2/s. The same units as the light from the PI curves.

```
par <- read.csv(file= "../raw_data/Light.csv")
par$time <- as.POSIXct(par$time)
```

```
par (mfrow=c(1,2), las=1)

with (CTDchl,
      plot(Chl, depth,
           type="l", ylim=c(100,0),
           ylab="water depth, m", xlab="ug/L", main = "Chl profile"))
with (par,
      plot(time, par,
           type= "l",
           ylab="uEinst/m2/s", main="Light intensity"))
```

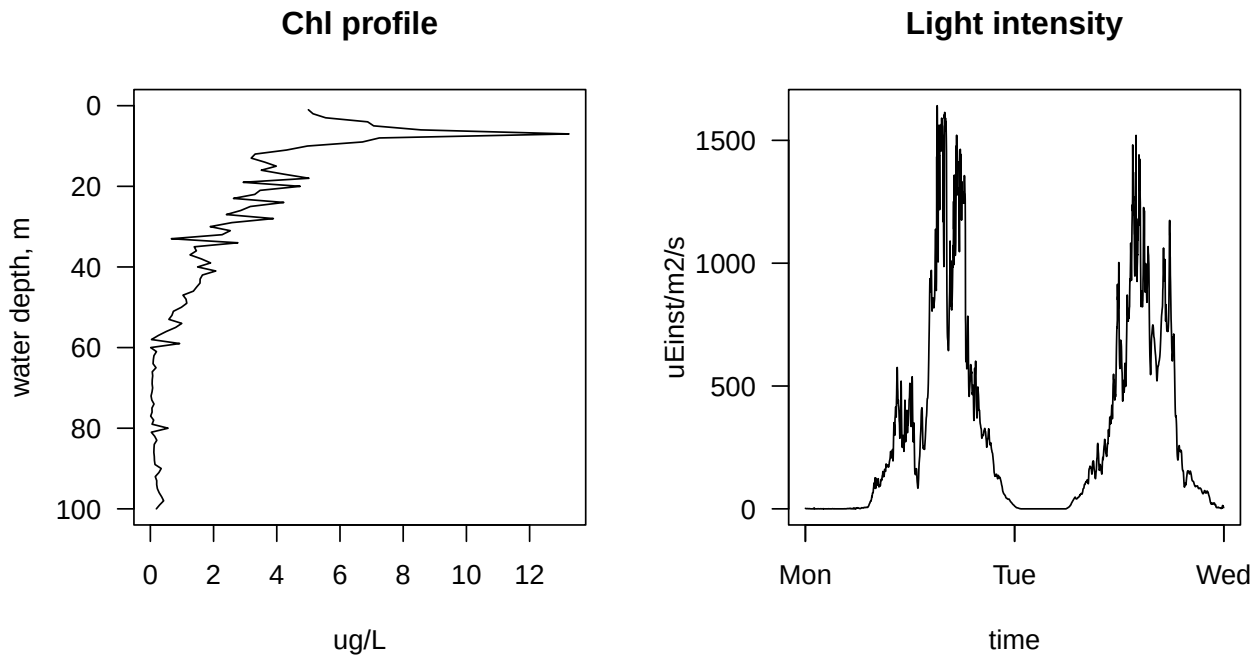


Figure 1: Accessory data needed to estimate depth-integrated PP

FRRF Fluorescence-light curves

Reading FRRF data

The Fluorescence-light curves from the FRRF (2 replicates) are read first.

We need the background fluorescence of the water to standardize the FRRF data. These data are inputted in a data.frame first.

```
dir   <- "../raw_data/FRRF/"   # directory with the data

FRRF.att <- data.frame(
  file = c(
    "A_1m_rep1.csv",  "A_1m_rep2.csv",
    "A_7m_rep1.csv",  "A_7m_rep2.csv",
    "A_40m_rep1.csv", "A_40m_rep2.csv"
    ),
  depth     = c(    1,     1,     7,     7,    40,    40),
  replicate = c(    1,     2,     1,     2,     1,     2),
  Fblanc    = c(0.194, 0.194, 0.175, 0.175, 0.156, 0.156)
)
```

All the FRRF files are read using function *readFRRF* from the package *dtPP*; they are pasted in one data.frame, and the water depth, replicate and blanc fluorescence are added to this data.frame.

```
FRRF <- NULL

for (fn in 1:nrow(FRRF.att))
  FRRF <- rbind(FRRF,
             data.frame(
               depth     = FRRF.att$depth    [fn],
               replicate = FRRF.att$replicate[fn],
               Fblanc    = FRRF.att$Fblanc   [fn],

               readFRRF(dir  = dir,
                        file = FRRF.att$file [fn])
                      )
                )
head(FRRF, n=2)
```

```
##   depth replicate Fblanc          file Saq  E Start   s   Chl ADC rP_measured rP_fitted  JPII  JVPII
## 1     1         1  0.194 A_1m_rep1.csv   1  0 00:44  44 5.426  72       0.000     0.000  0.00 0.0000
## 2     1         1  0.194 A_1m_rep1.csv   2 16 02:18 138 5.392  67       6.899     6.936 54.26 0.0139
##       C     p RSigma Sigma   CSQ TauES   NPQ   NSV    QR    Qo    Qm     QoSE     QmSE      QSE QSE.
## 1    NA 0.397 0.0387 5.855 0.570  2320 0.051 1.071 77.81 0.889 1.640 0.006683 0.006965 0.009652     
## 2 0.082 0.262 0.0372 5.632 0.564  2320 0.107 1.129 58.91 0.902 1.546 0.005887 0.009207 0.010900     
##   Qo.slope Qo.intercept Qm.points Qm.slope Qm.intercept             date
## 1   0.0184        0.889        36 0.000943        1.606 27/07/23 20:43:35
## 2   0.0162        0.902        36 0.000787        1.518 27/07/23 20:43:35
```

Standardizing the FRRF data

The FRRF data need to be standardized for the blanc values, *Fblanc*.

Standardization also uses the cross-sectional surface of the PSII system in the dark (*aLHII_0*). In case this is not passed as an argument, it is estimated either by using the *a_LHII* at E=0, or, when this is unavailable, by regressing a_LHII versus irradiance (E) for low values of E (< 100), and taken as the offset.

We first standardize all six Fluorescence-light curves at once, so that we estimate only ONE value for *a_LHII_0*.

We then check whether assuming one value is realistic, by plotting aLH_II versus E and looking at the offset.

The factor 3.6 (*convJVPII*) converts from $\mu mol\ e^-\ m^{-3}s^{-1}$ to $mmol\ e^-\ m^{-3}h^{-1}$.

```
FRRF_std_a <- standardizeFRRF(frrf       = FRRF,
                              Fblanc     = FRRF$Fblanc,
                              convJVPII  = 3.6)  # converts to mmol e-/m3/hour

# Show the attributes
head(attributes(FRRF_std_a)$processing)
```

```
## [1] "Standardized with Fblanc =  0.194 at 2024-01-23 13:12:36"
## [2] "Standardized with Fblanc =  0.175 at 2024-01-23 13:12:36"
## [3] "Standardized with Fblanc =  0.156 at 2024-01-23 13:12:36"
## [4] "JVPII calculated (absorption method), with conversion factor =  3.6 at 2024-01-23 13:12:36"
```

```
attributes(FRRF_std_a)$unit_JVPII
```

```
## [1] "mmol photons/m3/hour"
```

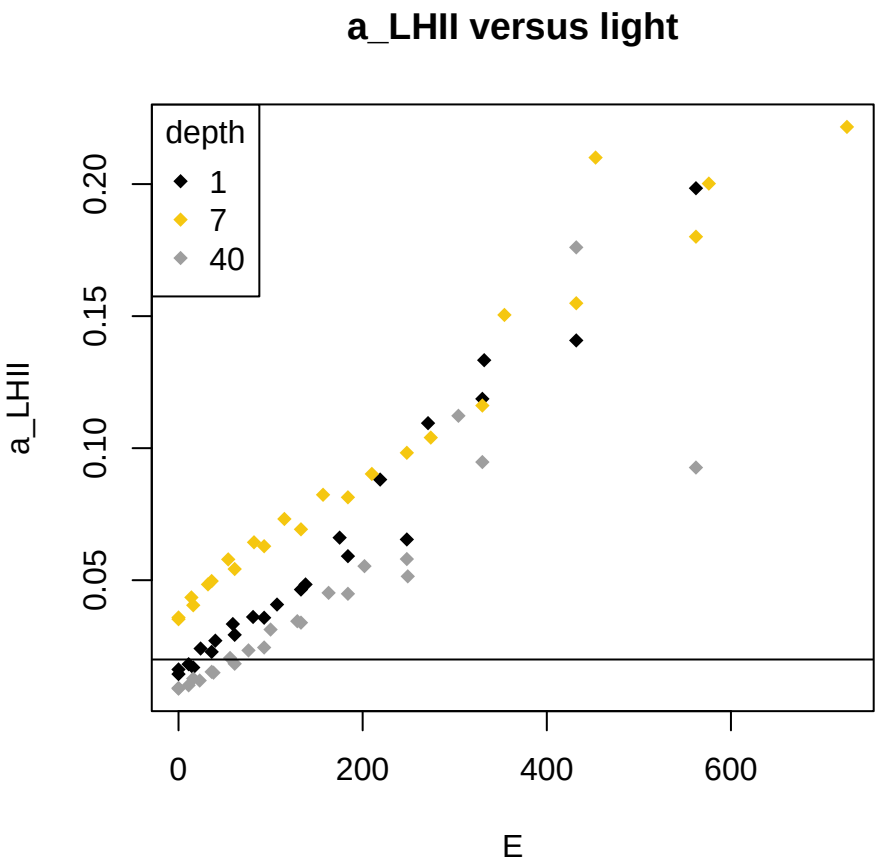
```
(aLHII_0 <- attributes(FRRF_std_a)$aLHII_0)
```

```
## [1] 0.01997596
```

```
attributes(FRRF_std_a)$ka
```

```
## NULL
```

```
with(FRRF_std_a,
     plot(E, a_LHII,
          main = "a_LHII versus light",
          col=depth, pch=18))
legend("topleft", legend = c(1, 7, 40), title="depth",
       pch=18, col=c(1, 7, 40))
abline(h=aLHII_0 )
```



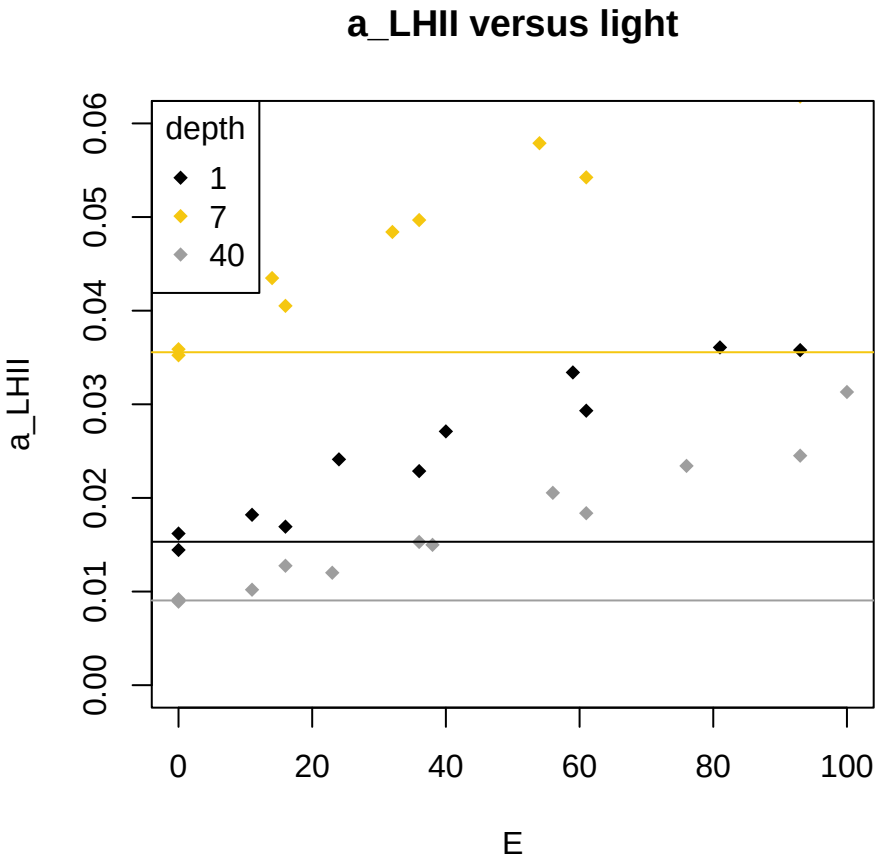
Based on the figure above, the samples at the three depths differ in the a_LHII versus E regression. Thus, a better option is to standardize the samples at the three depths separately, but combine the two replicates.

This way, a different value of a_LHII_0 is estimated for each PI dataset.

```
# take a suitable subset for each water depth

FRRF_1  <- subset(FRRF, subset = depth == 1)
FRRF_7  <- subset(FRRF, subset = depth == 7)
FRRF_40 <- subset(FRRF, subset = depth == 40)

# standardize
FRRF_1   <- standardizeFRRF(frrf      = FRRF_1,
                            Fblanc    = FRRF_1$Fblanc,
                            convJVPII = 3.6)
FRRF_7   <- standardizeFRRF(frrf      = FRRF_7,
                            Fblanc    = FRRF_7$Fblanc,
                            convJVPII = 3.6)
FRRF_40  <- standardizeFRRF(frrf      = FRRF_40,
                            Fblanc    = FRRF_40$Fblanc,
                            convJVPII = 3.6)

# check the aLHII_0 values
a1 <- attributes(FRRF_1)$aLHII_0
a2 <- attributes(FRRF_7)$aLHII_0
a3 <- attributes(FRRF_40)$aLHII_0

# combine the three standardized file
```

```
FRRF_std <- rbind(FRRF_1, FRRF_7, FRRF_40)

# fix the attributes
attributes(FRRF_std)$aLHII_0 <- c(a1, a2, a3)
```

```
with(FRRF_std,
     plot(E, a_LHII,
          main = "a_LHII versus light",
          xlim = c(0,100), ylim=c(0, 0.06),
          col=depth, pch=18))
legend("topleft", legend = c(1, 7, 40), title="depth",
       pch=18, col=c(1, 7, 40))

abline(h=attributes(FRRF_std)$aLHII_0, col=c(1, 7, 40))
```



Fitting the FRRF data

After standardization, the unit of the JVPII is now $mmol\ e^{-}m^{-3}h^{-1}$

The standardized fluorescence-light data can now be fitted with a PI function, using R-function *fitPI*. The default is to use the Eilers-Peeters model. We fit each depth and replicate separately.

It is easiest to write a function for fitting these data, as there are 6 cases to be fitted. The function also plots the fits, so as to see whether this worked properly.

```
# unique identifiers of the 6 cases to be fitted
Samples <- unique(FRRF_std[,c("depth", "replicate")])
```

```
PARS <- NULL

# -------------------------------------------
# Function for fitting (and plotting the fit)
# -------------------------------------------

fitProfile <- function(Depth, Replicate){

  # extract the data for the required depth and replicate
  Sub <- subset(FRRF_std,
                subset = depth     == Depth     &
                         replicate == Replicate)

  # fit it with the Eilers-Peeters model
  FIT <- fitPI(model    = "EP",
               response = Sub$JVPII,
               I        = Sub$E)

  # show goodness of fit
  plot(FIT,
       ylab= "JVPII, mmol e/m3/h",
       main = paste("depth=", Depth,", replicate=", Replicate))

  # return value
  c(depth=Depth, replicate=Replicate, FIT$par)
}

# call the function for each depth x replicate case
par(mfrow=c(3,2), las=1)

Fits <- NULL

for (i in 1: nrow(Samples)){

  fitcase <- fitProfile(Depth      = Samples$depth[i],
                        Replicate  = Samples$replicate[i])

  Fits <- rbind(Fits, fitcase)
}
```

```
Fits
```

```
##         depth replicate      alpha     eopt        ps
## fitcase     1         1 0.03207460 357.9246 2.3949212
## fitcase     1         2 0.02685324 202.9979 2.0167470
## fitcase     7         1 0.06521938 596.8451 8.3171432
## fitcase     7         2 0.06158890 418.0447 7.1873633
## fitcase    40         1 0.02022385 354.9111 0.8968518
## fitcase    40         2 0.01197941 162.4453 0.8819751
```

Chlorophyll-specific PI parameters

The values for *alpha*, *ps* and *eopt* are in $mmol\ e^{-}m^{-3}h^{-1}/[\mu Einst\ m^{-2}s^{-1}]$, $mmol\ e^{-}m^{-3}h^{-1}$ and $\mu Einst\ m^{-2}s^{-1}$ respectively.

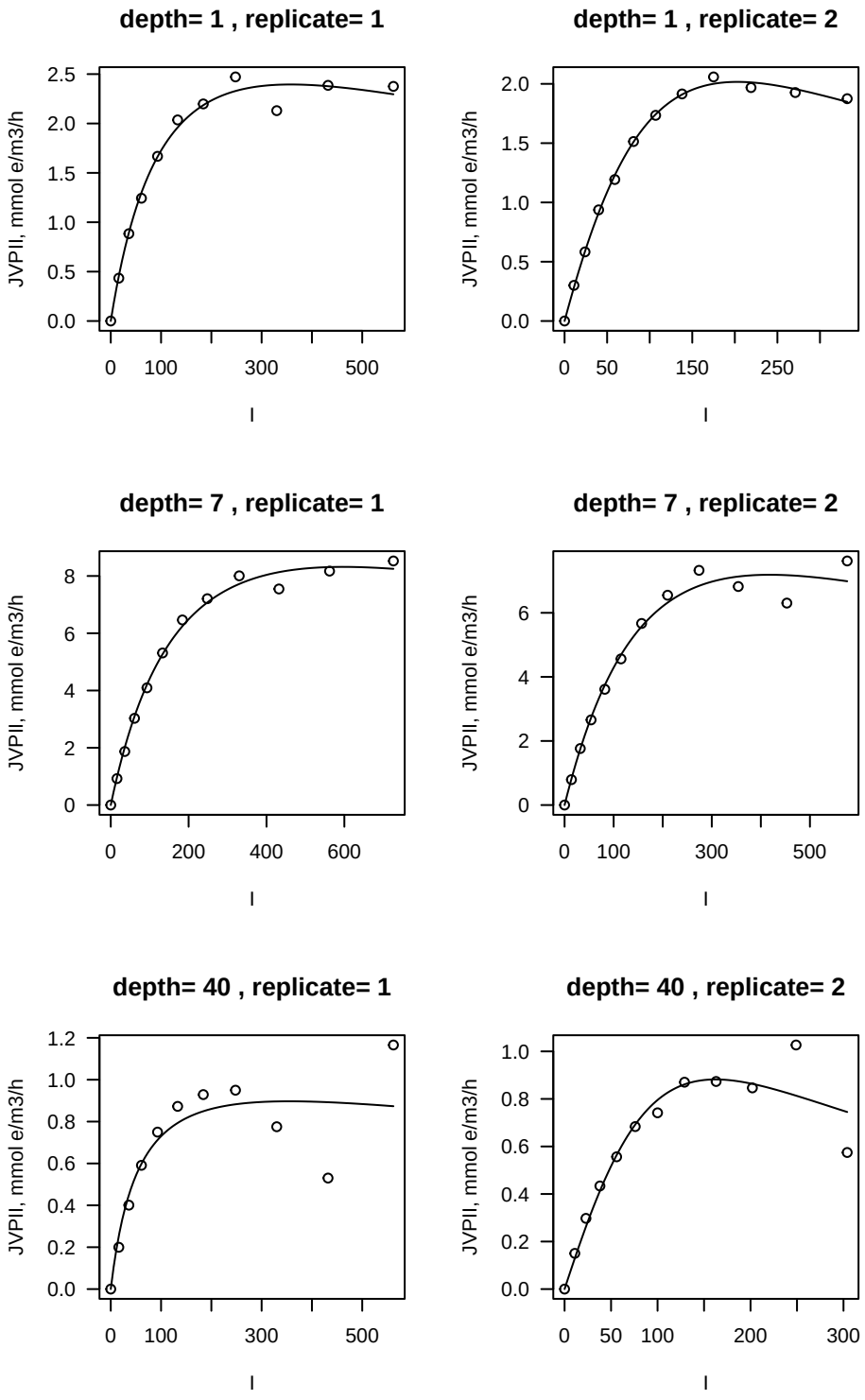


Figure 2: PI fits of the standardized FRRF data

The fitted values for *alpha* and *ps* show large differences, which partly reflect algal biomass. (Values for eopt are always variable). These parameters are now standardized per unit chlorophyll.

Here we have the choice to use the chlorophyll as it has been estimated by the FRRF apparatus, or to pick the chlorophyll as measured with the CTD. As we will recalculate total rates for the entire depth profile by multiplying with the CTD chlorophyll values, it makes sense to use CTD-derived chlorophyll values for standardizing.

To show diffenences between both Chl estimates, we merge the parameter file with both.

We first calculate the mean Chl concentration as measured with FRRF, and as stored in the FRRF_std data.frame:

```
# Mean chlorophyll concentration from FRRF
Chl <- aggregate(x   = FRRF_std$Chl,
                 by  = list(replicate = FRRF_std$replicate,
                            depth     = FRRF_std$depth),
                 FUN = mean)
names(Chl)[3] <- "Chl_FRRF"
```

To get the values of Chl from the CTD, we locate the closest depth point from the CTD cast, and then extract the corresponding Chl value:

```
# distance between all Chl measurements and the depth from the FRRF data
Distance <- outer(X = CTDchl$depth,     # depth of CTDs
                  Y = Samples$depth,    # depth of FRRF samples
                  FUN = function(x,y) abs(x-y))

# for each FRRF depth (columns): extract the Chl that is nearest

Chl$Chl_CTD <- apply(Distance,
                     MARGIN = 2,                  # use columns
                     FUN    = function(x)
                       CTDchl$Chl[which.min(x)])  # Chl from closest depth
```

Fitted parameters are now merged with the Chlorophyll estimates:

```
Fits <- merge(Fits, Chl)
Fits
```

```
##   depth replicate      alpha     eopt        ps  Chl_FRRF   Chl_CTD
## 1     1         1 0.03207460 357.9246 2.3949212  7.176909  5.000000
## 2     1         2 0.02685324 202.9979 2.0167470  7.253000  5.000000
## 3    40         1 0.02022385 354.9111 0.8968518  3.972727  1.496568
## 4    40         2 0.01197941 162.4453 0.8819751  4.022833  1.496568
## 5     7         1 0.06521938 596.8451 8.3171432 11.564500 13.245134
## 6     7         2 0.06158890 418.0447 7.1873633 11.957583 13.245134
```

As we will calculate the PI parameters for the entire water depth using the CTD-derived Chl measures, we calculate chlorophyll-specific alpha and ps parameters, by dividing with the CTD-derived chlorophyll values.

```
Fits$alpha_chl <- Fits$alpha / Fits$Chl_CTD
Fits$ps_chl    <- Fits$ps    / Fits$Chl_CTD
Fits
```

```
##   depth replicate      alpha     eopt        ps  Chl_FRRF   Chl_CTD   alpha_chl    ps_chl
## 1     1         1 0.03207460 357.9246 2.3949212  7.176909  5.000000 0.006414921 0.4789842
## 2     1         2 0.02685324 202.9979 2.0167470  7.253000  5.000000 0.005370647 0.4033494
## 3    40         1 0.02022385 354.9111 0.8968518  3.972727  1.496568 0.013513484 0.5992724
```

```
## 4    40         2 0.01197941 162.4453 0.8819751  4.022833  1.496568 0.008004586 0.5893318
## 5     7         1 0.06521938 596.8451 8.3171432 11.564500 13.245134 0.004924026 0.6279396
## 6     7         2 0.06158890 418.0447 7.1873633 11.957583 13.245134 0.004649927 0.5426418
```

We now use the average alpha_chl, eopt and ps_chl values for this station to estimate depth-varying PI parameters:

```
# station-averaged parameters
meanPIpar <- apply(X      = Fits[, c("alpha_chl", "eopt", "ps_chl")],
                   MARGIN = 2,
                   FUN    = mean)
meanPIpar
```

```
##    alpha_chl         eopt       ps_chl
## 7.146265e-03 3.488614e+02 5.402532e-01
```

Depth-varying PI parameters

Combining the Chl-specific PI parameters with the Chlorophyll measurements from the CTD, we now estimate depth-varying PI parameters.

```
PI.pars <- data.frame(depth = CTDchl$depth,
                      alpha = meanPIpar[["alpha_chl"]]*CTDchl$Chl,
                      eopt  = meanPIpar[["eopt"]],
                      ps    = meanPIpar[["ps_chl"]]   *CTDchl$Chl)
```

Depth-integrated photosynthesis

To estimate integrated production, we also need to input the light extinction in the water column (kz).

We convert from $mmol\ e^{-}m^{-3}h^{-1}$ to $mg\ C\ m^{-3}\ d^{-1}$ by assuming that we need 5 electrons per organic carbon produced, so the conversion factor becomes: $1/5\ 12\ 24$.

```
fac   <- 1/5*12*24        # from mmol e/m3/hr to mgC/m3/d
times <- par$time         # time over which to estimate integrated production
kz    <- 0.2              # extinction coefficient, [/m]
PS <- integratedPP(times   = times,
                   PI.par  = PI.pars,
                   It.data = par,
                   kz      = kz,
                   convFac = fac)
plot(PS, mass="mgC", time="d")
```

```
plot(PS, mass="mgC", time="d", which="profile")
```

```
par(mfrow=c(1, 3), las=1)
depths <- PS$profile$z
time   <- PS$ts$times
prod   <- PS$prod
prodV  <- colMeans(prod)
image2D(x = time, y = depths, z = prod,
        xlab = "time", ylab = "depth, m", clab = "mgC/m3/d",
        main = "photosynthesis",
        ylim = c(40,0))
plot(x = prodV, y = depths,
     xlab = "mgC/m3/d", ylab = "depth",
     main = "time-averaged photosynthesis",
```

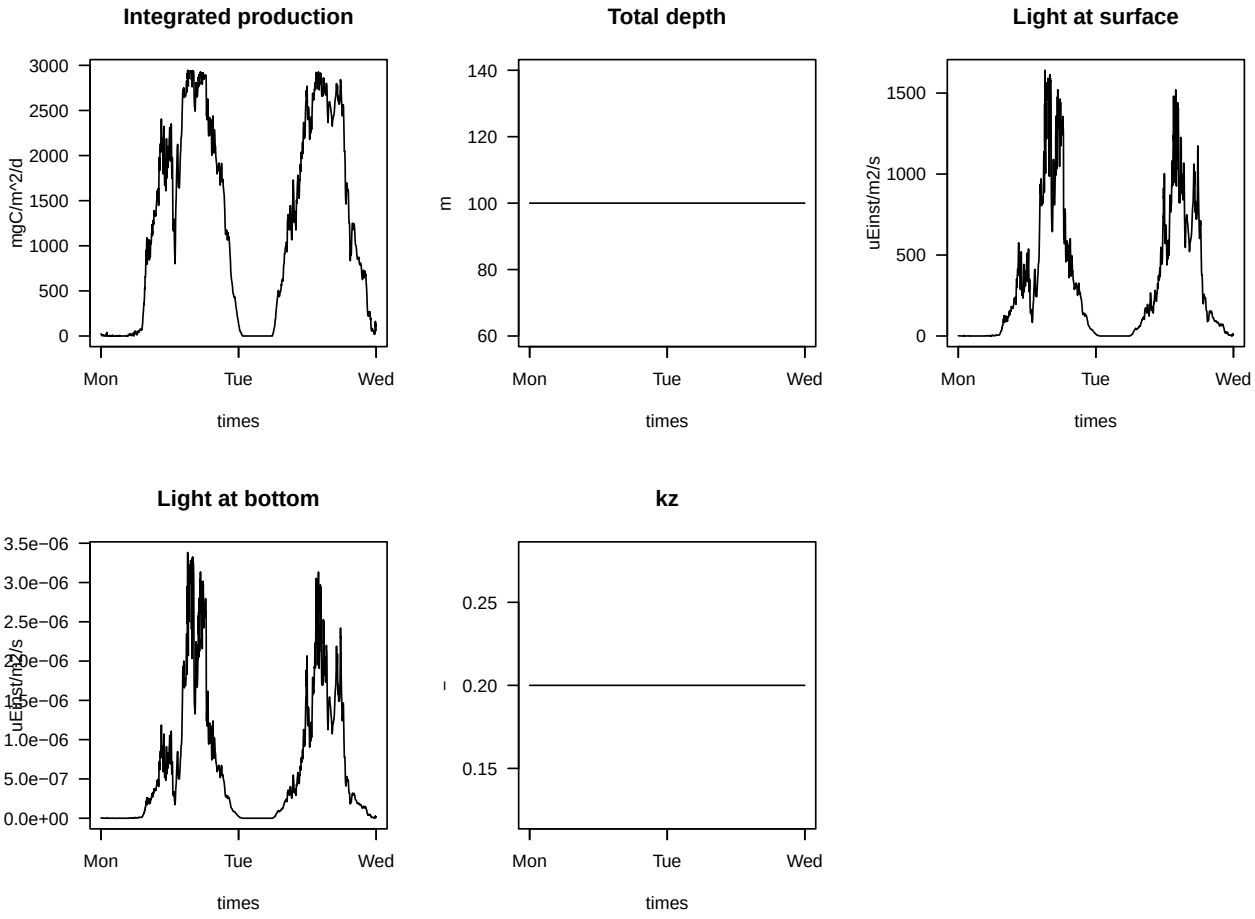


Figure 3: integrated production using FRRF data

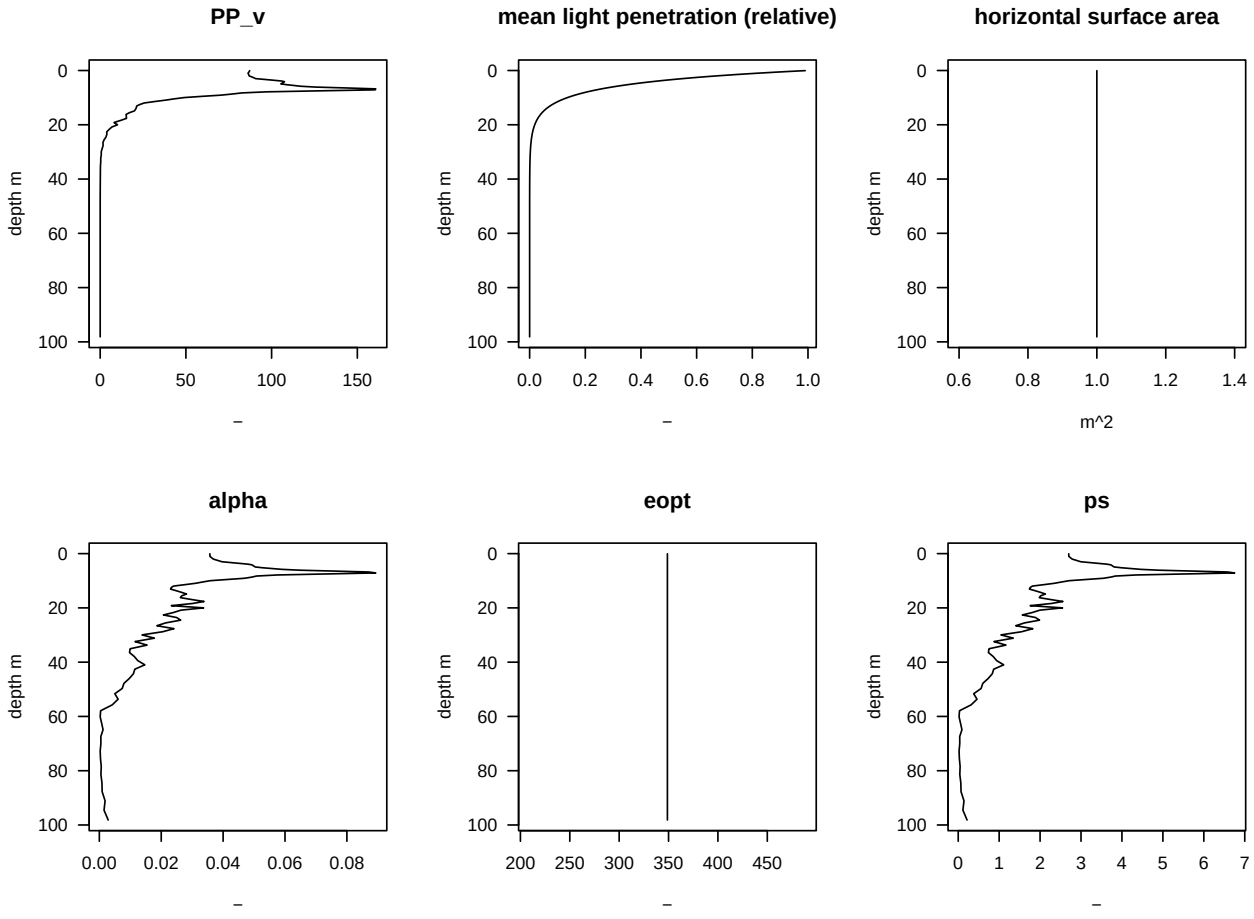


Figure 4: integrated production using FRRF data

```
     ylim = c(40,0), type = "l", lwd=2)
plot(x = CTDchl$Chl/max(CTDchl$Chl), y = CTDchl$depth,
     xlab = "-", ylab = "depth",
     main = "relative profiles", col=3,
     ylim = c(40,0), type = "l", lwd=2)
lines(x = PS$profile$Iz_I0, y = depths,
     col=2, type = "l", lwd=2)
lines(x = prodV/max(prodV), y = depths,
     col=4, type = "l", lwd=2)
legend("bottomright", legend=c("Chl/max(Chl", "Iz/max(Iz)", "phot/max(phot)"),
       lty=1, lwd=2, col=c(3,2,4))
```

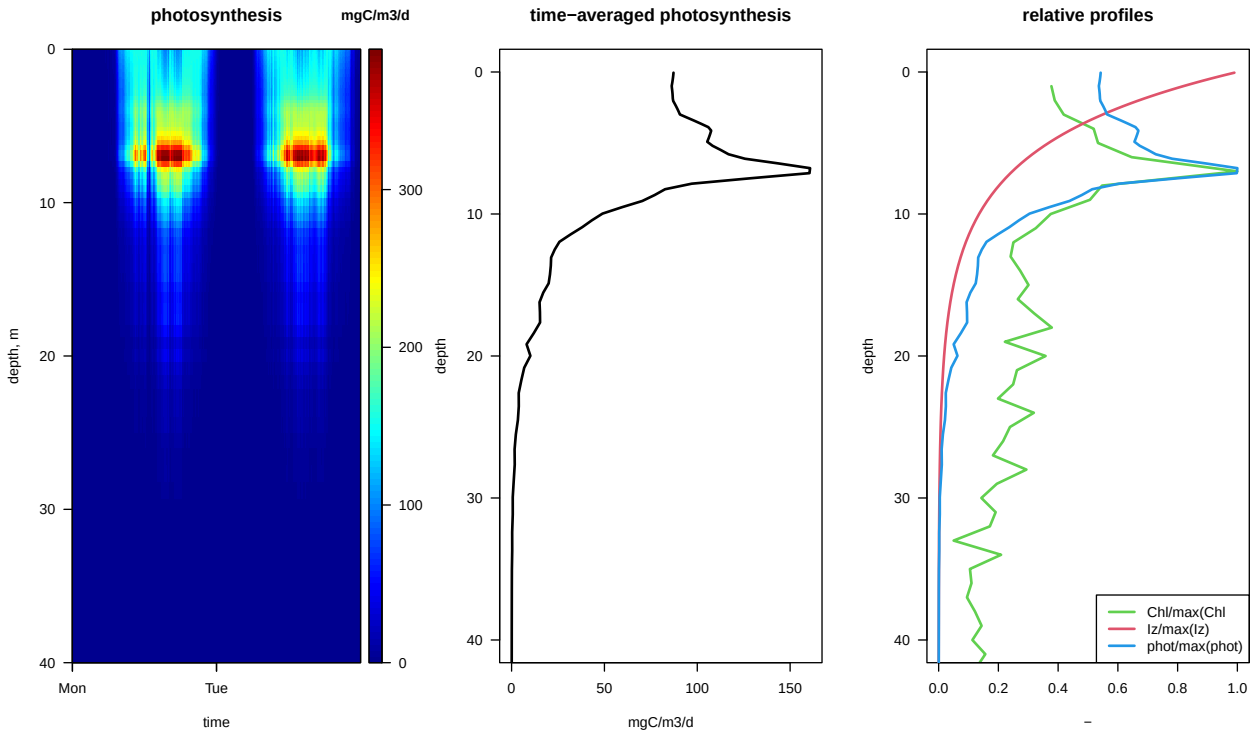


Figure 5: FRRf-derived photosynthesis

The vertical profiles show largest production at 6.8 m depth.

The daily mean production is now estimated from the timeseries in the PP list:

```
mean(PS$ts$PP) # mg C/m2/d
```

```
## [1] 1236.617
```

labstaf data

The LabSTAF data are treated similarly as the FRRf data.

```
dir   <- "../raw_data/labstaf"

LS.att <- data.frame(
  file = c(
             "B_1m.txt", "B_7m.txt",  "B_40m.txt"),
  depth     = c(1,                7,           40),
  Fblanc    = c(0.194,        0.175,        0.156)
)
```

The Labstaf data are stored with tab-separated format; this format can be read with R-function *read.delim*. This choice is passed while reading (*txt = "delim"*).

```
LabSTAF <- NULL

for (fn in 1:nrow(LS.att))
  LabSTAF <- rbind(LabSTAF,
             data.frame(
               depth     = LS.att$depth   [fn],
               Fblanc    = LS.att$Fblanc  [fn],
               readFRRF(dir  = dir,
                        file = LS.att$file[fn],
                        txt  = "delim")
                       )
                 )
head(LabSTAF, n=2)
```

```
##   depth Fblanc     file  E Fb  rP rP.fit JVPII  GOPII  JPII    Fo    Fm    Fq. Fq..Fm. Fq..Fmc. Fv..
## 1     1  0.194 B_1m.txt  0  0  NA    0.0    NA     NA    NA 0.899 1.517 0.6182  0.4075   0.4075   0.4
## 2     1  0.194 B_1m.txt 25  0 6.5    9.3 0.147 0.1323 41.25 0.834 1.327 0.4929  0.3715   0.3715   0.3
##   Ekt. Ekt AlphaPII. SigmaPII.   Rho. TauS. Taut. Fo.1 Fm.1 Fv Fv.Fm Fv.Fmc Fv..Fmc..1 EkS.1 AlphaPI
## 1   NA  NA   0.08927     3.426 0.2753  8752    NA   NA   NA NA    NA     NA         NA    NA       NA
## 2   NA  NA   0.08256     3.169 0.2477  3584    NA   NA   NA NA    NA     NA         NA    NA       NA
##    rP.1 JVPII.1 GOPII.1 JPII.1   F..1  Fm..1  Fq..1 Fq..Fm..1 Fq..Fmc..1 Fv..Fmc..2 Fq..Fv..1 EkS.2 
## 1    NA      NA      NA     NA 0.6206 0.9402 0.3196    0.3400     0.3400     0.2988        NA    NA 
## 2 4.926  0.1118  0.1007  40.11 0.7263 1.0390 0.3130    0.3012     0.3012     0.3202    0.9405   132 
##   AlphaPII..1 SigmaPII..1 Rho..1 TauS..1 Taut..1              date
## 1     0.09103       3.493 0.2567   10448      NA Jul 27, 2023 22:18
## 2     0.08443       3.240 0.1898    1861    4432 Jul 27, 2023 22:18
```

Converting the Labstaf data

For the LabSTAF apparatus, the volumetric electron flux (JVPII) is already calculated by the machine, assuming an inputted Fblanc. In theory, the data do not need to be standardized unless the actual blanc fluorescence deviates significantly from the inputted one.
In the data files considered, Fblanc was set to be 0, which is at odds with the actual values (0.156-0.194), so it makes sense to re-standardize the data.

First, we show that the standaridization procedure implemented in *dtPP* is consistent with what is done by the LabSTAF.

For the data at 1 m depth, the *aLHII_0* value provided by the LabSTF is 0.02271 (as given in the input file), so if we standardize the rates with this value, and assuming a $Fblanc = 0$, we should obtain the same value for JVPII, as provided by the LabSTAF:

```
Sub <- subset(LabSTAF, subset= depth==1)
SS  <- standardizeFRRF(Sub,
                       convJVPII = 1,
                       Fblanc    = 0,
                       aLHII_0   = 0.02271)

with (SS, cbind(JVPII, JVPII_uc))  # almost the same
```

```
##           JVPII JVPII_uc
##  [1,] 0.0000000       NA
##  [2,] 0.2109275   0.1470
##  [3,] 0.3852759   0.3858
##  [4,] 0.6098807   0.6116
##  [5,] 0.8128744   0.8128
##  [6,] 1.0002723   1.0020
##  [7,] 1.1480442   1.1470
##  [8,] 1.3024795   1.3010
##  [9,] 1.2878597   1.2870
## [10,] 1.2716532   1.2700
## [11,] 1.3760929   1.3760
## [12,] 1.3477295   1.3490
```

The original and standardized data are (almost) the same, except for the first few values. Deviations may be due to the limited precision of the data in the output files.

We thus re-standardize the data, using the correct blancs, and converting the estimated JVPII to $mmol\ e^{-1}m^{-3}h^{-1}$.

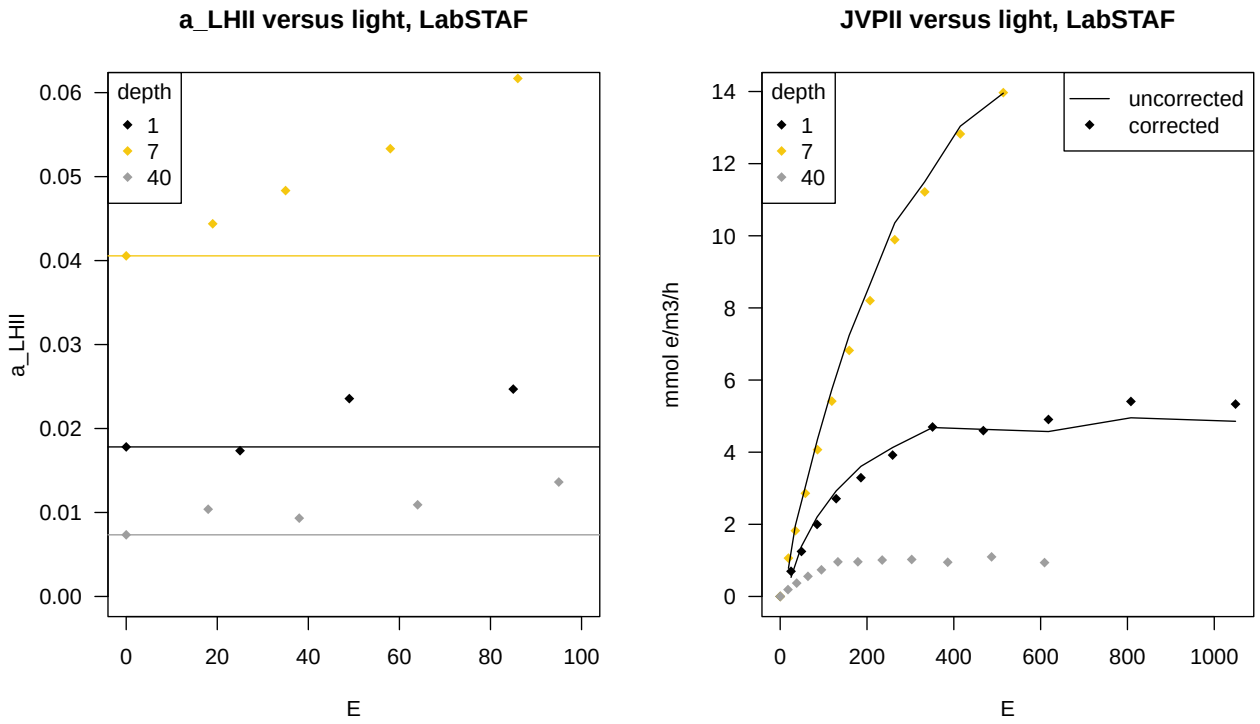
```
S1  <- standardizeFRRF(frrf      = subset(LabSTAF, subset= depth==1),
                       Fblanc    = 0.194,
                       convJVPII = 3.6)
S2  <- standardizeFRRF(frrf      = subset(LabSTAF, subset= depth==7),
                       Fblanc    = 0.175,
                       convJVPII = 3.6)
S3  <- standardizeFRRF(frrf      = subset(LabSTAF, subset= depth==40),
                       Fblanc    = 0.156,
                       convJVPII = 3.6)

aLHII_0 <- c(attributes(S1)$aLHII_0, attributes(S2)$aLHII_0, attributes(S3)$aLHII_0)
aLHII_0
```

```
## [1] 0.017809121 0.040562676 0.007339842
```

Note the relatively large difference between the aLHII_0 from the LabSTAF (resp. 0.02271, 0.0465 and NA) and the values generated with the standardization function ().

```
par(mfrow=c(1,2), las=1)
SS <- rbind(S1, S2, S3)
with(SS,
     plot(E, a_LHII,
          main = "a_LHII versus light, LabSTAF",
          xlim = c(0,100), ylim=c(0, 0.06),
          col = depth, pch=18))
legend("topleft", legend = c(1, 7, 40), title="depth",
       pch=18, col=c(1, 7, 40))
```

```
abline(h=aLHII_0, col=c(1, 7, 40))

with(SS,
     plot(E, JVPII,
          main = "JVPII versus light, LabSTAF",
          ylab = "mmol e/m3/h",
          col = depth, pch=18))
with(SS,
     lines(E, JVPII_uc*3.6,
          col = depth))

legend("topleft", legend = c(1, 7, 40), title="depth",
       pch=18, col=c(1, 7, 40))
legend("topright", legend = c("uncorrected", "corrected"),
       pch=c(NA,18), lty=c(1, NA))
```



Restandardizing the data led to actual estimates of JVPII for depth=40 m, while this did not give a value in the LabSTAF generated file. For the two other depths, the blanc-corrected and uncorrected values are quite similar.

The rest of the anlysis is very similar to the treatment of the FRRF data:

```
par(mfrow=c(2,2), las=1)
FitLS1 <- fitPI(model    = "EP",
                response = S1$JVPII,
                I        = S1$E)

plot(FitLS1,
     main = paste("depth=1"))
with(S1,
      points(E, JVPII_uc*3.6,
             pch=18))
```

```
legend("bottomright",
       legend = c("corrected", "uncorrected"),
       pch    = c(1, 18))

FitLS2 <- fitPI(model    = "EP",
                response = S2$JVPII,
                I        = S2$E)

plot(FitLS2,
     main = paste("depth=7"))
with(S2,
     points(E, JVPII_uc*3.6,
            pch=18))

FitLS3 <- fitPI(model    = "EP",
                response = S3$JVPII,
                I        = S3$E)
plot(FitLS3,
     main = paste("depth=40"))
with(S3, points(E, JVPII_uc*3.6,
                pch=18))

FitLS <- as.data.frame(rbind(FitLS1$par, FitLS2$par, FitLS3$par))
```

The LabSTAF does not generate estimates for Chlorophyll, so we use those from the CTD; these were already extracted for the FRRf data, so we hard-code them here:

```
FitLS$Chl       <- c(5, 13.245134, 1.496568)

# standardize
FitLS$alpha_chl <- FitLS$alpha/FitLS$Chl
FitLS$ps_chl    <- FitLS$ps/FitLS$Chl

# estimate mean values
meanLSpar       <- apply(FitLS, MARGIN=2, FUN=mean)
```

```
# estimate depth profile of parameters
PI.parsLS <- data.frame(
                    depth = CTDchl$depth,
                    alpha = meanLSpar[["alpha_chl"]]*CTDchl$Chl,
                    eopt  = meanLSpar[["eopt"]],
                    ps    = meanLSpar[["ps_chl"]]   *CTDchl$Chl
                    )
```

```
# depth-integrated production
fac   <- 1/5*12*24            # from mmol e/m3/hr to mgC/m3/d
times <- par$time
kz    <- 0.2  # /m
PSLS <- integratedPP(times   = times,
                    PI.par  = PI.parsLS,
                    It.data = par,
                    kz      = kz,
                    convFac = fac)

plot(PSLS, mass="mgC", time="d")
```

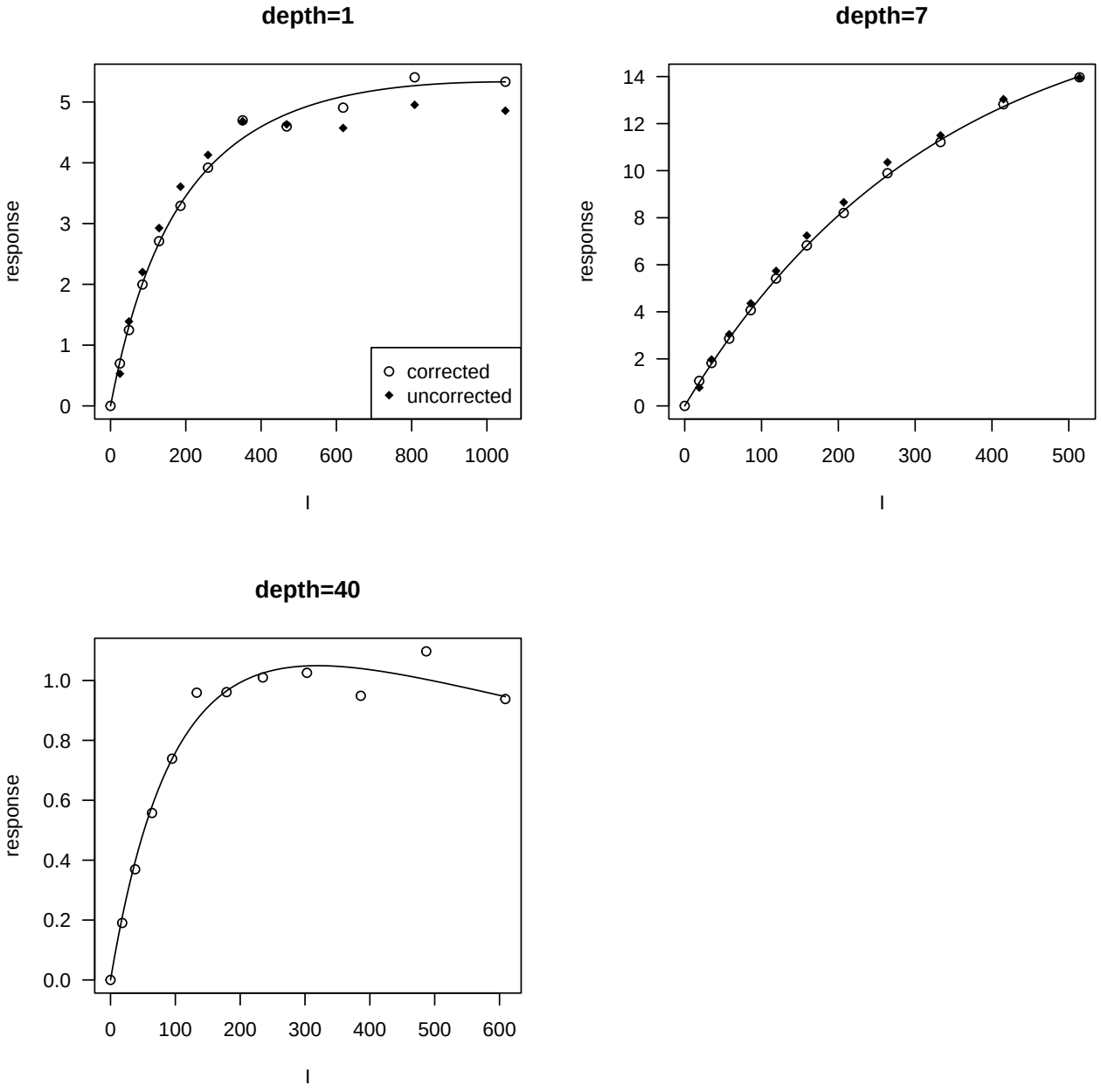


Figure 6: PI fits for the standardized labSTAF data (also shows uncorrected data)

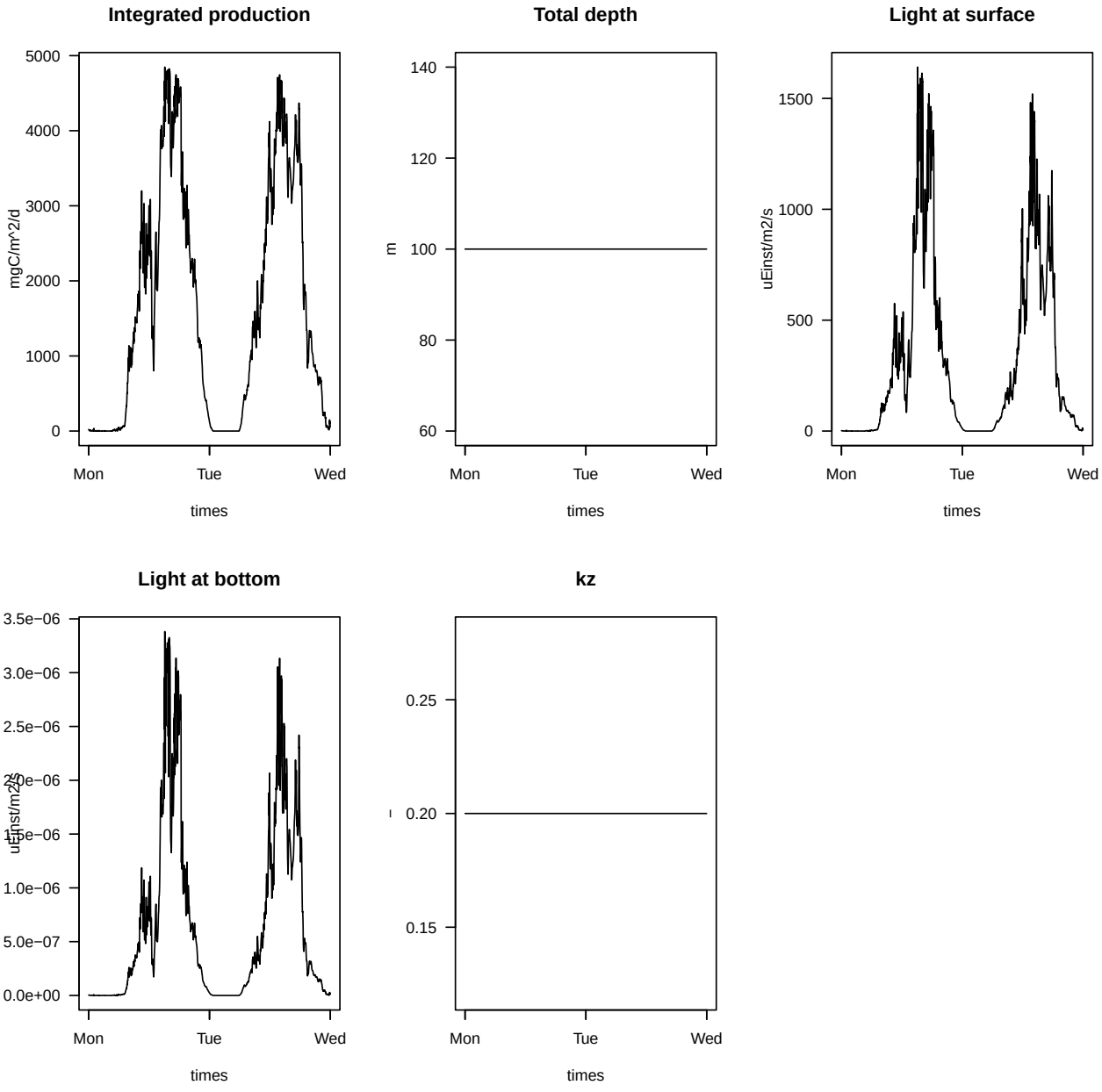


Figure 7: integrated production using LabSTAF data

```
plot(PSLS, mass="mgC", time="d", which="profile")
```

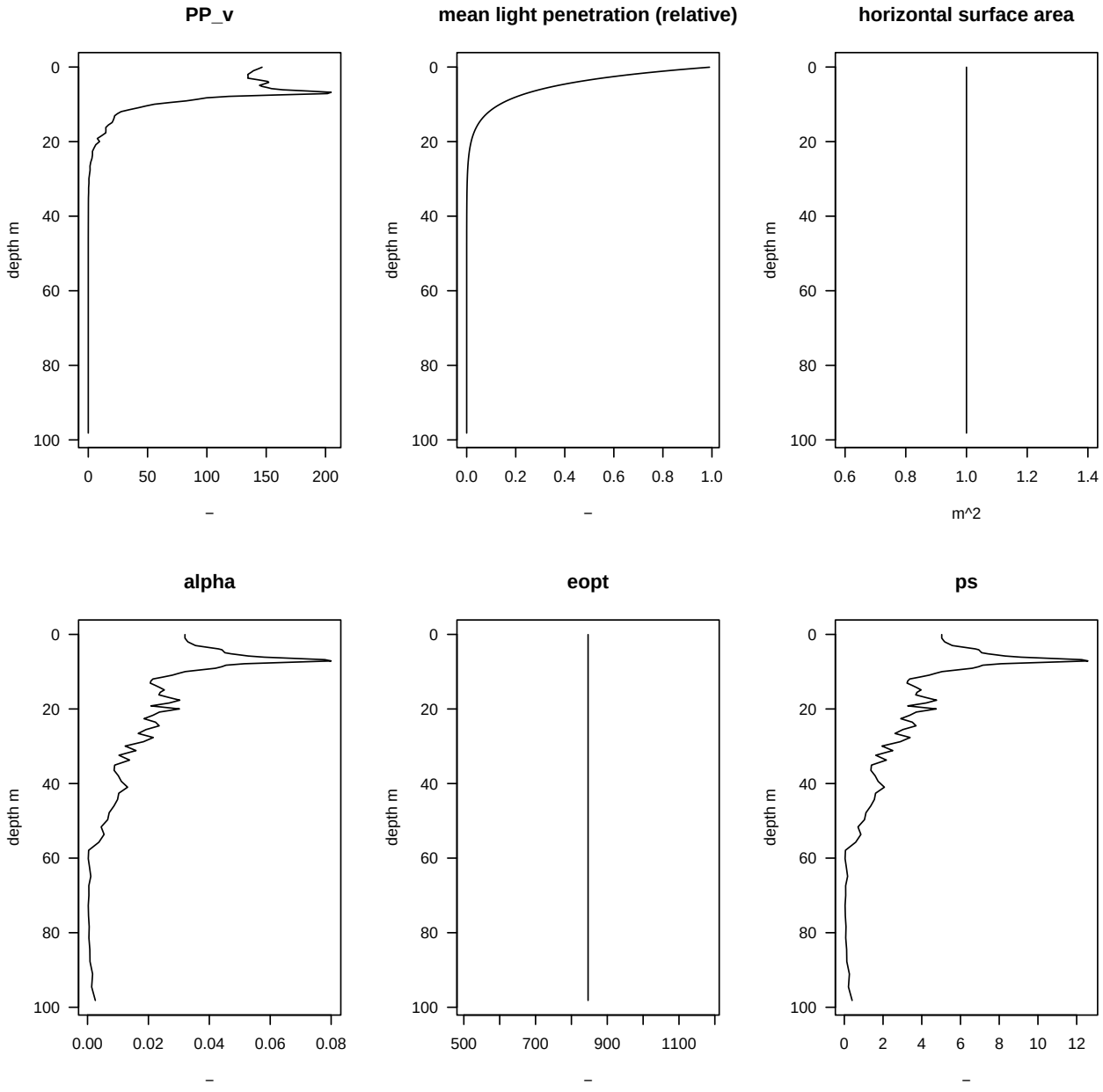


Figure 8: integrated production using LabSTAF data

```
par(mfrow=c(1, 3), las=1)
depths <- PSLS$profile$z
time   <- PSLS$ts$times
prod   <- PSLS$prod
prodV  <- colMeans(prod)
image2D(x = time, y = depths, z = prod,
        xlab = "time", ylab = "depth, m", clab = "mgC/m3/d",
        main = "photosynthesis",
        ylim = c(40,0))
plot(x = prodV, y = depths,
```

```
     xlab = "mgC/m3/d", ylab = "depth",
     main = "time-averaged photosynthesis",
     ylim = c(40,0), type = "l", lwd=2)
plot(x = CTDchl$Chl/max(CTDchl$Chl), y = CTDchl$depth,
     xlab = "-", ylab = "depth",
     main = "relative profiles", col=3,
     ylim = c(40,0), type = "l", lwd=2)
lines(x = PSLS$profile$Iz_I0, y = depths,
     col=2, type = "l", lwd=2)
lines(x = prodV/max(prodV), y = depths,
     col=4, type = "l", lwd=2)
legend("bottomright", legend=c("Chl/max(Chl", "Iz/max(Iz)", "phot/max(phot)"),
       lty=1, lwd=2, col=c(3,2,4))
```

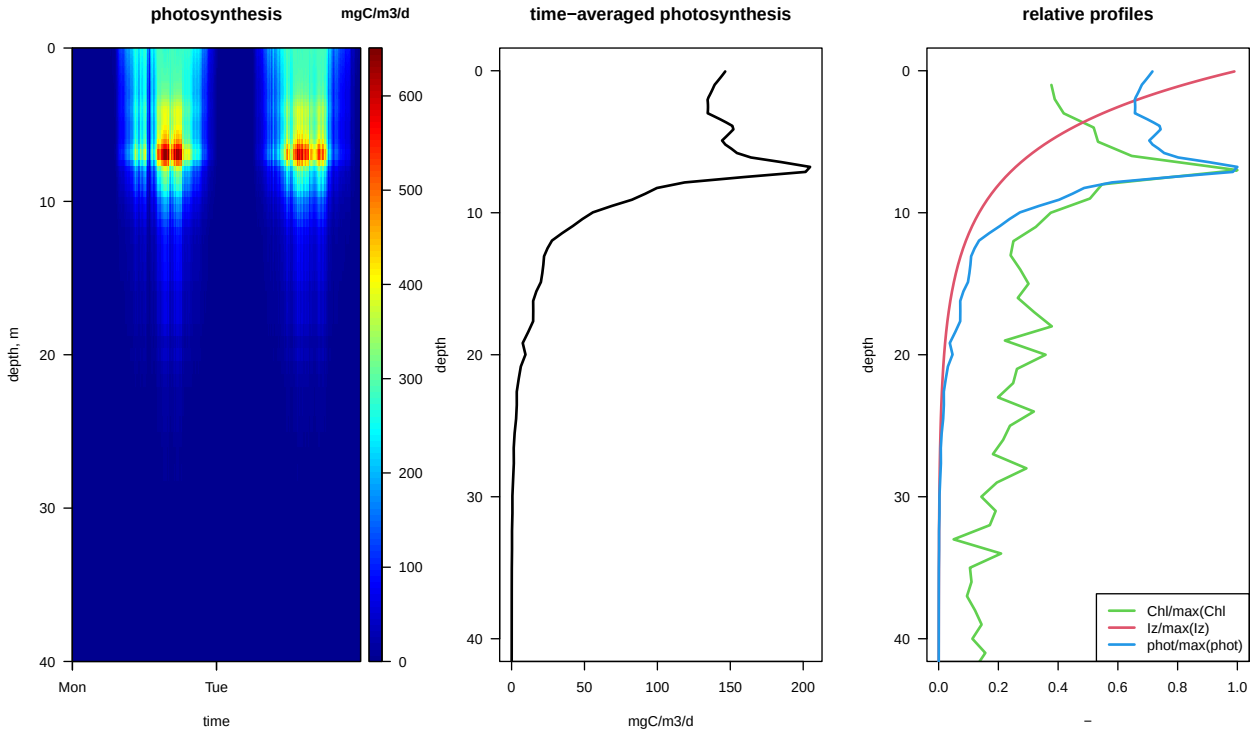


Figure 9: LabSTAF-derived photosynthesis

The daily mean production is now estimated from the timeseries in the PP list:

```
mean(PSLS$ts$PP) # mg C/m2/d
```

```
## [1] 1617.297
```