Time series analysis

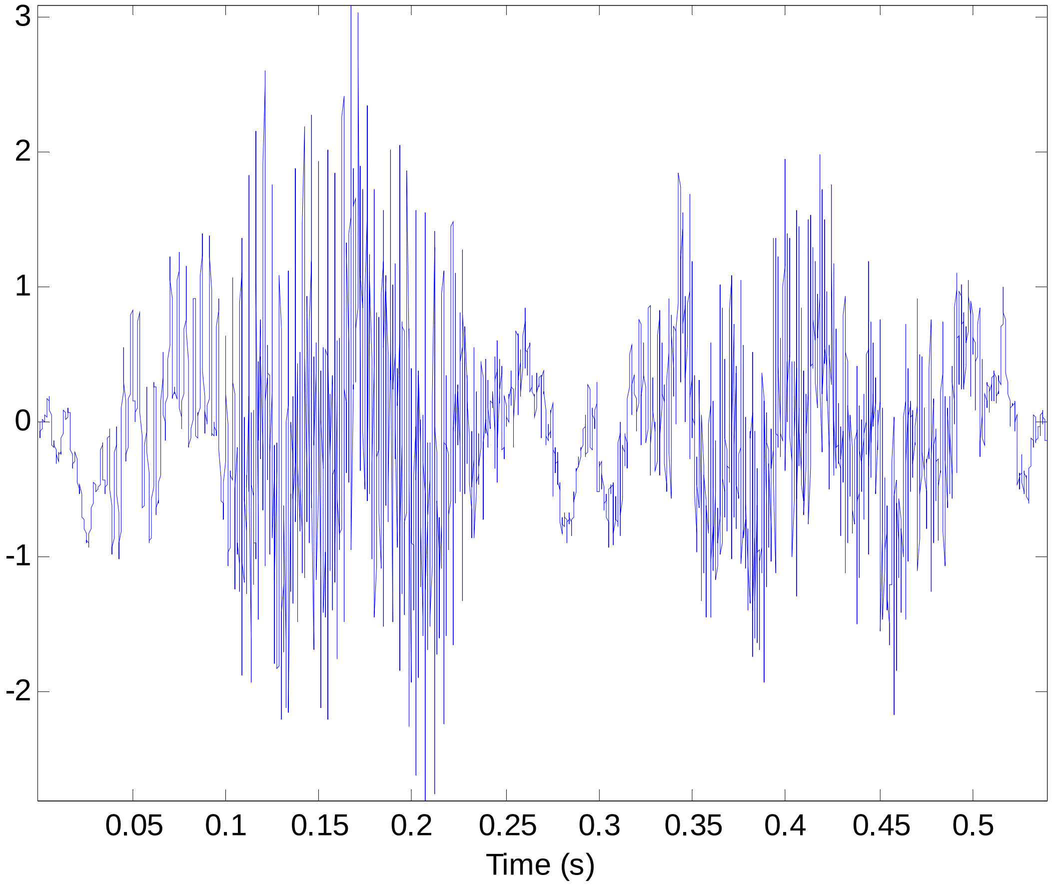
Session III

Outline

- Time-Frequency Representations (TFR)
- short-time fourier transform
- uncertainty principle
- Wavelets
- Phase analysis

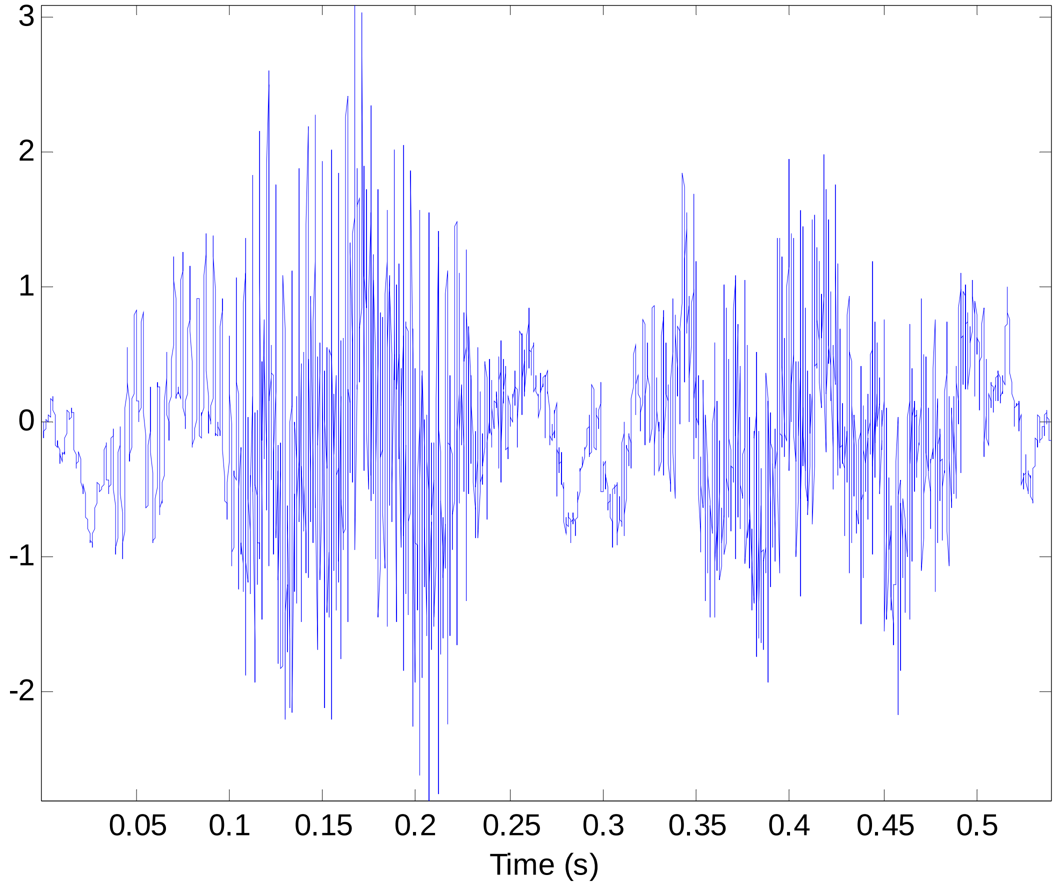
Time-frequency analysis

- Why TFR?
- all the information is in the TS or in the FT of TS
- maybe not easily accessible
- TFR increase redundancy and intelligibility!

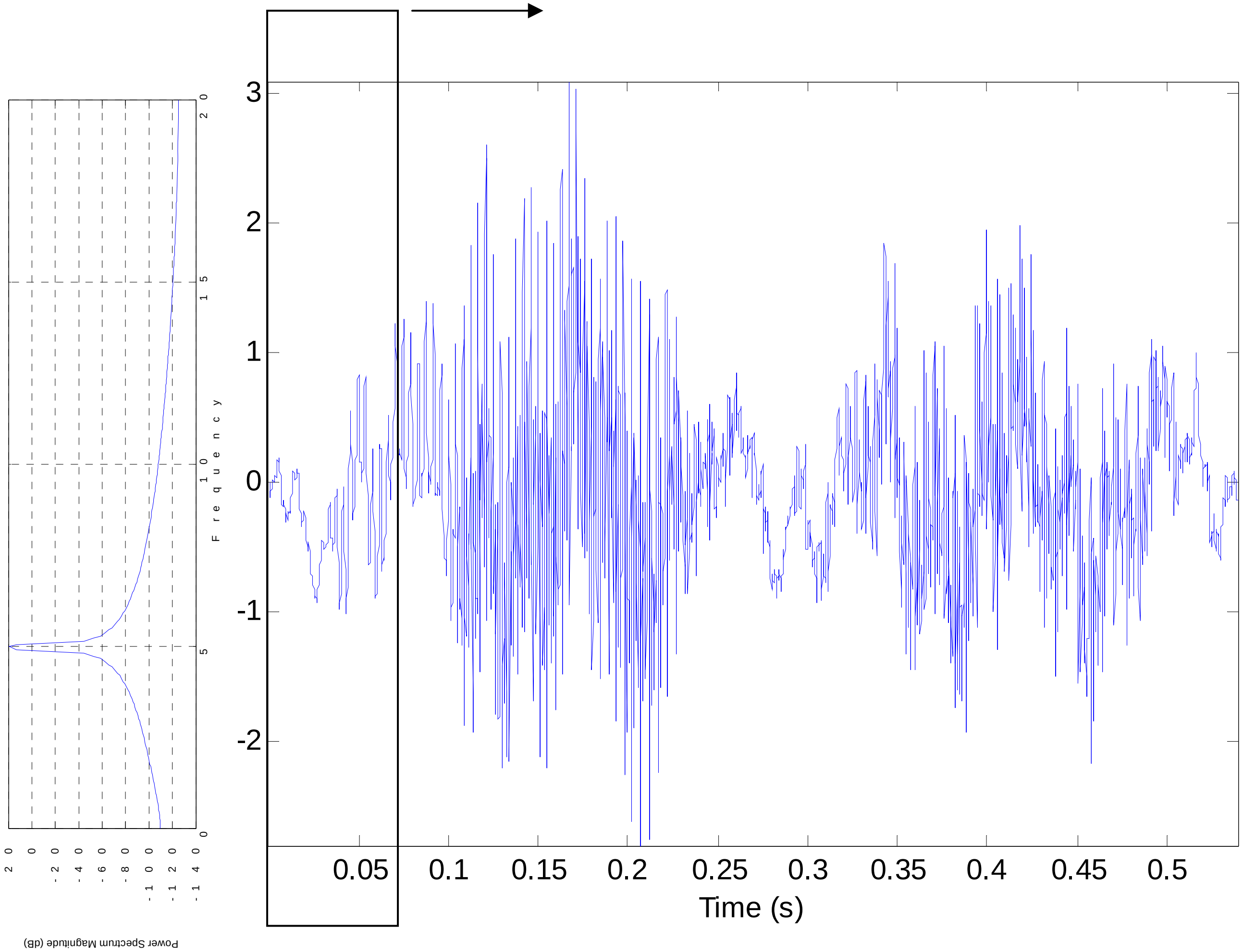


Time-frequency analysis

- Why TFR?
- FFT assumes stationarity!
- no time information



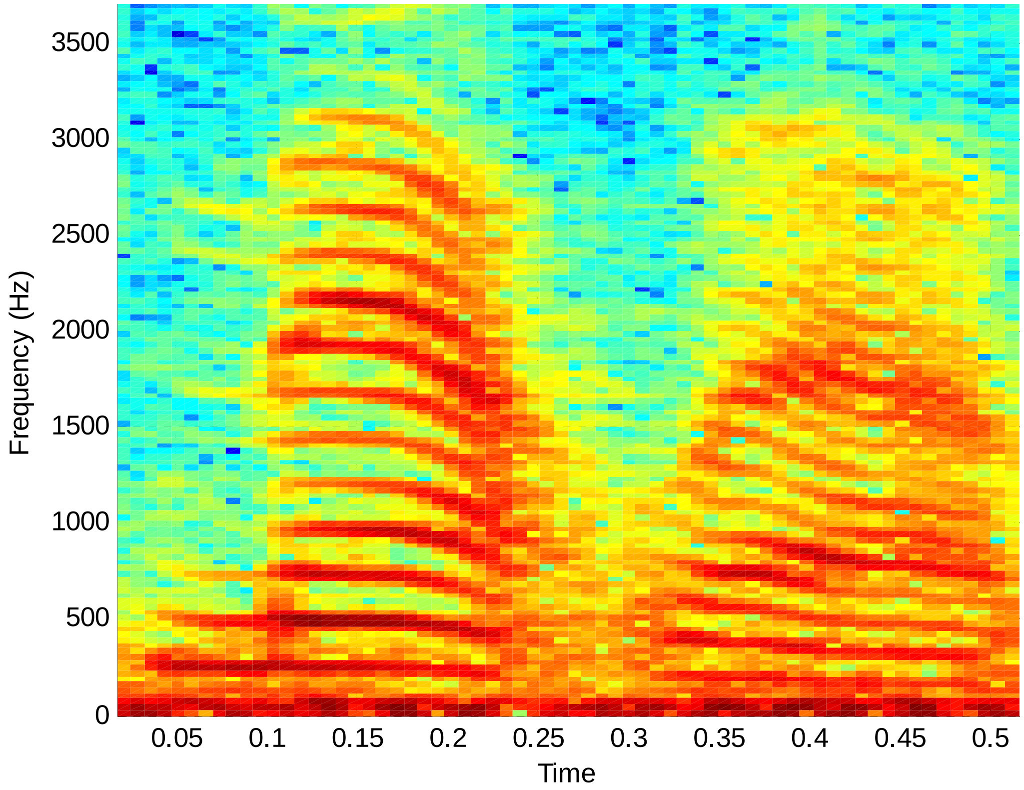
TFR with FFT



3
2
1
0
-1
-2
0.05
0.1
0.15
0.2
0.25
0.3
0.35
0.4
0.45
0.5
Time (s)
Frequency
Power Spectrum Magnitude (dB)

Short-time FT

- S = spectrogram(x,window,noverlap,nfft,fs);
- spectrogram(mtlb,hanning(256),200,256,Fs);



Exercise

- specgramdemo

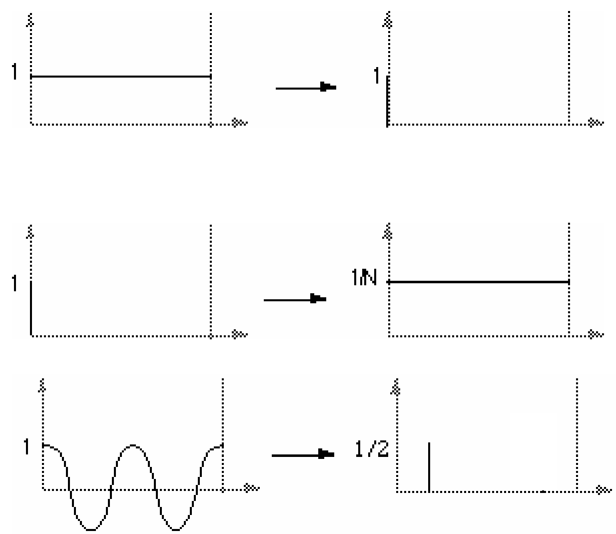
Problem with STFT

- time-frequency resolution is not optimal
- shorter window => better time resolution => decreased frequency resolution

Uncertainty principle

- trade-off: localisation in time or frequency domain

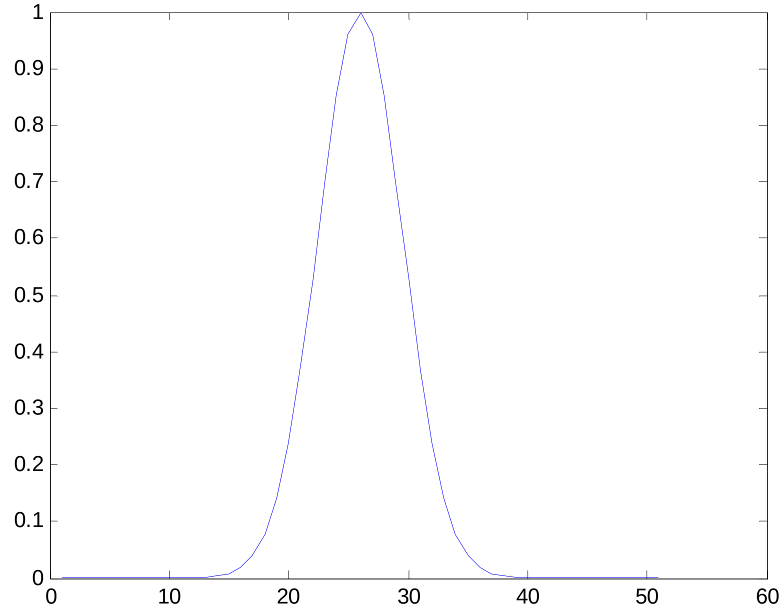
Time domain   Frequency domain



Uncertainty principle

- trade-off: localisation in time or frequency domain
- gaussian has same localisation in time and frequency domain

Time domain   Frequency domain

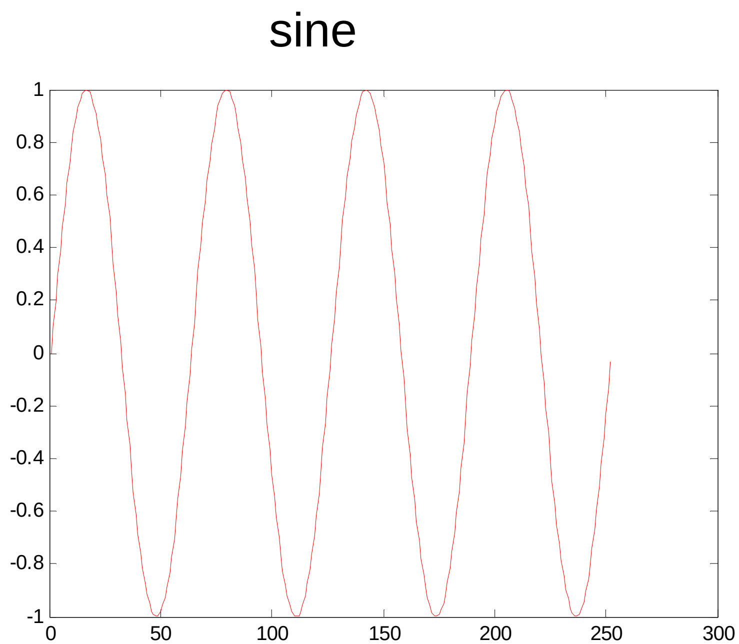


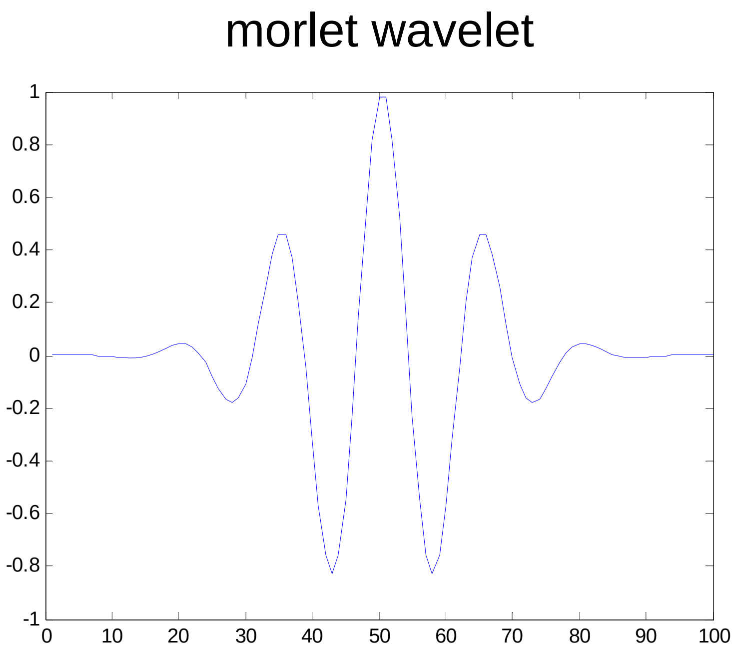
Uncertainty principle

- duration*bandwidth >= constant
- signals can not be localised with arbitrary precision in time and frequency domain

Wavelets

- introduced 1983 (Morlet, Grossmann)
- Morlet wavelet: cosine weighted with a Gaussian





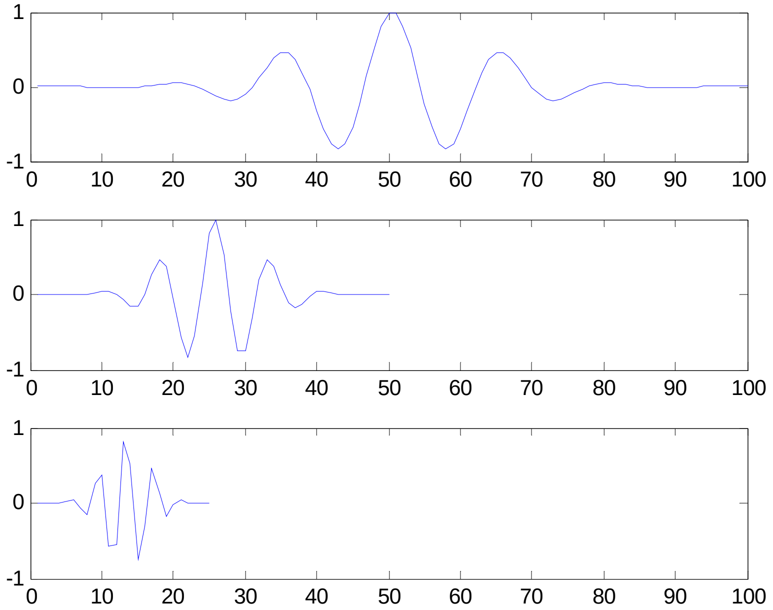
TFR with Wavelets

- correlating the signal with scaled and shifted wavelets
- what is scale? similar to frequencies in fourier analysis
- stretching or compressing a wavelet

scale=1

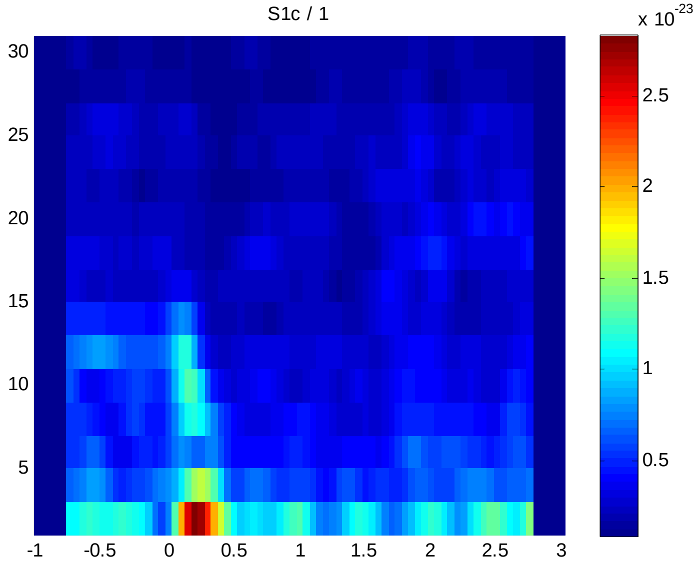
scale=0.5

scale=0.25



computing the TFR

1. correlate your wavelet with the beginning of your TS
2. shift your wavelet and correlate again
3. repeat step 2 until you reach the end of TS
4. scale your wavelet and repeat steps 1-3
5. repeat 1-4 for all desired scales (frequencies)



TFRs in Fieldtrip

easy comparison between methods

```
cfg              = [];
cfg.output       = 'pow';
cfg.method       = 'mtmconvol';
cfg.taper        = 'hanning';
cfg.foi          = 2:2:30;
cfg.t_ftimwin    = ones(length(cfg.foi),1).*0.5;
cfg.toi          = -1:0.05:3;
freq = freqanalysis(cfg, data);
```

TFRs in Fieldtrip

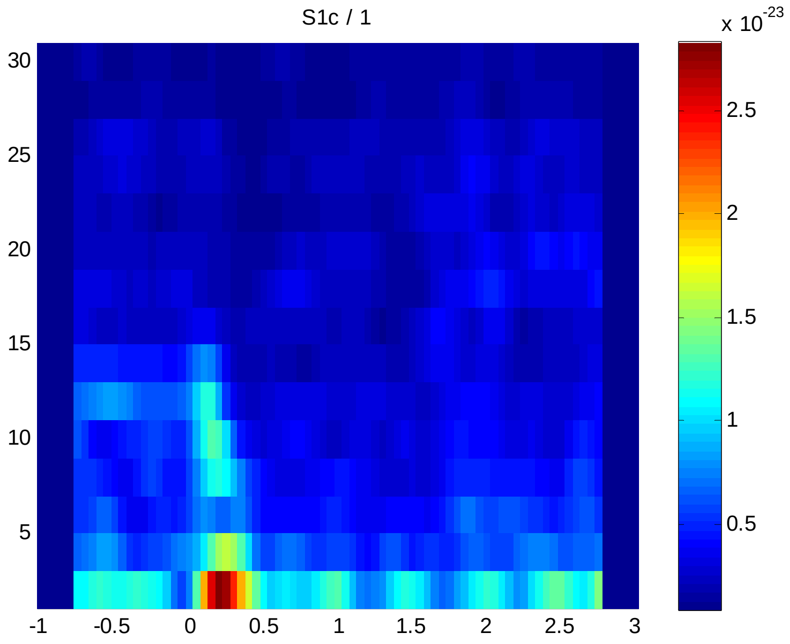
easy comparison between methods

```
cfg              = [];
cfg.output       = 'pow';
cfg.method       = 'mtmconvol';
cfg.taper        = 'hanning';
cfg.foi          = 2:2:30;
cfg.t_ftimwin    = 5/cfg.foi;  %5 cycles !
cfg.toi          = -1:0.05:3;
freq = freqanalysis(cfg, data);
```

similar to wavelet analysis (windowing is different)

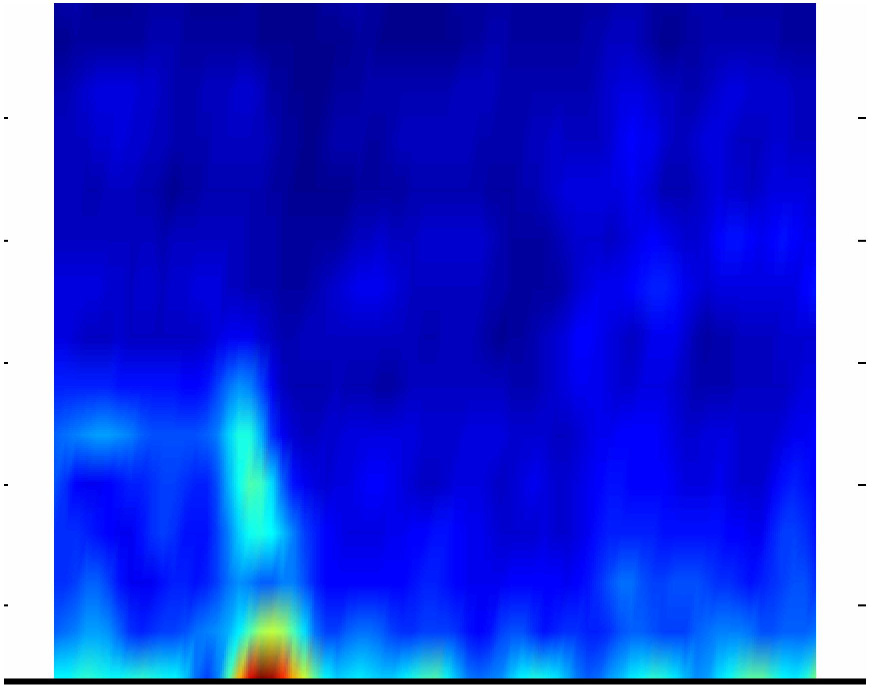
Plotting the TFR

```
cfg = [];
%cfg.baseline = [-0.5 -0.1];
%cfg.baselinetype = 'relative';
cfg.channel   = 'S1c';
singleplotTFR(cfg, freq);
```



Plotting the TFR

```
pcolor(freq.time,freq.freq,squeeze(freq.powspctrm(1,:,:)));
shading interp
```



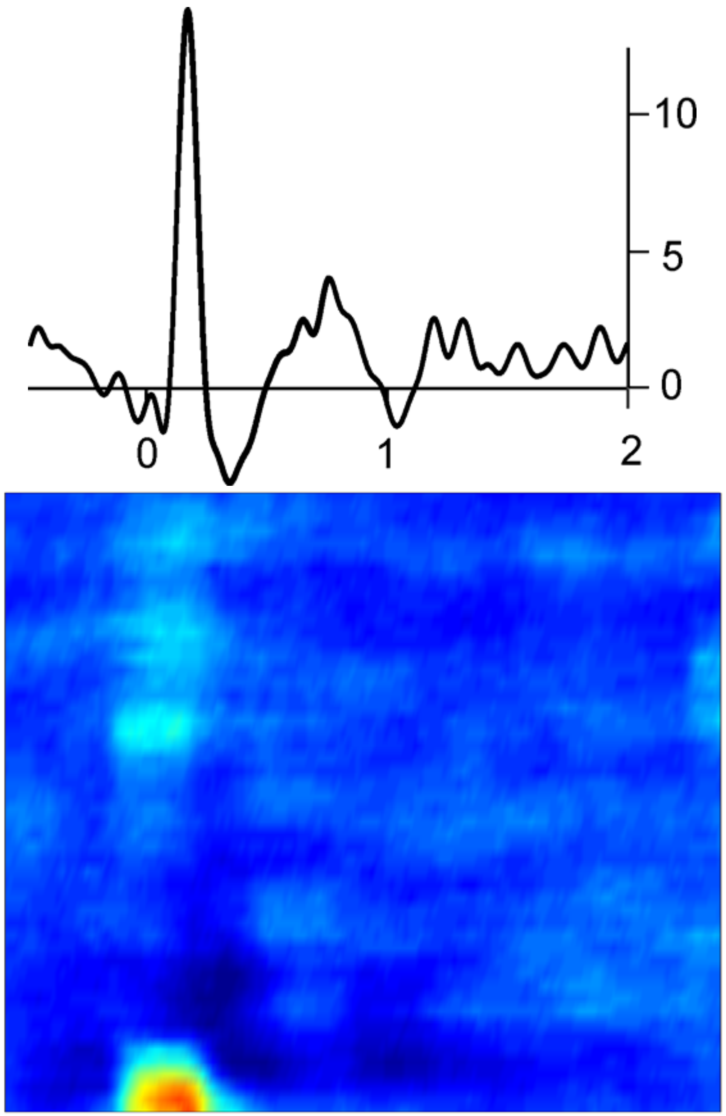
TFRs with wavelet

easy comparison between methods

```
cfg              = [];
cfg.output       = 'pow';
cfg.method       = ‘wltconvol';
cfg.width        = 5;
cfg.foi          = 2:2:30;
cfg.toi          = -1:0.05:3;
freq = freqanalysis(cfg, data);
```

Exercise

1. compute and plot TFR with STFT
   (modify window length)
2. repeat step 1 with wavelets
3. repeat step 1 with multitapers
4. use singleplot and compare absolute and relative power



Practical considerations

- Wavelets
  - length of wavelet determines time-frequency resolution (short wavelet – poor frequency resolution)
  - rule of thumb: not less than two cycles
  - wavelets provide better time-frequency resolution than STFT (they adapt with frequency, long for low frequencies, short for high frequencies)
  - product of frequency and time resolution remains constant for different frequencies (not true for STFT).

Practical considerations

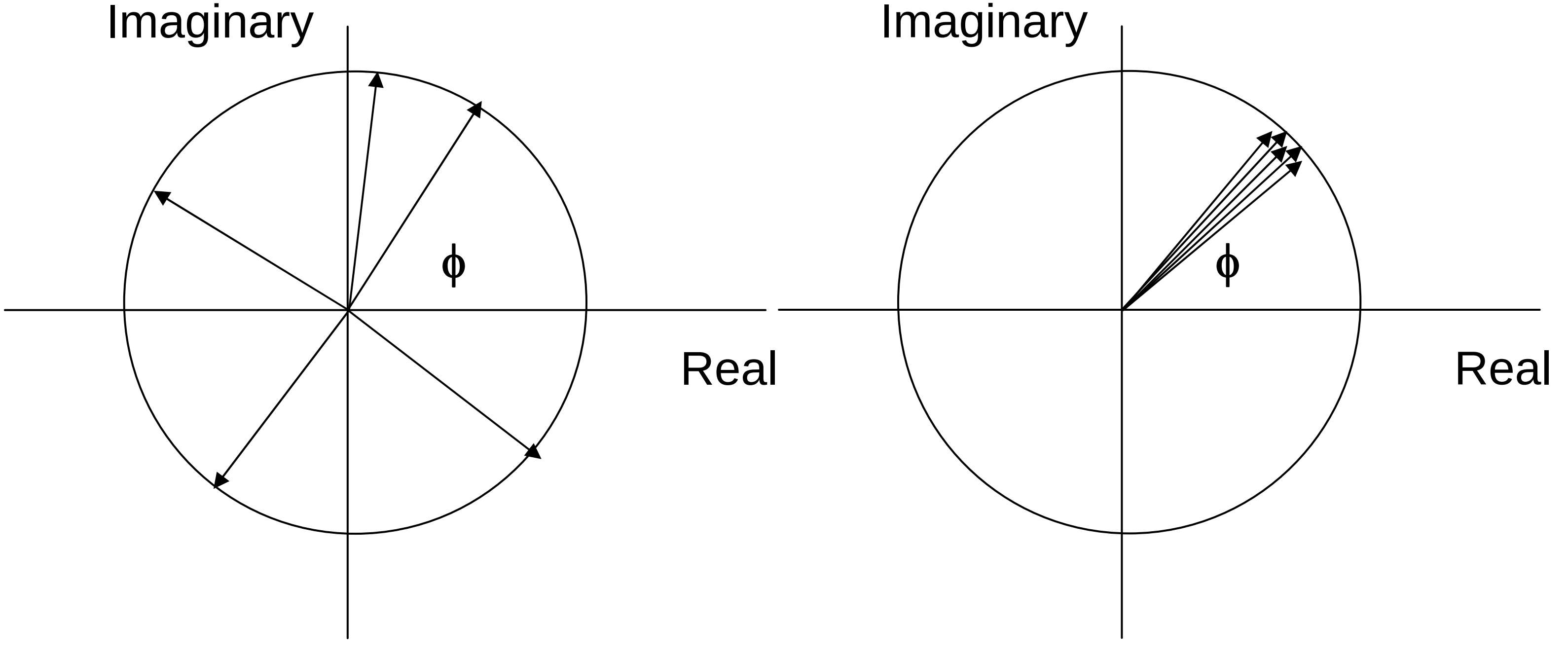
- Multi-taper
  - allows smoothing in frequency domain
  - in fieldtrip time windows can be adapted

Remarks

- there is no single correct TFR
- each method emphasises different aspects of the signal
- ideally: use several techniques and vary parameters

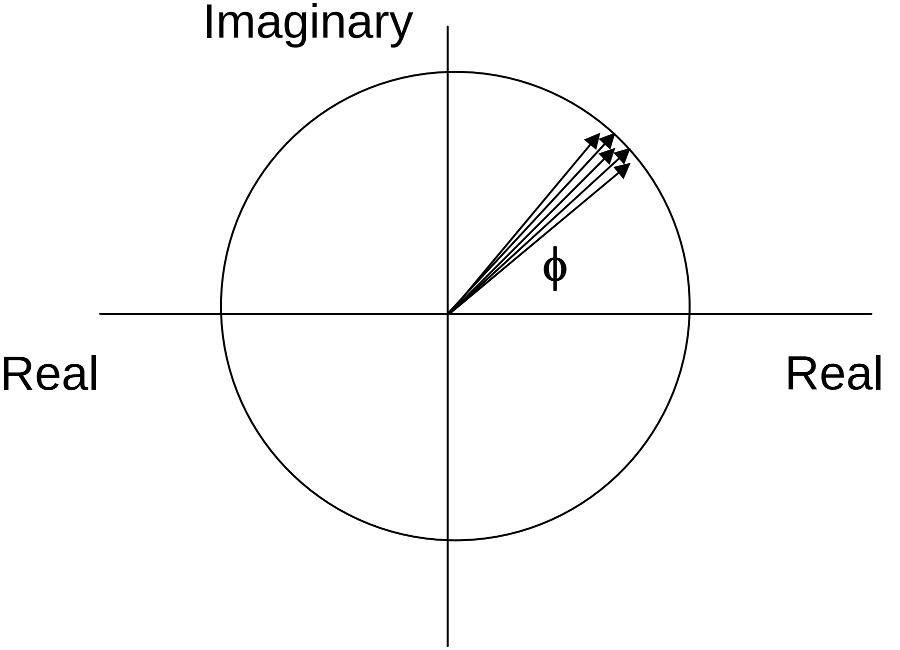
Phase analysis

- aim: detect phase-locking (preferred, non-random phase)
- for example: phae locking to stimulus



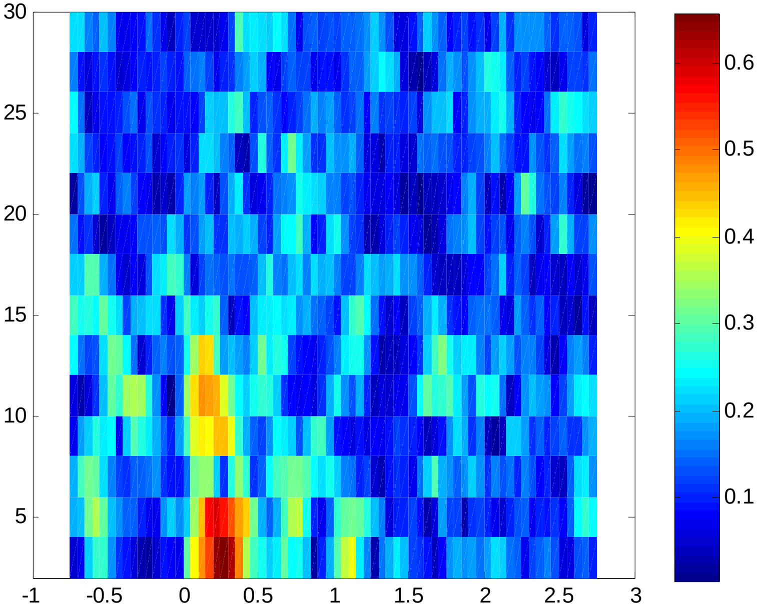
Phase analysis

1. normalize complex number of 1
2. compute mean of complex numbers
3. compute absolute value



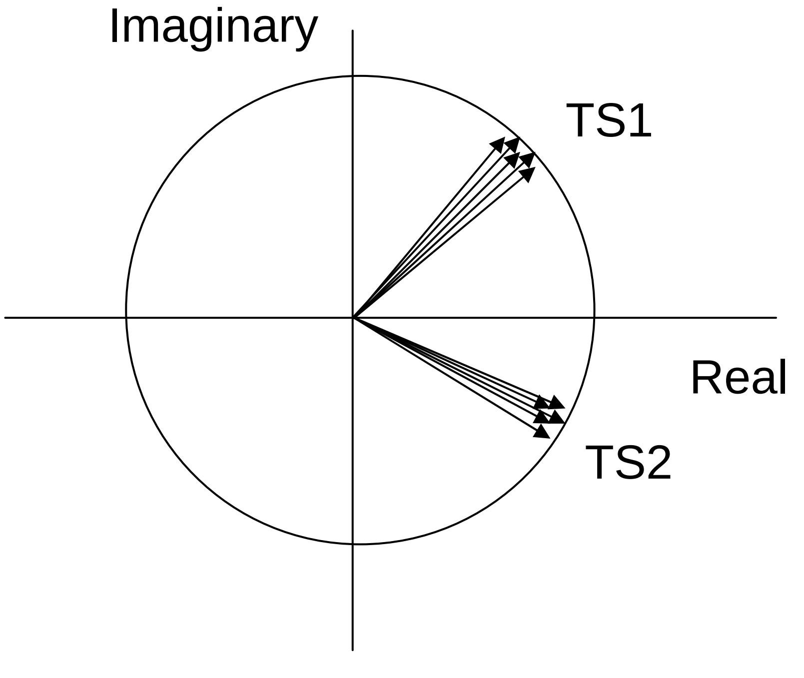
Phase-locking

```
%freq = freqanalysis(cfg, data);
 freqnorm=freq.fourierspctrm./abs(freq.fourierspctrm);%still complex
 PLV=abs(mean(freqnorm,1));%average over trials
 pcolor(freq.time,freq.freq,PLV);shading interp; colorbar
```



Phase-locking

- can be computed between two time series
- here you want to detect preferred phase difference



Phase-locking

```
cfg.output       = ‘powandcsd';
freq = freqanalysis(cfg, data);
cfg.cohmethod    = ‘PLV’;
PLV = freqdescriptives(cfg, freq);
```