Wavelet Analysis with R

Why wavelet analysis in astronomy?

- How can we deal with non linear frequencies, non stationary signals...and compute all spectral characteristics of a time series taking account all this non linearities
- Wavelet analysis is a tool to analyze a signal in the time frequency domain
- Challenging since the uncertainty principle holds ...

Basics on wavelet analysis

- The 1D continuous wavelet transform (CWT) : ψ = wavelet, $f \in L^2(\mathbb{R})$,

$$W_f(\tau, s) = \frac{1}{\sqrt{s}} \int_{-\infty}^{+\infty} f(x) \overline{\psi\left(\frac{t-\tau}{s}\right)} dt \ .$$

- The parameter s is a scale parameter whereas τ s a position parameter
- The pointwise (or Morlet) reconstruction formula:
For f analytic (i.e. $\hat{f}$ vanishes on $\mathbb{R}_-$),

$$f(\tau) = \frac{1}{c_\psi} \int_0^{+\infty} W_f(\tau, s) \frac{ds}{s^{3/2}}$$

with $c_\psi = \int_{-\infty}^{+\infty} \frac{\overline{\hat{\psi}(\xi)}}{\xi} d\xi$.

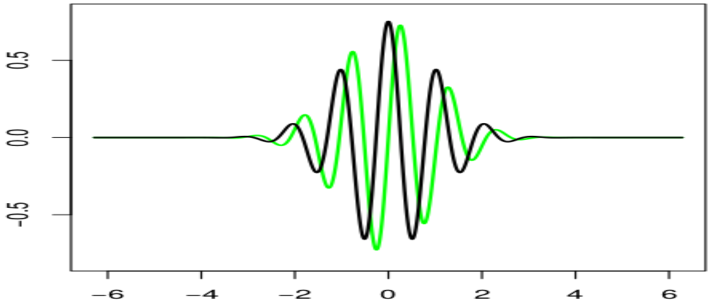
Basics on wavelet analysis

- A classical wavelet is the Morlet wavlet which the omplex valued Morlet function defined as

$$\psi(t) = \pi^{-1/4} e^{i\omega t} e^{-t^2/2} \text{ with } \omega = 6$$

- The value $\omega = 6$ is chosen so that the wavelet is quasi-analytic

Basics on wavelet analysis



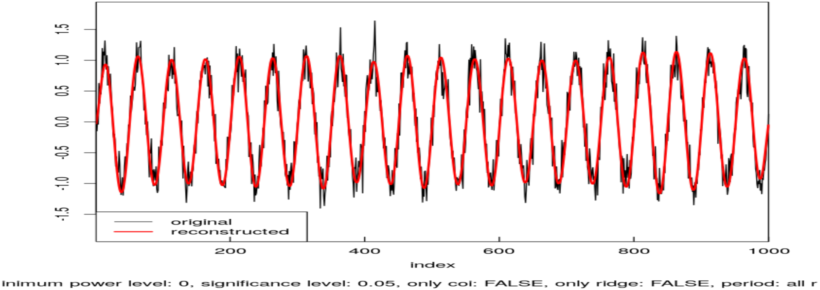
Basics on wavelet analysis

- When considering a time series (x_t) we need to discretize the itegral involved in the definition of wavelet coefficients and define
$$w(\tau, s) = \frac{1}{\sqrt{s}} \sum_t x_t \overline{\psi\left(\frac{t-\tau}{s}\right)}$$
which is a complex number if the signal is real valued
- The wavelet spectrum is then $|w(\tau, s)|^2$. Roughly speaking it is the energy at time τ and at scale s of the time series (x_t)
- One can also compute the phase of $w(\tau, s)$. It is the local wavelet phase

```
library(WaveletComp)
x = periodic.series(start.period = 50, length = 1000)
x = x + 0.2*rnorm(1000)
reconstruct(my.w, plot.waves = FALSE, lwd =
c(1,2),legend.coords = "bottomleft", ylim = c(-1.8,
1.8))
```

```
my.data <- data.frame(x = x)
my.w <- analyze.wavelet(my.data, "x",loess.span =
0,dt = 1, dj = 1/250,lowerPeriod = 16,upperPeriod =
128,make.pval = TRUE, n.sim = 10)
```

Basics on wavelet analysis



Arguments

- loess.span = 0 : no detrending
- dt = 1. Defines the time unit
- lowerPeriod = 16, upperPeriod = 128. Range of periods involved in the wavelet transform
- $dj = 1/250$. Number of suboctave in each octave
- make.pval = TRUE, n.sim = 10: The region of significant periods in x for each t is found using 10 simulations

Case of an harmonic signal $f(t) = A\cos(\omega t)$.

- Since the Morlet wavelet is quasi analytic, the CWT of f the real part of the CWT of its analytic signal $F(t) = A\, e^{i\omega t}$:

$$W_F(\tau, s) = \frac{1}{\sqrt{s}} \int_{-\infty}^{+\infty} F(t)\overline{\psi\left(\frac{t-\tau}{s}\right)} dt = s\sqrt{s}\,\overline{\hat{\psi}(s\omega)}\, e^{i\omega\tau}$$

- Then

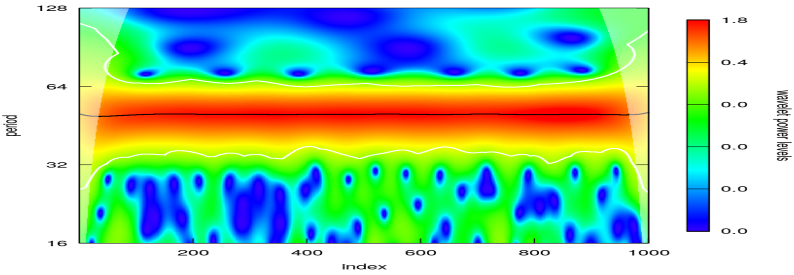
$$\partial_\tau W_F(\tau, s) = i\omega W_F(\tau, s) \rightarrow \omega = -i\frac{\partial_b W_F(\tau; s)}{W_F(a\tau; s)} ,$$

and $W_F(\tau, s_0) = \lambda_\psi \sqrt{s_0} F(\tau)$ where $s_0 = \frac{k_0}{\omega}$, (k_0 peak wavenumber of ψ).

- Take home message : in its wavelet coefficients are hidden the spectral content of an harmonic signal

Plot the wavelet power spectrum of $x_t = sin(100\pi t) + noise$

```
wt.image(my.w, color.key = "quantile", n.levels =
250,legend.params = list(lab = "wavelet power
levels", mar = 4.7))
```



A more general setting

- **Monocomponent complex signal**: $f(x) = A(x)\exp(i\varphi(x))$, with slowly varying A, φ (IMF).
- Candidate instantaneous frequency:

$$\omega_F(a,b) = -i\frac{\partial_b W_F(a,b)}{W_F(a,b)}, \quad \text{when } |W_F(a;b)| > \varepsilon$$

- Estimate :

$$|\omega_F(a,b) - \varphi'(b)| < \varepsilon \ ,$$

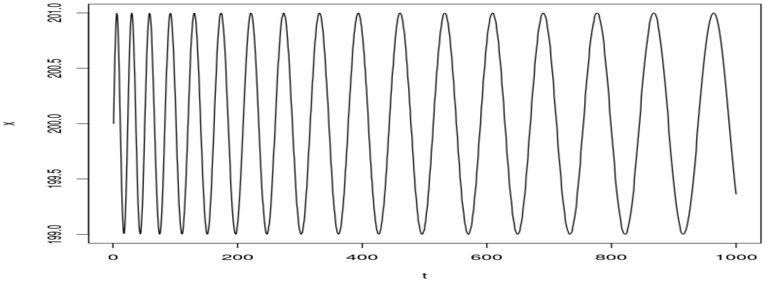
with suitable conditions (C_ε) on A, φ.

- For $a\phi'(b) \equiv k_0$

$$W_F(a,b) \equiv \lambda_\psi \sqrt{a}F(b) \ ,$$

(k_0 peak wavenumber of ψ).

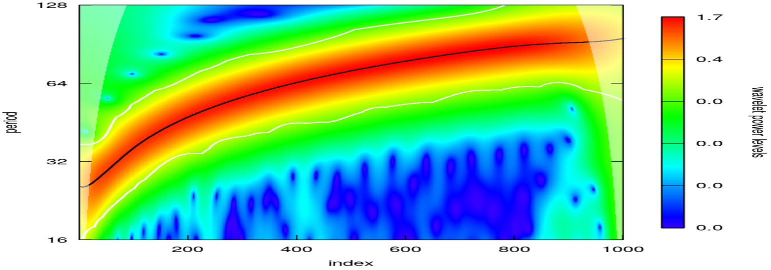
```
t=1:1000
x = periodic.series(start.period = 20, end.period =
100, length = 1000)+0.2*norm(1000)
plot(t,x,'l')
```



Basics on wavelet analysis

Wavelet analysis with R for a non-periodic time series

```
my.data <- data.frame(x = x)
my.w <- analyze.wavelet(my.data, "x", loess.span = 0,
dt = 1, dj = 1/250, lowerPeriod = 16, upperPeriod =
128, make.pval = TRUE, n.sim = 10) wt.image(my.w,
n.levels = 250, legend.params = list(lab = "wavelet
power levels"))
```

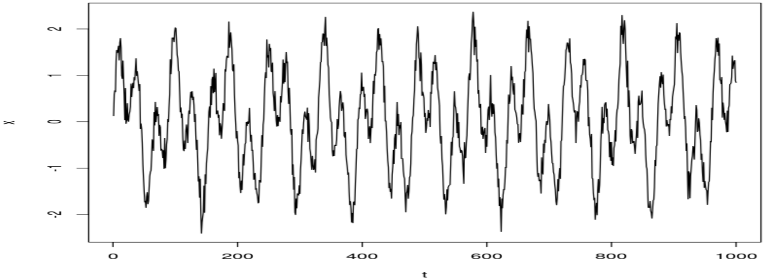


- **Multicomponent complex signal** $f(t)$: superposition of several IMFs assumed to be slowly varying and well separated in time-frequency domain:

$$f(t) = \sum_{\ell=1}^{L} A_\ell(t)\, e^{i\varphi_\ell(t)}$$

- The wavelet transform is linear. The spectral content of the signal is hidden in the wavelet transform of each component

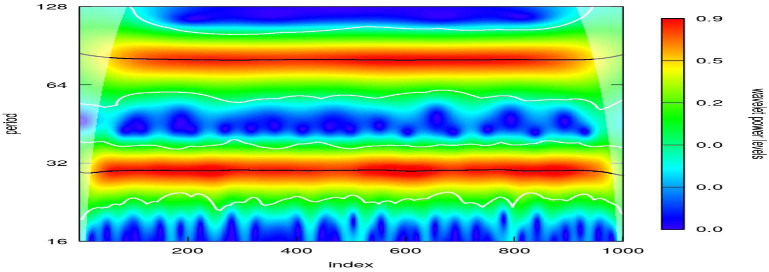
```
t=1:1000
x1 <- periodic.series(start.period = 80, length =
1000)
x2 <- periodic.series(start.period = 30, length =
1000)
x <- x1 + x2 + 0.2*rnorm(1000)
plot(t,x,'l')
```



```
my.data <- data.frame(x = x)
my.w <- analyze.wavelet(my.data, "x", loess.span = 0,
dt = 1, dj = 1/250, lowerPeriod = 16, upperPeriod =
128, make.pval = TRUE, n.sim = 10) wt.image(my.w,
n.levels = 250, legend.params = list(lab = "wavelet
power levels") )
```

Basics on wavelet analysis

Wavelet analysis with R for a superposition of time series



Basics on wavelet analysis

- A vertical slice through a wavelet plot is a measure of the local spectrum
- The time-averaged wavelet spectrum over a certain period is

$$\overline{w}(\tau, s) = \frac{1}{2m+1} \sum_{\ell=-m}^{m} w(\tau + \ell, s)$$

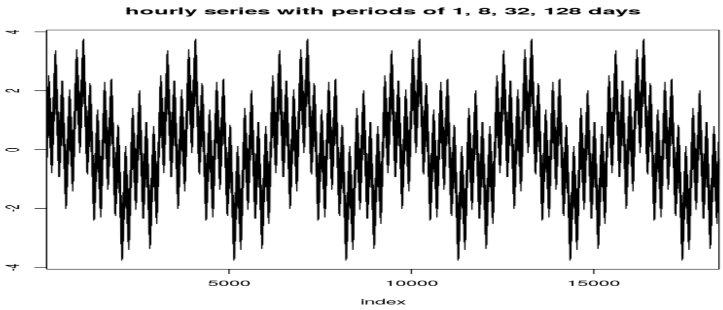
- We can average over the whole time range and obtained the averaged power spectrum
- It could be useful to compare the spectral content of two time series differently sampled

Basics on wavelet analysis

```
## The following example is adopted from Liu et al., 2007:
```

```
series.length <- 6*128*24
x1 <- periodic.series(start.period = 1*24, length =
series.length)
x2 <- periodic.series(start.period = 8*24, length =
series.length)
x3 <- periodic.series(start.period = 32*24, length =
series.length)
x4 <- periodic.series(start.period = 128*24, length =
series.length)
x <- x1 + x2 + x3 + x4
plot(x, type = "l", xlab = "index", ylab = "", xaxs =
"i", main = "hourly series with periods of 1, 8, 32,
128 days")
```

Basics on wavelet analysis

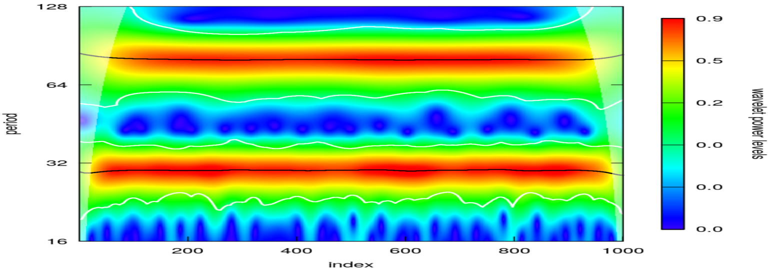


Basics on wavelet analysis

```
my.data <- data.frame(x = x)
```

```
my.wt <- analyze.wavelet(my.data, "x", loess.span =
0, dt = 1/24, dj = 1/20, lowerPeriod = 1/4, make.pval
= TRUE, n.sim = 10)
wt.image(my.wt, color.key = "i", main = "wavelet
power spectrum", legend.params = list(lab = "wavelet
power levels (equidistant levels)"), periodlab =
"period (days)", timelab = "time elapsed (days)",
spec.time.axis = list(at = index.ticks, labels =
index.labels))
```

Basics on wavelet analysis



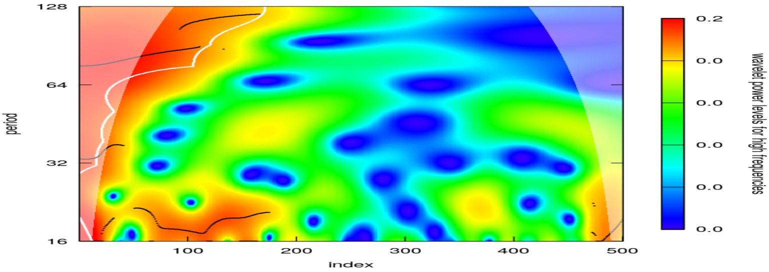
- On simple cases it seems that one can visualize different components on wavelet spectrum
- But how separate these different components?
- The EMD answer!

The procedure of extracting an IMF is called sifting. The sifting process is as follows:

- Identify all the local extrema in the test data.
- Connect all the local maxima by a cubic spline line as the upper envelope.
- Repeat the procedure for the local minima to produce the lower envelope.

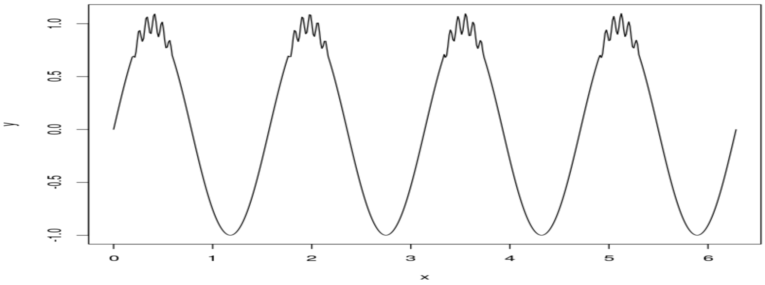
Basics on wavelet analysis

More on multicomponents signals



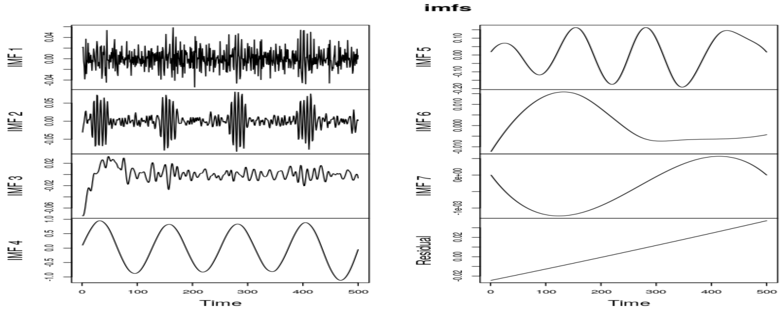
```
library(Rlibeemd)
x <- seq(0, 2*pi, length.out = 500)
signal <- sin(4*x)
intermittent <- 0.1 * sin(80 * x)
y <- signal * (1 + ifelse(signal > 0.7, intermittent,
0))
plot(x = x,y = y,type = "l")
```

Basics on wavelet analysis



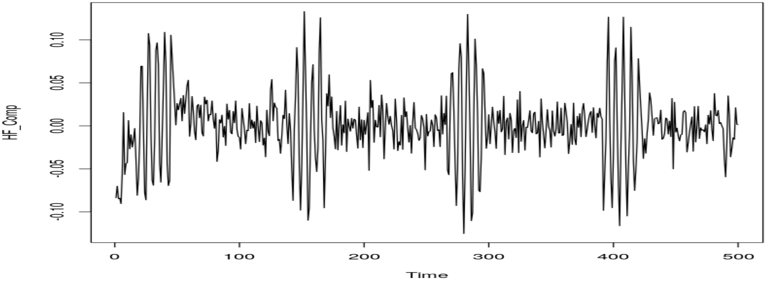
Basics on wavelet analysis

```
imfs <- eemd(y, num_siftings = 10, ensemble_size = 50,
threads = 1)
plot(imfs)
```

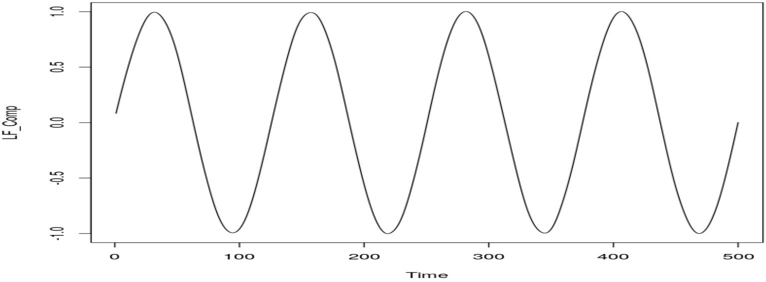


Basics on wavelet analysis

```
#Plot high frequencies component
HF_Comp<-rowSums(imfs[, 1:3])
ts.plot(HF_Comp)
```

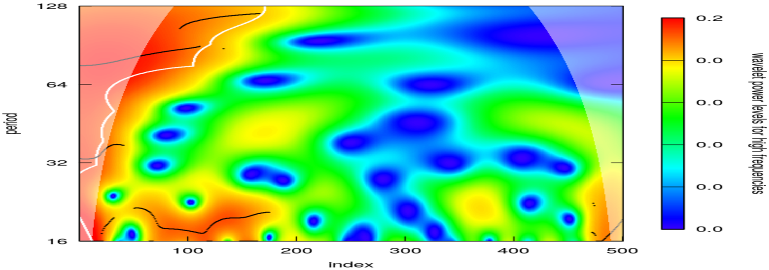


```
#Plot low frequencies component
LF_Comp<-rowSums(imfs[, 1:3])
ts.plot(LF_Comp)
```



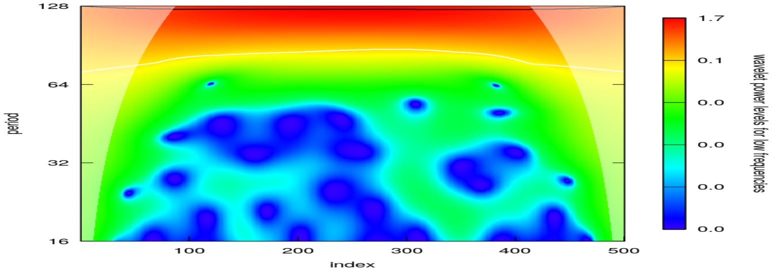
Basics on wavelet analysis

```
my.data <- data.frame(HF_Comp = HF_Comp)
my.w <- analyze.wavelet(my.data, "HF_Comp",
loess.span = 0, dt = 1, dj = 1/250, lowerPeriod = 16,
upperPeriod = 128, make.pval = TRUE, n.sim = 10)
wt.image(my.w, n.levels = 250, legend.params =
list(lab = "wavelet power levels for high
frequencies") )
```



Basics on wavelet analysis

```
my.data <- data.frame(LF_Comp = LF_Comp)
my.w <- analyze.wavelet(my.data, "LF_Comp",
loess.span = 0, dt = 1, dj = 1/250, lowerPeriod = 16,
upperPeriod = 128, make.pval = TRUE, n.sim = 10)
wt.image(my.w, n.levels = 250, legend.params =
list(lab = "wavelet power levels for low
frequencies") )
```



Bivariate wavelet analysis

- When considering bivariate time series $(x_t^{(1)}, x_t^{(2)})$, one wants to focus on cross spectrum analysis on the two components The right way to do so s to use the function `analyse coherency`

Bivariate wavelet analysis

sig1 ¡- periodic.series(start.period = 1*24, length = 24*96) sig2 ¡- periodic.series(start.period = 2*24, length = 24*96) sig3 ¡- periodic.series(start.period = 4*24, length = 24*96) sig4 ¡- periodic.series(start.period = 8*24, length = 24*96) sig5 ¡- periodic.series(start.period = 16*24, length = 24*96) x1 ¡- sig1 + sig2 + 3*sig3 + sig4 + sig5 + 0.5*rnorm(24*96) x2 ¡- sig1 + sig2 - 3*sig3 + sig4 + 3*sig5 + 0.5*rnorm(24*96)

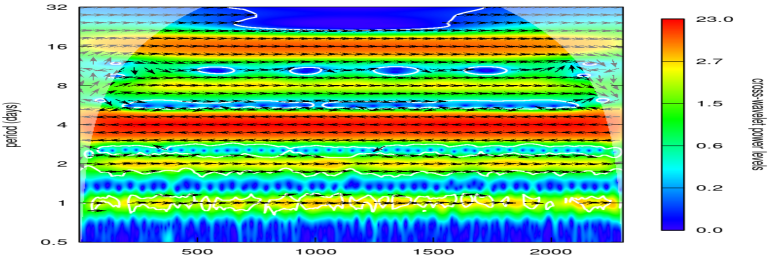
Bivariate wavelet analysis

```
my.data <- data.frame(x1 = x1, x2 = x2)
```

```
my.wc <- analyze.coherency(my.data, my.pair =
c("x1","x2"), loess.span = 0, dt = 1/24, dj = 1/100,
lowerPeriod = 1/2, make.pval = TRUE, n.sim = 10)
```

Bivariate wavelet analysis

```
wc.image(my.wc, n.levels = 250, legend.params =
list(lab = "cross-wavelet power levels"), timelab =
"", periodlab = "period (days)")
```



Bivariate wavelet analysis

```
wc.avg(my.wc, siglvl = 0.01, sigcol = "red", sigpch =
20, periodlab = "period (days)")
```