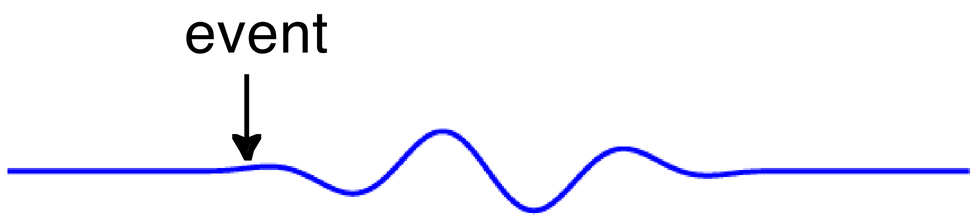
Fundamentals of neuronal oscillations and synchrony

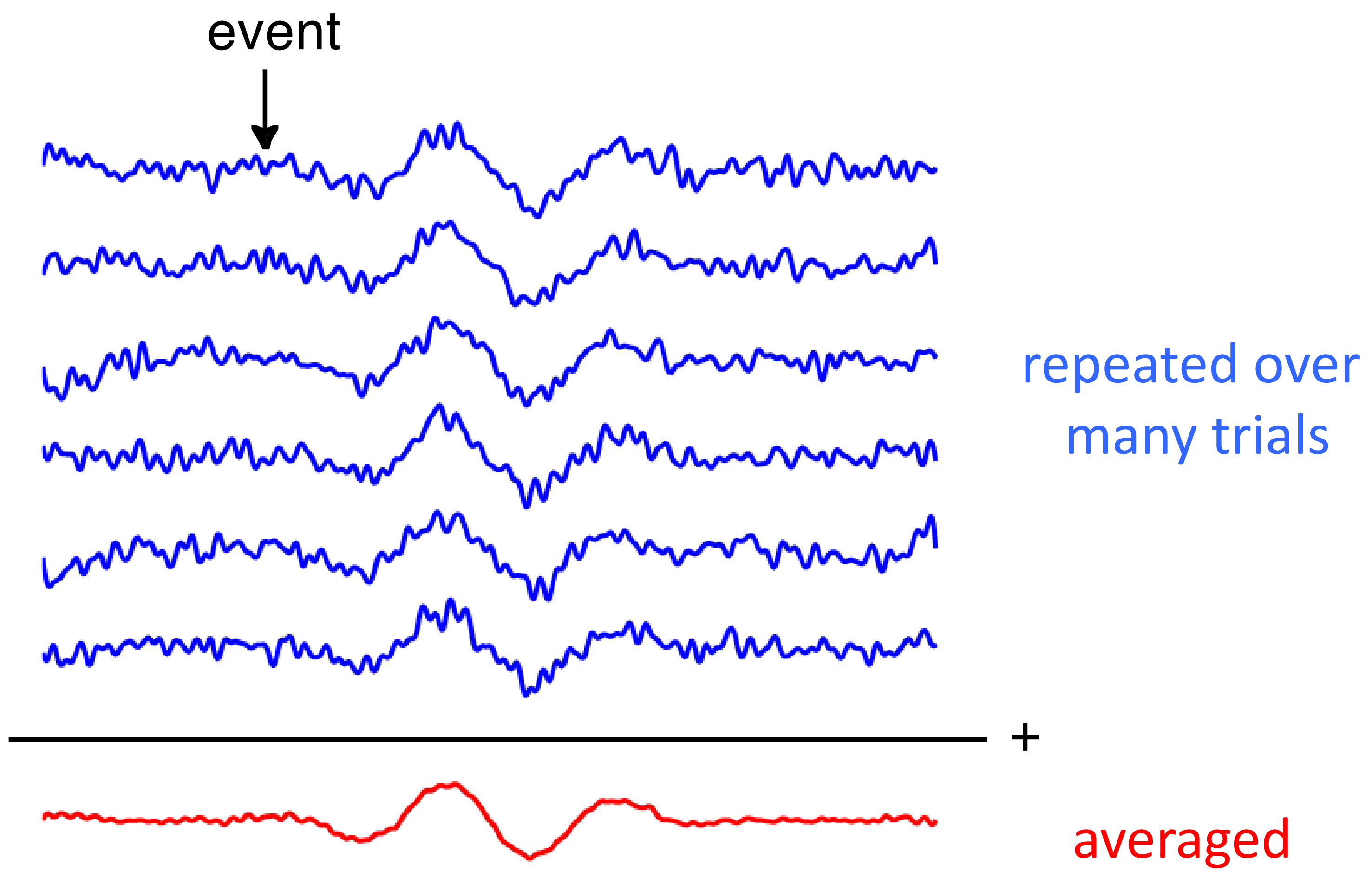
Mats van Es

Donders Institute, Radboud University, Nijmegen, NL

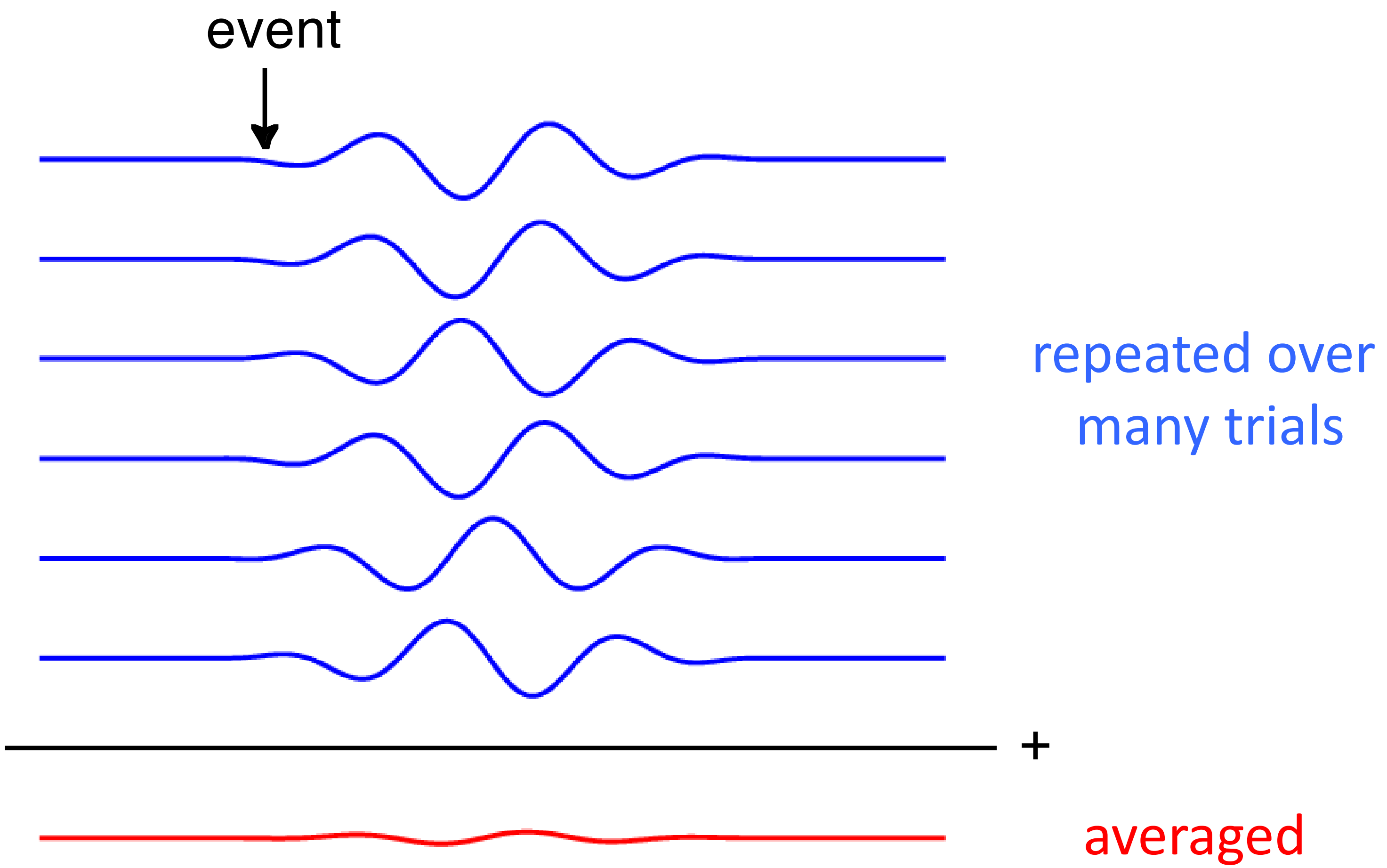
Evoked activity



Evoked activity



Induced activity



M/EEG signal characteristics considered during analysis

timecourse of activity
-> ERP

spectral characteristics
-> power spectrum

temporal changes in power
-> time-frequency response (TFR)

spatial distribution of activity over the head
-> source reconstruction

M/EEG signal characteristics considered during analysis

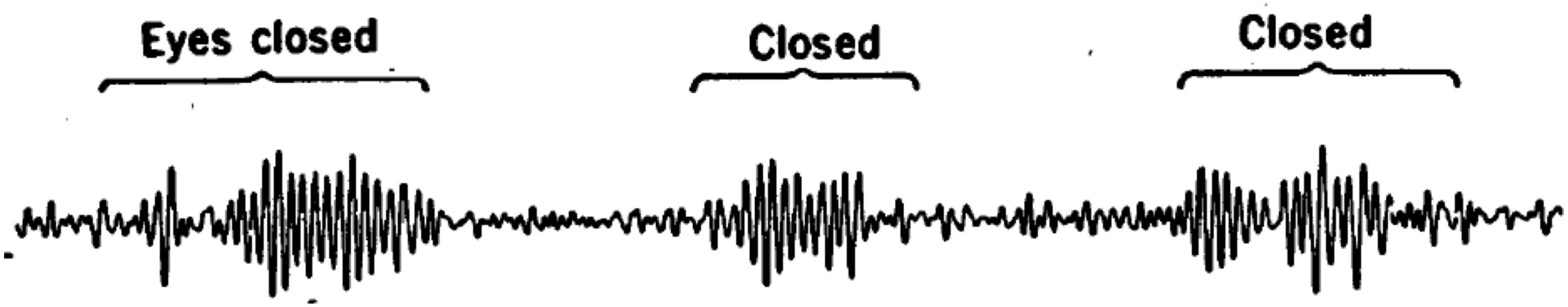
timecourse of activity
-> ERP

spectral characteristics
-> power spectrum

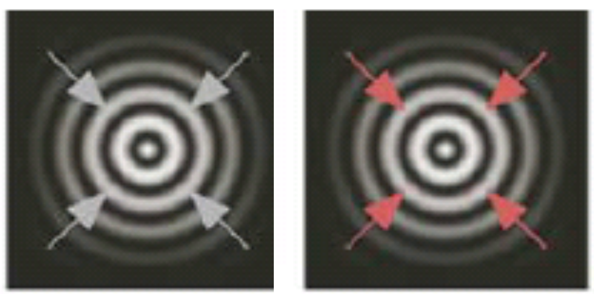
temporal changes in power
-> time-frequency response (TFR)

spatial distribution of activity over the head
-> source reconstruction

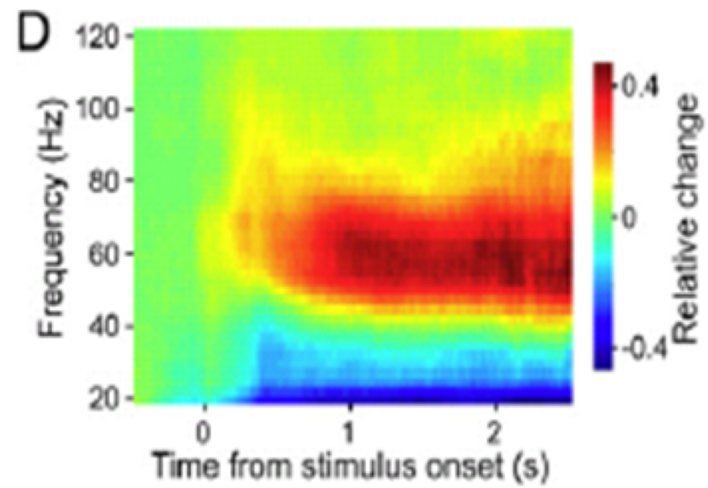
Brain signals contain oscillatory activity at multiple frequencies



Cohen, 1972



Hoogenboom et al, 2006



Outline

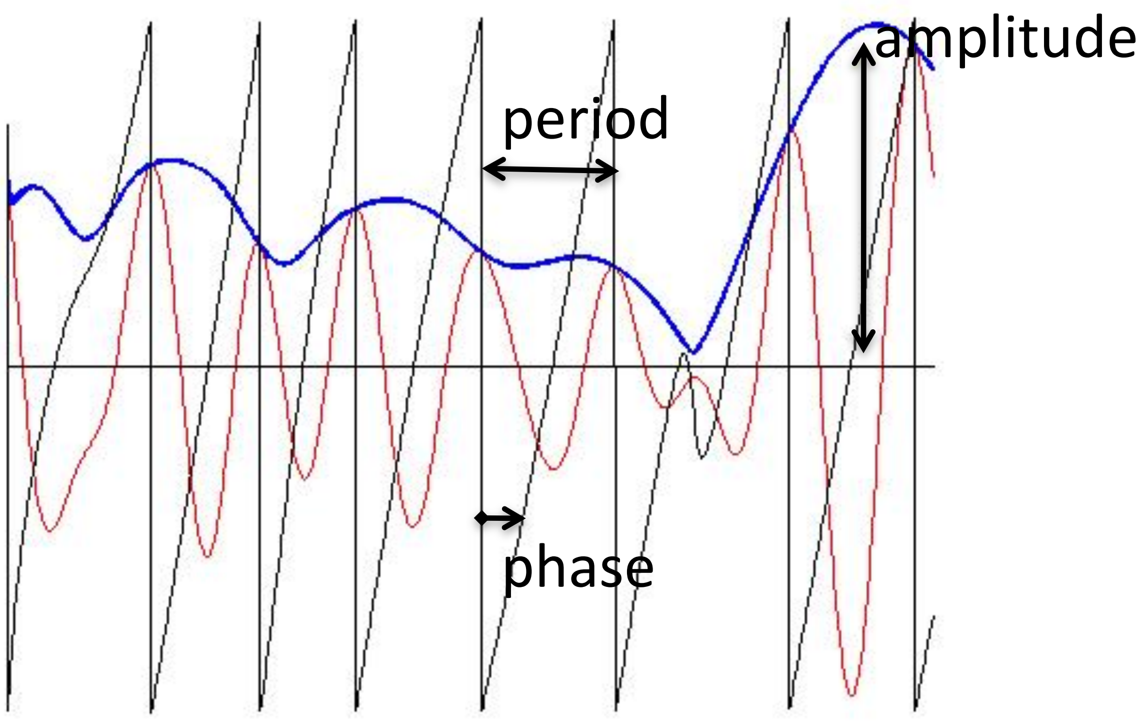
Spectral analysis: going from time to frequency domain

Issues with finite and discrete sampling

Spectral leakage and (multi-)tapering

Time-frequency analysis

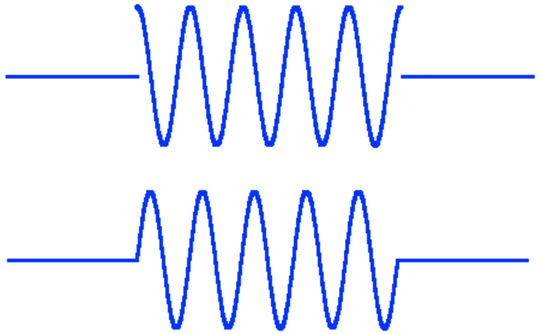
A background note on oscillations



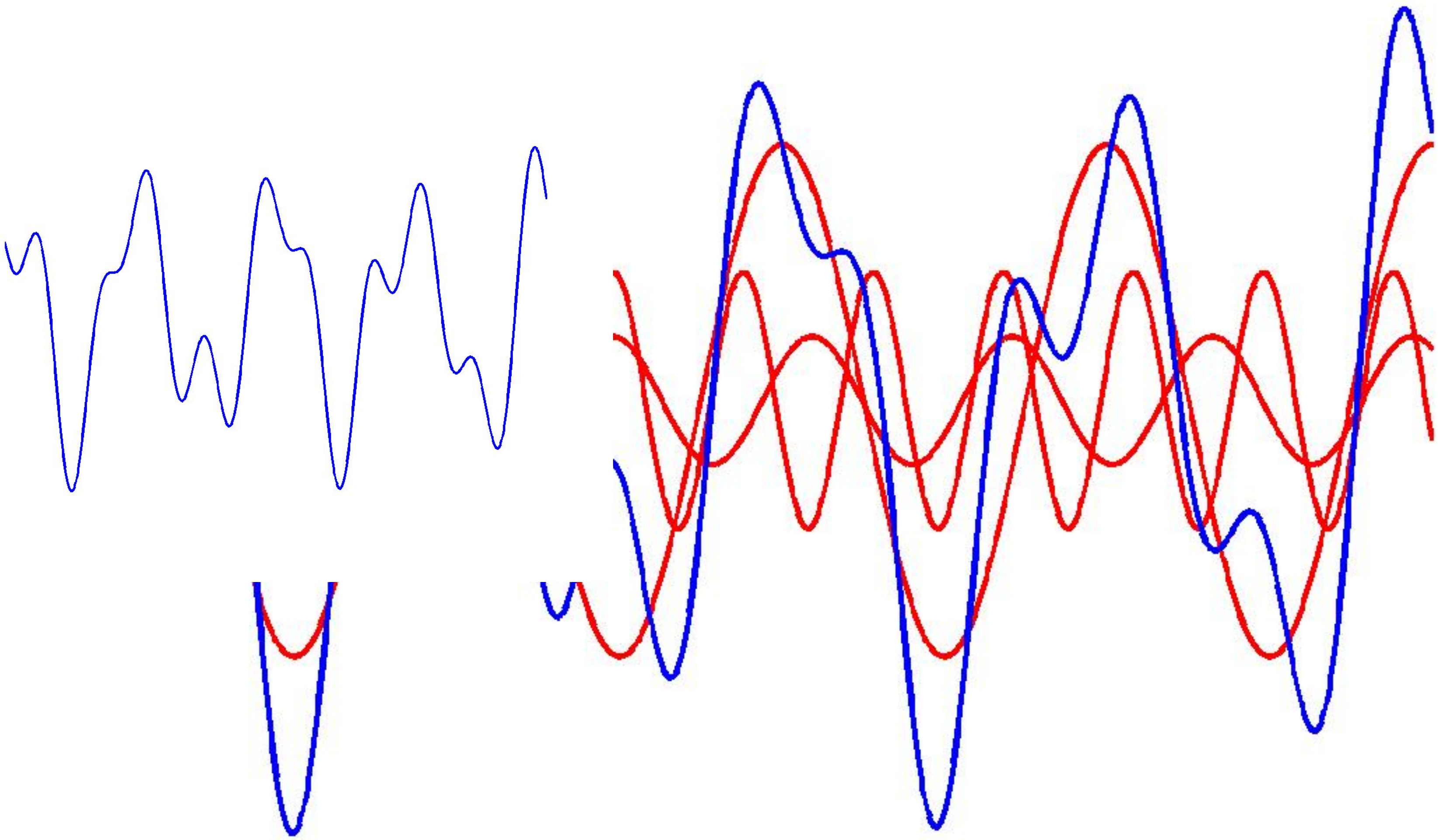
Spectral analysis

Deconstructing a time domain signal into its constituent oscillatory components, a.k.a. Fourier analysis

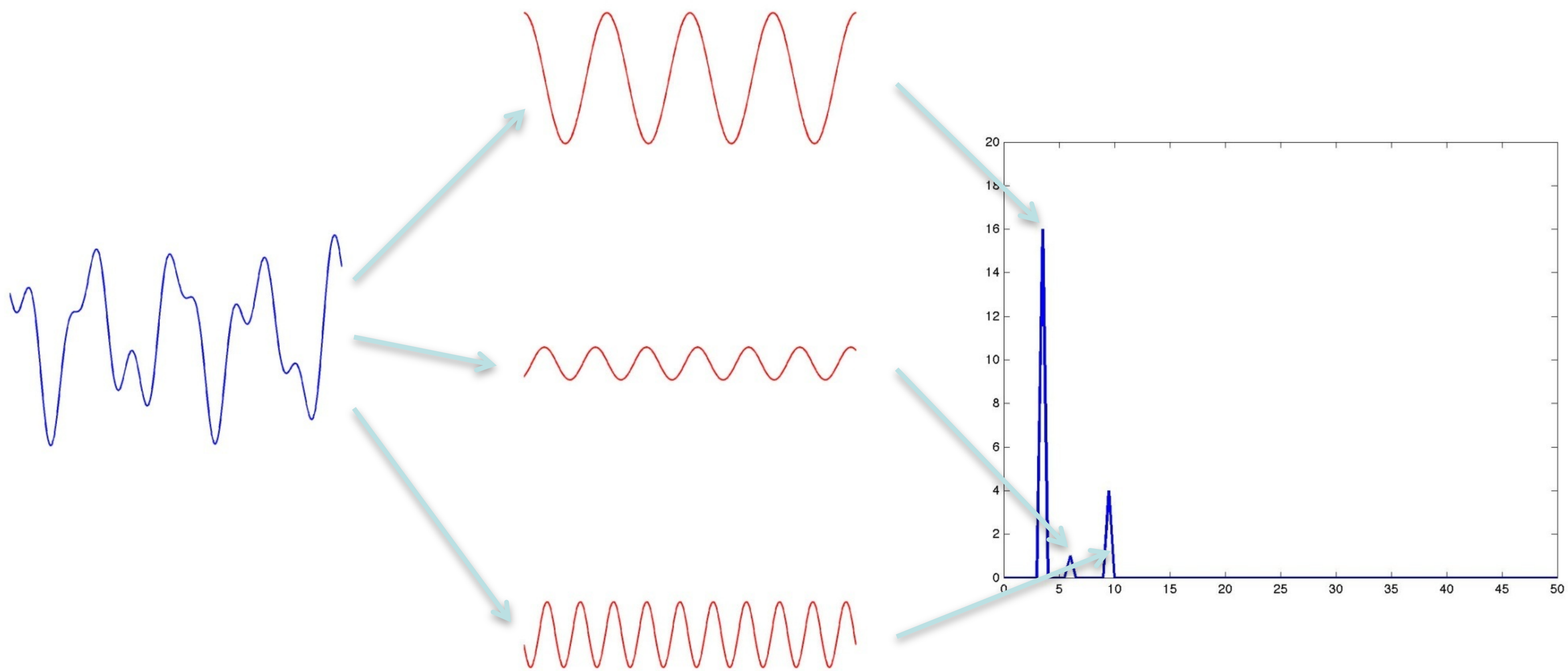
Using simple oscillatory functions: cosines and sines



Spectral decomposition: the principle



Spectral decomposition: the power spectrum



Spectral analysis

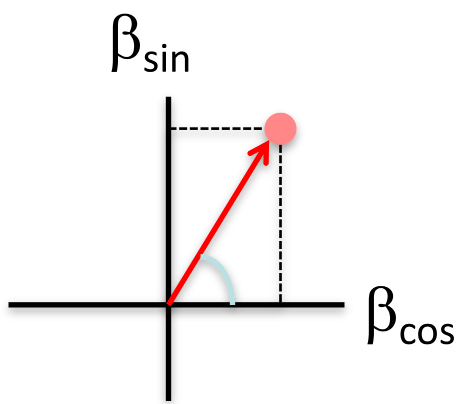
Deconstructing a time domain signal into its constituent oscillatory components, a.k.a. Fourier analysis

Using simple oscillatory functions: cosines and sines

Express signal as function of frequency, rather than time

Concept: linear regression using oscillatory basis functions

Spectral analysis ~ GLM



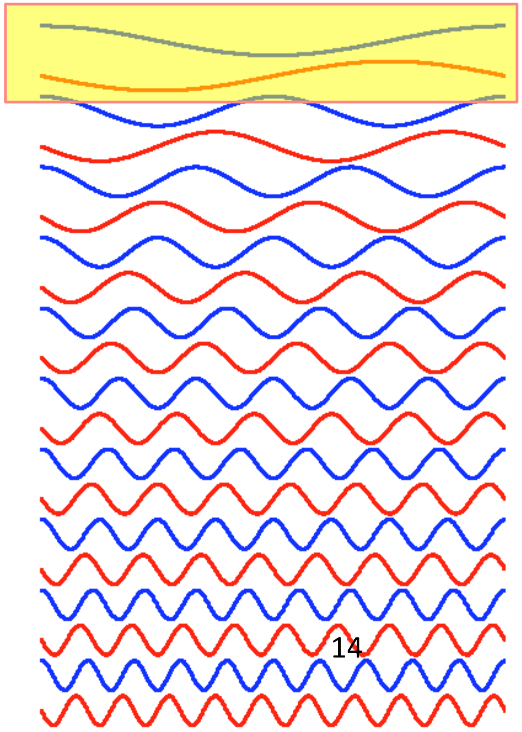
Y = β * **X**

X set of basis functions

β_i contribution of basis function *i* to the data.

β for cosine and sine components for a given frequency map onto amplitude and phase estimate.

Restriction: basis functions should be 'orthogonal'



Consequence 1: frequencies not arbitrary
-> integer amount of cycles should fit into N points.

Consequence 2: N-point signal
-> N basis functions

Time-frequency relation

Consequence 1: frequencies not arbitrary
-> integer amount of cycles should fit into N samples of Δt each.

The frequency resolution is determined by the total length of the data segments ($N * \Delta t = T$)

Rayleigh frequency = $1/T = \Delta f$ = frequency resolution

Time window: **1 s**	→ **Frequencies:** **(0) 1 2 3 4 5 6 .. Hz**
Time window: **0.2 s**	→ **Frequencies:** **(0) 5 10 15 20 .. Hz**

Time-frequency relation

Consequence 2: N-point signal
-> N basis functions

N basis functions -> N/2 frequencies

The highest frequency that can be resolved depends
on the sampling frequency F

Nyquist frequency = F/2

Sampling freq 1 kHz **Time window 1 s**	→	**Frequencies:** **(0) 1 2 … 499 500 Hz**
Sampling freq 400 Hz **Time window 0.25 s**	→	**Frequencies:** **(0) 4 8… 196 200 Hz**

Spectral analysis

Deconstructing a time domain signal into its constituent oscillatory components, a.k.a. Fourier analysis

Using simple oscillatory functions: cosines and sines

Express signal as function of frequency, rather than time

Concept: linear regression using oscillatory basis functions

Each oscillatory component has an amplitude and phase

Discrete and finite sampling constrains the frequency axis

Goal and challenges

- Estimate the true oscillations from the observed data
- Limited time available for Fourier transform
- You are looking at the activity through a time restricted window

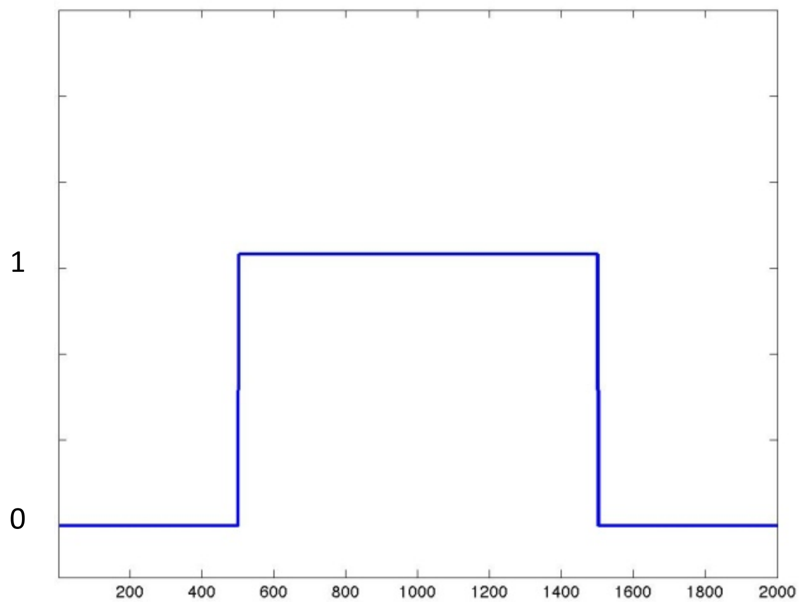
Goal and challenges

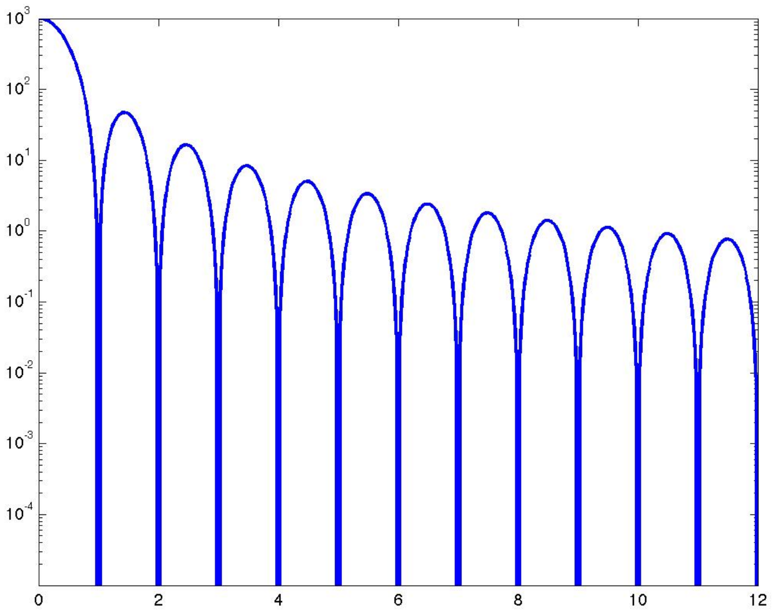
- Estimate the true oscillations from the observed data
- Limited time available for Fourier transform
- You are looking at the activity through a time restricted window

- This implicitly means that the data are 'tapered' with a boxcar
- Furthermore, data are discretely sampled

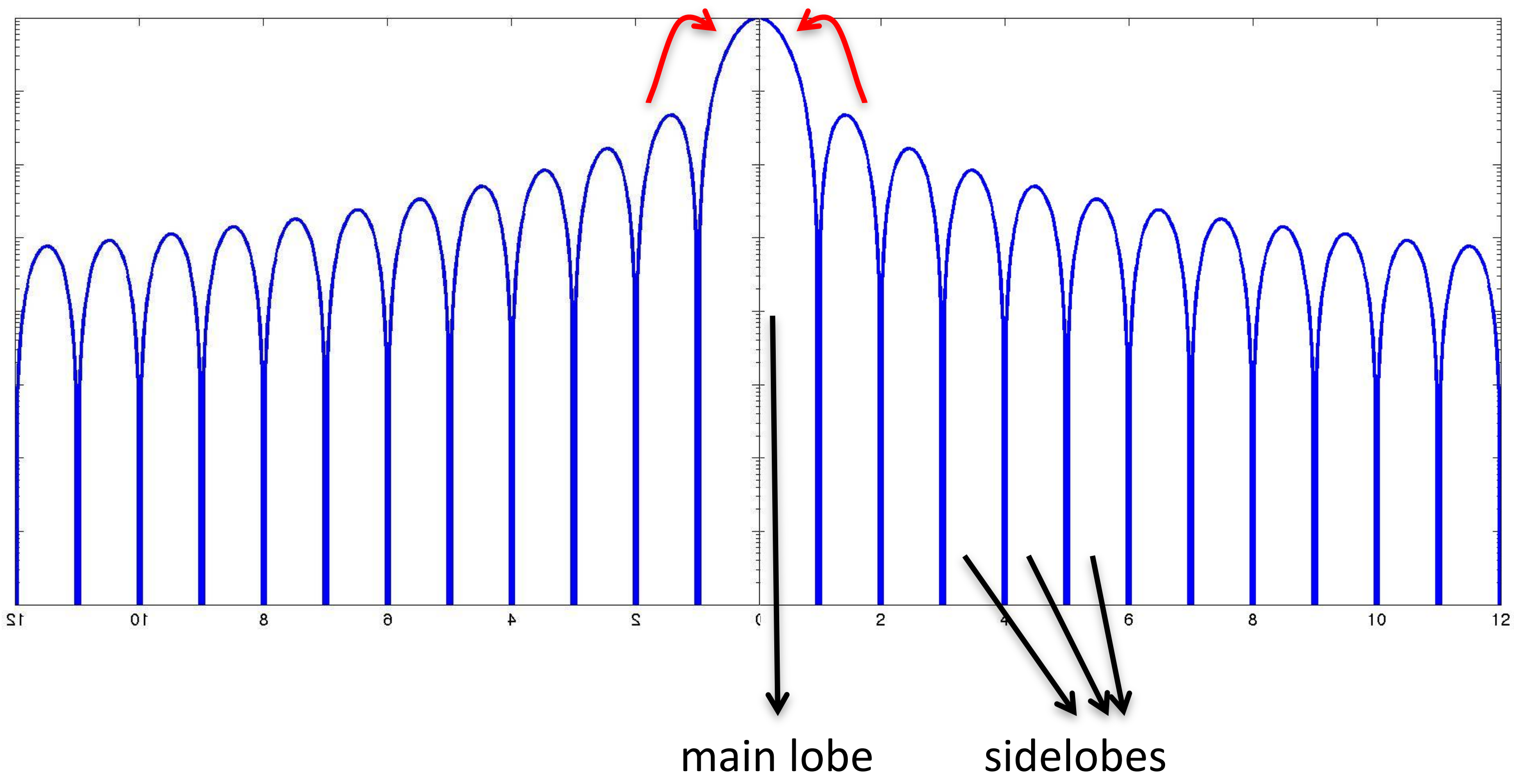
Spectral leakage and tapering

- True oscillations in data at frequencies **not sampled** with Fourier transform **spread their energy** to the sampled frequencies
- Not tapering is equal to applying a "boxcar" taper
- Each type of taper has a specific leakage profile

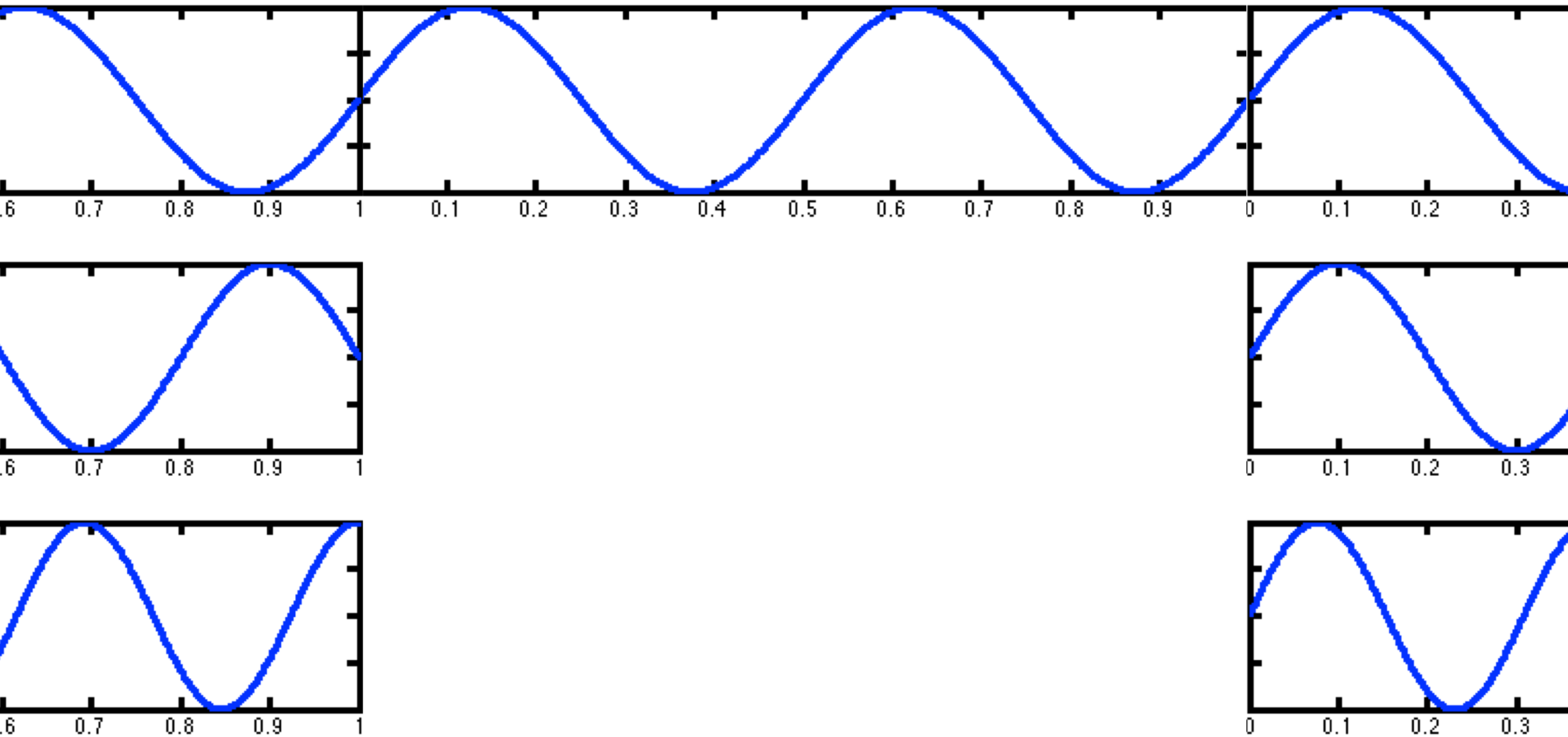




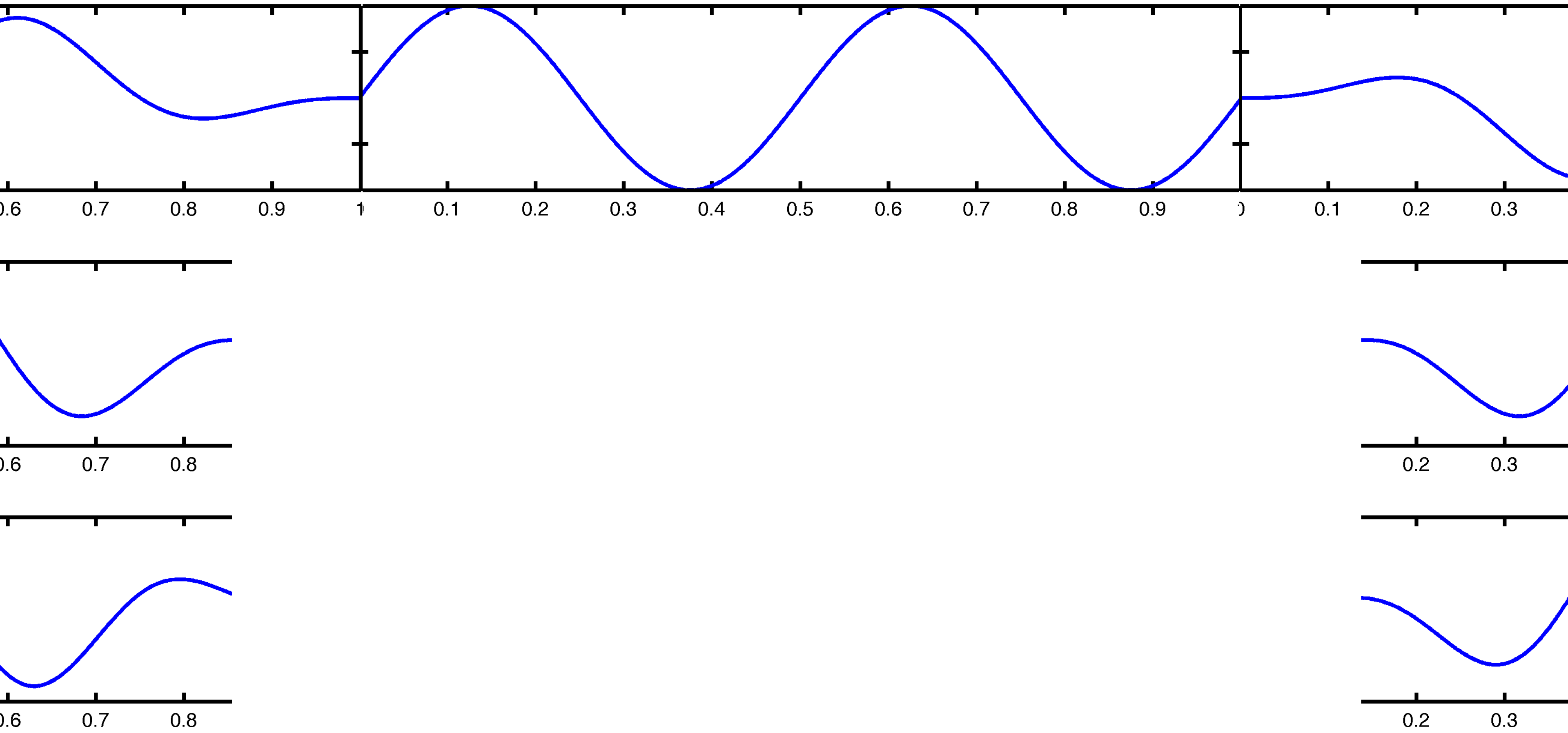
Spectral leakage



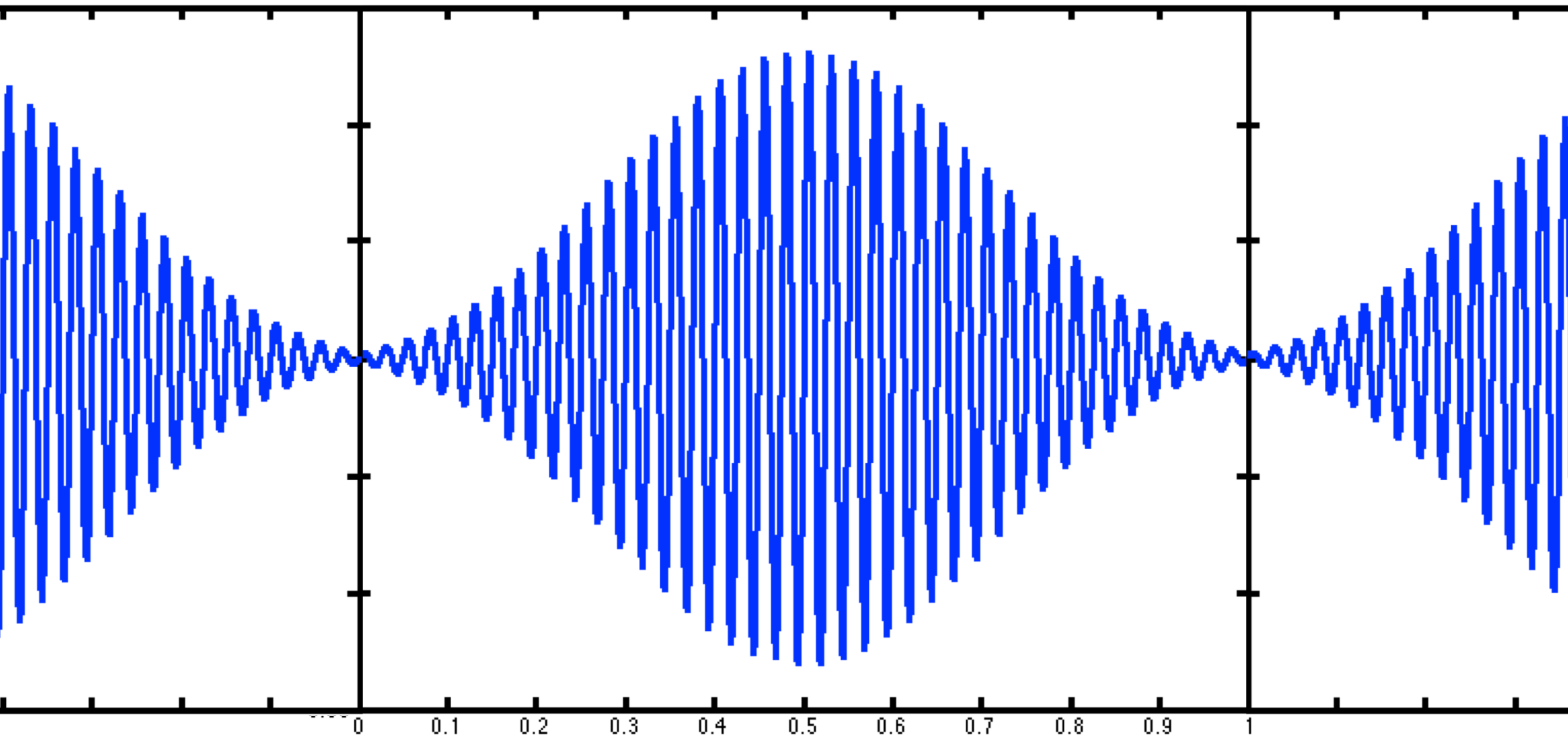
Tapering in spectral analysis



Tapering in spectral analysis

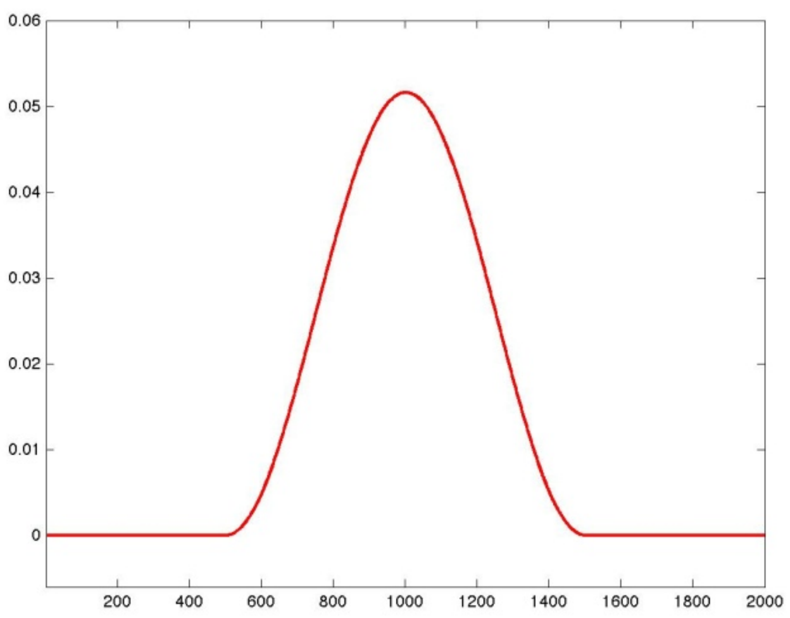


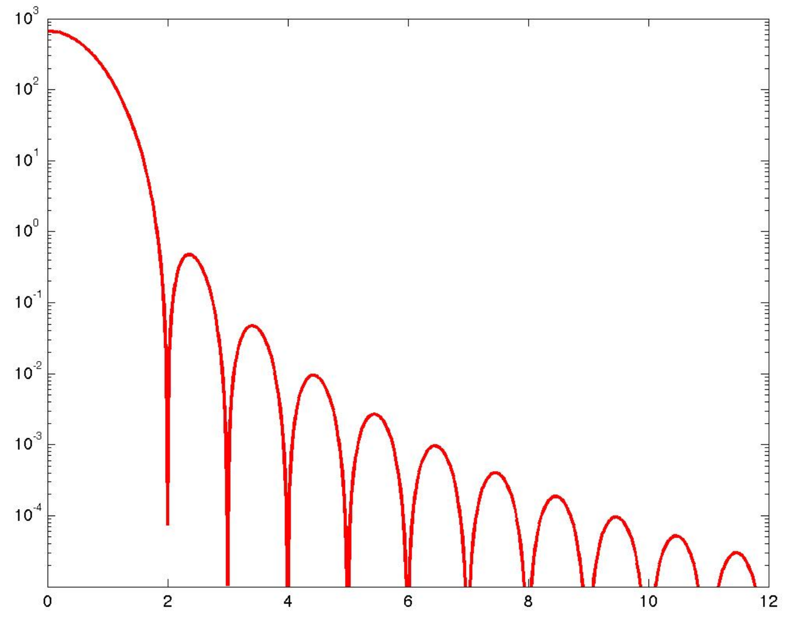
Tapering in spectral analysis



Spectral leakage and tapering

- True oscillations in data at frequencies not sampled with Fourier transform spread their energy to the sampled frequencies
- Not tapering is equal to applying a boxcar taper
- Each type of taper has a specific leakage profile





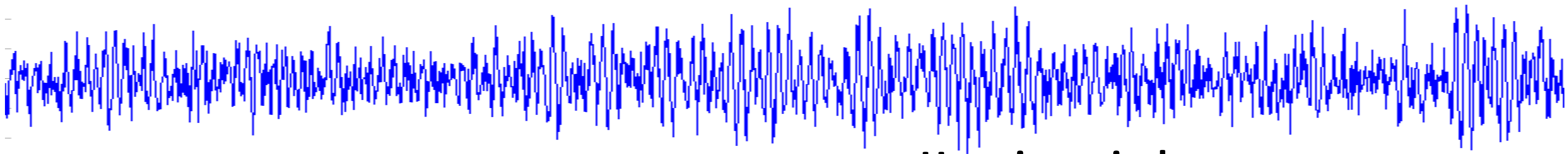
Multitapers

Make use of more than one taper and combine their properties

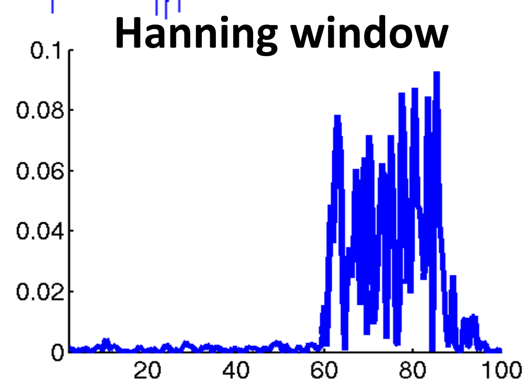
Used for smoothing in the frequency domain

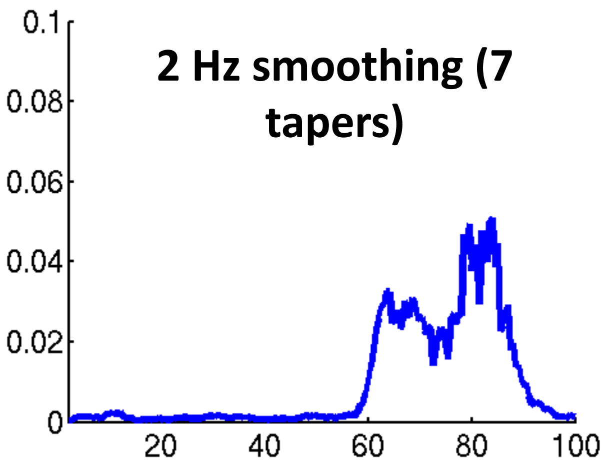
Instead of “smoothing” one can also say “controlled leakage”

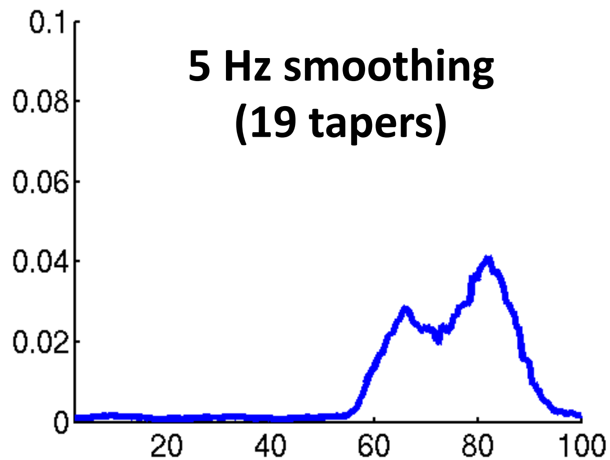
Multitapered spectral analysis

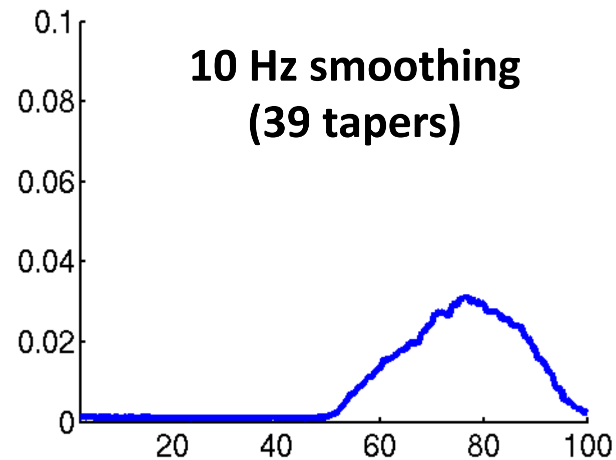


broadband activity between 60-90 Hz



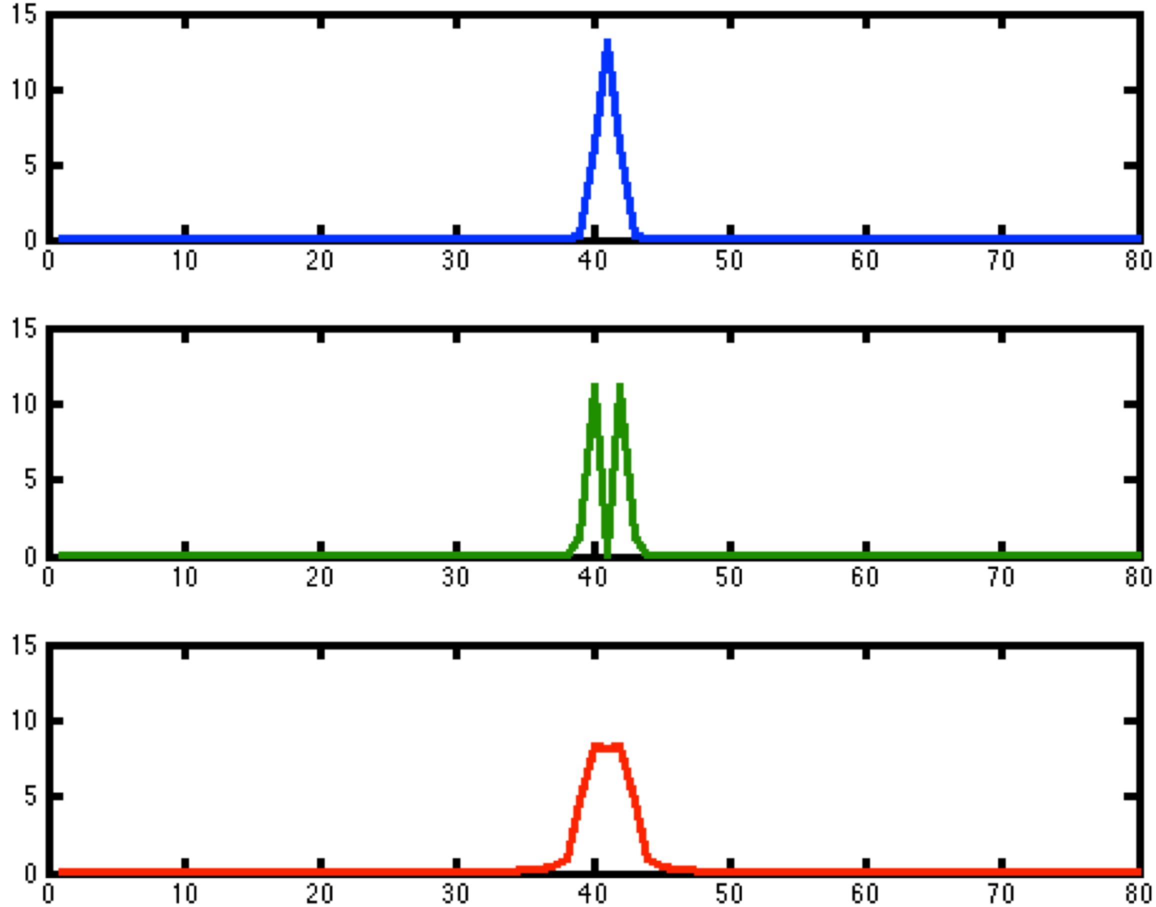






Mitra & Pesaran, 1999, Biophys J

Multitapered spectral analysis



Multitapers

Multitapers are useful for reliable estimation of high frequency components

Low frequency components are better estimated using a single (Hanning) taper

```
%estimate low frequencies

cfg  = [];
cfg.method = 'mtmfft';
cfg.foilim = [1 30];
cfg.taper  = 'hanning';
      .
      .
      .
freq=ft_freqanalysis(cfg, data);
```

```
%estimate high frequencies

cfg  = [];
cfg.method     = 'mtmfft';
cfg.foilim     = [30 120];
cfg.taper      = 'dpss';
cfg.tapsmofrq = 8;
      .
      .
freq=ft_freqanalysis(cfg, data);
```

Interim summary

- Spectral analysis
  - Decompose signal into its constituent oscillatory components
  - Focused on ‘stationary’ power
- Tapers
  - Boxcar, Hanning, Gaussian
- Multitapers
  - Control spectral leakage/smoothing

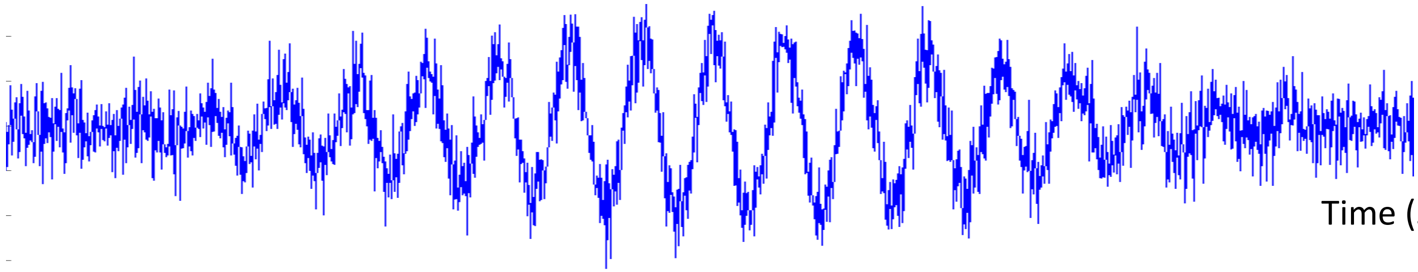
Time-frequency analysis

Typically, brain signals are not 'stationary'

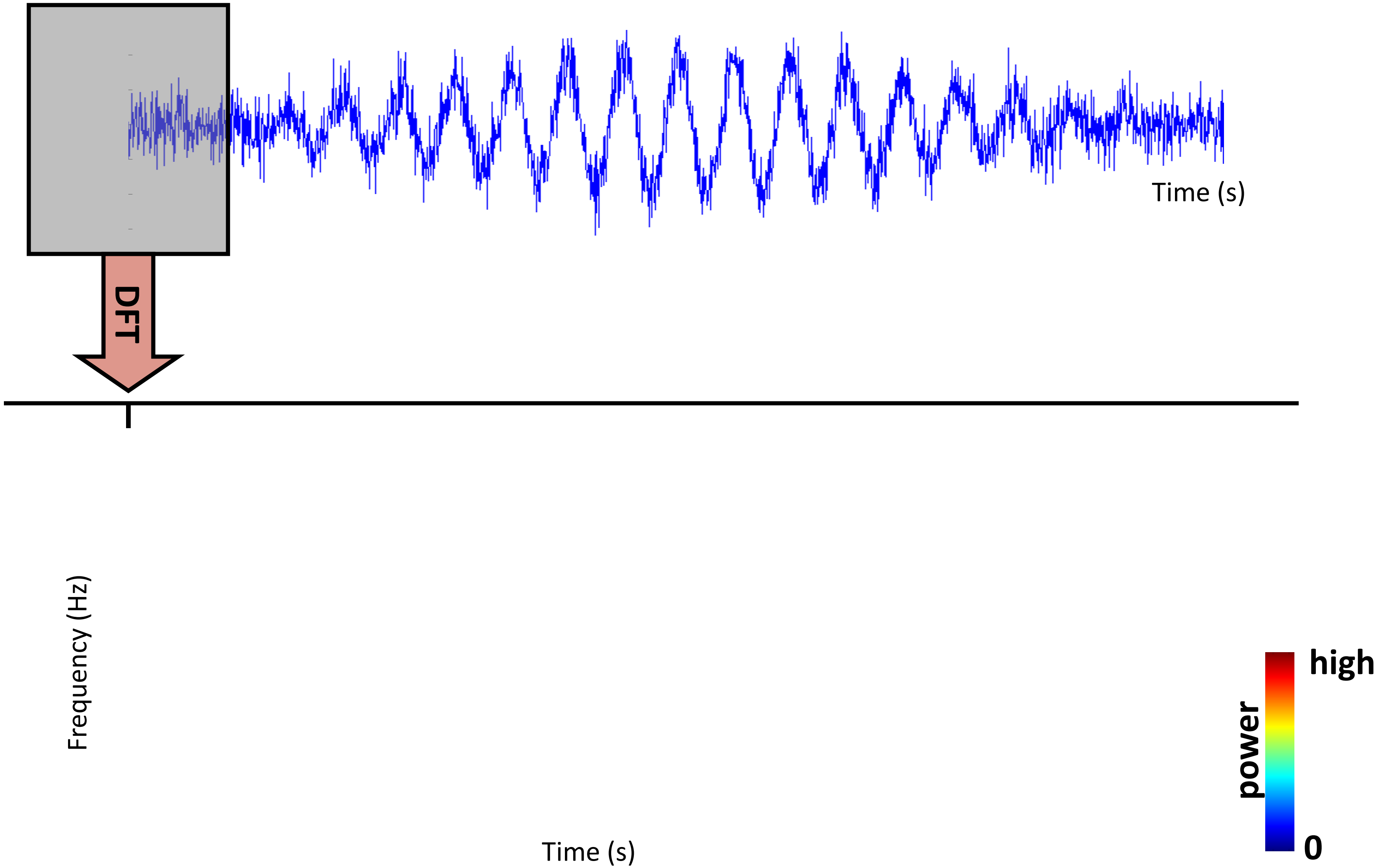
- Divide the measured signal in shorter time segments and apply Fourier analysis to each signal segment
- Everything we saw so far with respect to frequency resolution applies here as well

```
cfg  = [];
cfg.method = 'mtmconvol';
        .
        .
        .
freq = ft_freqanalysis(cfg, data);
```

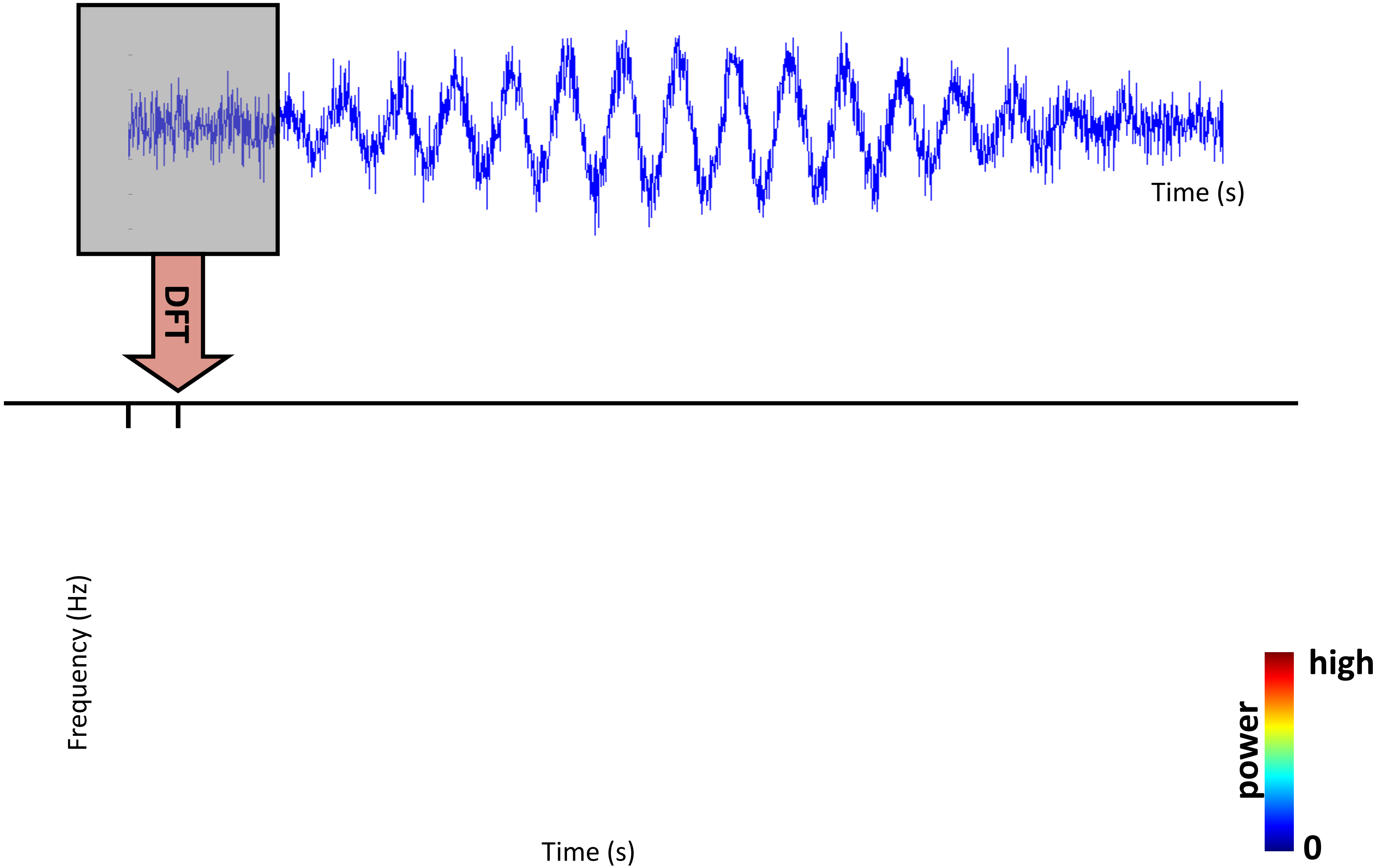
Time frequency analysis



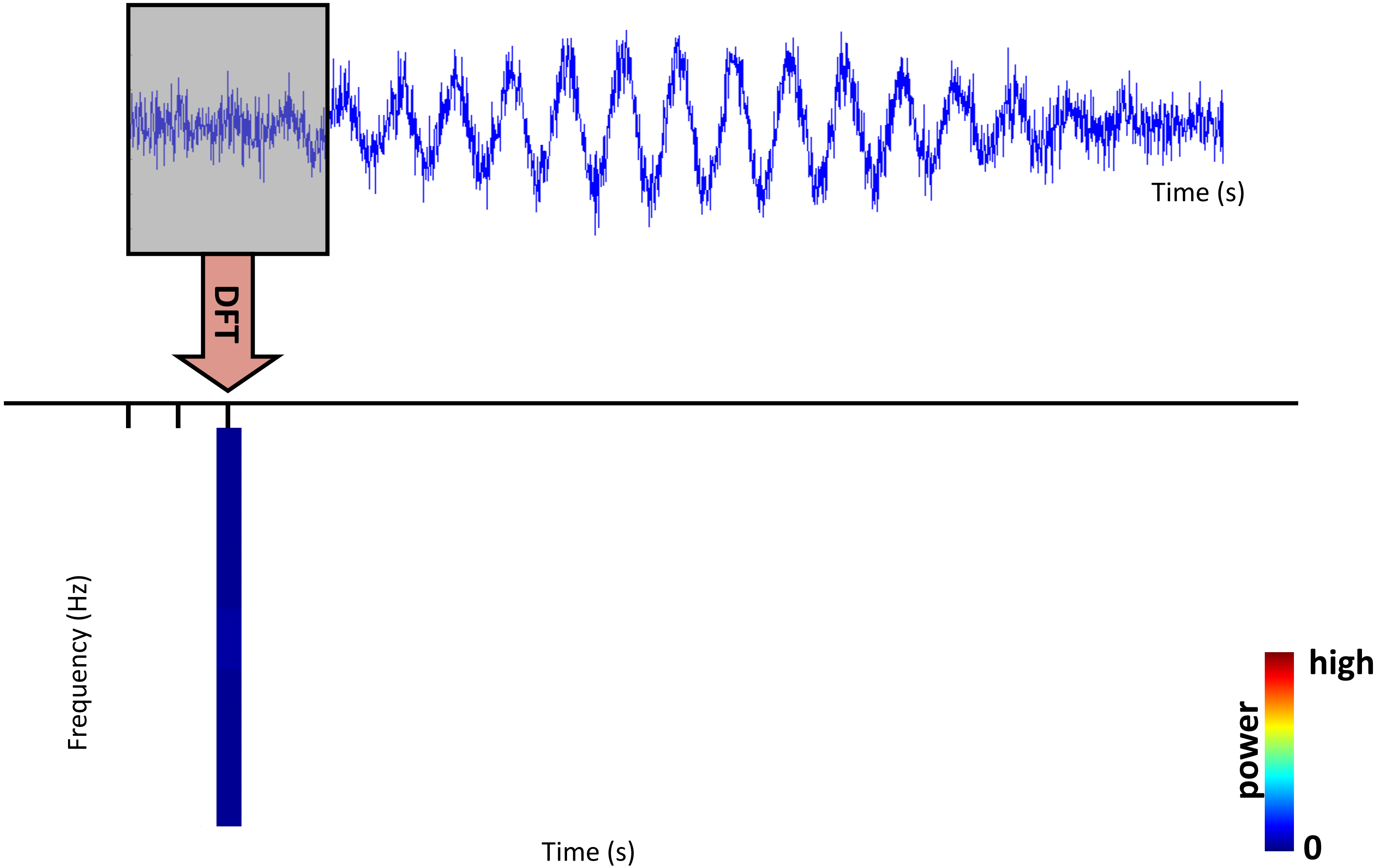
Time frequency analysis



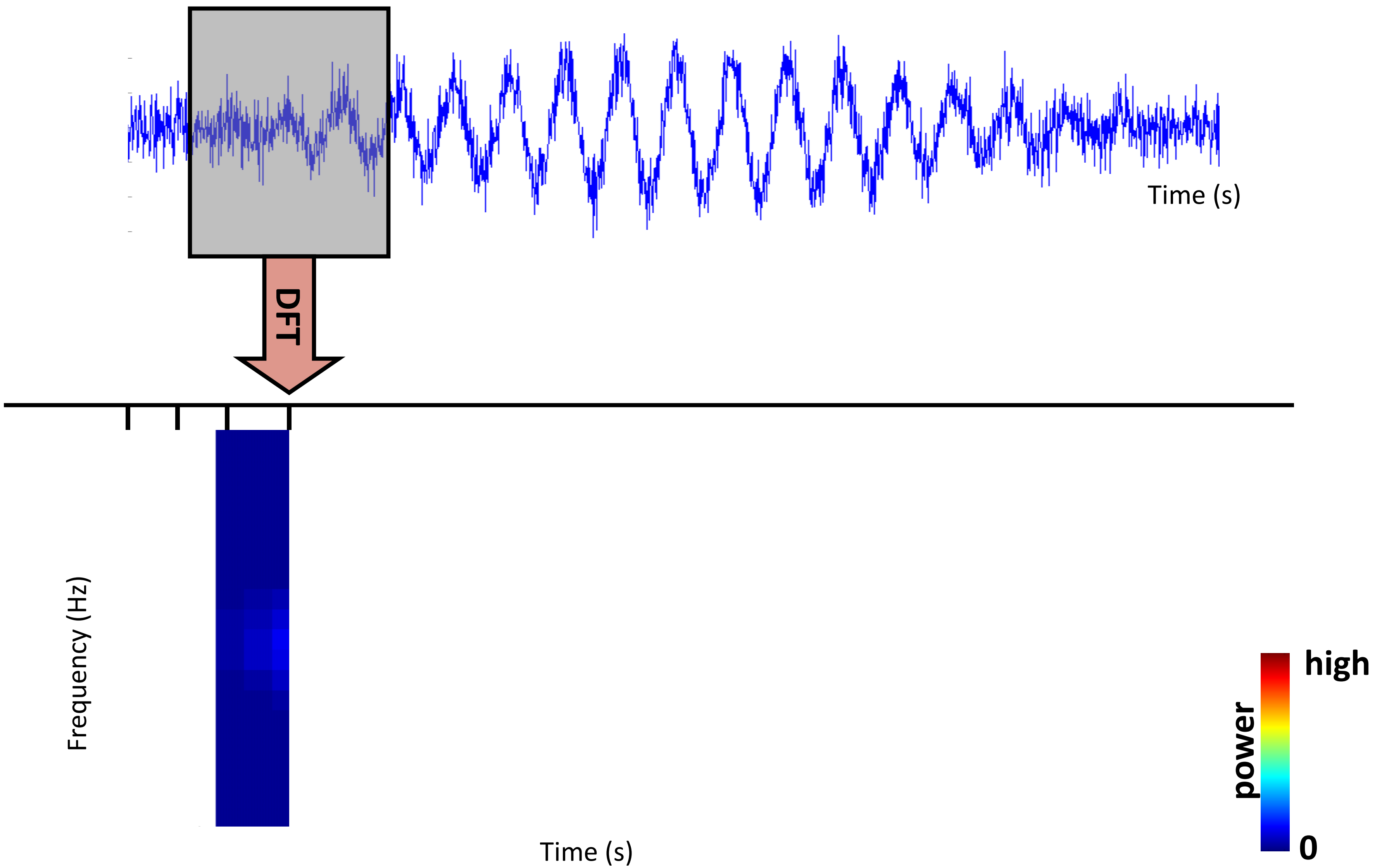
Time frequency analysis



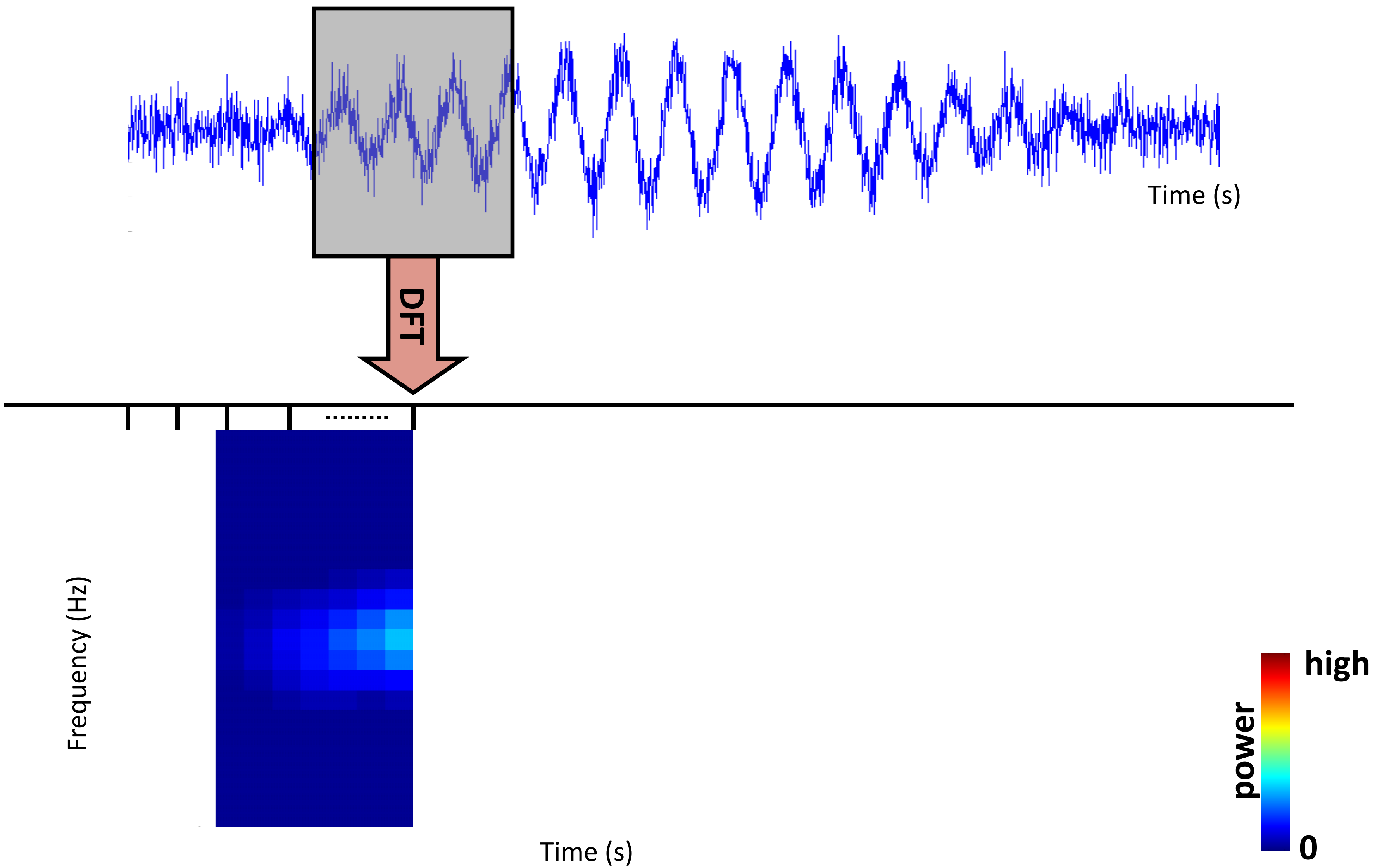
Time frequency analysis



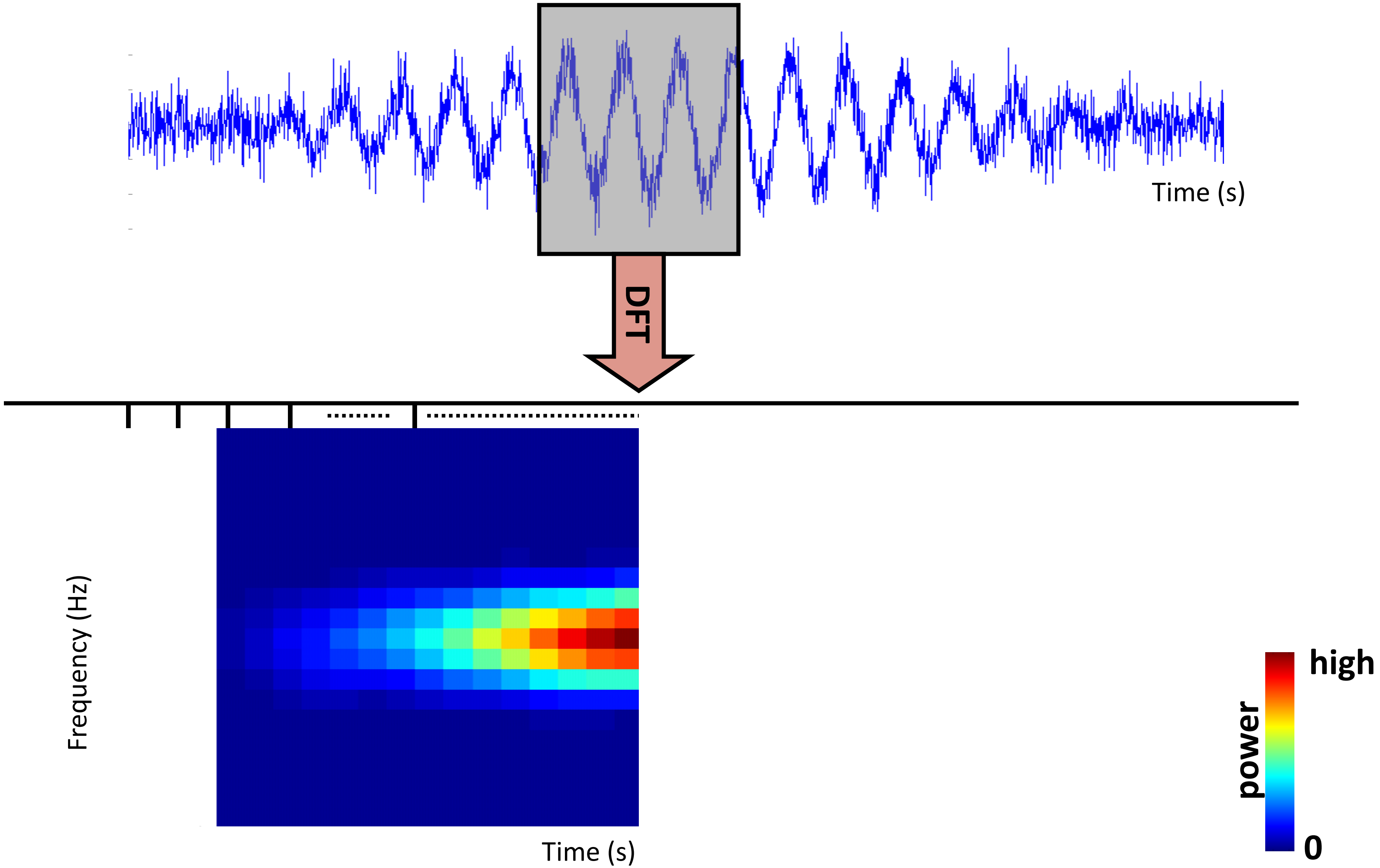
Time frequency analysis



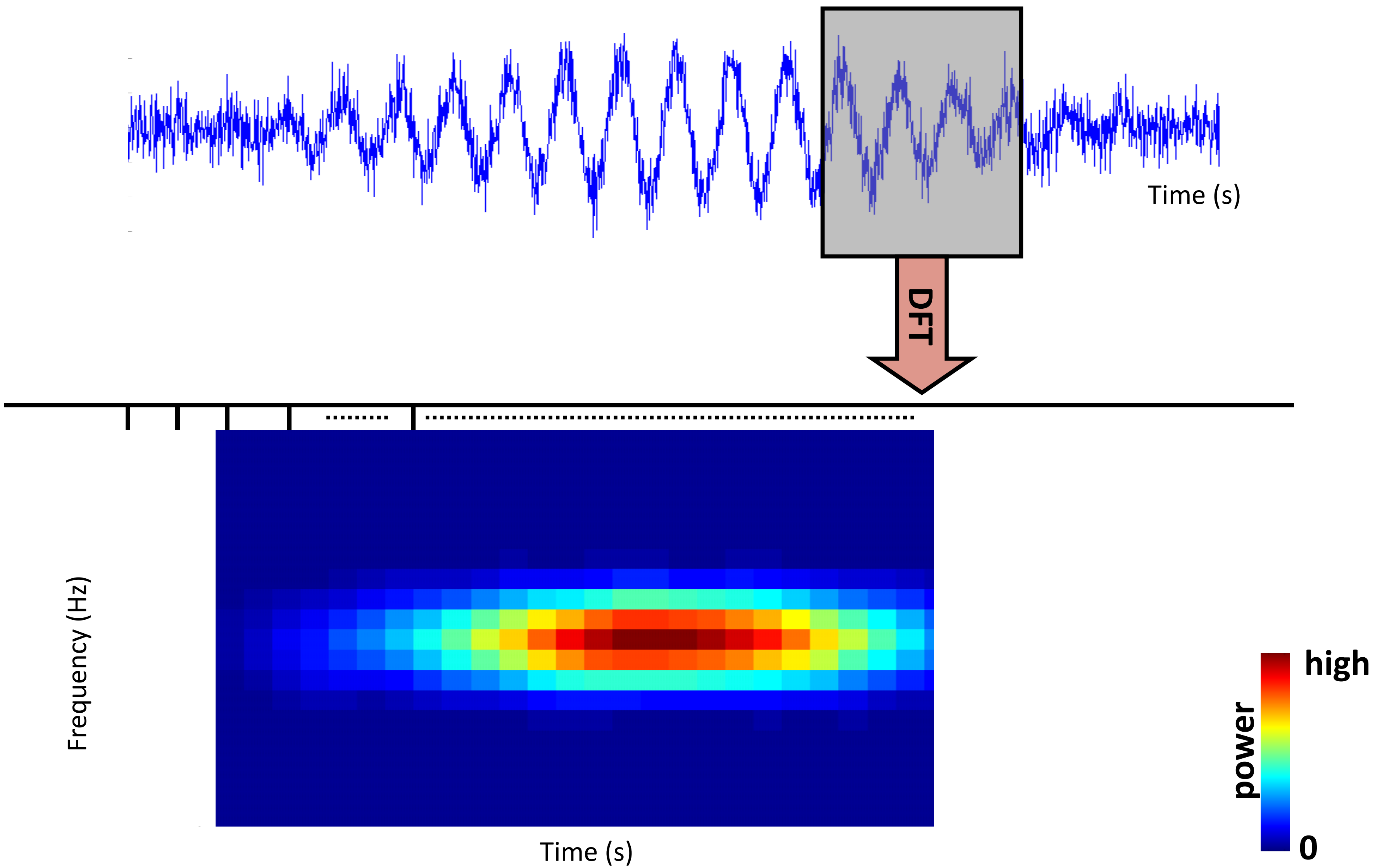
Time frequency analysis



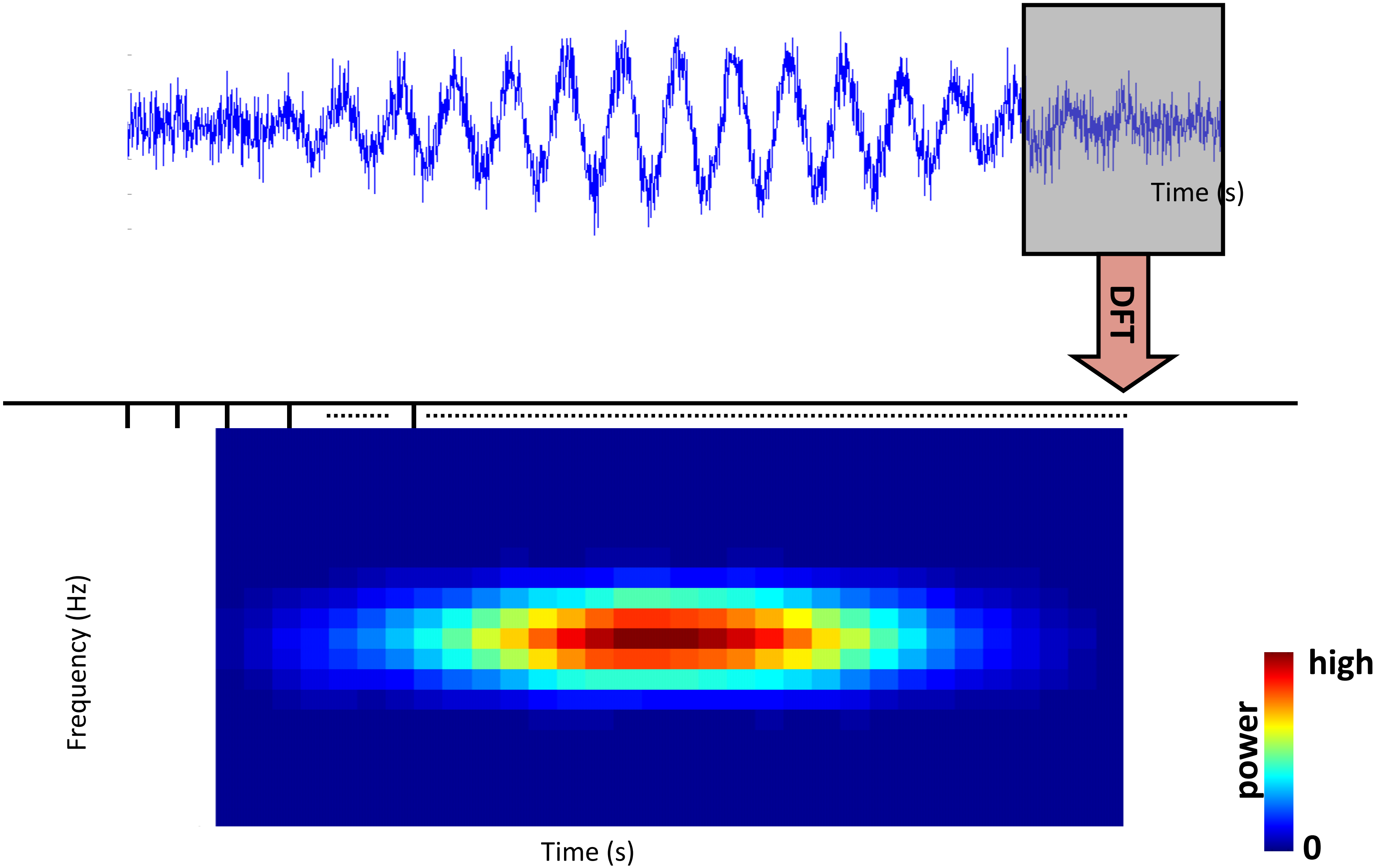
Time frequency analysis



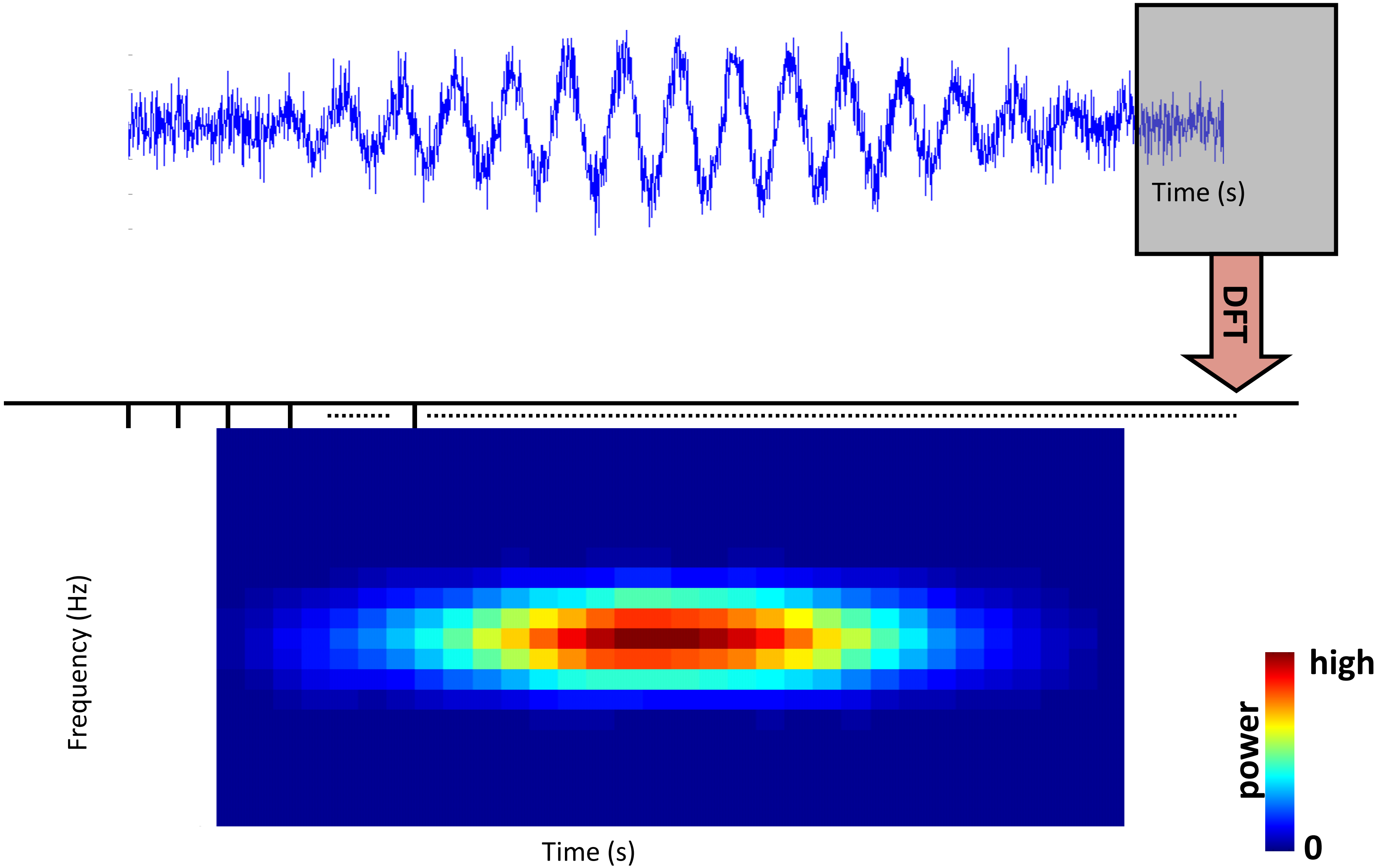
Time frequency analysis



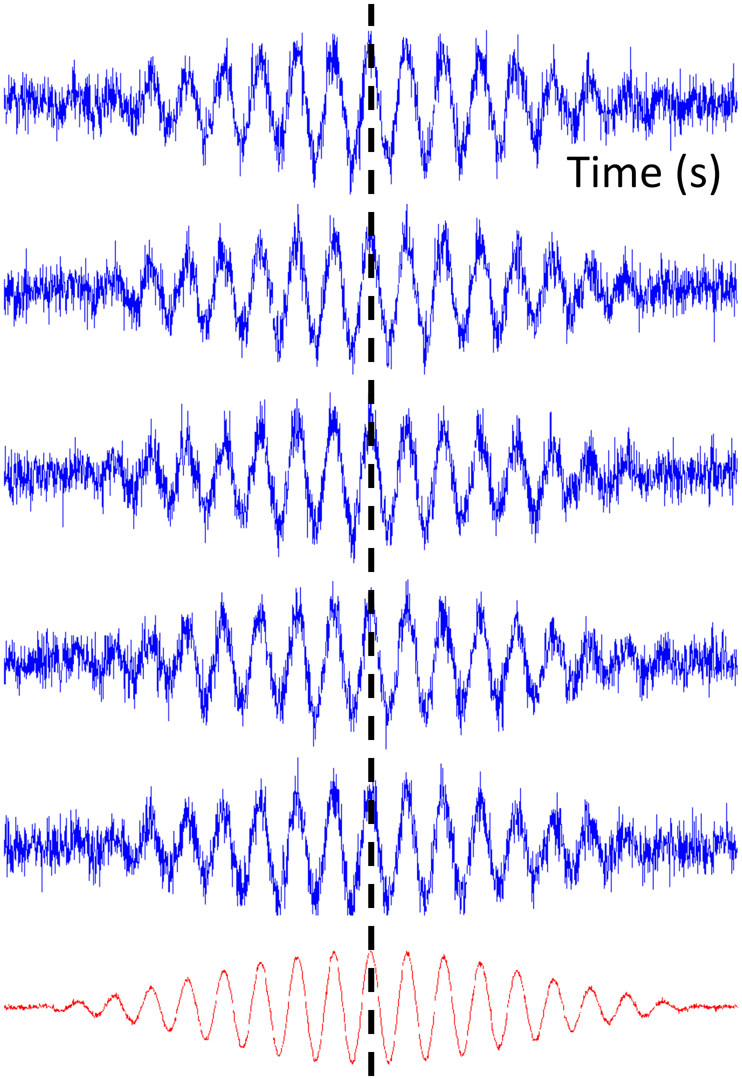
Time frequency analysis

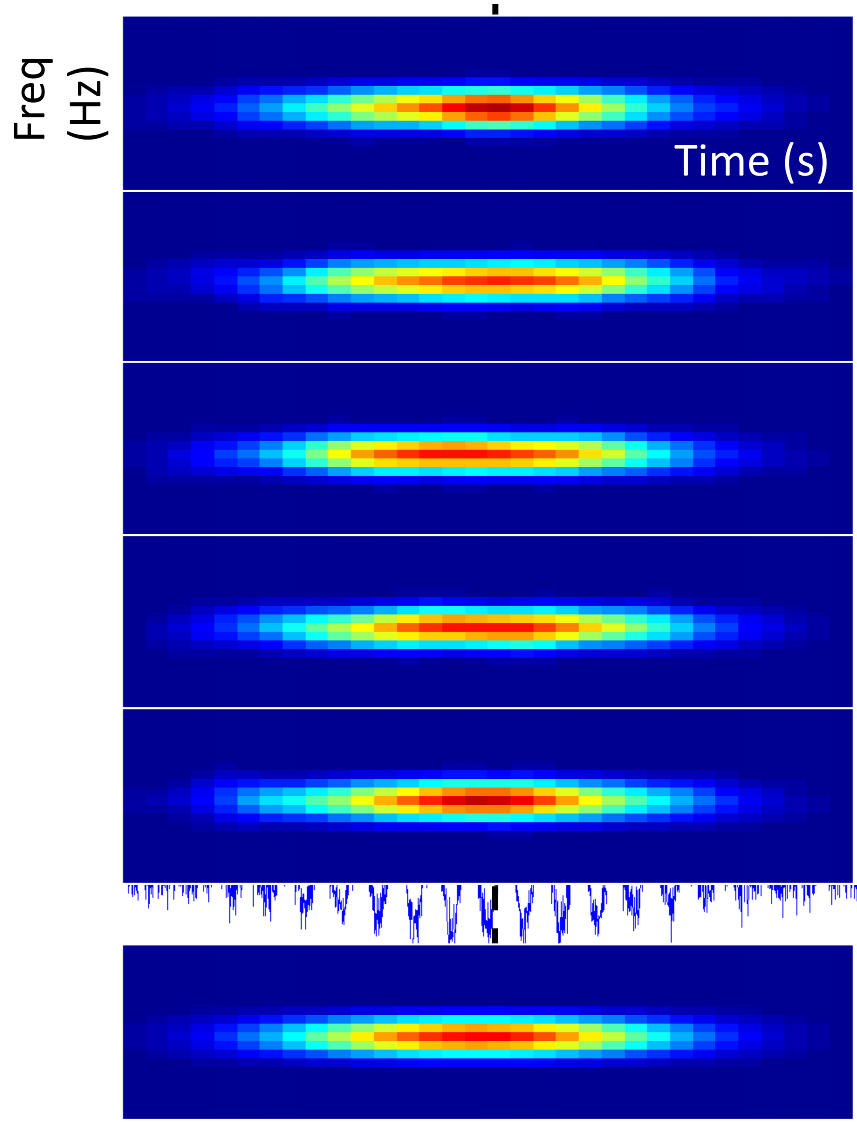


Time frequency analysis

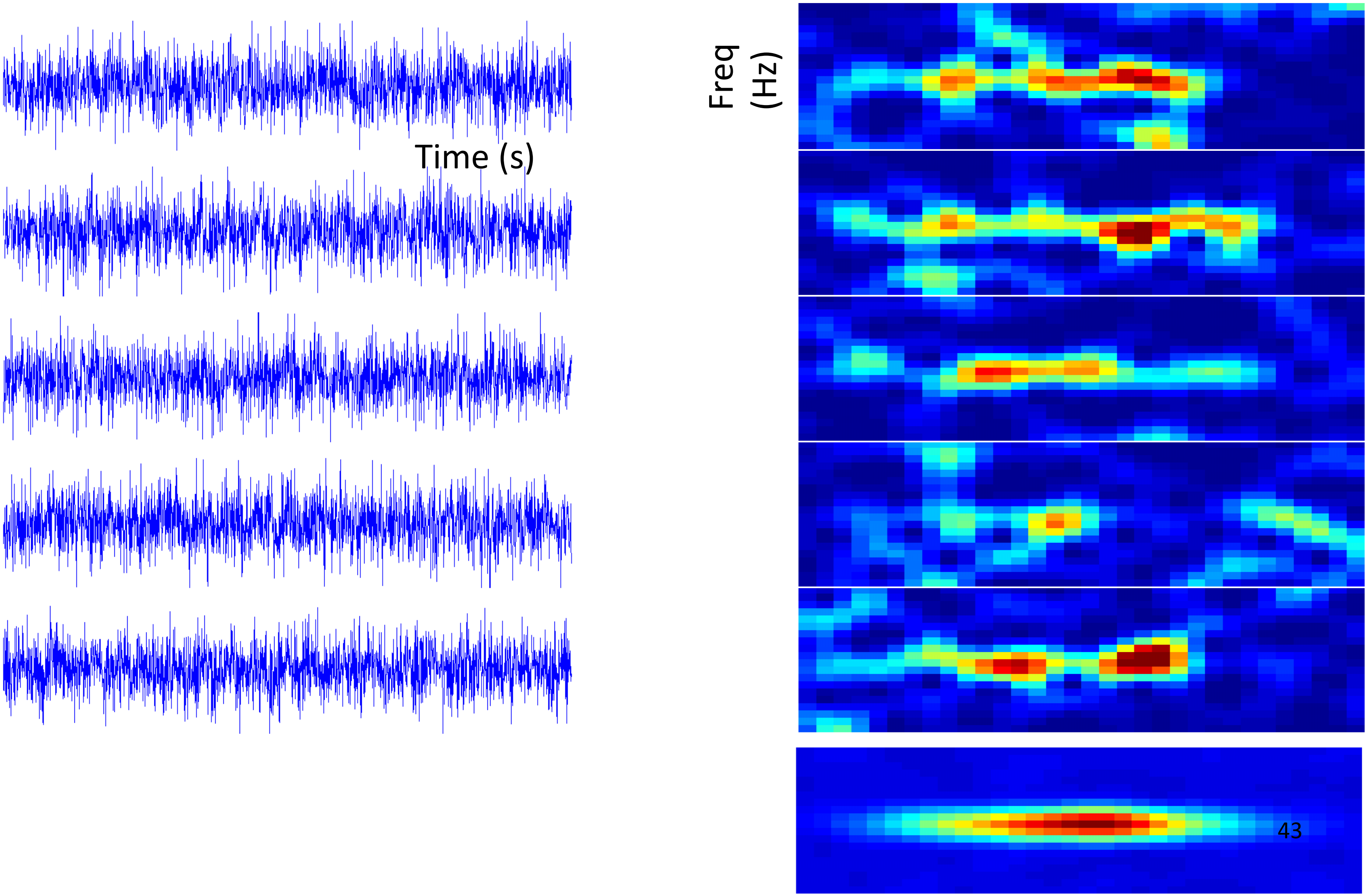


Evoked versus induced activity

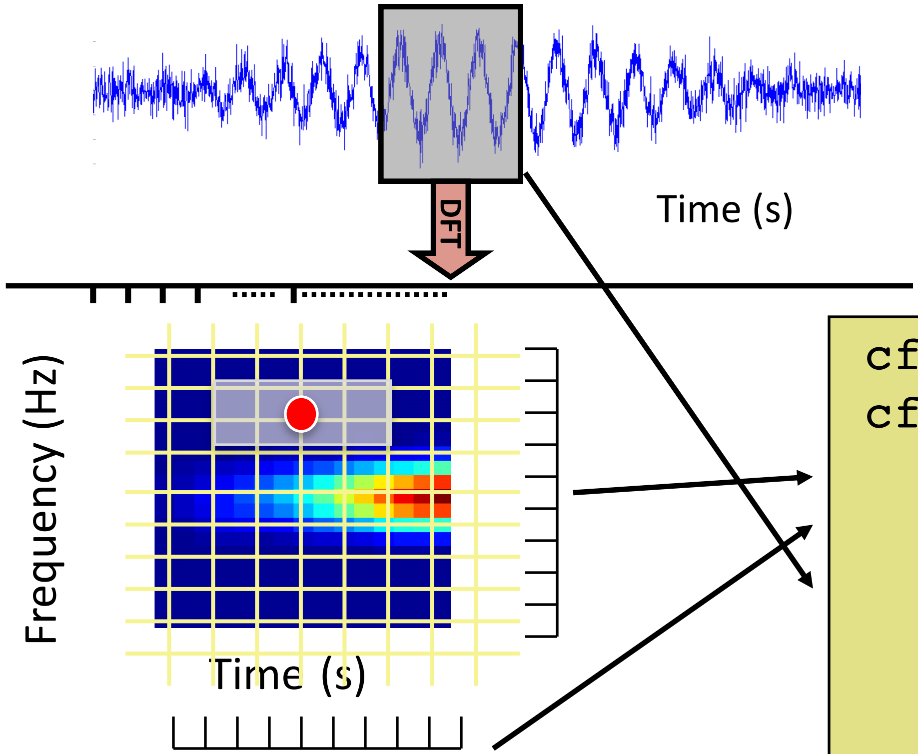




Noisy signal -> many trials needed



The time-frequency plane



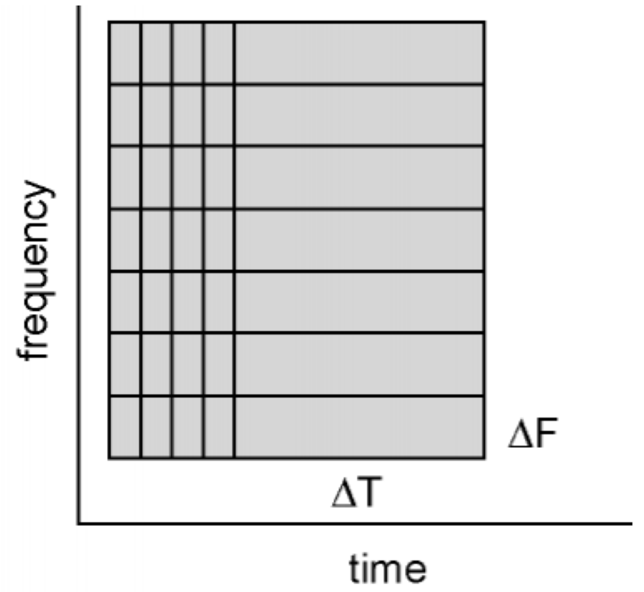

```
cfg = [];
cfg.method = 'mtmconvol';



                .
                .
                .
freq = ft_freqanalysis(cfg,data);
```

The time-frequency plane

The division is ‘up to you’

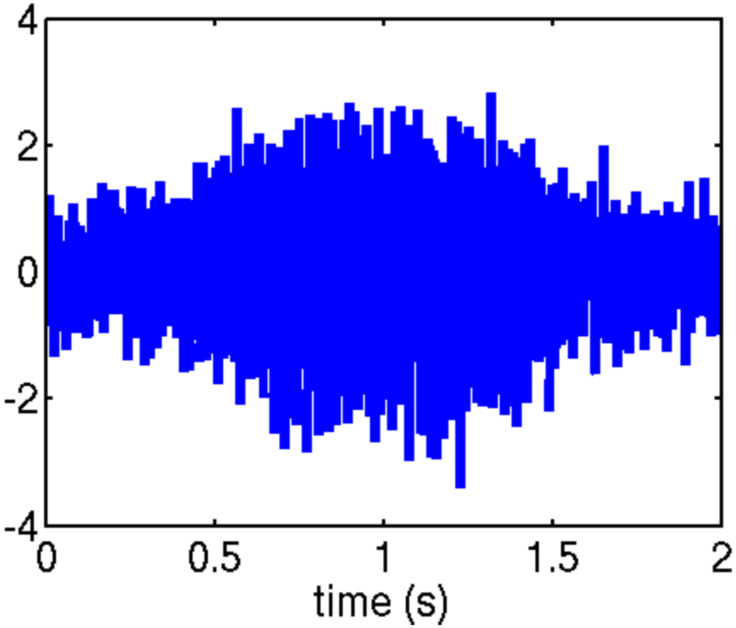
Depends on the phenomenon you want to investigate:

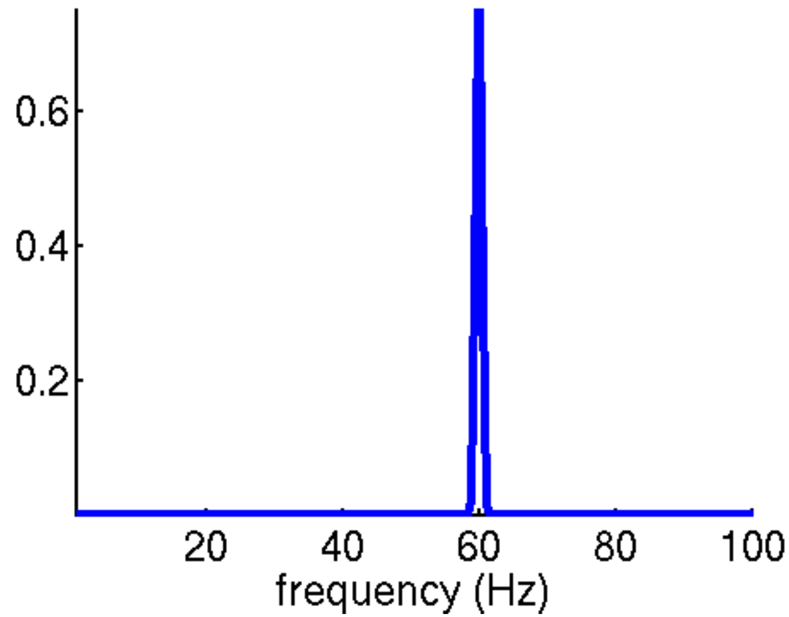
- Which frequency band?
- Which time scale?




```
cfg = [];
cfg.method     = ‘mtmconvol’;
cfg.foi        = [2 4 … 40];
cfg.toi        = [0:0.050:1.0];
cfg.t_ftimwin = [0.5 0.5 … 0.5];
cfg.tapsmofrq = [ 4   4  …  4 ];
          .
          .
freq = ft_freqanalysis(cfg,data);
```

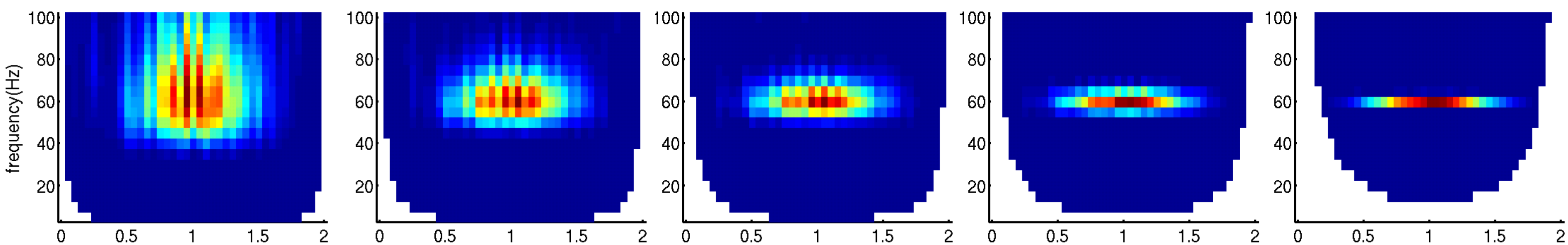
Time versus frequency resolution





short timewindow

long timewindow



Interim summary

Time frequency analysis

- Fourier analysis on shorter sliding time window

Evoked & Induced activity

Time frequency resolution trade off

Wavelet analysis

Popular method to calculate time-frequency representations

Is based on convolution of signal with a family of ‘wavelets’ which capture different frequency components in the signal

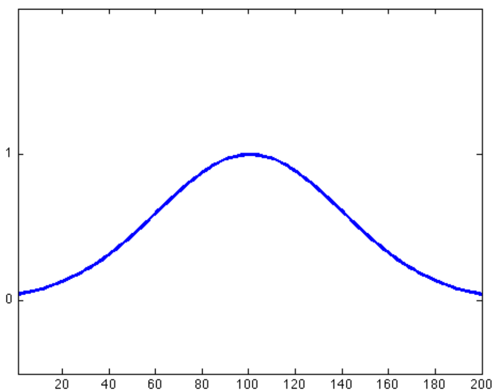
Convolution ~ local correlation

Wavelet analysis

```
cfg  = [];
cfg.method = 'wavelet';
       .
       .
       .
freq=ft_freqanalysis(cfg, data);
```

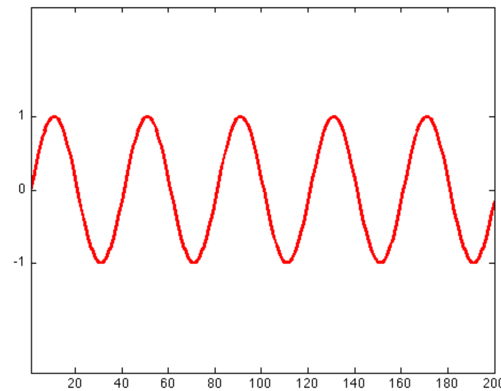
Wavelets

Taper

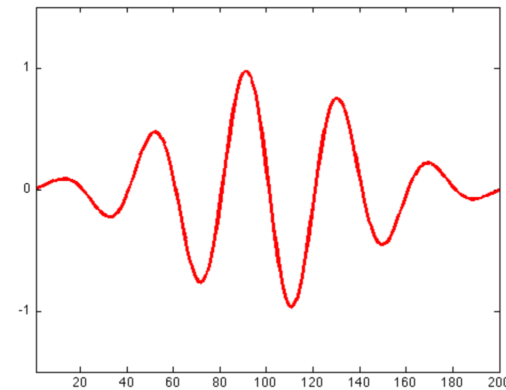


X

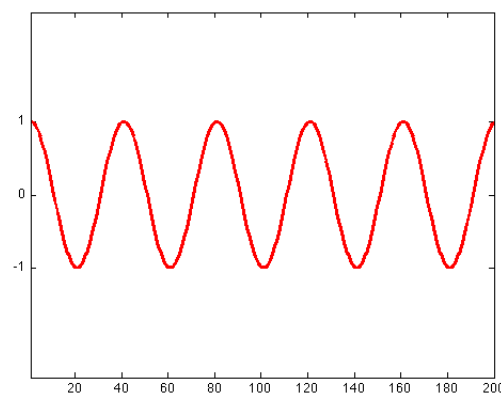
Sine wave



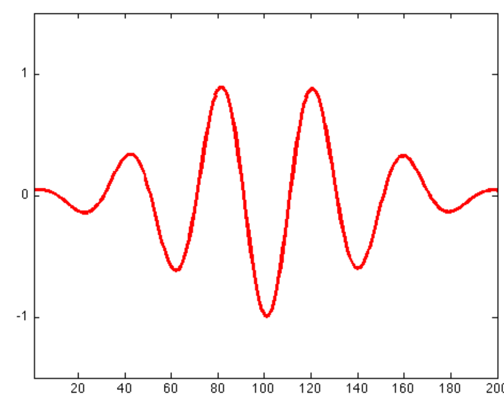
=

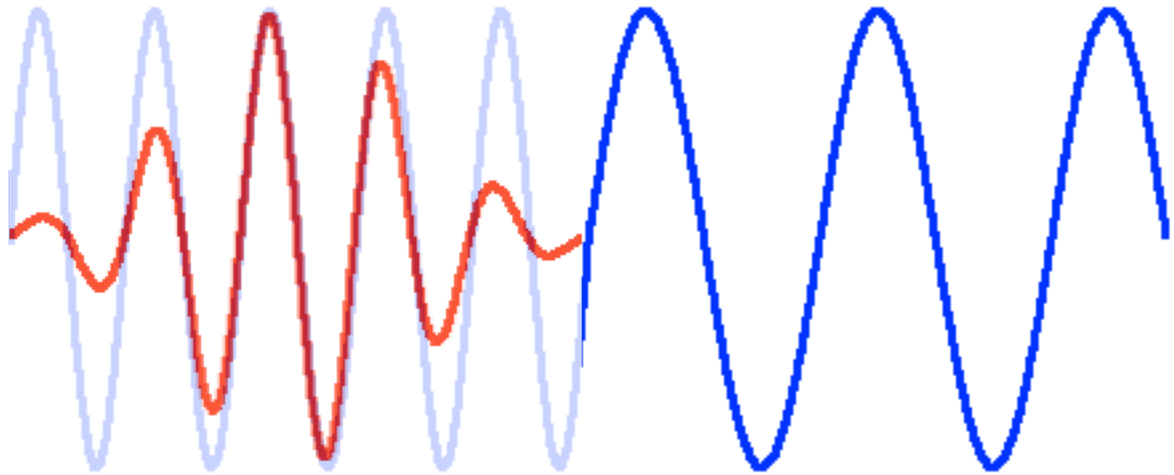


Cosine wave



=





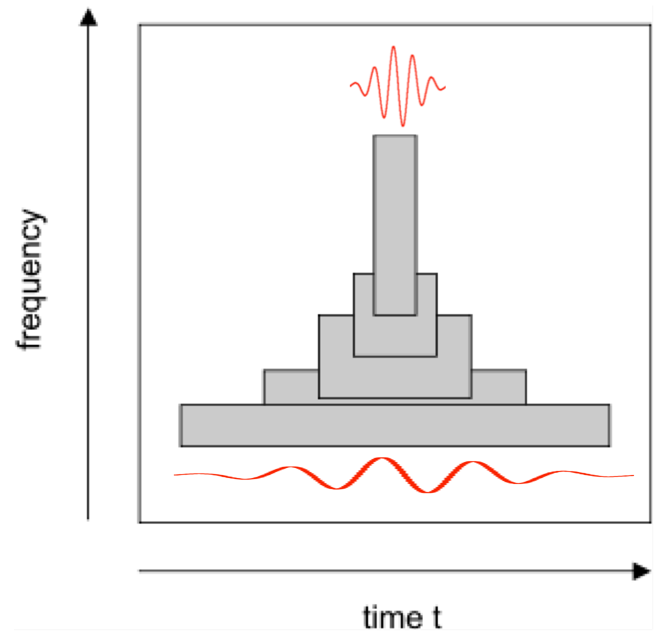
Wavelet analysis

Wavelet width determines the time-frequency resolution

Width is a function of frequency (often 5 cycles)

'Long' wavelet at low frequencies leads to relatively narrow frequency resolution but poor temporal resolution

'Short' wavelet at high frequencies leads to broad frequency resolution but more accurate temporal resolution



Wavelet analysis

Similar to Fourier analysis, but

- Can be computationally slower
- Tiles the time frequency plane in a particular way with fewer degrees of freedom

```
%time frequency analysis with
%multitapers

cfg  = [];
cfg.method     = ‘mtmconvol’;
cfg.toi        = [0:0.05:1];
cfg.foi        = [ 4   8  …  80];
cfg.t_ftimwin = [0.5 0.5 … 0.5];
cfg.tapsmofrq = [ 2   2  …  10];
      .
      .
freq=ft_freqanalysis(cfg, data);
```

```
%time frequency analysis with
%wavelets

cfg  = [];
cfg.method = ‘wavelet’;
cfg.toi    = [0:0.05:1];
cfg.foi    = [4 8 … 80];
cfg.width  = 5;
      .
      .
      .
freq=ft_freqanalysis(cfg, data);
```

Summary

- Spectral analysis
  - Relation between time and frequency domains
  - Tapers
- Time frequency analysis
  - Time vs frequency resolution
- Wavelets

Tomorrow morning: hands-on

- Time-frequency analysis
- Different methods
- Parameter tweaking
- Power versus baseline
- Visualization