Time series & signal processing

BILD 62

Objectives for today

- Identify the types of time series you may encounter in biology
- Implement common signal processing techniques for these time series

Anything recorded continuously over time is a **time series** (a set of data points generated from successive measurements over time)

Commonly encountered time series data in biology

- Gene expression data over time
- Neurophysiology recordings (e.g. electrophysiology, imaging)
- Circadian rhythm data
- Medical observations over time
- Animal movement
- Physiology data (e.g. heart rate/ECG, pulse rate, respiration, etc.)
- Molecules/proteins/cells moving

White blood cell tracking bacteria
Image info

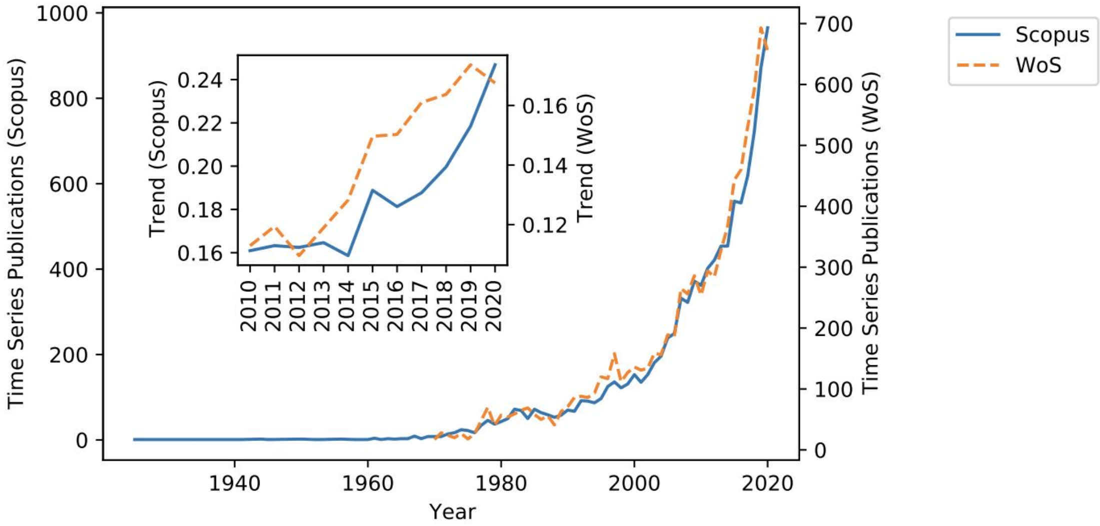


Fig. 1: Number of documents retrieved by Scopus (left axis) and Web Of Science (WoS, right axis) with the search string (restricted to documents titles): "time series" AND ("analysis" OR "data mining" OR "machine learning") over time. The inlet represents the trend (in %) over the last 20 years (the trend is normalized over the total publications in DBLP (the data is accessible in `https://dblp.org/statistics/publicationsperyear.html`).

More and more people developing time series analyses!
(Siebert et al., 2021)

Sample Python packages to work with time series

- BioSPPy
  https://github.com/PIA-Group/BioSPPy
- Obspy (seismology data)
  https://github.com/obspy/obspy
- yasa (sleep data)
  https://github.com/raphaelvallat/yasa
- pastas (groundwater)
  https://github.com/pastas/pastas
- exoplanet (astronomy)
  https://github.com/exoplanet-dev/exoplanet
- PyEMMA (molecular dynamics)
  https://github.com/markovmodel/PyEMMA

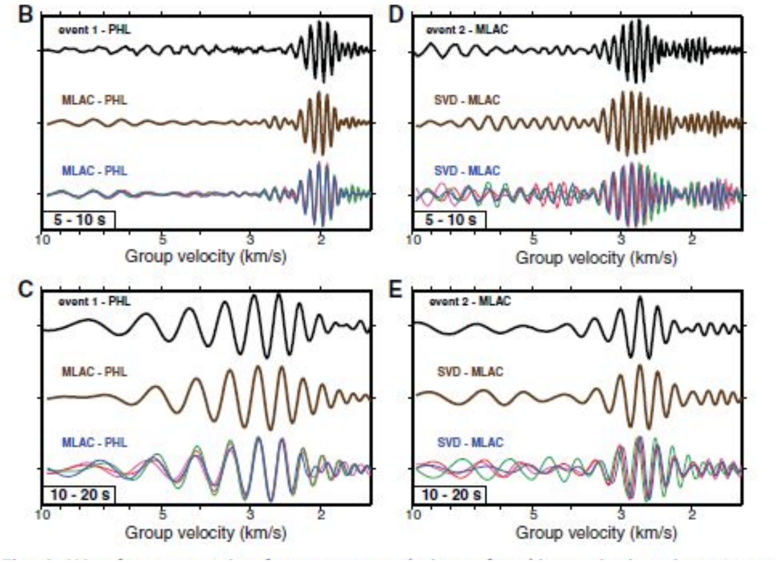
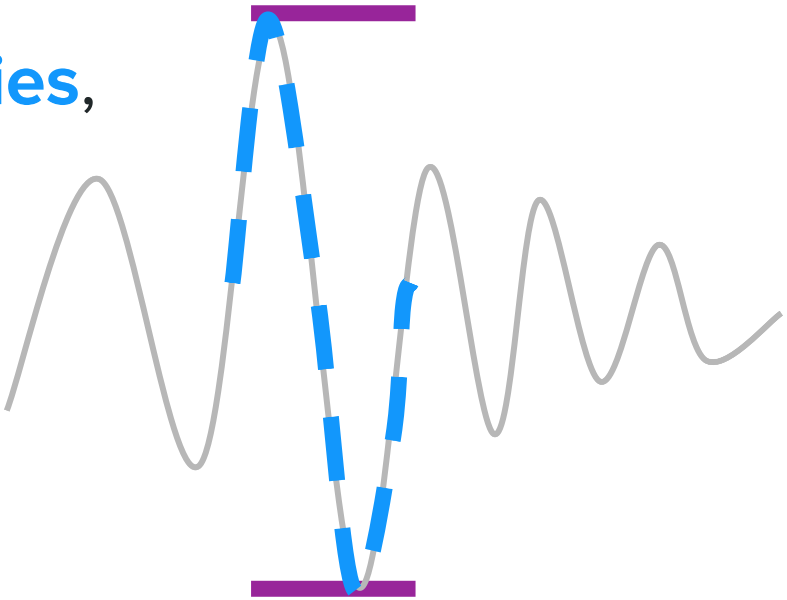


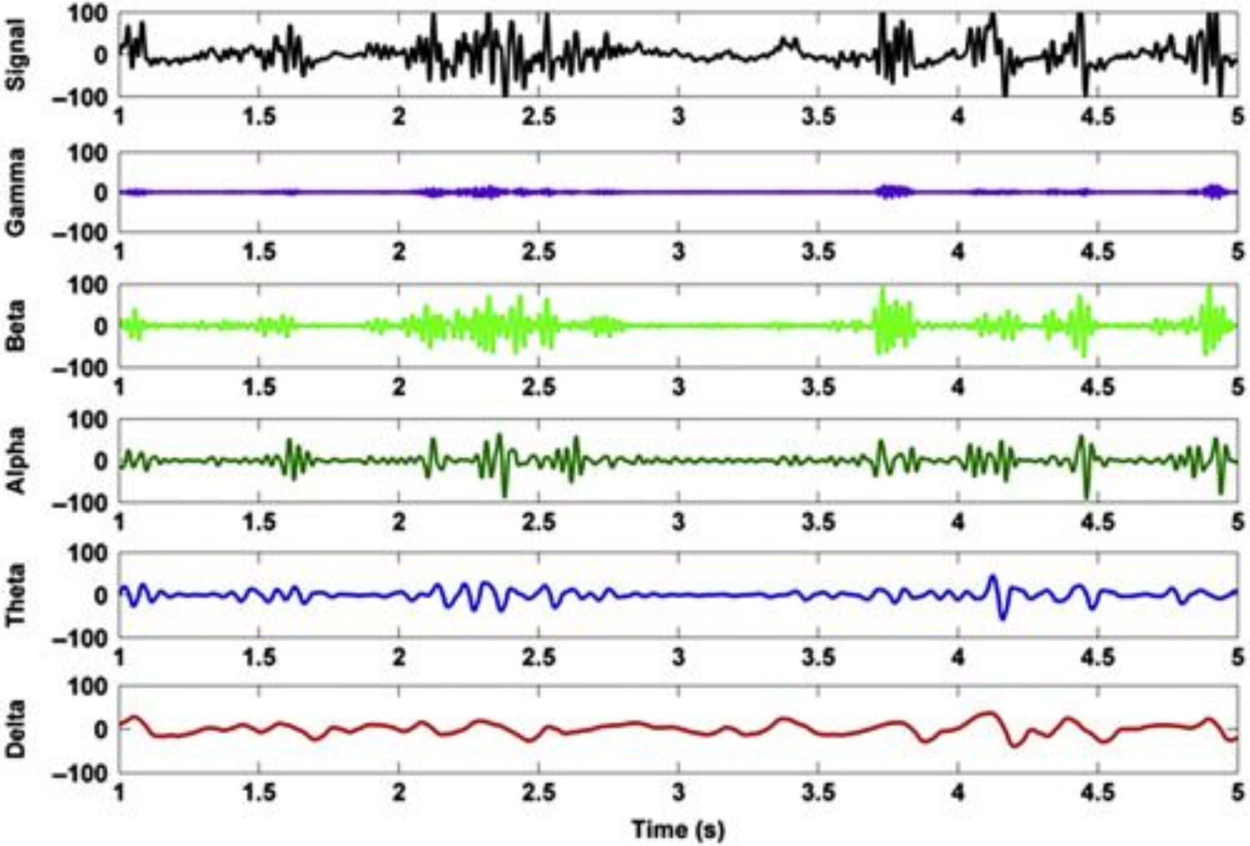
Image from obspy

Common signal processing approaches

- Preprocessing & data cleaning
  - Removing outliers and/or noise
- **Filtering**
  - **High, low, bandpass**
- Looking for correlations in time
- Clustering & classification
- Dimensionality reduction or segmentation
- Prediction
- Anomaly or peak detection

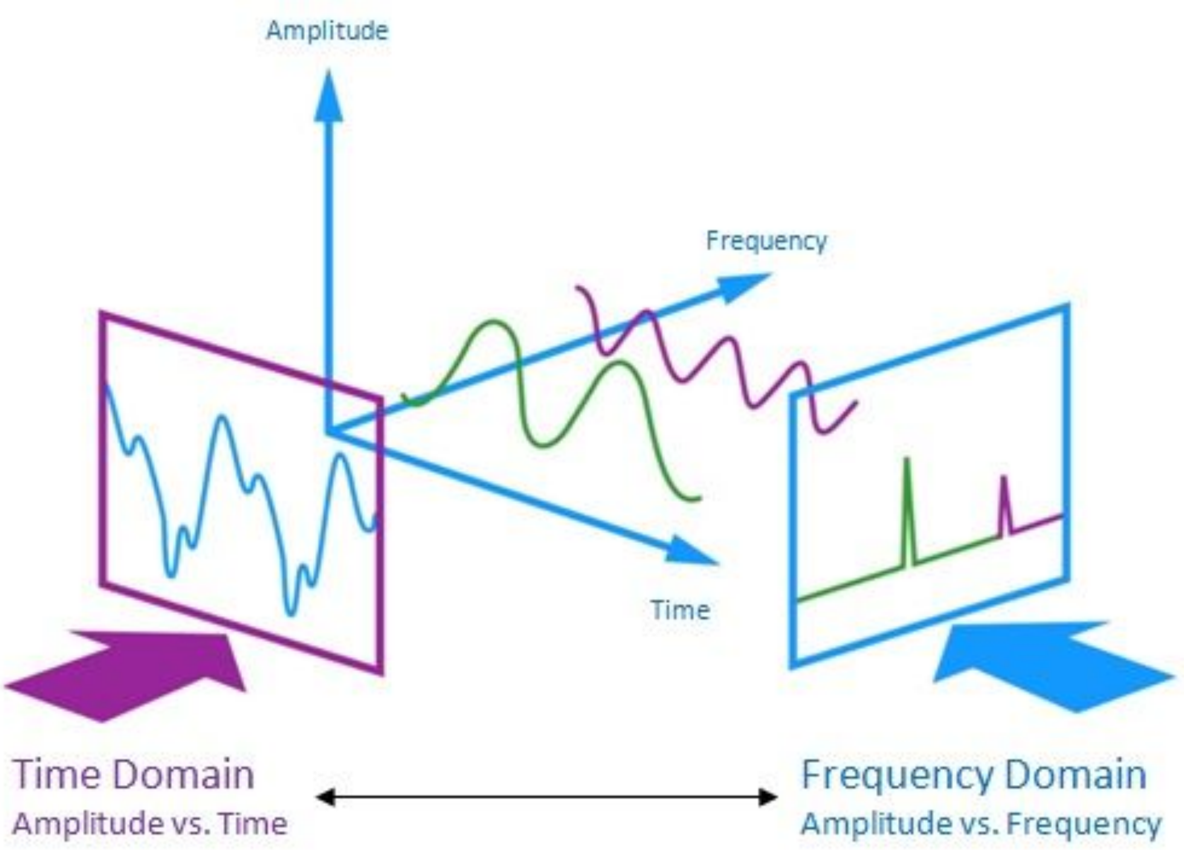
If a signal oscillates, it is convenient to describe **amplitudes of different frequencies**, rather than **amplitudes at different times**.





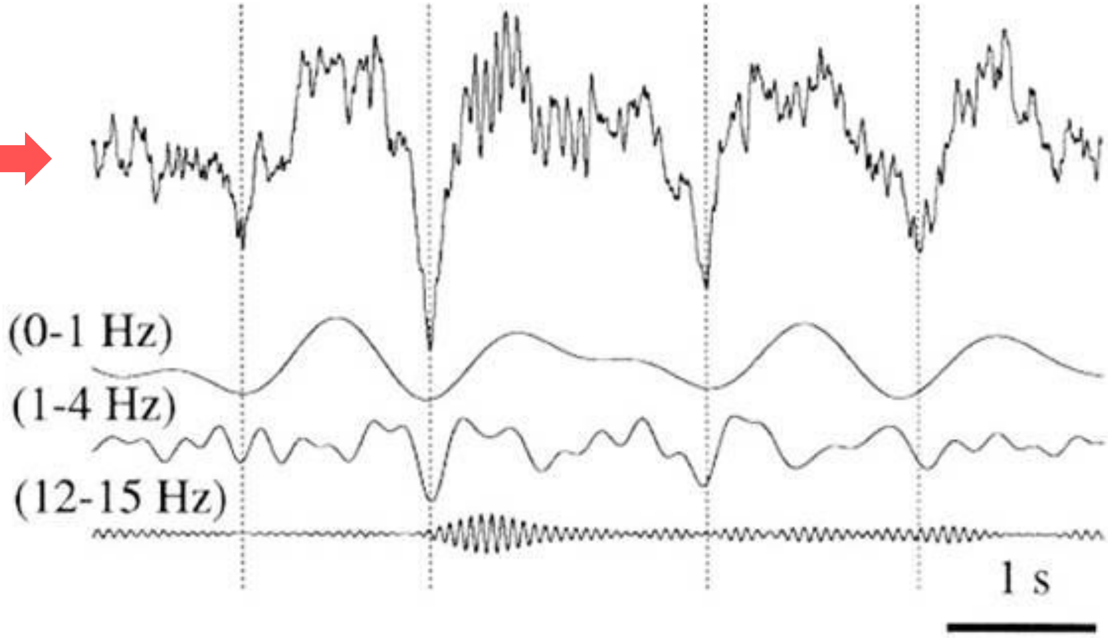
EEG Waves can be broken down into component frequencies

The goal: transform our data from the **time domain** to the **frequency domain**



Fourier transforms can decompose time dependent signals (including EEG), determine the power at specific frequency bands

original EEG signal



Side note: a Fourier transform can also be used for images

Use in x-ray crystallography!

- Crystals have periodic structure!
- So, the Fourier transform of the diffraction pattern will reveal the structure of the scattering object.

For more information:
http://www.ysbl.york.ac.uk/~cowtan/fourier/ftheory.html

Side note: a Fourier transform can also be used for images

Use in

x-ray crystallography

Crystallography uses the diffraction of X-rays, electrons or neutrons by crystals to provide invaluable data on the atomic structure of matter, from single atoms to ribosomes. Much of crystallography's success is due to the software packages developed to enable automated processing of diffraction data. However, the analysis of unconventional diffraction experiments can still pose significant challenges – many existing programs are closed source, sparsely documented, or challenging to integrate with modern libraries for scientific computing and machine learning. Described here is *reciprocalspaceship*, a Python library for exploring reciprocal space. It provides a tabular representation for reflection data from diffraction experiments that extends the widely used *pandas* library with built-in methods for handling space groups, unit cells and symmetry-based operations. As is illustrated, this library facilitates new modes of exploratory data analysis while supporting the prototyping, development and release of new methods.

Recently published tool:

RECIPROCALSPACESHIP: A PYTHON LIBRARY FOR CRYSTALLOGRAPHIC DATA ANALYSIS

Additional Resources

https://mark-kramer.github.io/Case-Studies-Python/03.html

https://voyteklab.com/oscillations/publications/interpreting-spectrum/

Related UCSD classes:

COGS 118C. Neural Signal Processing

DSC 120. Signal Processing for Data Analysis