Detrending the signal

A rough estimate of the sedimentation rate from biostratigraphy is 2 - 6 cm/kyr, assuming there are no major hiatusses in the record. Even when the highest sedimentation rate would turn out to be true, variability with a wavelength of >50 m would be on million-year timescales. We detrend the signal to get rid of the low-frequency variability that more than likely occurs on the million-year scale, and thus not on astronomical timescales.

```
library(astrochron)
```

```
## Welcome to astrochron v0.9 (2019-01-08)
```

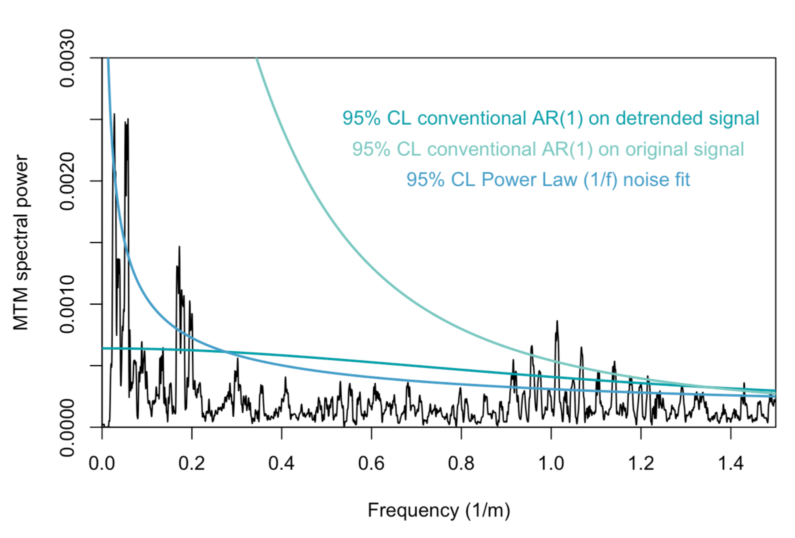
```
signal_used=read.csv(paste0(DIRECTORY,"Case_3_Signal.csv"))
signal_detrended=lowpass(signal_used, fcut = 1/50, win = 0, addmean = F, genplot = F, verbose = F)
signal_detrended[,2]=signal_used[,2]-signal_detrended[,2]
```

Multitaper method (MTM) spectral analysis

A first MTM power spectrum to get an idea of potential cyclic components in the signal. We first do an MTM spectral analysis of the detrended signal, while estimating conventional AR(1) confidence levels. However, correctly estimating noise in a spectrum is statistically challenging: This is illustrated by a few sensitivity experiments. Finally, we decide that we want to filter the ~18 meter cycles (at 0.055 cycles/m), scrutinizing the working-hypothesis that this spectral peak reflects the imprint of 405-kyr long eccentricity. But exactly how wide are we going to make our filter? Different possible widths are indicated by red, black and blue boxes.

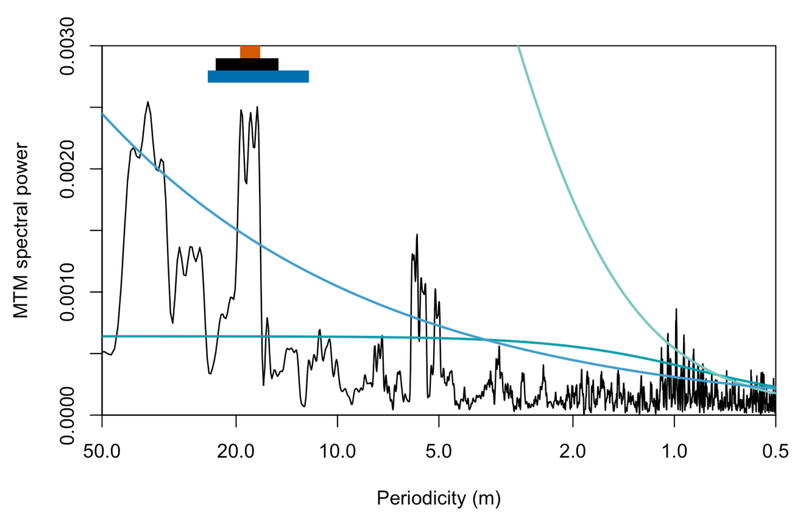
```
mtm_HadSM_signal_2=mtm(signal_detrended,demean = T, ar1 = T, tbw = 2,output = 1, genplot = F, verbose = F)
mtm_HadSM_signal_o=mtm(signal_used,demean = T, ar1 = T, tbw = 2,output = 1, genplot = F, verbose = F)
mtm_HadSM_signal_PL=mtmPL(signal_used,demean = T, tbw = 2,output = 1, genplot = F, verbose = F)

plot(mtm_HadSM_signal_2[,1],mtm_HadSM_signal_2[,2],lwd = 1.2, type="l", xlim = c(0,1.5),ylim = c(0,0.003), xlab =
'Frequency (1/m)', ylab = 'MTM spectral power', xaxs="i", yaxs = "i")
lines(mtm_HadSM_signal_2[,1],mtm_HadSM_signal_2$AR1_95_power,type = "l", col = rgb(8/255,164/255,172/255), lwd =
2)
lines(mtm_HadSM_signal_o[,1],mtm_HadSM_signal_o$AR1_95_power,type = "l", col = rgb(123/255,204/255,196/255), lwd
= 2)
lines(mtm_HadSM_signal_PL[,1],mtm_HadSM_signal_PL$PowerLaw_95_power,type = "l", col = rgb(67/255,162/255,202/255)
, lwd = 2)
text(x = 1, y = 0.0025, labels = "95% CL conventional AR(1) on detrended signal", col = rgb(8/255,164/255,172/255
), lwd = 2)
text(x = 1, y = 0.00225, labels = "95% CL conventional AR(1) on original signal ", col = rgb(123/255,204/255,196/
255), lwd = 2)
text(x = 1, y = 0.002, labels = "95% CL Power Law (1/f) noise fit ", col = rgb(67/255,162/255,202/255), lwd = 2)
```




```
plot(1/mtm_HadSM_signal_2[,1],mtm_HadSM_signal_2[,2],lwd = 1.2, type="l", xlim = c(50,0.5),ylim = c(0,0.003), xla
b = 'Periodicity (m)', ylab = 'MTM spectral power', xaxs="i", yaxs = "i", log = "x")
lines(1/mtm_HadSM_signal_2[,1],mtm_HadSM_signal_2$AR1_95_power,type = "l", col = rgb(8/255,164/255,172/255), lwd
= 2)
lines(1/mtm_HadSM_signal_o[,1],mtm_HadSM_signal_o$AR1_95_power,type = "l", col = rgb(123/255,204/255,196/255), lw
d = 2)
lines(1/mtm_HadSM_signal_PL[,1],mtm_HadSM_signal_PL$PowerLaw_95_power,type = "l", col = rgb(67/255,162/255,202/25
5), lwd = 2)

rect(xleft=19.5,ybottom=0.0029,xright=17,ytop=0.0030,col=rgb(213/255,94/255,0),border=NA)
rect(xleft=23,ybottom=0.0028,xright=15,ytop=0.0029,col="black",border=NA)
rect(xleft=24.3,ybottom=0.0027,xright=12.15,ytop=0.0028,col=rgb(0,111/255,178/255),border=NA)
```

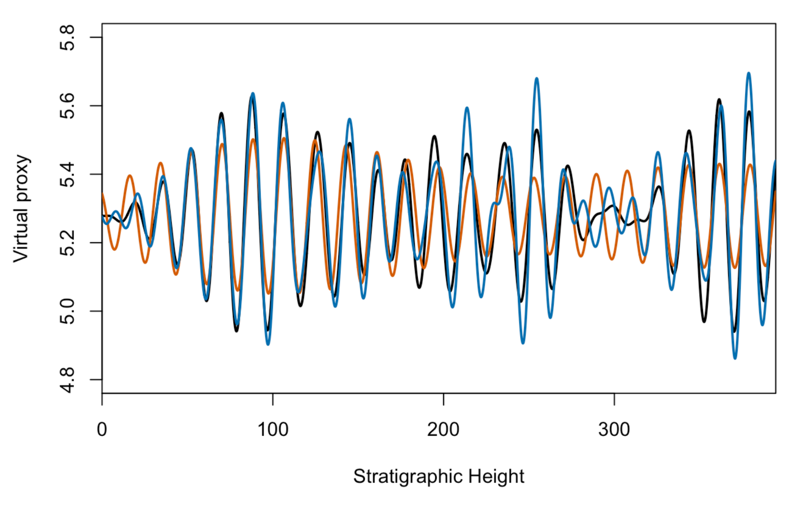


Different bandpass filter settings

In the next few lines, we explore how different or similar the different ~18-meter filters are. It becomes clear that around 300 m, the filters disagree.

```
black_filter = bandpass(signal_detrended, flow = 1/23, fhigh = 1/15,xmax = 0.3, genplot = F, verbose = F)
red_filter = bandpass(signal_detrended, flow = 1/19.5, fhigh = 1/17, xmax = 0.3, genplot = F, verbose = F)
blue_filter = bandpass(signal_detrended, flow = 1/24.3, fhigh = 1/12.15, xmax = 0.3, genplot = F, verbose = F)

plot(black_filter, type = "l", lwd = 2, col = "black", xlab = "Stratigraphic Height", ylab = "Virtual proxy", xax
s = "i", ylim = c(4.8,5.8))
lines(red_filter, type = "l", lwd = 2, col=rgb(213/255,94/255,0))
lines(blue_filter, type = "l", lwd = 2, col=rgb(0,111/255,178/255))
```



The precession envelope

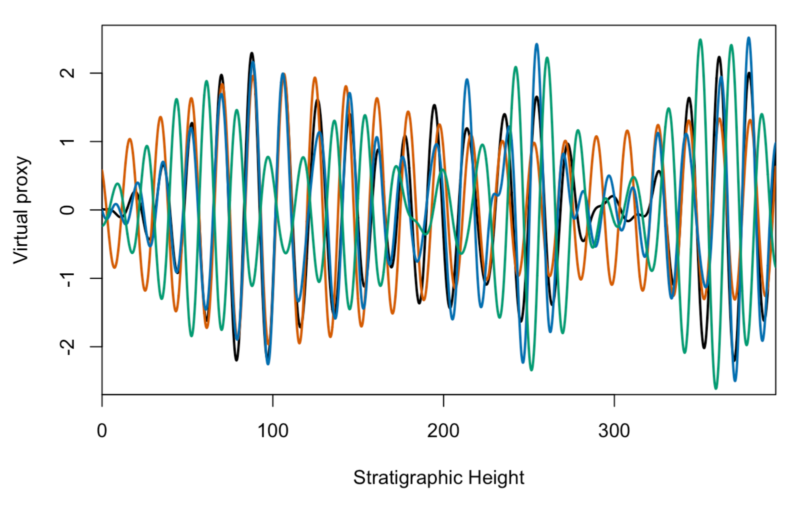
The precession envelope might be useful in determining which bandpass filter is most thrustworthy. First, we filter the precession-scale variablity (according to our working hypothesis). Then, we do a hilbert transform to obtain the precession envelope. Finally, we filter that envelope to isolate the 405-kyr eccentricity imprint on the precession envelope.

Plotting up the precession envelope with the ~18-m filters reveals that they are anti-correlated. In other words, maxima in the direct filters correspond to precession envelope minima, and thus to eccentricity minima. We thus figured our the phase-relationship between our virtual proxy and eccentricity: They are anti-phased!

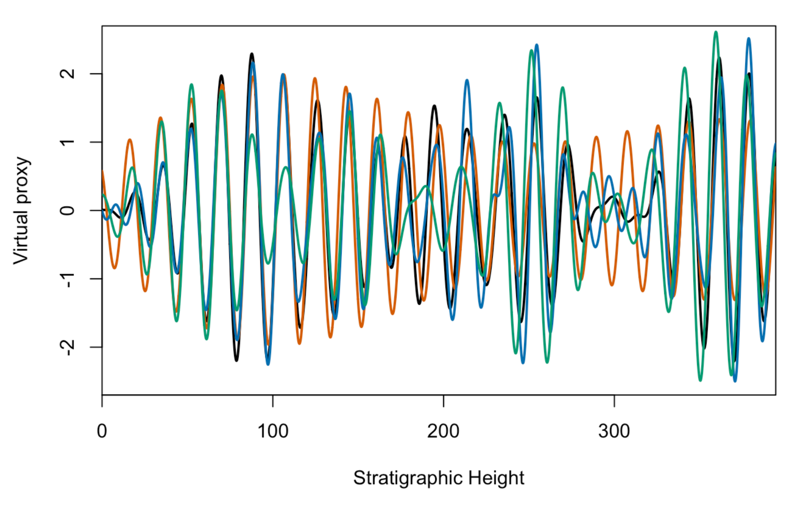
Once we figured that out, we'll make the same plot, but with the y-axis of the precession envelope signal flipped (green)

```
precession_filter=bandpass(signal_detrended, flow = 0.9, fhigh = 1.25, xmax = 2, genplot = F, verbose = F)
precession_envelope=hilbert(precession_filter, genplot = F, verbose = F)
precession_envelope_405=bandpass(precession_envelope, flow = 1/23, fhigh = 1/15,xmax = 0.5, genplot = F, verbose
= F)

plot(black_filter[,1], scale(black_filter[,2]), type = "l", lwd = 2, col = "black", xlab = "Stratigraphic Height"
, ylab = "Virtual proxy", xaxs = "i", ylim = c(-2.5,2.5))
lines(red_filter[,1],scale(red_filter[,2]), type = "l", lwd = 2, col=rgb(213/255,94/255,0))
lines(blue_filter[,1],scale(blue_filter[,2]), type = "l", lwd = 2, col=rgb(0,111/255,178/255))
lines(precession_envelope_405[,1],scale(precession_envelope_405[,2]), type = "l", lwd = 2, col=rgb(0,158/255,115/
255))
```




```
plot(black_filter[,1], scale(black_filter[,2]), type = "l", lwd = 2, col = "black", xlab = "Stratigraphic Height"
, ylab = "Virtual proxy", xaxs = "i", ylim = c(-2.5,2.5))
lines(red_filter[,1],scale(red_filter[,2]), type = "l", lwd = 2, col=rgb(213/255,94/255,0))
lines(blue_filter[,1],scale(blue_filter[,2]), type = "l", lwd = 2, col=rgb(0,111/255,178/255))
par(new=T)
lines(precession_envelope_405[,1],-scale(precession_envelope_405[,2]), type = "l", lwd = 2, col=rgb(0,158/255,115
/255))
```

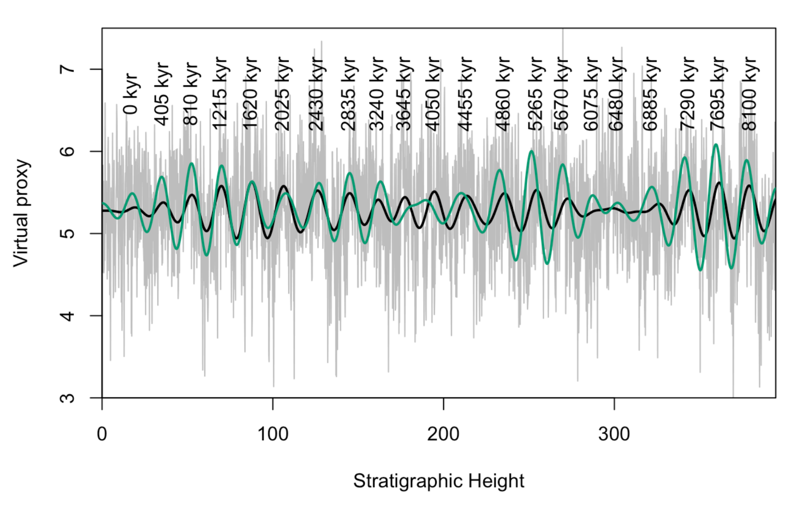


Cycle counting

Based on the previous plot, it seemed that the black filter and the precession amplitude cycles largely agreed as to where the 405-kyr cycle extremes are. We'll thus use those two indicators to construct a first age-depth model. The corresponding figure shows my personal interpretation of where 405-kyr eccentricity minima are located (corresponding to virtual proxy maxima!)

We then make a matrix that reflects that astronomical interpretation based on 405-kyr cycle counting

```
plot(signal_detrended,type="l", col = "grey",xlab = "Stratigraphic Height", ylab = "Virtual proxy", xaxs = "i", y
lim = c(3,7.5), yaxs = "i")
lines(black_filter, col = "black", lwd = 2)
lines(precession_envelope_405[,1],5.3-0.3*scale(precession_envelope_405[,2]), col=rgb(0,158/255,115/255), lwd = 2
)
text(x = 17.5, y = 6.7, labels = "0 kyr", srt = 90)
text(x = 36, y = 6.7, labels = "405 kyr", srt = 90)
text(x = 52.8, y = 6.7, labels = "810 kyr", srt = 90)
text(x = 70.05, y = 6.7, labels = "1215 kyr", srt = 90)
text(x = 87.9, y = 6.7, labels = "1620 kyr", srt = 90)
text(x = 106.5, y = 6.7, labels = "2025 kyr", srt = 90)
text(x = 126.3, y = 6.7, labels = "2430 kyr", srt = 90)
text(x = 145.2, y = 6.7, labels = "2835 kyr", srt = 90)
text(x = 161.85, y = 6.7, labels = "3240 kyr", srt = 90)
text(x = 177.45, y = 6.7, labels = "3645 kyr", srt = 90)
text(x = 194.7, y = 6.7, labels = "4050 kyr", srt = 90)
text(x = 213.75, y = 6.7, labels = "4455 kyr", srt = 90)
text(x = 235.8, y = 6.7, labels = "4860 kyr", srt = 90)
text(x = 254.7, y = 6.7, labels = "5265 kyr", srt = 90)
text(x = 269.85, y = 6.7, labels = "5670 kyr", srt = 90)
text(x = 287.4, y = 6.7, labels = "6075 kyr", srt = 90)
text(x = 301.95, y = 6.7, labels = "6480 kyr", srt = 90)
text(x = 322.05, y = 6.7, labels = "6885 kyr", srt = 90)
text(x = 343.95, y = 6.7, labels = "7290 kyr", srt = 90)
text(x = 361.5, y = 6.7, labels = "7695 kyr", srt = 90)
text(x = 380, y = 6.7, labels = "8100 kyr", srt = 90)
```

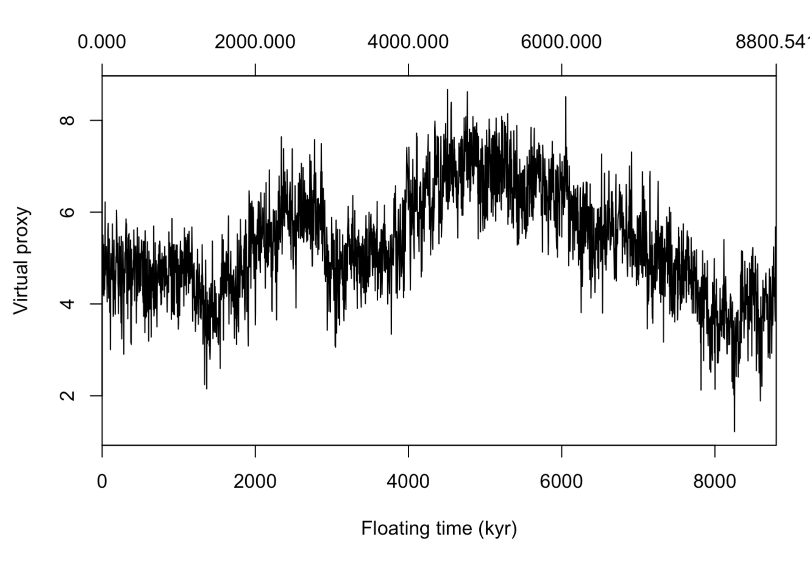



```
agemodel=matrix(nrow = 21, ncol = 2, data = c(17.5,36,52.8,70.05,87.9,106.5,126.3,145.2,161.85,177.45,194.70,213.
75,235.80,254.7,269.85, 287.4,301.95,322.05,343.95,361.5,380,0,405,810,1215,1620,2025,2430,2835,3240,3645,4050,44
55,4860,5265,5670,6075,6480,6885,7290,7695,8100))
```

From depth to time domain

We'll use our first age-depth model to go from the depth to the time domain, and we'll see whether our minimal tuning approach resulted in spectral peaks that occur at the expected astronomical frequencies

```
signal_T1=tune(signal_used,agemodel,extrapolate = T, genplot = F, verbose = F)
signal_T1[,1]=signal_T1[,1]-signal_T1[1,1]
plot(signal_T1,type="l", xlab = "Floating time (kyr)", ylab = "Virtual proxy", xaxs = "i")
axis(3, at = c(0,2000,4000,6000,signal_T1[length(signal_T1[,1]),1]))
```




```
signal_T1=linterp(signal_T1, dt = 3, verbose = F, genplot = F)
signal_T1_detrended=lowpass(signal_T1, fcut = 1/800, win = 0, addmean = F, genplot = F, verbose = F)
signal_T1_detrended[,2]=signal_T1[,2]-signal_T1_detrended[,2]

mtm_signal_T1=mtm(signal_T1_detrended,demean = T, ar1 = T, tbw = 2,output = 1, xmax = 0.07, genplot = F, verbose
= F)
plot(mtm_signal_T1[,1],mtm_signal_T1[,2],lwd = 1.2, type="l", xlim = c(0,0.07), xlab = 'Frequency (1/kyr)', ylab
= 'MTM spectral power', xaxs="i")
lines(mtm_signal_T1[,1],mtm_signal_T1$AR1_95_power,type = "l", col = rgb(8/255,164/255,172/255))
```