Processing CANOE LabStaf data

Karline Soetaert

2023-07-19

Abstract

This document pre-processes the Labstaf data: it reads the separate data files, adds the blanc values, and finds the station positions. The PI data are then fitted and the data are normalised. The normalized labstaf data are written as one long data file; the PI parameters are written to a binary file and as a csv file.

Underway data

The Labstaf dates are in Greenland time + 1 hour.

Time in the underway data is in universal time (UTC), which is two hours later than Greenland time.

Labstaf data

The Labstaf was operated by Jens Dujardin.

Its settings were, e.g. for station 6B:

STAF system	STAF setup	pulses	Temperatures at start
LabSTAF: 21-2937-012	416 nm: 0.4141	1st pulse: 100 μs	Sample: 19.7 ^{o} C
Mode: Auto FLC	452 nm: 0.5139	2nd pulse: 100 μs	System: 25.7 ^{o} °C
Date: Jul 19, 2023	452 nm: 0.5289	Start gap: 400 μs	
Time: 17:59	473 nm: 0.4531	End gap: 12800 μs	
Duration: 27:26 s	495 nm: 0.2844	Gap steps: 11	
Acqs: 1375	534 nm: 0.3785	Seq interval: 120 ms	
Saqs: 1375	594 nm: 0.4263	Seq / Acq: 8	
Groups: 125	622 nm: 0.2273	Acq / Saq: 1	
Group time: 11 s			
Set Fv/Fm: 0.5			
Ka: 11800			
PEC: 0.3758			
cPEC: 1.174			

Reading LabSTAF station data

The LabSTAF data have been exported in two ways: as excel files and as text file. For the excel files, the relevant data section was copy-pasted and written as a csv file. These two types of data need to be handled differently.

```
DIR <- "../raw_data"
ff <- list.files(DIR, pattern=".csv")
LABSlist <- NULL
```

```
for (f in ff)
  LABSlist <- rbind(LABSlist,
                    data.frame(file=f,
                               read.csv2(paste(DIR,f, sep="/"))[,1:22]))

cf <- colnames(LABSlist)
cf[cf %in% c("F.","Fm.")] <-
           c("Fo","Fm")

colnames(LABSlist) <- cf

LABSlist[,-(1:2)] <- lapply(LABSlist[,-(1:2)], FUN=as.numeric)

DIRS <- c("../raw_data/45O", "../raw_data/48F", "../raw_data/49O",
          "../raw_data/50B", "../raw_data/51O", "../raw_data/53M",
          "../raw_data/54O", "../raw_data/56F", "../raw_data/58B",
          "../raw_data/59B", "../raw_data/60B")

rLABs <- function(fn, dir){
  specs <- strsplit(fn, "m.txt")[[1]]
  specs <- strsplit(specs, "_")[[1]]
  specs <- specs[c(1, length(specs))]
  names(specs) <- c("station", "depth")
  fr <- readFRRF(file=fn, dir=dir,
                 specs=specs ,
                 txt  = "delim")
  fr
}

Flist <- NULL

for (d in DIRS){
  ff <- list.files(d, pattern=".txt")
  for (f in ff)
    Flist <- rbind(Flist,rLABs(f, d))
}

FF                 <- Flist[,colnames(LABSlist)]
LabSTAFlist        <- rbind(LABSlist, FF)

LabSTAFlist$statNr    <- substr(LabSTAFlist$station, 1, nchar(LabSTAFlist$station)-1)
LabSTAFlist$longitude <- LabSTAFlist$latitude <- NA
LabSTAFlist[,3:24]    <- lapply(LabSTAFlist[,3:24], FUN=as.numeric)
```

add Fblanc

```
FRRFsettings <- read.csv("../raw_data/settings/FRRFsettings.csv")

Fblanc <- aggregate(FRRFsettings$Fblanc, by=list(FRRFsettings$nr), FUN=mean)
Fblanc <- rbind(Fblanc, c(3, 0.1))  # Fblanc is unknown for station 3

colnames(Fblanc) <- c("statNr", "Fblanc")

LabSTAFlist <- merge(LabSTAFlist, Fblanc, all.x=TRUE)
```

add latitude and longitude

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

  ii   <- which (LabSTAFlist$station == stations$station[i])

  if (length(ii)){
    LabSTAFlist$latitude[ii]  <- stations$latitude[i]
    LabSTAFlist$longitude[ii] <- stations$longitude[i]
  }
}
```

Processing the LabSTAF data

The labstaf already calculates the volumetric electron flux (JVPII), assuming a blanc value = 0.

Based on measurements, required for the FRRF, the blanc fluorescence was around 0.1-0.2. We thus restandardize the data, using the estimated Fblanc for the FRRF data.

Standardization

The data are standardized with respect to the measured blancs (fluorescence in the absence of Chl).

The volumetric e-flux in mmol/m3/hour is estimated ($JVPII$), using the absorption method:

$$JV_{PII} = \frac{F_q}{F_m} \cdot a_{LHII} \cdot E \cdot 3.6$$

where $F_q = F'_m - F'_o$, $F'_m = F_m - F_b$, $F'_o = F_o - F_b$, $aLH_{II} = \frac{F'^d_o \cdot F'^d_m}{F_q} \cdot \frac{K_a}{1e^6}$, and where E is the incident light, F_o, and F_m are measured by the FRRF, and F_b is the blanc. K_a is a machine-dependent constant, with units of m^{-1}. For the FRRF, this is equal to 11800; for the LabSTAF the same value is used. The superscript d means that the measurement is done in the dark.

The volumetric electron flux $JVPII$ (in $\mu mol\ e^- m^{-3} s^{-1}$) is converted to units of $mmol\ e^- m^{-3} h^{-1}$.

```
LabSTAFlist$JVPII <- LabSTAFlist$JVPII*3600/1000
```

```
profile <- unique(LabSTAFlist[,c("station", "depth")])
LSlist <- NULL

for (i in 1: nrow(profile)){
   sub <- subset(LabSTAFlist,
                 subset=station == profile[i,1]  &
                        depth   == profile[i,2]  )

   std <-  standardizeFRRF(sub,
                           Fblanc    = sub$Fblanc,
                           convJVPII = 3.6)
  LSlist <- rbind(LSlist, std)

}

# negative JVPII values are set to NA
LSlist$JVPII[LSlist$JVPII <= 0 ] <- NA
```

```
save(file = "../processed_data/LSlist.rda",      LSlist)
save(file = "../processed_data/LabSTAFlist.rda", LabSTAFlist)
write.csv(file = "../processed_data/LSlist.csv", LSlist)
```

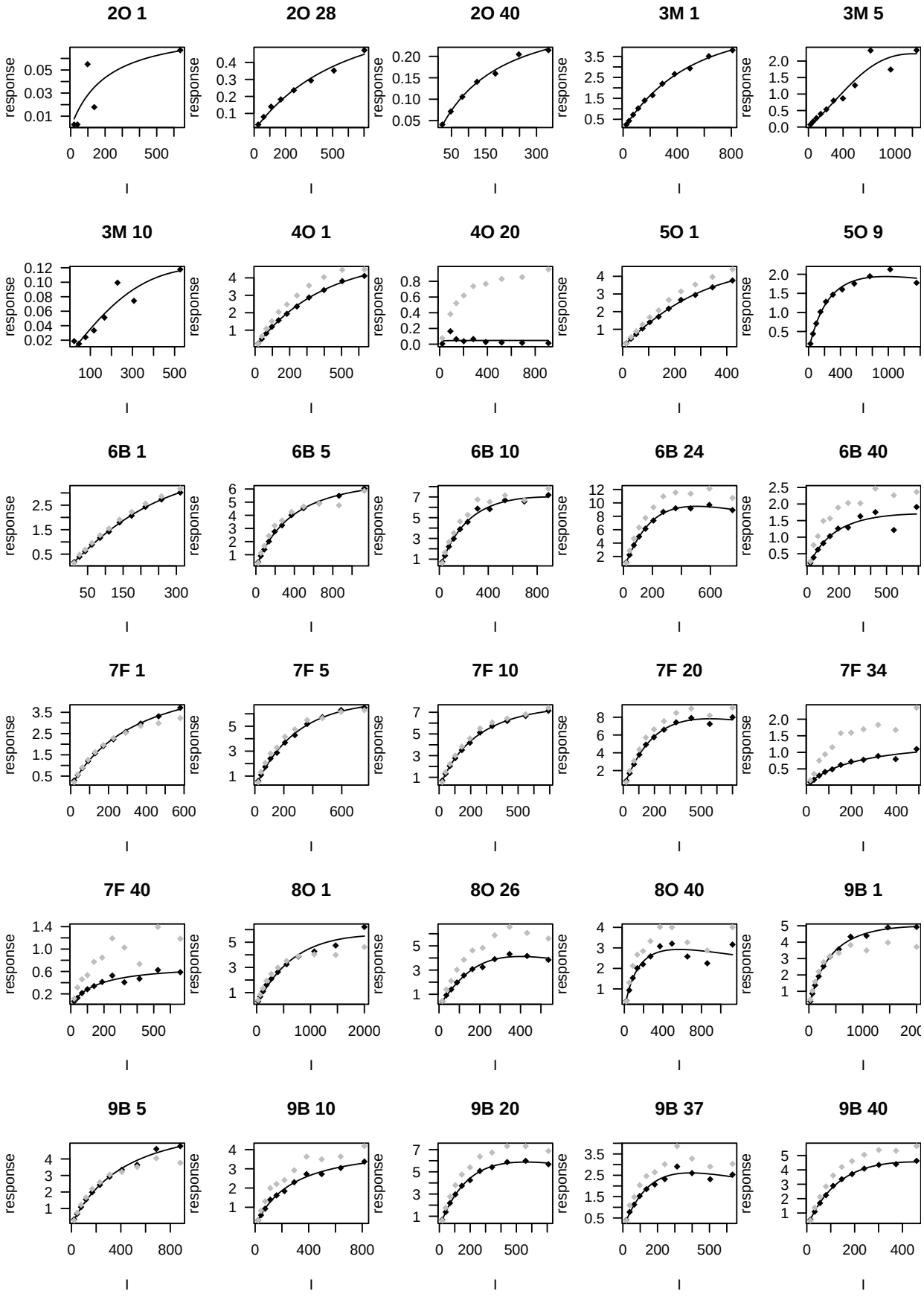
Fitting the PI curves

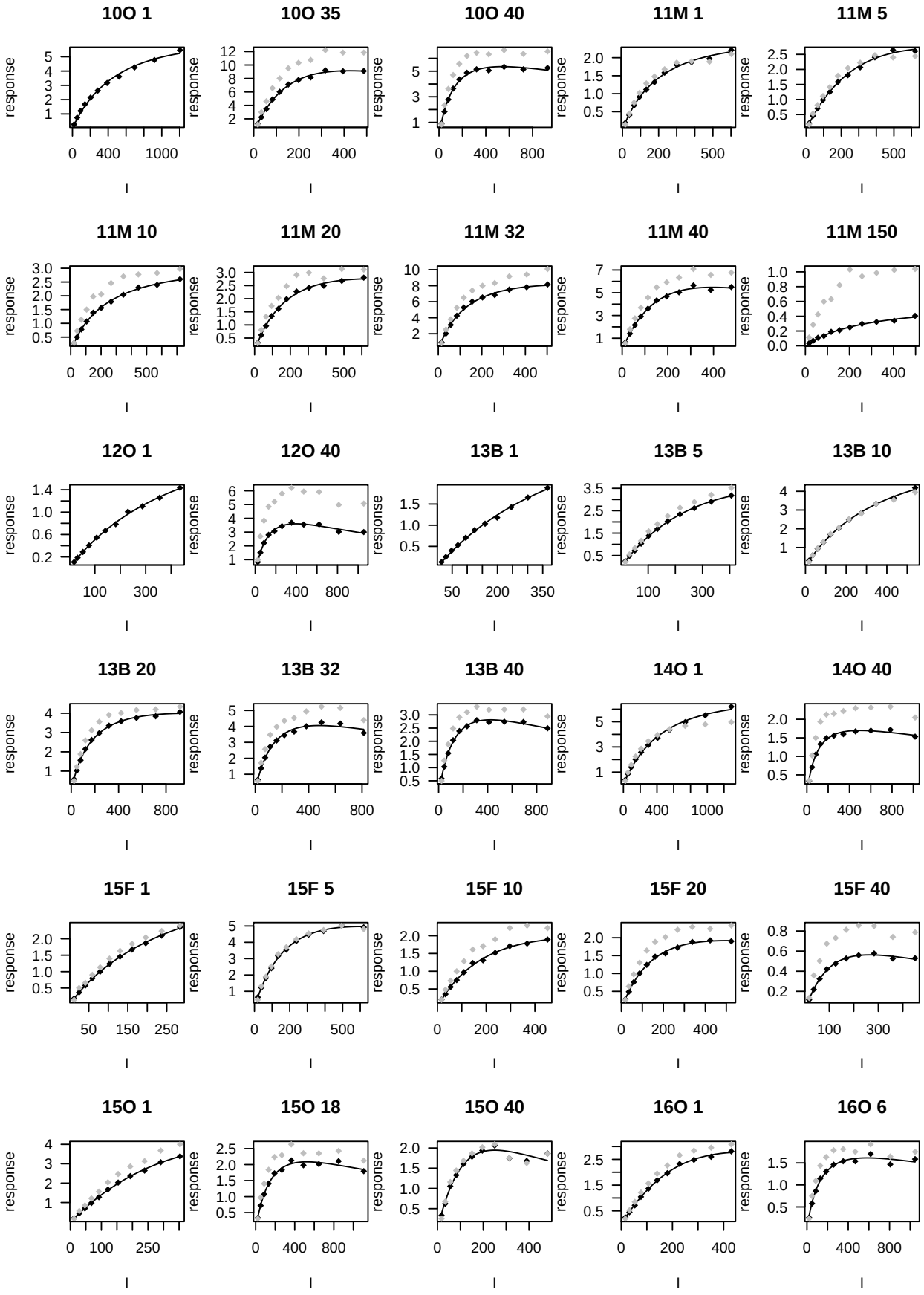
The JVPII variable, in units of mmol electrons $m^{-3} h^{-1}$, is fitted as a function of light using the Eilers-Peeters model.

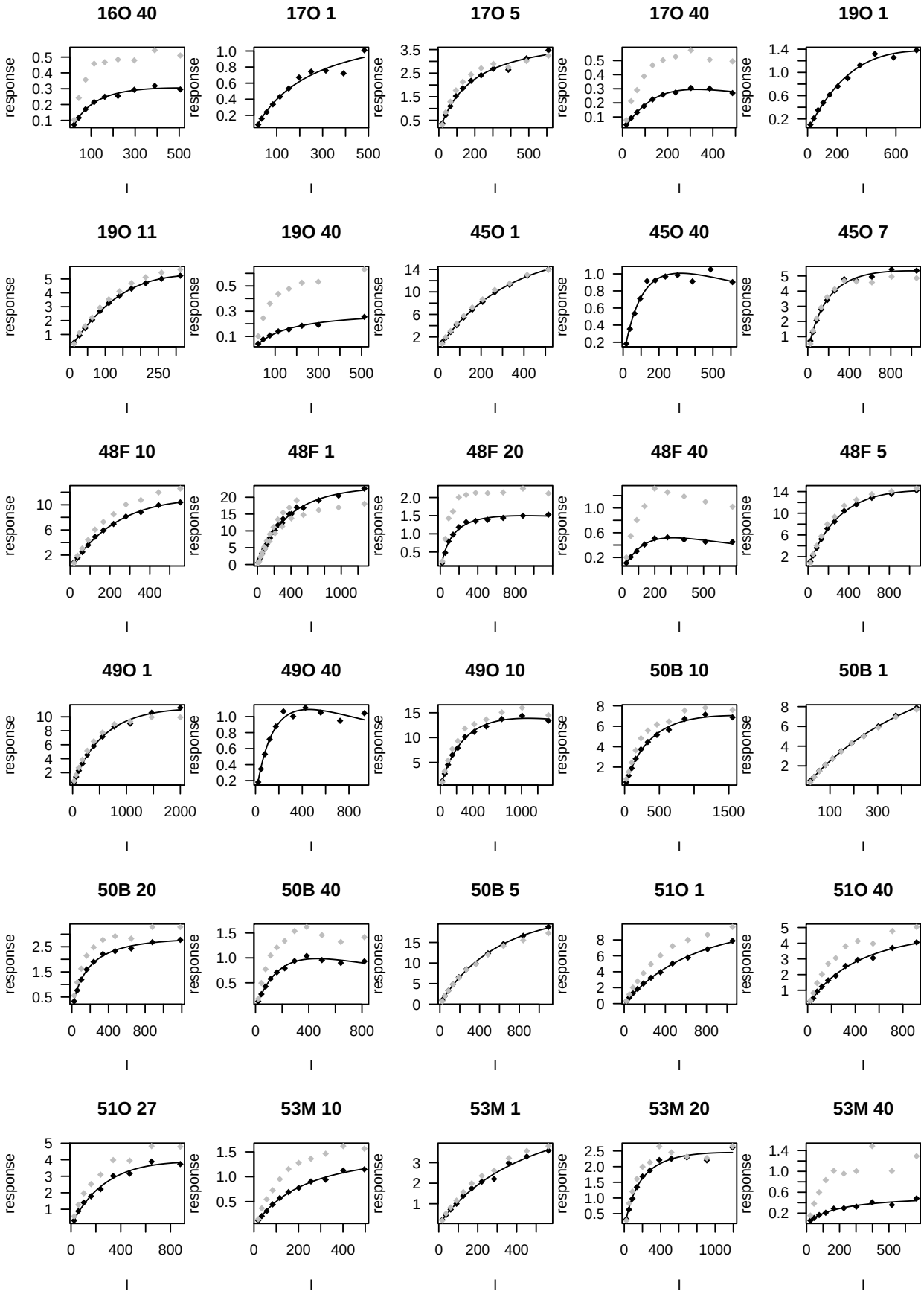
Datasets that contain less than 4 datapoints are ignored.

We fit the reprocessed data points (circles), but also show the uncorrected values. In some cases, they differ significantly.

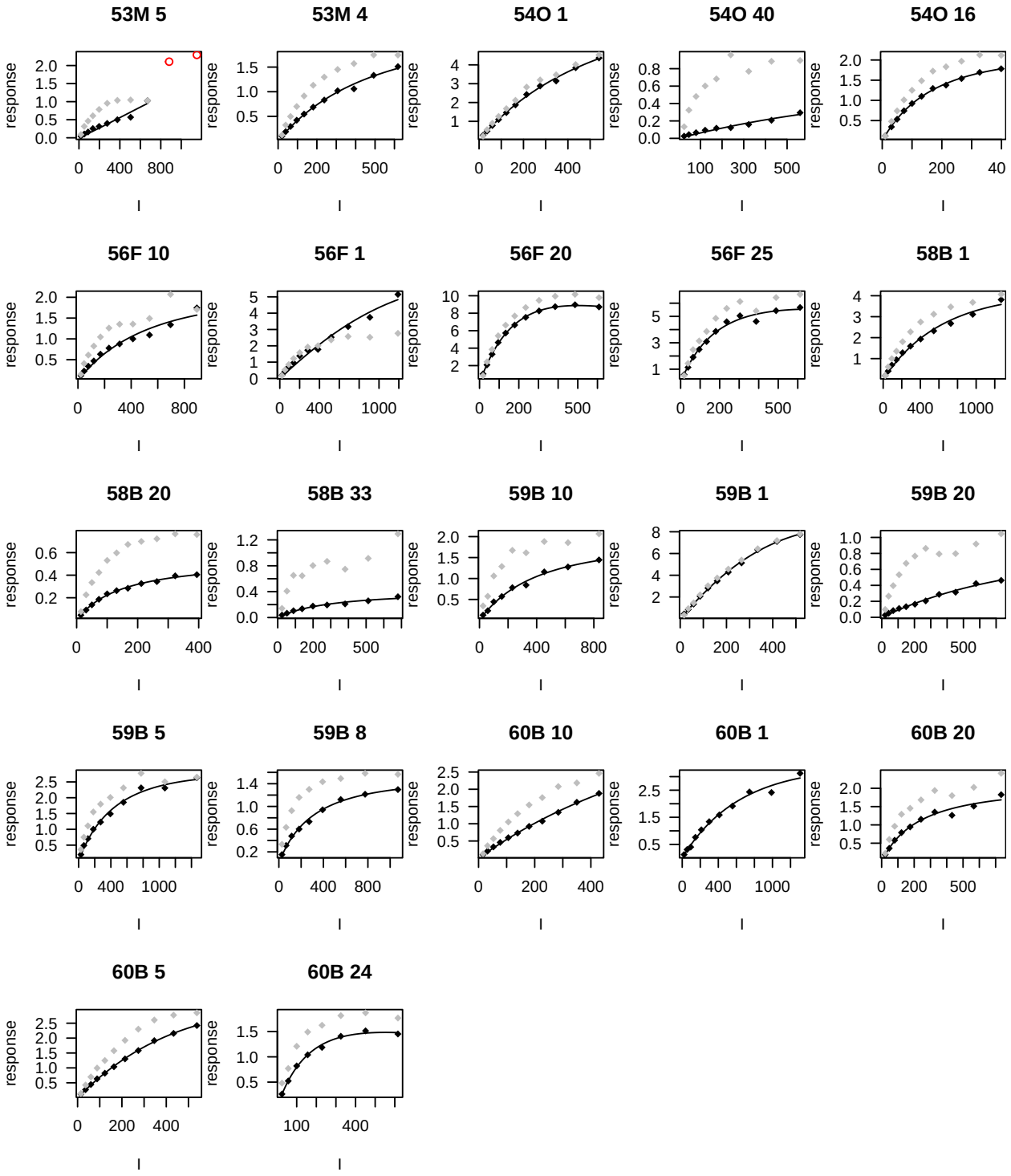
```r
par(mfrow=c(6,5), mar=c(4,4,4,0), las=1)

# unique combinations of station and measurement depth
Uin <- unique(data.frame(station=LSlist$station,
                         depth  =LSlist$depth))

# Fit all PI curves and put in one list
FIT  <- NULL
Pmax <-1000     # ignore measurements above 1000 uEinst/m2/s (i.e. no)
minRows  <- 4

Uin$used <- TRUE

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

  frrf <- subset(LSlist,
                 subset = station==Uin$station[i] &
                          depth  ==Uin$depth[i])
  if(Uin$station[i] == "53M" & Uin$depth[i]==5){
    nf <- nrow(frrf)
    NAremove <- frrf[(nf-1):nf,]
    frrf <-   frrf[1:(nf-2),]
  } else NAremove <- list(E=NA, JVPII=NA)

  frrf <- frrf[! is.na(frrf$JVPII), ]

  if (nrow(frrf) > minRows){
    STAT <- paste(Uin$station[i], Uin$depth[i]) # to label the plot
    fitone <-  fitPI(I = frrf$E, response = frrf$JVPII, model="EP")
    xlim <- range(c(frrf$E, NAremove$E), na.rm=TRUE)
    ylim <- range(c(frrf$JVPII, frrf$JVPII_uc, NAremove$JVPII), na.rm=TRUE)
    plot(fitone, main=STAT, ylim=ylim, pch=18, xlim=xlim)
    points(frrf$E, frrf$JVPII_uc, pch=18, col="grey")
    if (!is.na(NAremove$E[1])) points(NAremove$E, NAremove$JVPII, col="red")
    FIT <- rbind(FIT,
                 data.frame(
                   station = Uin$station[i],
                   depth   = Uin$depth[i],
                   alpha   = fitone$par["alpha"],
                   eopt    = fitone$par["eopt"],
                   ps      = fitone$par["ps"],
                   rsq     = r.squared(fitone))
                )
  } else Uin$used[i] <- FALSE
}
```


2O 1
2O 28
2O 40
3M 1
3M 5
3M 10
4O 1
4O 20
5O 1
5O 9
6B 1
6B 5
6B 10
6B 24
6B 40
7F 1
7F 5
7F 10
7F 20
7F 34
7F 40
8O 1
8O 26
8O 40
9B 1
9B 5
9B 10
9B 20
9B 37
9B 40
response


10O 1
10O 35
10O 40
11M 1
11M 5
11M 10
11M 20
11M 32
11M 40
11M 150
12O 1
12O 40
13B 1
13B 5
13B 10
13B 20
13B 32
13B 40
14O 1
14O 40
15F 1
15F 5
15F 10
15F 20
15F 40
15O 1
15O 18
15O 40
16O 1
16O 6
response


16O 40
17O 1
17O 5
17O 40
19O 1
19O 11
19O 40
45O 1
45O 40
45O 7
48F 10
48F 1
48F 20
48F 40
48F 5
49O 1
49O 40
49O 10
50B 10
50B 1
50B 20
50B 40
50B 5
51O 1
51O 40
51O 27
53M 10
53M 1
53M 20
53M 40
response

```
FIT <- as.data.frame(FIT)

FIT$statnr <- as.integer(substr(FIT$station, 1, nchar(FIT$station)-1))
```




```
fs <- unique(FIT$station)
FIT$chl_ctd <- NA
chl_c <- NULL
```

```
for (s in fs){
  ii  <- which(FIT$station==s)
  ctd <- subset(CTDall, subset=station==s)
  if (nrow(ctd) > 0){

    for (i in ii){
      D     <- FIT$depth[i]
      Dmin  <- max(0, D-1)
      ctd_s <- subset(ctd, subset=
                        Depth >= Dmin  & Depth <= D+1  &
                        ! is.na(Chlorophyll) & Chlorophyll > 0)
      chl_c <- mean(ctd_s$Chlorophyll, na.rm=TRUE)
      FIT$chl_ctd[i] <- chl_c
    }
  }
}
```

Of the 154 datasets, 112 were fitted; 42 did not contain good data.

Here is a list of the depth strata per station whose data were either successfully fitted or ignored.

```
USE <- cbind(aggregate(Uin$used,   by=list(Uin$station), FUN=sum),
             aggregate(1-Uin$used, by=list(Uin$station), FUN=sum)[,2])
names(USE)  <- c("station", "used", "ignored")
USE <- merge(stations[, 1:2], USE)
USE <- USE[order(USE$statNr),]
knitr::kable(USE, row.names = FALSE,
             caption="Depth levels per station that were fitted (used) or had no good JVPII values (igno
```

Table 2: Depth levels per station that were fitted (used) or had no good JVPII values (ignored)

station	statNr	used	ignored
2O	2	3	1
3M	3	3	3
4O	4	2	1
5O	5	2	2
6B	6	5	1
7F	7	6	2
8O	8	3	0
9B	9	6	1
10O	10	3	1
11M	11	7	1
12O	12	2	1
13B	13	6	2
14O	14	2	0
15F	15	5	2
15O	15	3	1
16O	16	3	1
17O	17	3	1
19O	19	3	1
45O	45	3	1
48F	48	5	4
49O	49	3	1

station	statNr	used	ignored
50B	50	5	2
51O	51	3	1
54O	54	3	0
56F	56	4	4
58B	58	3	4
59B	59	5	2
60B	60	5	0

for stations the LabSTAF failed completely, so their photosynthesis cannot be estimated.

Resulting parameters

```
fit_Labstaf <- FIT
fit_Labstaf <- merge(fit_Labstaf,
               stations[,c("station", "statNr", "latitude", "longitude", "Fjord")],
               by="station")

knitr::kable(fit_Labstaf[,c("station", "depth", "alpha", "eopt", "ps", "chl_ctd", "rsq")],
    digits=c(0,0,4,0,2, 2, 2),
    caption="Photosynthesis parameters, derived from the LabSTAF measurements")
```

Table 3: Photosynthesis parameters, derived from the LabSTAF measurements

station	depth	alpha	eopt	ps	chl_ctd	rsq
10O	40	0.0540	540	5.36	6.56	0.99
10O	1	0.0136	2500	5.73	0.97	1.00
10O	35	0.0780	424	9.15	7.39	1.00
11M	1	0.0132	1259	2.34	0.66	1.00
11M	10	0.0144	2500	2.96	1.69	1.00
11M	20	0.0197	684	2.76	6.08	1.00
11M	32	0.0765	640	8.13	12.33	1.00
11M	5	0.0121	797	2.72	1.46	1.00
11M	150	0.0021	2500	0.51	NaN	0.99
11M	40	0.0474	396	5.48	6.53	1.00
12O	40	0.0399	397	3.60	7.16	0.99
12O	1	0.0060	2500	2.27	0.42	1.00
13B	1	0.0085	2500	3.30	0.39	1.00
13B	20	0.0287	1019	4.00	2.35	1.00
13B	32	0.0386	479	4.05	5.75	0.99
13B	5	0.0164	805	3.58	1.70	1.00
13B	10	0.0172	2500	5.64	2.21	1.00
13B	40	0.0298	424	2.81	3.25	1.00
14O	1	0.0163	2500	6.41	1.37	1.00
14O	40	0.0219	497	1.70	3.05	0.99
15F	1	0.0159	2500	3.91	0.71	1.00
15F	20	0.0163	502	1.92	2.02	1.00
15F	40	0.0073	265	0.56	1.81	1.00
15F	5	0.0359	610	4.97	2.07	1.00
15F	10	0.0123	665	1.95	1.46	1.00

station	depth	alpha	eopt	ps	chl_ctd	rsq
15O	1	0.0161	779	4.10	1.12	1.00
15O	18	0.0178	523	2.09	1.87	0.99
15O	40	0.0228	250	1.95	1.14	0.97
16O	1	0.0153	495	2.79	0.92	1.00
16O	6	0.0162	583	1.61	2.54	0.98
16O	40	0.0037	519	0.31	0.73	0.99
17O	1	0.0055	2500	1.19	0.19	0.96
17O	5	0.0233	2500	3.80	3.26	0.99
17O	40	0.0025	329	0.30	0.47	0.99
19O	40	0.0020	2500	0.28	0.53	0.98
19O	1	0.0050	801	1.37	0.22	0.99
19O	11	0.0371	368	5.27	4.41	1.00
2O	28	0.0013	2500	0.61	0.36	0.99
2O	40	0.0019	1328	0.27	0.49	0.99
2O	1	0.0004	2500	0.08	0.09	0.65
3M	1	0.0103	1920	4.51	0.30	1.00
3M	5	0.0023	1208	2.22	0.25	0.93
3M	10	0.0004	686	0.12	0.39	0.90
45O	1	0.0543	1138	16.54	3.24	1.00
45O	40	0.0124	321	1.01	1.83	0.97
45O	7	0.0337	987	5.34	7.90	0.99
48F	1	0.0692	1901	22.74	2.52	1.00
48F	20	0.0144	910	1.50	3.60	0.99
48F	10	0.0475	780	10.75	3.65	1.00
48F	5	0.0512	1202	14.16	2.37	1.00
48F	40	0.0046	313	0.52	1.10	0.98
49O	1	0.0220	2500	11.11	0.97	1.00
49O	40	0.0094	457	1.09	1.19	0.97
49O	10	0.0577	1077	13.89	7.34	1.00
4O	1	0.0136	1462	4.90	1.16	1.00
4O	20	0.0124	2498	0.05	0.36	0.00
50B	1	0.0283	2500	13.27	0.57	1.00
50B	20	0.0181	1983	2.81	1.74	1.00
50B	40	0.0069	464	0.98	0.82	0.99
50B	10	0.0228	1643	7.05	1.47	1.00
50B	5	0.0403	2500	21.38	1.21	1.00
51O	40	0.0124	2500	4.65	1.67	0.99
51O	27	0.0154	1032	3.86	2.16	0.99
51O	1	0.0151	2500	9.26	0.42	1.00
54O	1	0.0140	2500	6.69	0.95	1.00
54O	40	0.0006	2500	0.50	0.77	0.97
54O	16	0.0134	888	1.98	2.16	1.00
56F	10	0.0041	2500	1.91	0.68	0.97
56F	1	0.0056	2500	6.02	0.59	0.98
56F	20	0.0636	501	8.88	2.27	1.00
56F	25	0.0368	697	5.56	5.44	0.99
58B	1	0.0068	2500	3.98	0.27	0.99
58B	20	0.0037	2500	0.50	0.66	1.00
58B	33	0.0013	2500	0.36	1.26	0.97
59B	10	0.0048	2500	1.70	0.98	0.99
59B	1	0.0246	867	8.65	0.49	1.00
59B	20	0.0010	2500	0.70	0.46	0.99

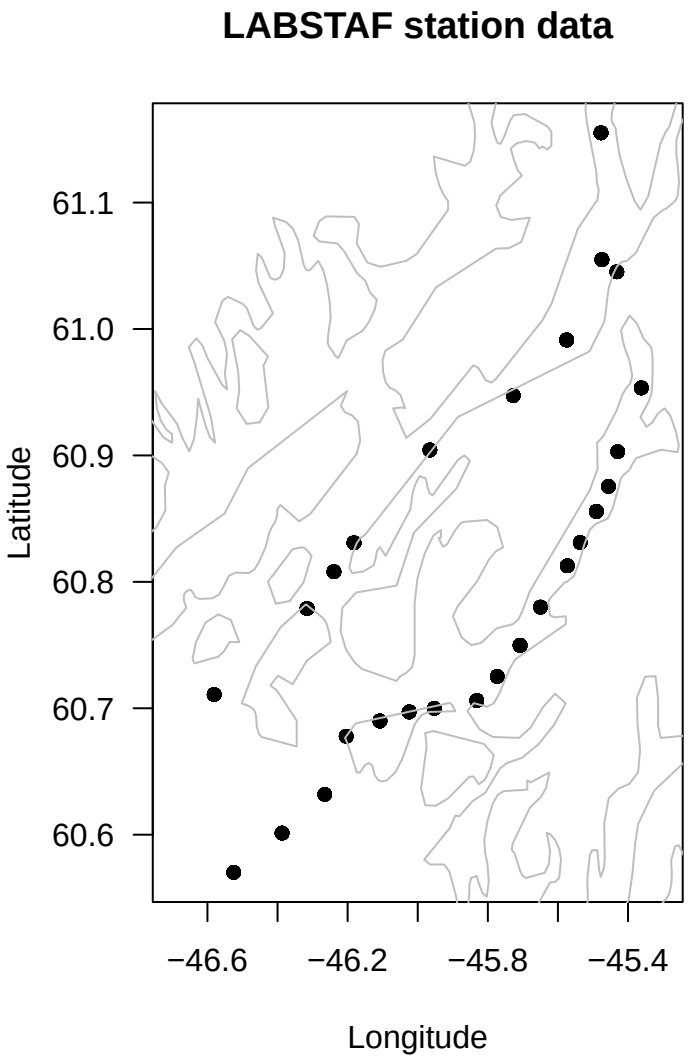
station	depth	alpha	eopt	ps	chl_ctd	rsq
59B	5	0.0070	2500	2.69	1.04	0.99
59B	8	0.0052	2500	1.41	1.22	1.00
5O	9	0.0094	972	1.94	2.12	0.98
5O	1	0.0149	819	4.36	1.36	1.00
60B	10	0.0058	2500	3.81	0.53	1.00
60B	1	0.0055	2500	3.25	0.30	0.99
60B	20	0.0092	2500	1.92	0.87	0.97
60B	5	0.0075	1231	3.00	0.26	1.00
60B	24	0.0115	552	1.48	1.16	1.00
6B	1	0.0151	645	3.65	1.85	1.00
6B	5	0.0197	2500	6.44	1.67	1.00
6B	40	0.0122	805	1.71	0.81	0.90
6B	10	0.0346	933	7.01	2.62	0.99
6B	24	0.0591	505	9.49	6.65	1.00
7F	1	0.0158	2500	4.76	1.41	1.00
7F	5	0.0266	1178	6.85	1.58	1.00
7F	34	0.0065	2500	1.26	2.47	0.97
7F	40	0.0044	2500	0.66	0.67	0.93
7F	10	0.0364	1242	7.51	3.02	1.00
7F	20	0.0473	555	7.83	2.30	0.99
8O	26	0.0253	399	4.12	4.78	0.99
8O	40	0.0254	570	2.92	2.55	0.89
8O	1	0.0089	2500	5.56	0.35	0.97
9B	40	0.0391	469	4.56	4.17	1.00
9B	20	0.0374	561	5.88	4.54	1.00
9B	37	0.0208	391	2.61	6.35	0.97
9B	1	0.0147	2353	4.97	1.01	1.00
9B	5	0.0153	2500	5.58	1.22	0.99
9B	10	0.0157	2500	3.73	2.24	0.99

```
fit_Labstaf$station <- factor(fit_Labstaf$station,
                         levels = c("2O",  "3M" ,
                                    "4O", "5O", "6B", "7F", "8O", "9B",
                                    "10O", "11M", "12O", "13B", "14O", "15F",
                                    "15O", "16O", "17O", "19O", "45O", "48F",
                                    "49O", "50B", "51O", "52B", "53B", "54O",
                                    "55O", "56F", "57O", "58B", "59B", "60B" ))
fit_Labstaf <- fit_Labstaf[order(as.integer(fit_Labstaf$statNr)),]
```

```
save(file="../processed_data/fit_Labstaf.rda", fit_Labstaf)
write.csv(file="../processed_data/fit_Labstaf.csv", fit_Labstaf, row.names = FALSE)
```

Plotting the measurement positions

```
J0 <- 19553  # Julian day for the first day of the cruise

par(mfrow=c(1,2), las=1)
asp <- 1/cos((mean(LSlist$latitude) * pi)/180)

 with(LSlist,
     plot(longitude, latitude, asp=asp, pch=16,
                          main="LABSTAF station data", xlim=c(-46.7, -45.3),
                          xlab="Longitude",
                          ylab="Latitude"))
 lines2D(LATLONcoast$longitude, LATLONcoast$latitude,
        type="l", col="grey", add=TRUE)
```



Specifics of the LabSTAF measurements are printed here:

```
LL <- aggregate(LSlist$depth, by=list(LSlist$station),
                FUN=function(x) length(unique(x)))
names(LL) <- c("station", "numDepths")
LL <- merge(LL, stations[, c(1:5, ncol(stations))], by="station")
```

```
LL <- LL[order(as.numeric(LL$statNr)),]
knitr::kable(LL, digits=c(0,0,0,3,3,0), row.names=FALSE,
             caption="position of stations and number of Labstaf measurment depths")
```

Table 4: position of stations and number of Labstaf measurment depths

station	numDepths	statNr	latitude	longitude	depth	subarea
2O	4	2	60.570	-46.525	143	Igaliku_up
3M	6	3	60.601	-46.387	556	Igaliku_up
4O	3	4	60.632	-46.266	216	Igaliku_up
5O	4	5	60.678	-46.204	200	Igaliku_up
6B	6	6	60.690	-46.108	243	Igaliku_up
7F	8	7	60.697	-46.024	289	Igaliku_up
8O	3	8	60.700	-45.953	93	Igaliku_up
9B	7	9	60.706	-45.831	400	Igaliku_down
10O	4	10	60.725	-45.773	426	Igaliku_down
11M	8	11	60.750	-45.708	404	Igaliku_down
12O	3	12	60.780	-45.650	421	Igaliku_down
13B	8	13	60.813	-45.573	416	Igaliku_down
14O	2	14	60.831	-45.536	406	Igaliku_down
15F	7	15	60.856	-45.490	383	Igaliku_down
15O	4	15	60.856	-45.490	382	Igaliku_down
16O	4	16	60.875	-45.456	368	Igaliku_down
17O	4	17	60.903	-45.430	311	Igaliku_down
19O	4	19	60.953	-45.362	220	Igaliku_down
45O	4	45	60.711	-46.581	251	Tunulliarfik_up
48F	9	48	60.779	-46.316	384	Tunulliarfik_up
49O	4	49	60.808	-46.239	332	Tunulliarfik_up
50B	7	50	60.831	-46.182	321	Tunulliarfik_up
51O	4	51	60.904	-45.965	387	Tunulliarfik_down
54O	3	54	60.947	-45.727	292	Tunulliarfik_down
56F	8	56	60.991	-45.575	278	Tunulliarfik_down
58B	7	58	61.055	-45.475	219	Tunulliarfik_down
59B	7	59	61.155	-45.477	150	Tunulliarfik_down
60B	5	60	61.045	-45.432	66	Tunulliarfik_down

About the standardization

For some of the profiles, these is a large difference between Labstaf-generated estimates of JVPII (assuming blanc of 0), and the corrected values. We select the data for station 54O, at depth of 40m.

```
par(mfrow=c(2,2))
f_ori <- subset(LabSTAFlist, subset = station== "54O" & depth == 40)
frrf  <- standardizeFRRF(f_ori, Fblanc=f_ori$Fblanc)

f_ori2 <- subset(LabSTAFlist, subset = station== "54O" & depth == 1)
frrf2  <- standardizeFRRF(f_ori2, Fblanc=f_ori2$Fblanc)

with(frrf, matplot(x=E, y=cbind(JVPII, JVPII_uc), pch=16:17,
                   ylab="JVPII", main="station 54O, 40m"))
legend("right", legend=c("Fblanc=0", "Fblanc=0.132"), col=1:2, pch=16:17)
```

```r
with(frrf2, matplot(x=E, y=cbind(JVPII, JVPII_uc), pch=16:17,
                    ylab="JVPII", main="station 540, 1m"))
legend("right", legend=c("Fblanc=0", "Fblanc=0.132"), col=1:2, pch=16:17)

ZZ <- standardizeFRRF(f_ori,
                      Fblanc  = 0,
                      aLHII_0 = 0.007194)
ZZb <- standardizeFRRF(f_ori,
                      Fblanc  = 0)

Ztrue <- standardizeFRRF(f_ori,
                         Fblanc  = f_ori$Fblanc)

data.frame(corrected=Ztrue$JVPII, corrFb0=ZZ$JVPII, uncorrected = ZZ$JVPII_uc)  # difference
```

```
##     corrected   corrFb0 uncorrected
## 1  0.00000000 0.0000000          NA
## 2  0.02525763 0.1929556    0.133128
## 3  0.04285212 0.3287681    0.324288
## 4  0.06503265 0.4786503    0.480960
## 5  0.09334959 0.6021345    0.601560
## 6  0.11642463 0.6826313    0.682920
## 7  0.12271933 0.9573348    0.958320
## 8  0.15959469 0.7682798    0.769320
## 9  0.20751703 0.8854494    0.884880
## 10 0.29398889 0.8907857    0.894240
```

```r
(aLHII_0_true <- attributes(Ztrue)$aLHII_0)  # compare with 0.007194  !!!
```

```
## [1] 0.0003691267
```

```r
(aLHII_0_LS   <- attributes(ZZ)$aLHII_0)  # compare with 0.007194  !!!
```

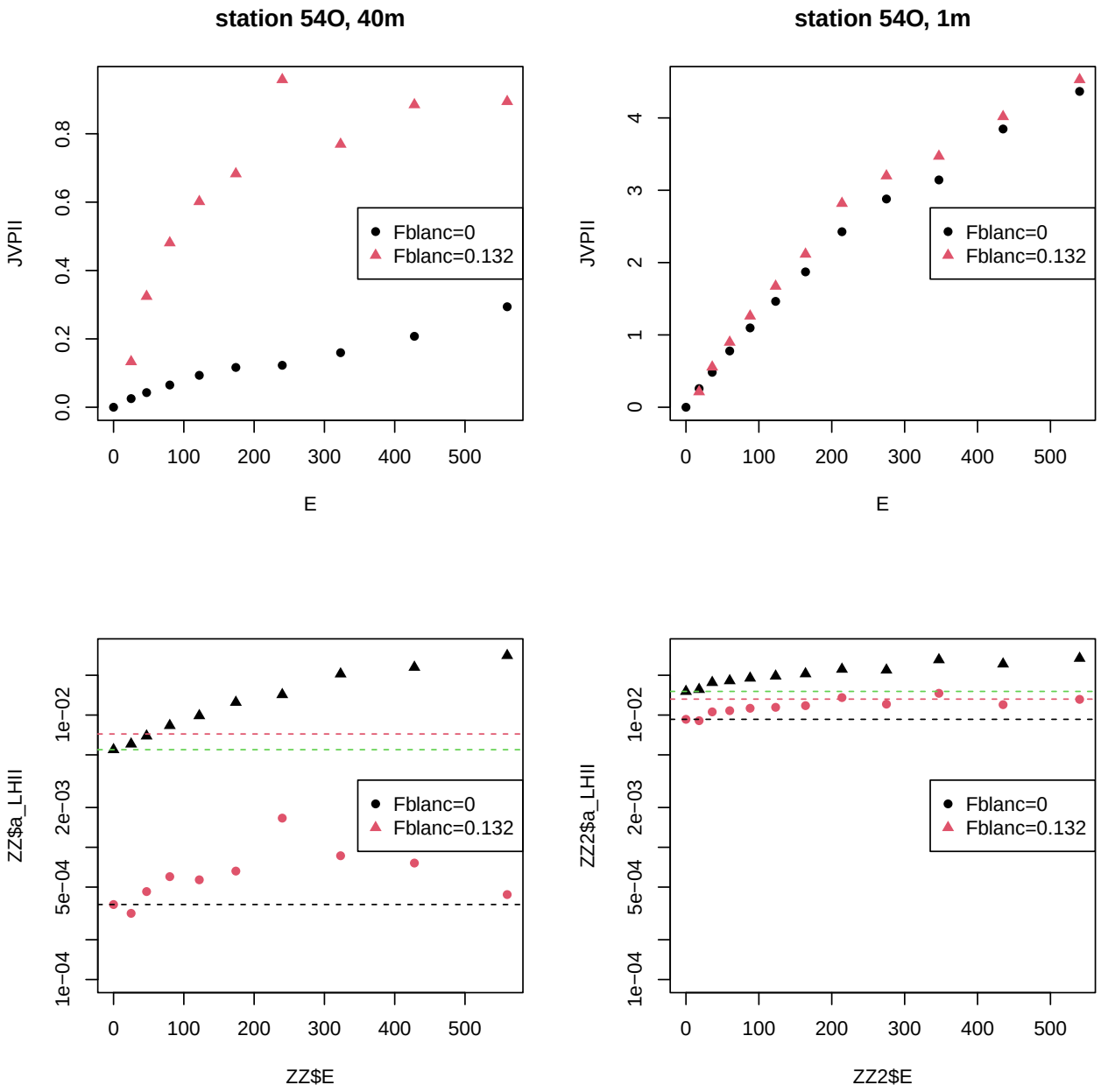
```
## [1] 0.007194
```

```r
(aLHII_0_b0   <- attributes(ZZb)$aLHII_0)  # compare with 0.007194  !!!
```

```
## [1] 0.005471202
```

```r
plot(ZZ$E, ZZ$a_LHII, ylim=c(0.0001, 0.03), log="y", pch=17, col=1)
points(Ztrue$E, Ztrue$a_LHII, pch=16, col=2)
abline(h=c(aLHII_0_true,aLHII_0_LS,aLHII_0_b0), col=1:3, lty=2)
legend("right", legend=c("Fblanc=0", "Fblanc=0.132"), col=1:2, pch=16:17)


ZZ2 <- standardizeFRRF(f_ori2,
                      Fblanc  = 0,
                      aLHII_0 = 0.0132)
ZZb2 <- standardizeFRRF(f_ori2,
                      Fblanc  = 0)

Ztrue2 <- standardizeFRRF(f_ori2,
                         Fblanc  = f_ori2$Fblanc)

aLHII_0_true2 <- attributes(Ztrue2)$aLHII_0  # compare with 0.0132  !!!
aLHII_0_LS2   <- attributes(ZZ2)$aLHII_0  # compare with 0.0132  !!!
```

```
aLHII_0_b02   <- attributes(ZZb2)$aLHII_0  # compare with 0.0132  !!!

plot(ZZ2$E, ZZ2$a_LHII, ylim=c(0.0001, 0.03), log="y", pch=17, col=1)
points(Ztrue2$E, Ztrue2$a_LHII, pch=16, col=2)
abline(h=c(aLHII_0_true2,aLHII_0_LS2,aLHII_0_b02), col=1:3, lty=2)
legend("right", legend=c("Fblanc=0", "Fblanc=0.132"), col=1:2, pch=16:17)
```



The used value for aLHII_0 for this station at 40 m is 0.007194; using the true blanc was 0.132 instead of 0 as input to the machine.

If we "standardize" the data with these values, we obtain values that are very similar to the original values, except for the first few values.