GPPFourier - An R package implementing the Fourier method to estimate Gross Primary Production from high frequency O2 data.

Tom J.S. Cox

August 30, 2017

1 Introduction

This package implements the Fourier method to estimate Gross Primary Production (GPP) from high frequency O2 data. This method estimates time averaged GPP as

$$\overline{GPP(t)} \approx \qquad 2\omega_1 \frac{\sin\theta - \theta\cos\theta}{\theta - \frac{1}{2}\sin 2\theta} A_{\mathrm{O}_2} \tag{1}$$

where A_{O_2} represents the Fourier amplitude at the diurnal frequency of the oxygen time series; $\theta = \pi \mathrm{f}_{\mathrm{DL}}$, with f_{DL} the relative fraction of daylight hours over a 24h period. ω_1 denotes the diurnal frequency. For more details on the assumptions and performance of the method we refer to (Cox et al., 2015).

When GPP varies slowly with time, so will A_{O_2}. In that case we can use a mathematical procudure called called *complex demodulation* (Bloomfield, 2000) to extract the slowly varying amplitude of a signal. Applied to the above relation between A_{O_2} and GPP we have (Cox et al., 2017)

$$\begin{aligned} GPP'(t) \approx &\qquad 4\pi|y'(t)|\frac{\sin\theta - \theta\cos\theta}{\theta - \frac{1}{2}\sin 2\theta} \qquad (2) \\ \text{with} &\qquad y(t) = O_2(t)\exp 2\pi i t \end{aligned}$$

This relation allows to estimate slowly varying GPP based on high frequency O_2 time series. For more details we refer to Cox et al. (2017).

The two core functions implementing the Fourier method are `GPPFourier()` implementing (1) and `GPPFourier_t()` implementing (2). Additionally, 2 higher level functions are provided to apply (1) to a long data series in consecutive blocks of pre-defined length (`WindowGPPFourier()`) and to time series of O_2 with gaps (`WindowGPPFourier.gts()`). In all cases O_2 time series are first preconditioned with `GPPFourierPreprocess()`. This function allows for detrending and low pass filtering (simple moving average) the time series, and it confines the data set to an integer number of days (it is called from within `GPPFourier()` and `GPPFourier_t()`).

A crucial parameter in Fourier methods to estimate GPP is the relative daylength (ratio of light hours over 24h). The function `fDLfun()`) implements an algorithm from the Astronomical Almanac to calculate daylength based on a Date, for a given Longitude and Lattitude. `fDLfun()`

is automatically invoked by `GPPFourier()` and `GPPFourier_t` when time is in the first column of the data frame x. Alternatively, the user can manually provide fDL.

It is crucial that $O2$ time series are provided with a constant sampling interval. The `GPPFourier`-package contains functions for resampling and gapfilling time series with non-constant sampling intervals (`gapfill()` and `resample()`).

Four example time series are provided

- `Kruibeke`: A time series observed in 2010 in the Scheldt estuary (Kruibeke pontoon, N51.176 E4.326), used in Cox et al. (2015)
- `Hoernum`: A time series observed in 2008 at Hoernum Tief in the German Wadden Sea (Hoernum Tief pole, N54.783 E8.45), used in Cox et al. (2017)
- `watercolumn`: A depth averaged O2 and GPP time series, simulated with a 1D vertical model of O2 dynamics (see Cox et al. (2017))
- `estuary`: O2 and GPP time series at a fixed estuarine location simulated with 1D longitudinal estuary model of O2 dynamics (see Cox et al. (2017))

2 Calculate average GPP from a single oxygen time series

Figure 1 shows a 10 day data set of oxygen observed at a fixed location in the Scheldt estuary (Kruibeke pontoon, N51.176 E4.326). The major visible oscilations are semi-diurnal due to the tides, that transport an oxygen gradient past the sensor. Nevertheless, there is a diel component in this time series, which is readily visible in a plot of the Fourier transform of the series (plotted with `plotF()`, Figure 1)

```
> par(mfrow=c(1,2))
> DO <- Kruibeke[Kruibeke$time>="2010-06-03"&Kruibeke$time<="2010-06-13",]
> plot(DO,ylab="O2",type="l")
> plotF(detrend(DO$O2),
+                 dt=as.numeric(difftime(DO$time[2], DO$time[1], units="days")),
+                 xlim=c(0,3),
+                 type="b")
```

The function `GPFourier()` estimates average GPP over the period of this time series from the amplitude of the peak at the diel frequency.

```
> dt <- difftime(DO$time[2], DO$time[1], units="days")
> GPP <- GPPFourier(DO$O2,
+         dt=dt,
+         Nfilt=1/as.numeric(dt, unit="days"),
+         fDL=fDLfun(DO$time[1], phi=51.176, lambda=4.326)
+ )
> GPP # volume specific primary production, i.e. mmol O2/m3/d
```

```
[1] 43.66706
```

This results in the volume specific gross primary production, with units given by the unit of the O2 concentration in the time series (μM) and the time unit of `dt` (days). Thus the GPP

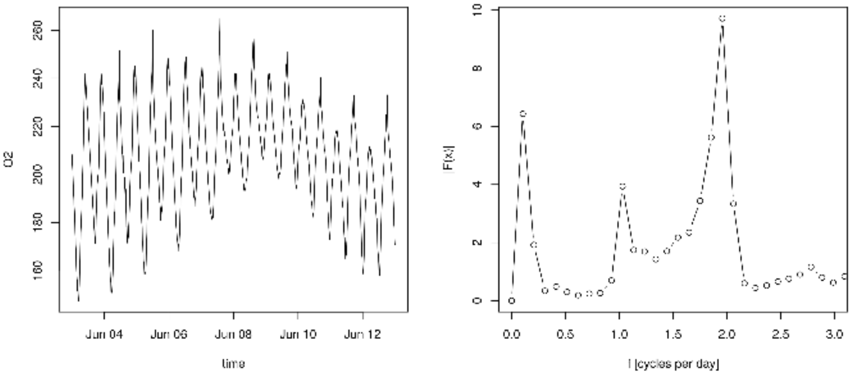

Figure 1: Example O2 time series (left) and Fourier transform (right)

estimate from this O_2 time series is 43.7 mmol $O_2 m^{-3} d^{-1}$. The Fourier method assumes that the O_2 time series represent the depth averaged O_2 concentration, either because the system is considered vertically well mixed, or because the depth averaged concentrations is determined is explicitely observed, e.g. with multiple sensors. Thus, the surface specific productivity is obtained by simply multiplying by the depth of the system. The example data set is recorded at Kruibeke pontoon in the Scheldt estuary where the water depth is approximately 9.3m. Thus the surface specific productivity is estimated as 406.1 mmol $O_2 m^{-2} d^{-1}$.

3 Calculate average GPP in consecutive windows, on $O2$ time series with gaps

Often, O_2 sensor data consists of multiple consecutive series, each of them sampled at regular time intervals, interrupted by periods where there is no data available (due to sensor failure, maintenance,...). As an example, figure 2 presents the O_2 data recorded in 2010 at a fixed location in the Scheldt estuary (Kruibeke pontoon, N51.176 E4.326).

The function `WindowGPPFourier.gts()` analyses such interrupted O_2 series with a single function call. It will split the time series at gaps that are larger that a specified number of data points (`gapMaxN`). On each individual series, GPP is caculated in consecutive time windows of prescribed length (`Width`). Missing values or gaps smaller than `gapMaxN` are automatically filled by linear interpolation. The result is shown in the bottom panel of figure 2.

```
> par(mfrow=c(2,1))
> plot(Kruibeke,
+       pch=20,
+       xlim=as.POSIXct(c("2010-01-01", "2010-12-31")))
> GPPAll_4 <- WindowGPPFourier.gts(Kruibeke,
+                                  gapMaxN = 10,
+                                  Width = 10,
+                                  filtWidth=1*24,
+                                    phi=51.176,
+                                    lambda=4.326,
+                                    Detrend=TRUE,
+                                    filter=TRUE,
+                                    filtcorrect=FALSE)
> plot(GPPAll_4$time,
+       GPPAll_4$GPP*9.3/1.3,
+       col="black",
+       pch=19,
+       type="b",
+       xlab="time",
+       ylab="GPP",
+       xlim=as.POSIXct(c("2010-01-01", "2010-12-31")))
>
```

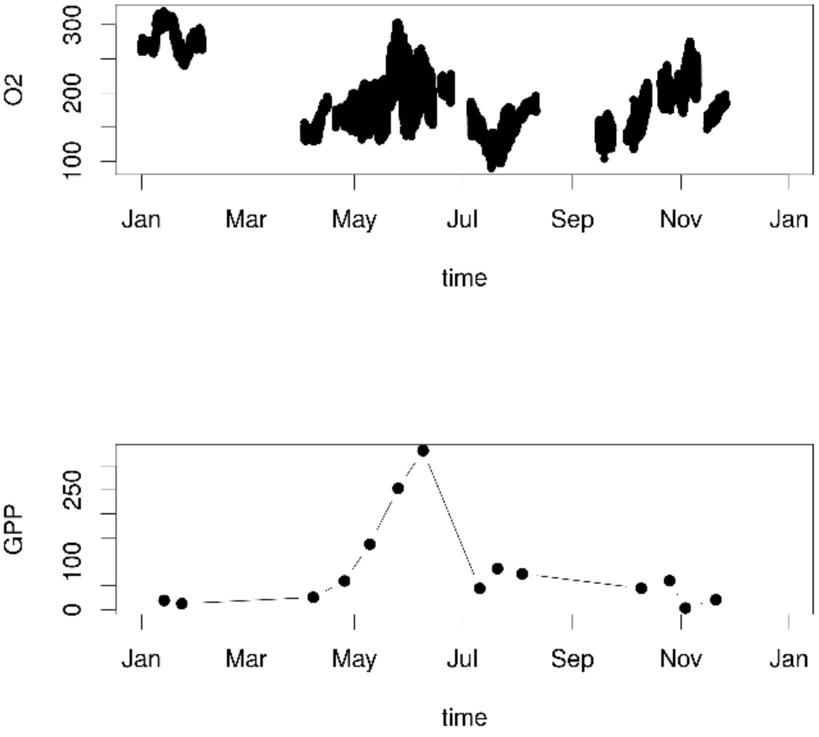

Figure 2: Calculate average GPP on O_2 series with gaps

4 Estimate time varying GPP by complex demodulation

4.1 Simulated water column

When long uninterrupted time series of O_2 are available, complex demodulation can be used to calculate time varying GPP with `GPPFourier_t()`. As demonstrated by the application to a simulated water column O_2 time series, the time varying estimate closely tracks the true GPP (Figure 3).

```
> # Calculate GPP(t) by complex demodulation of simulated water column time series
>
> dt <- as.numeric(difftime(watercolumn$time[2], watercolumn$time[1] , units="days"))
> Nfilt <- 1/dt
> GPPt <- GPPFourier_t(watercolumn[,c("time", "O2")],
+                       dt=dt, units="days",
+                      Detrend=TRUE,
+                      filter=TRUE,
+                      Nfilt=Nfilt,
+                      NLowPass=Nfilt,
+                      fDL=NULL,
+                      circular=FALSE,
+                      sides=2,
+                      nf=2)
> par(mfrow=c(2,1), cex=1.2)
> plot(watercolumn[,c("time", "O2")],
+      type="l",
+      xlab="",
+      ylab=expression(paste(O[2], " [", mu, "M]")))
> title(main="Water column")
> plot(GPPt, type="l", lwd=3, ylim=c(0,30), xlab="", ylab="GPP")
> lines(watercolumn[,c("time","GPP")], col="red")
> legend("topleft",
+        lty=1,
+        col=c("red", "black"),
+        legend=c( "Simulated GPP",
+                  expression(paste("Complex demodulated ", O[2], " series"))),
+        bty="n")
```

4.2 Simulated estuary

Tidal systems are more complicated and there can sometimes be large mismatches (Figure 4). The presence of tidal harmonics with close to diel frequency (such as O1, K1, Q1) result in spurious diel O_2 fluctuations that can not be attributed to GPP. The magnitude of those fluctuations depend on the magnitude of those tidal harmonics in the water movement, as well as on the magnitude of horizontal O_2 gradients.

```
> # Calculate GPP(t) by complex demodulation of simulated estuary time series
>
> dt <- as.numeric(difftime(estuary$time[2], estuary$time[1] , units="days"))
> Nfilt <- 1/dt
> GPPt <- GPPFourier_t(estuary[,c("time", "O2")],
+                     dt=dt,
+                     Detrend=FALSE,
+                     filter=TRUE,
+                     Nfilt=Nfilt,
+                     NLowPass=Nfilt,
+                     fDL=NULL,
+                     circular=FALSE,
+                     sides=2,
+                     nf=2)
> par(mfrow=c(2,1), cex=1.2)
> plot(estuary[,c("time", "O2")],
+      type="l",
+      xlab="",
+      ylab=expression(paste(O[2], " [", mu, "M]")))
> title(main="Estuary")
> plot(GPPt,
+      type="l",
+      lwd=3,
+      xlab="",
+      ylab="GPP",
+      ylim=c(0,30))
> lines(estuary$time,
+       filter(filter(estuary$GPP, rep(1/Nfilt, Nfilt)),rep(1/Nfilt, Nfilt)),
+       type="l",
+       lwd=3,
+       ylim=c(0,30),
+       xlab="",
+       ylab="GPP",
+       col="red")
> legend("topleft",
+        lty=1,
+        col=c("black", "red"),
+        legend=c(expression(paste("Demodulated ", O[2], " series")), "GPP"),
+        bty="n")
>
```

4.3 Real world O_2 time series

However, these spurious fluctuations mostly average out to zero over a time period of 15 days (see Cox et al. (2017)). This can be achieved by setting the `NLowPass` parameter in `GPPFourier_t()` to $15/dt$ where dt is the sampling interval (Figure 5).

```
> # Calculate GPP(t) by complex demodulation of Hoernum Tief O2 time series
> dt <- as.numeric(diff(Hoernum$time)[1], units="days")
> Nf <- 1/dt
> phi <- 54.783
> lambda <- 8.45
> # Diurnal harmonic
> T1 <- 1
> f1 <- 1/T1
> # Tidal harmonics
> TO1 <- 25.81933871/24        # O1 period
> fO1 <- 1/TO1
> TQ1 <- 26.868350/24   # Large lunar elliptic diurnal
> fQ1 <- 1/TQ1
> GPPt <- GPPFourier_t(gapfill(Hoernum),
+                      Nf=Nf,
+                      nf=2,
+                      NLowPass=Nf,
+                      phi=phi,
+                      lambda=lambda)
> GPPt15 <- GPPFourier_t(gapfill(Hoernum),
+                        Nf=Nf,
+                        nf=2,
+                        NLowPass=15/dt,
+                        phi=phi,
+                        lambda=lambda)
> par(mfrow=c(2,1))
> plot(GPPt, type="l", ylab="GPP(t)")
> title(main="Hoernum Tief")
> lines(GPPt15, lwd=3)
> legend("topright", legend=c("1 Day filter", "15 Day filter"),lwd=c(1,3), bty="n")
> plotF((GPPt$GPPt-GPPt15$GPPt)[!is.na(GPPt15$GPPt)], dt=dt, xlim=c(0,0.5), type="b", pch=20)
> abline(v=c(f1-fO1, f1-fQ1))
> title(main="Spectrum of difference")
>
```

References

Bloomfield, P. (2000). *Fourier Analysis of Time Series. Second Edition.* New York: J. Wiley & Sons.

Cox, T., Soetaert, K., Maris, T., Kromkamp, J., & Meire, P.and Meysman, F. (2015). Estimating primary production from oxygen time series: a novel approach in the frequency domain. *Limnology And Oceanography:Methods*, 13, 529–552.

Cox, T. J. S., van Beusekom, J. E. E., & Soetaert, K. (2017). Tune in on 11.57 μ hz and listen to primary production. *Biogeosciences Discussions*, 2017, 1–15.

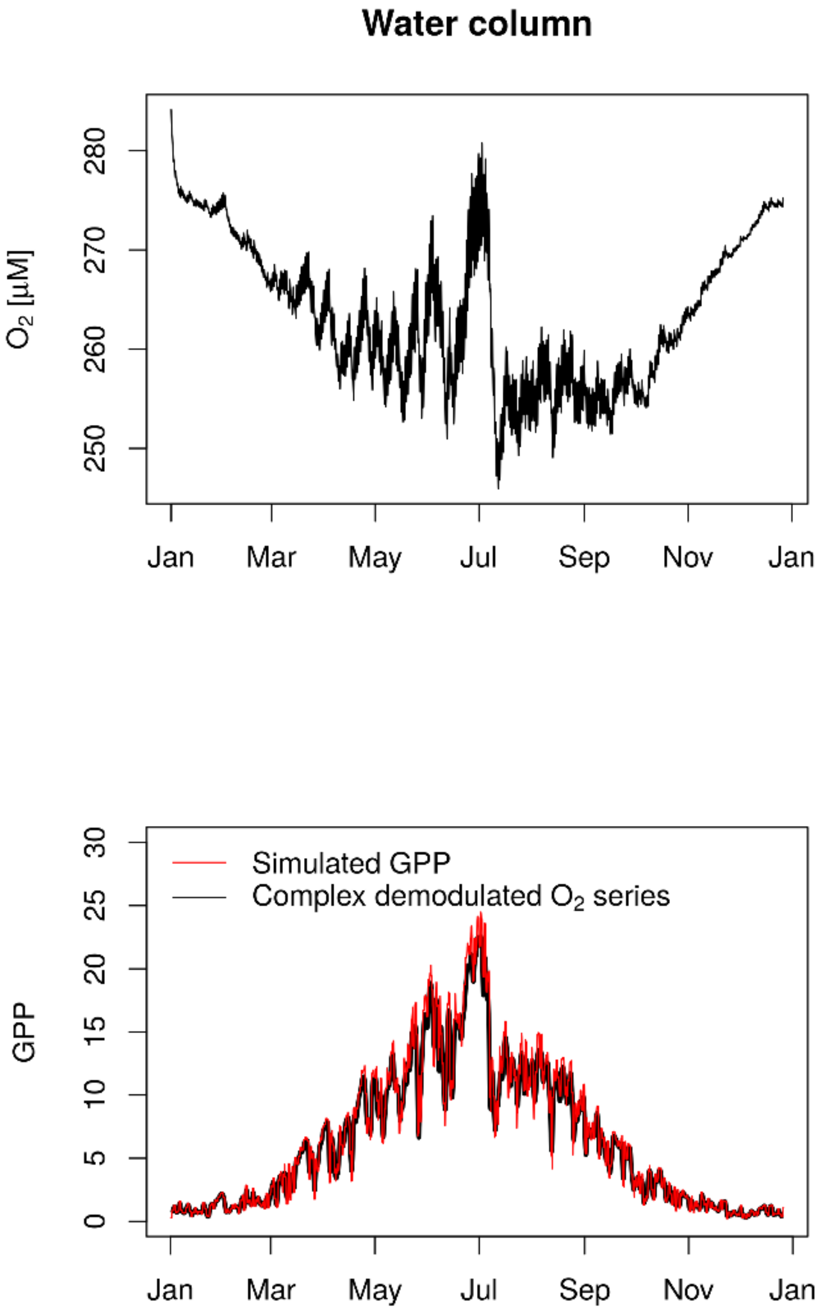

Figure 3: Time varying GPP estimated by complex demodulation of simulated water column O_2 time series

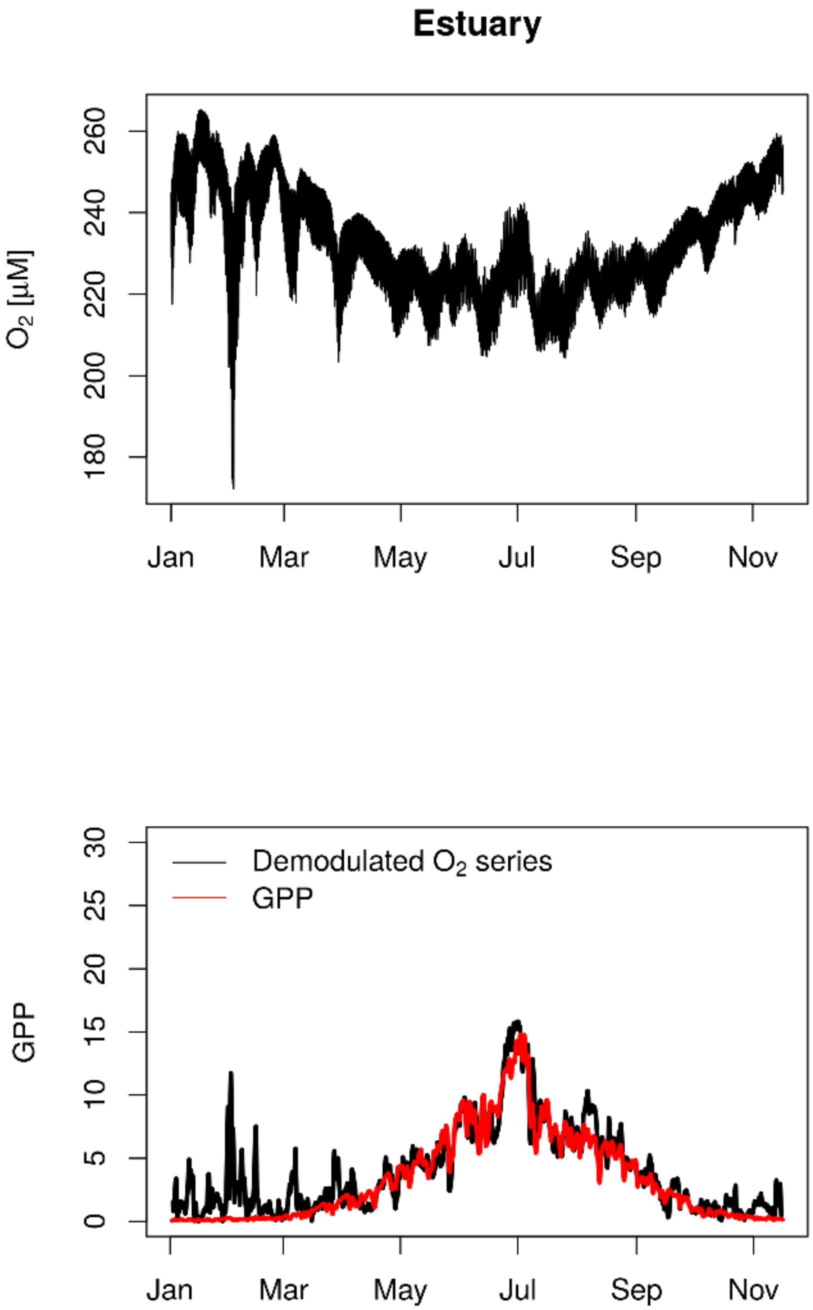

Figure 4: Time varying GPP estimated by complex demodulation of simulated estuarine O_2 time series

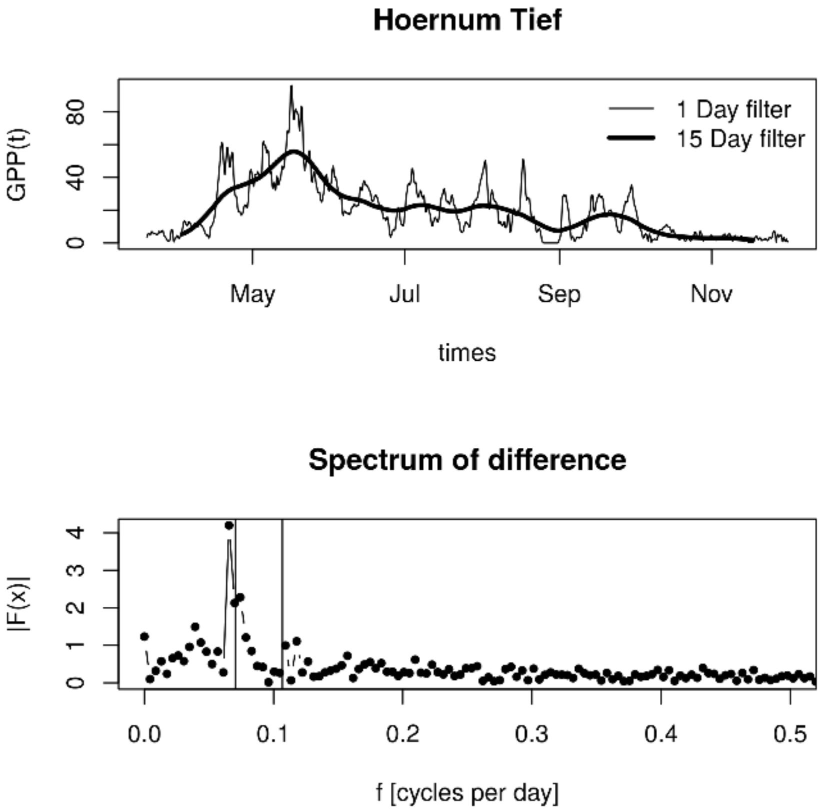

Figure 5: Complex demodulated time series of Hoernum Tief in 2008