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

Karline Soetaert

20-December-2023

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. Measurements were done with the FRRF and Labstaf.

Apart from these 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 CTD will be used to estimate depth-dependent PI parameters.

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

```
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
```

```
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"))
```

Reading FRRF data

The PI curves from the FRRF (2 replicates) are read first.

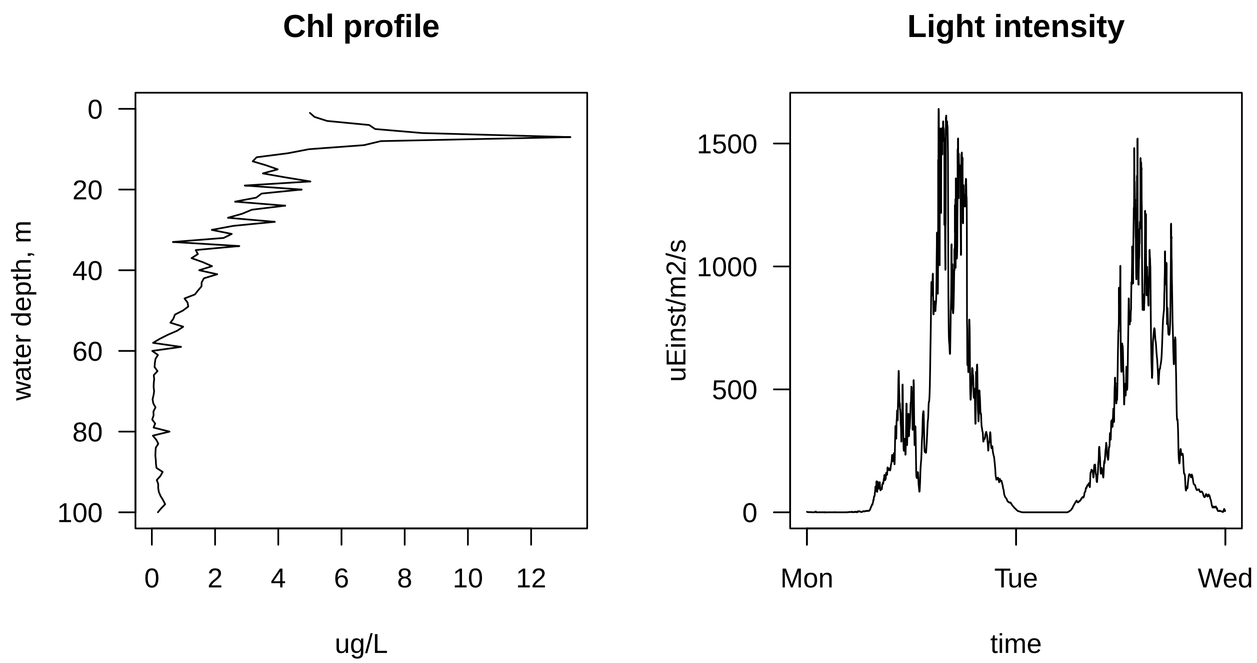


Figure 1: Accessory data needed to estimate depth-integrated PP

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

```
dir   <- "../raw_data/FRRF/"
files <- list.files(dir)

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 and pasted in one data.frame.

The 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.i
## 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     0
## 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     0
##   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

To standardize the FRRF data, the blanc values, *Fblanc* need to be passed. Also the cross-sectional surface of the PSII system in the dark (*aLHII_0*) is needed. In case this is not passed as an argument, it is estimated by regressing a_LHII versus irradiance (E) for low values of E (< 100), and taken as the offset.

We first standaridize all replicates at once, so that we estimate ONE value for a_LHII_0. We then check whether one value is realistic, by plotting aLH_II versus E.

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

# Show the attributes
attributes(FRRF_std_a)$unit_JVPII
```

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

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

```
## (Intercept)
##  0.01966198
```

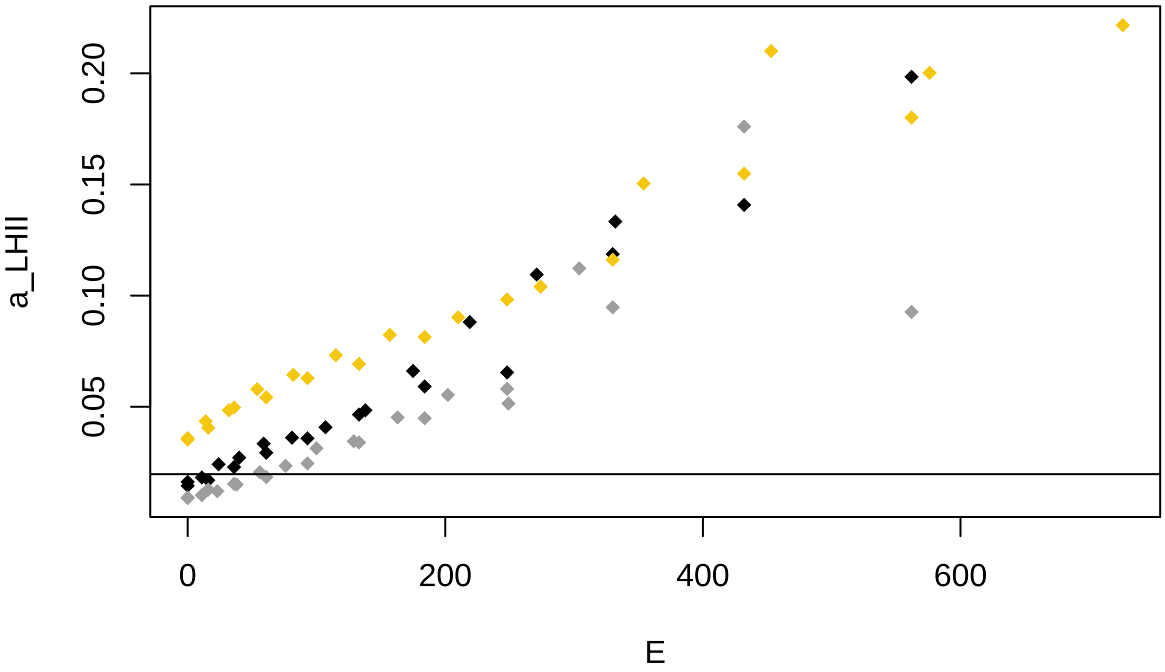
```
attributes(FRRF_std_a)$ka
```

```
## NULL
```

```
head(attributes(FRRF_std_a)$processing)
```

```
## [1] "Standardized with Fblanc =  0.194 at 2023-12-22 14:31:37"
## [2] "Standardized with Fblanc =  0.194 at 2023-12-22 14:31:37"
## [3] "Standardized with Fblanc =  0.194 at 2023-12-22 14:31:37"
## [4] "Standardized with Fblanc =  0.194 at 2023-12-22 14:31:37"
## [5] "Standardized with Fblanc =  0.194 at 2023-12-22 14:31:37"
## [6] "Standardized with Fblanc =  0.194 at 2023-12-22 14:31:37"
```

```
with(FRRF_std_a,
     plot(E, a_LHII,
          col=depth, pch=18))
abline(h=aLHII_0 )
```



Based on the difference in the a_LHII versus E regression for the different water depths, a better option is to standardize each dataset separately (but combine the two replicates). This way, a different value of a_LHII_0 is estimated for each PI dataset.

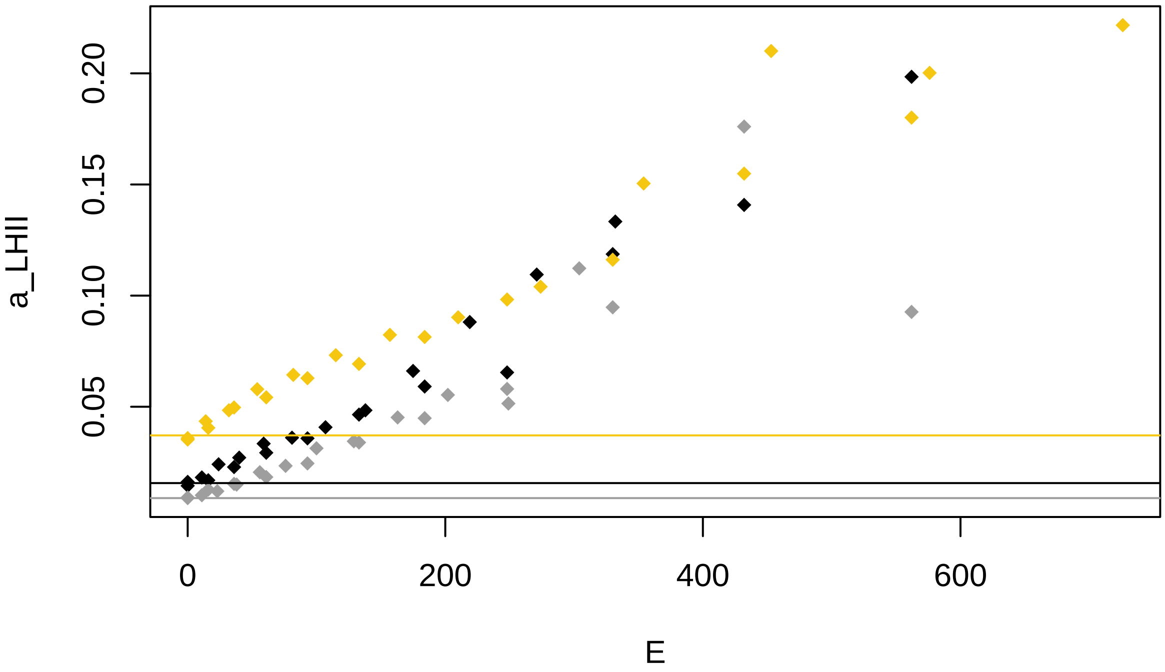
```
FRRF_1  <- subset(FRRF, subset = depth == 1)
FRRF_7  <- subset(FRRF, subset = depth == 7)
FRRF_40 <- subset(FRRF, subset = depth == 40)

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

a1 <- attributes(FRRF_1)$aLHII_0
a2 <- attributes(FRRF_7)$aLHII_0
a3 <- attributes(FRRF_40)$aLHII_0

FRRF_std <- rbind(FRRF_1, FRRF_7, FRRF_40)
attributes(FRRF_std)$aLHII_0 <- c(a1, a2, a3)

with(FRRF_std,
     plot(E, a_LHII,
          col=depth, pch=18))
abline(h=attributes(FRRF_std)$aLHII_0, col=c(1,7,40))
```



Fitting the FRRF data

The standardized data can now be fitted with a PI function. 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, as there are 9 cases to be fitted. The function also plots the fits, so as to see whether this worked properly.

```
profile <- unique(FRRF_std[,c("depth", "replicate")])

PARS <- NULL

# Function for fitting (and plotting the fit)
fitProfile <- function(Depth, Replicate){

  Sub <- subset(FRRF_std,
                subset = depth     == Depth    &
                         replicate == Replicate)

  FIT <- fitPI(model    = "EP",
               response = Sub$JVPII,
               I        = Sub$E)

  plot(FIT,
       main = paste("depth=", Depth,", replicate=", Replicate))

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

# call the function for each depth x replicate case
par(mfrow=c(3,2))
Fits <- NULL
```

```
for (i in 1: nrow(profile)){

  fitcase <- fitProfile(Depth      = profile[i,1],
                        Replicate  = profile[i,2])

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

```
Fits
```

```
##         depth replicate      alpha     eopt         ps
## fitcase     1         1 0.03278811 357.9244 2.4481967
## fitcase     1         2 0.02745059 202.9979 2.0616100
## fitcase     7         1 0.06808442 596.8451 8.6825081
## fitcase     7         2 0.06429444 418.0445 7.5030978
## fitcase    40         1 0.01989708 354.9617 0.8823484
## fitcase    40         2 0.01178562 162.4453 0.8677081
```

Chlorophyll-specific PI parameters

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

Here we have the choice to use the chlorophyll as it has been measured by the FRRF apparatus, or to pick the chlorophyll as measured with the CTD.

We merge the parameter file with both Chl measures.

We first calculate the mean Chl concentration as measured with FRRF:

```
# 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:

```
# Corresponding Chl from the CTD
Distance <- outer(X = CTDchl$depth,
                  Y = profile$depth,
                  FUN = function(x,y) abs(x-y))

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 0.03278811 357.9244 2.4481967  7.176909  5.000000
## 2     1         2 0.02745059 202.9979 2.0616100  7.253000  5.000000
```

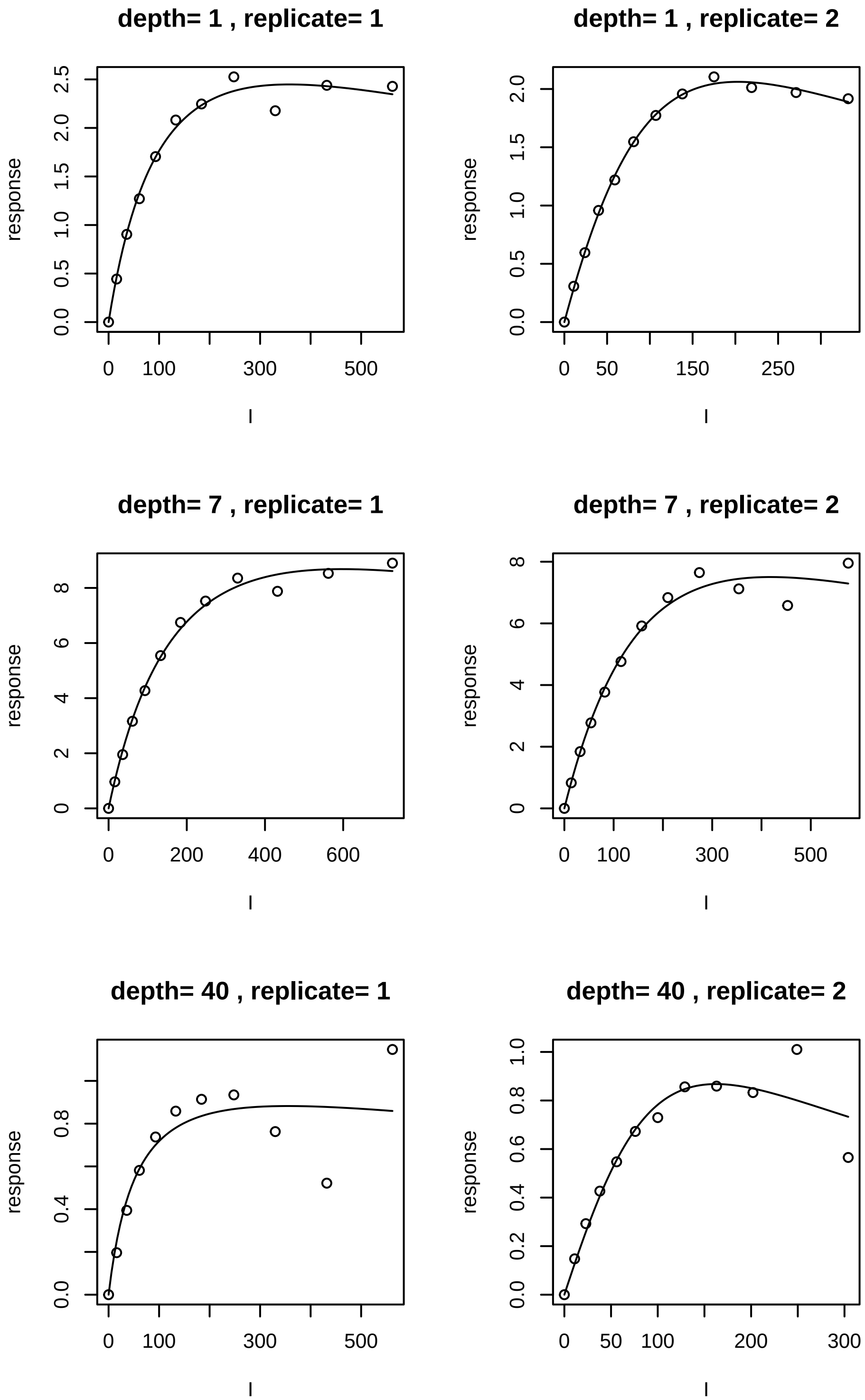


Figure 2: PI fits of the standardized FRRF data

```
## 3    40         1 0.01989708 354.9617 0.8823484  3.972727  1.496568
## 4    40         2 0.01178562 162.4453 0.8677081  4.022833  1.496568
## 5     7         1 0.06808442 596.8451 8.6825081 11.564500 13.245134
## 6     7         2 0.06429444 418.0445 7.5030978 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 0.03278811 357.9244 2.4481967  7.176909  5.000000 0.006557623 0.4896393
## 2     1         2 0.02745059 202.9979 2.0616100  7.253000  5.000000 0.005490118 0.4123220
## 3    40         1 0.01989708 354.9617 0.8823484  3.972727  1.496568 0.013295139 0.5895812
## 4    40         2 0.01178562 162.4453 0.8677081  4.022833  1.496568 0.007875101 0.5797987
## 5     7         1 0.06808442 596.8451 8.6825081 11.564500 13.245134 0.005140335 0.6555244
## 6     7         2 0.06429444 418.0445 7.5030978 11.957583 13.245134 0.004854193 0.5664796
```

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

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

```
##    alpha_chl         eopt       ps_chl
## 7.202085e-03 3.488698e+02 5.488909e-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)
```

```
## Warning in data.frame(depth = CTDchl$depth, alpha = meanPIpar["alpha_chl"] * : row names were found
## variable and have been discarded
```

Depth-integrated photosynthesis

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

We convert from mmol electrons per m3 per hour to mg C/m3/d by assuming that we need 5 electrons per carbon, 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
kz    <- 0.2  # /m
PS <- integratedPP(times=times, PI.par=PI.pars, It.data=par, kz=kz, convFac=fac)
plot(PS, mass="mgC", time="d")
```

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

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

```
## [1] 1253.414
```

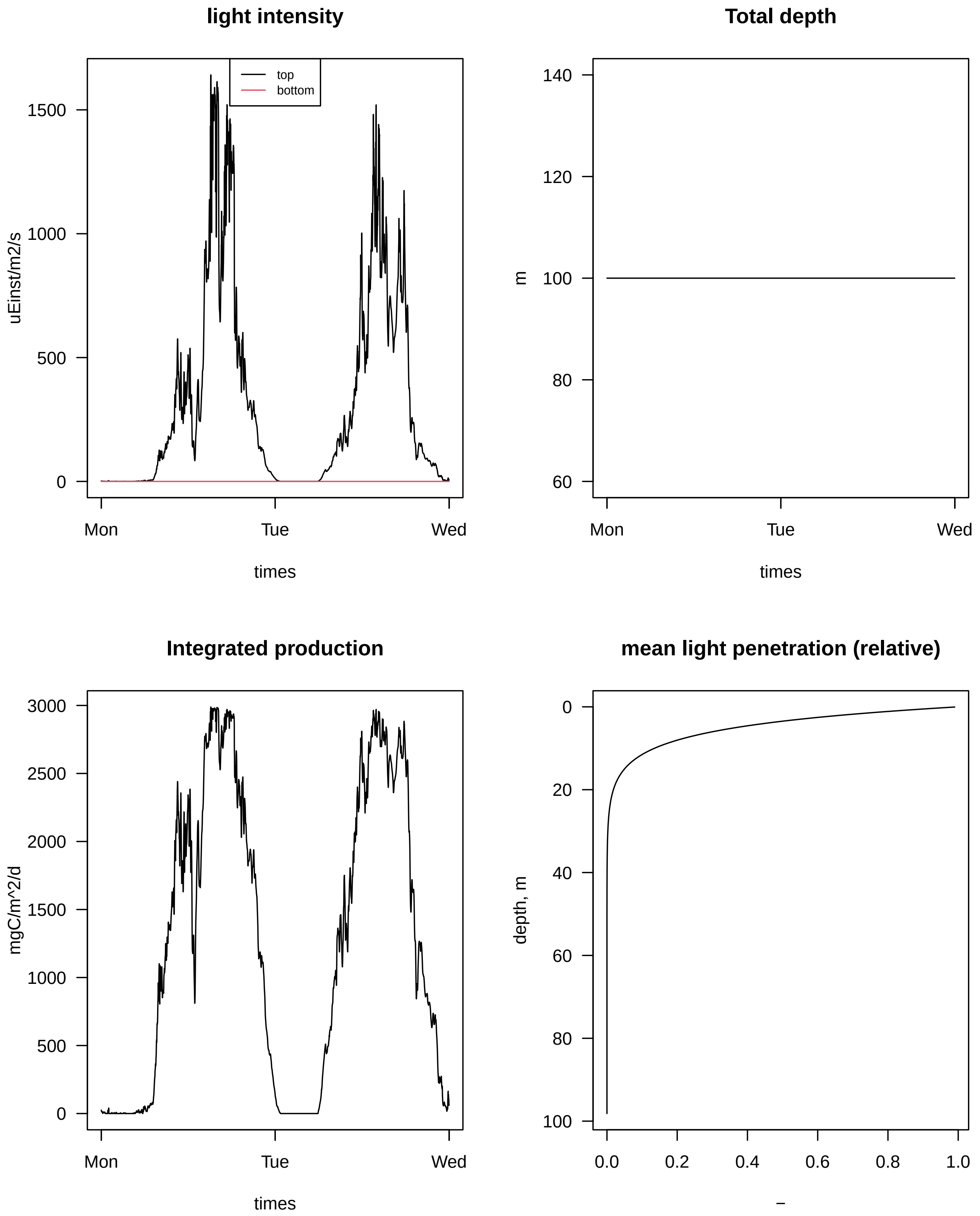


Figure 3: integrated production using FRRF data

labstaf data

```
dir   <- "../raw_data/labstaf"
files <- list.files(dir)

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)
)
```

For reading Labstaf data, the data files are stored with tab-separated; they can be read with read.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        N
## 2   NA  NA   0.08256     3.169 0.2477  3584    NA   NA   NA NA    NA     NA         NA    NA        N
##    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, the volumetric electron flux (JVPII) is already calculated by the machine, assuming an inputted Fblanc, and so 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 *dtBioG* is consistent with what is done by the LabSTAF.

For the depth=1, the aLHII_0 values is 0.0465 (as given in the input file), so if we standardize the rates with this value, and assuming a blanc = 0, we obtain quasi 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   = 0.0465)

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

```
##           JVPII JVPII_uc
##  [1,] 0.0000000       NA
##  [2,] 0.4318858   0.1470
##  [3,] 0.7888740   0.3858
##  [4,] 1.2487649   0.6116
##  [5,] 1.6644059   0.8128
##  [6,] 2.0481136   1.0020
##  [7,] 2.3506849   1.1470
##  [8,] 2.6668998   1.3010
##  [9,] 2.6369651   1.2870
## [10,] 2.6037813   1.2700
## [11,] 2.8176275   1.3760
## [12,] 2.7595518   1.3490
```

We thus re-standardize the data, using the correct blancs, and converting the estimated JVPII to *mmol e/m3/hour*.

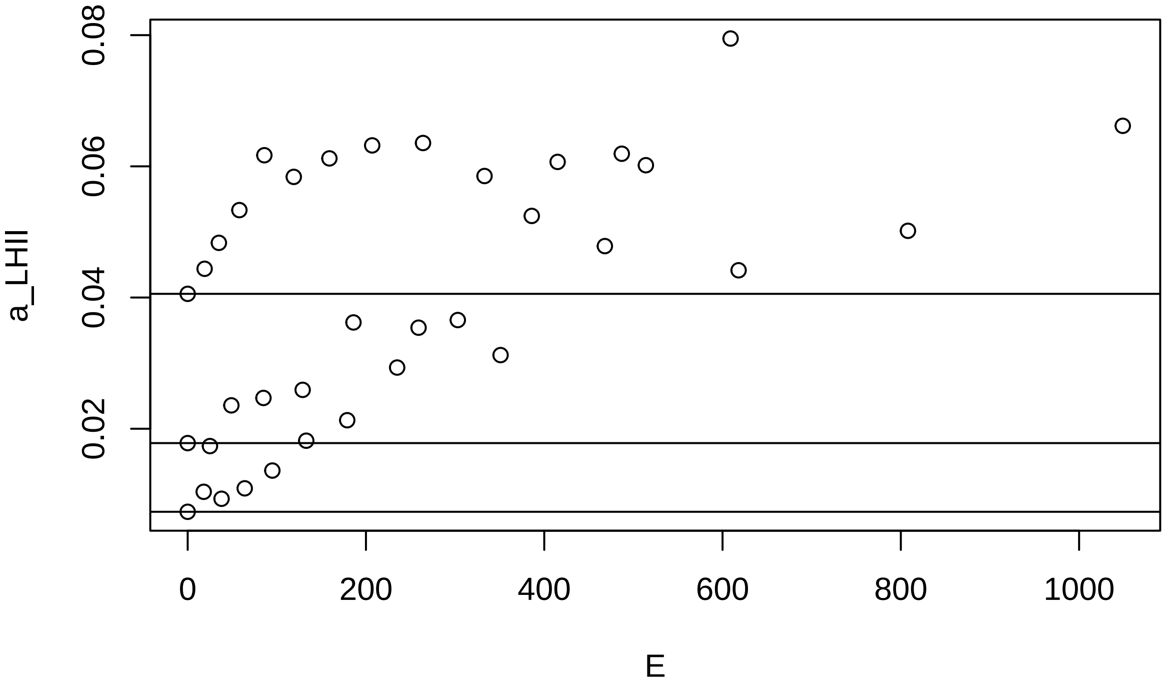
```
S1  <- standardizeFRRF(subset(LabSTAF, subset= depth==1),
                       Fblanc    = 0.194,
                       convJVPII = 3.6)
S2  <- standardizeFRRF(subset(LabSTAF, subset= depth==7),
                       Fblanc    = 0.175,
                       convJVPII = 3.6)
S3  <- standardizeFRRF(subset(LabSTAF, 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 reltively large difference between the aLHII_0 from the LabSTAF and the values generated with the standardization function.

```
with(rbind(S1, S2, S3), plot(E, a_LHII))
abline(h=aLHII_0)
```



The rest 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", pch=c(1, 18), legend=c("corrected", "uncorrected"))

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:

```
FitLS$Chl <- c(5, 13.245134, 1.496568)
FitLS$alpha_chl <- FitLS$alpha/FitLS$Chl
FitLS$ps_chl <- FitLS$ps/FitLS$Chl
meanLSpar <- apply(FitLS, MARGIN=2, FUN=mean)
```

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

```
## Warning in data.frame(depth = CTDchl$depth, alpha = meanLSpar["alpha_chl"] * : row names were found f
## variable and have been discarded
```

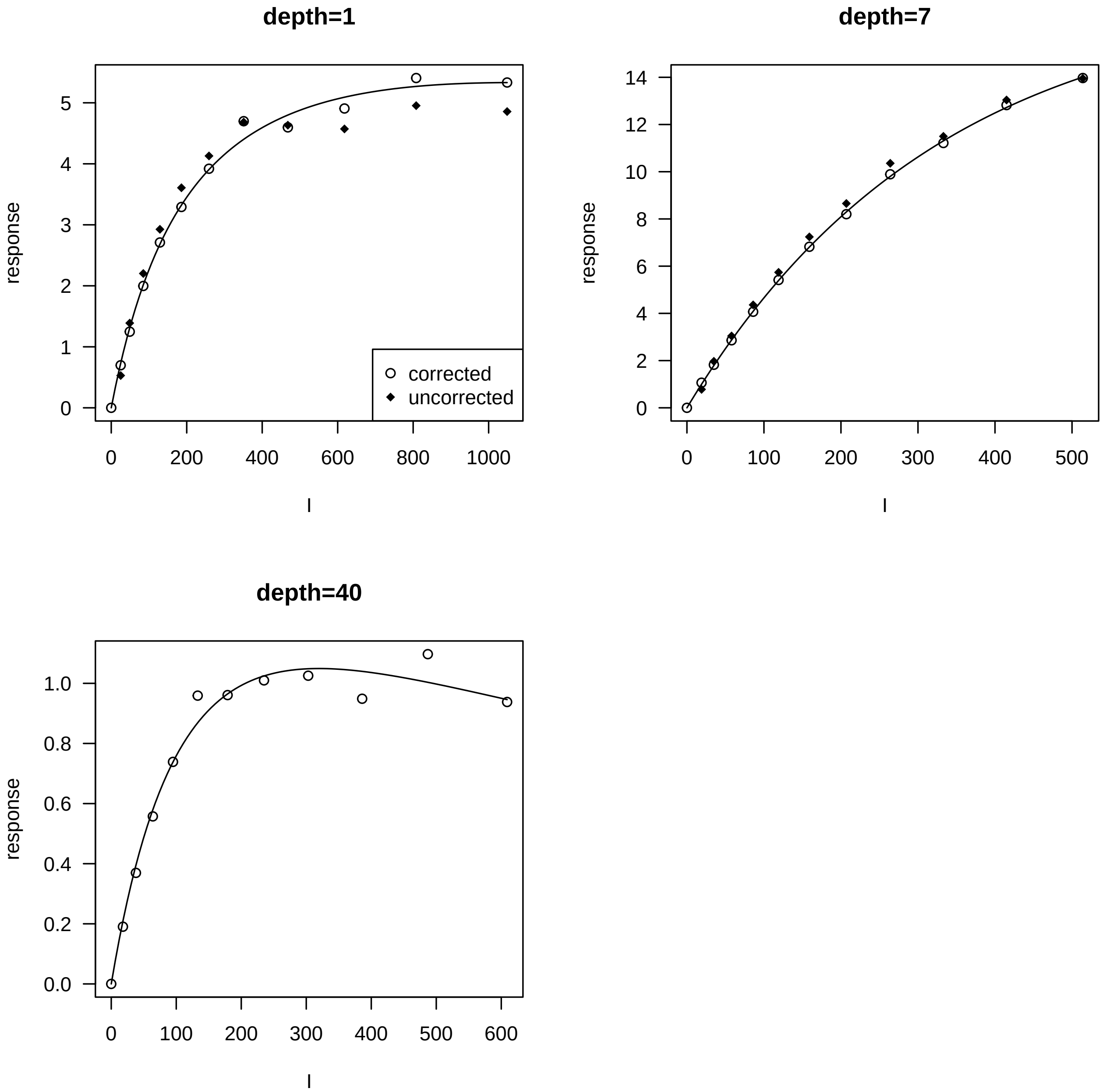


Figure 4: PI fits for the standardized labSTAF data (also shows uncorrected data)

```
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")
```

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
```

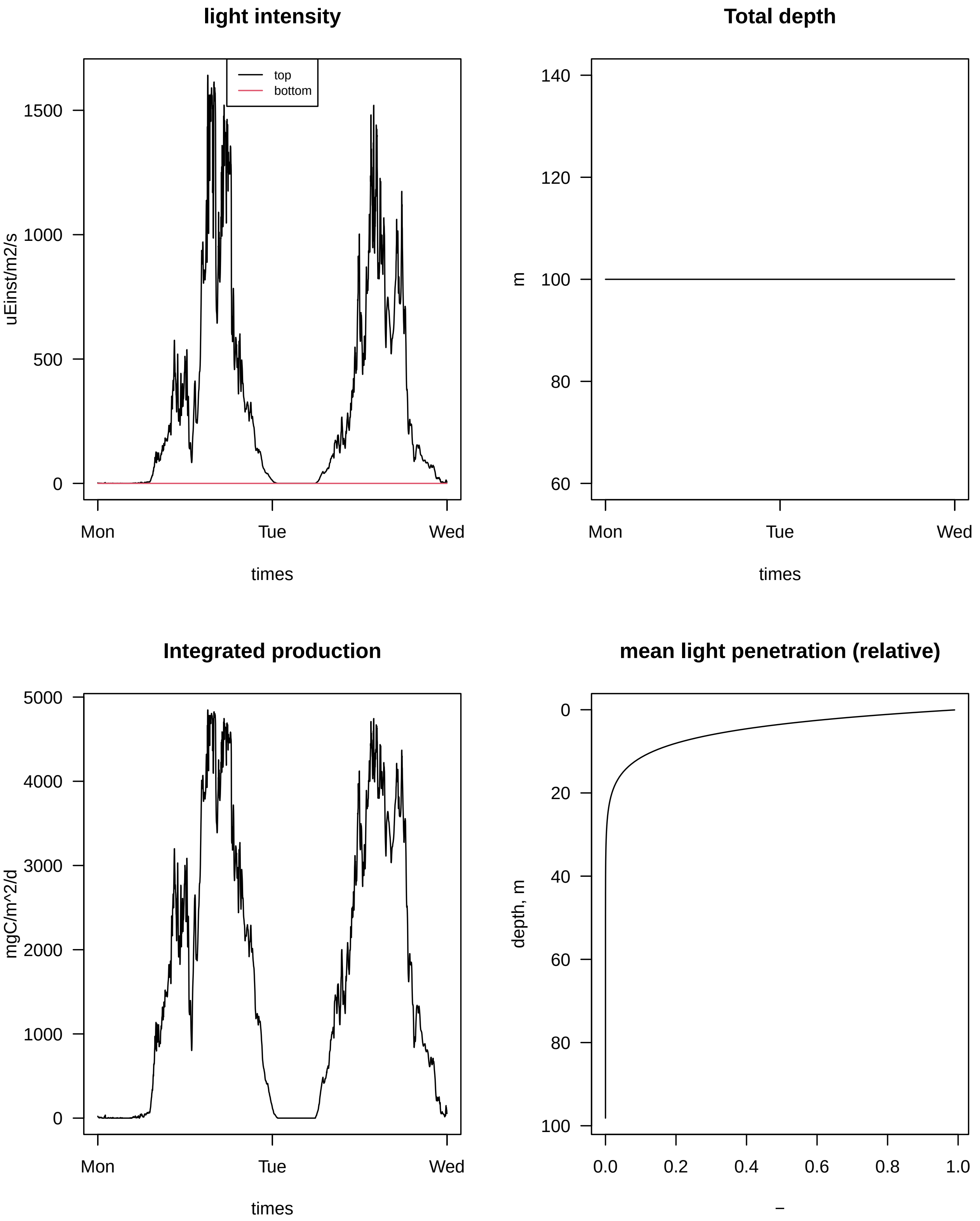


Figure 5: integrated production using LabSTAF data