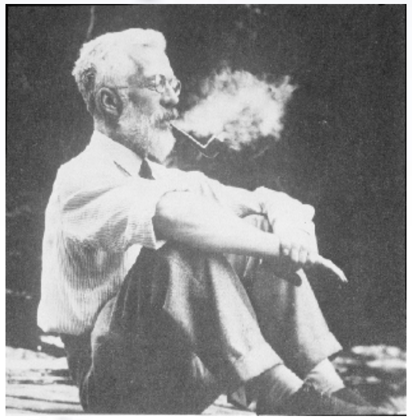
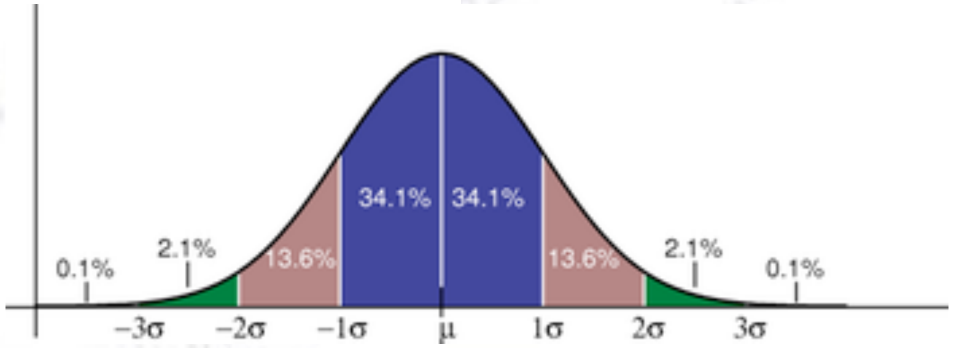
Applied Statistics

Time series analysis

Mathias Luidor Heltberg (NBI)



"Statistics is merely a quantisation of common sense"

Overview

In this introductory lecture we will go through a series of fundamental tools when working with time series. The topics will include:

- Stationarity, de-trending and de-noising
- Signal decomposition and Fourier coefficients
- Phase space analysis

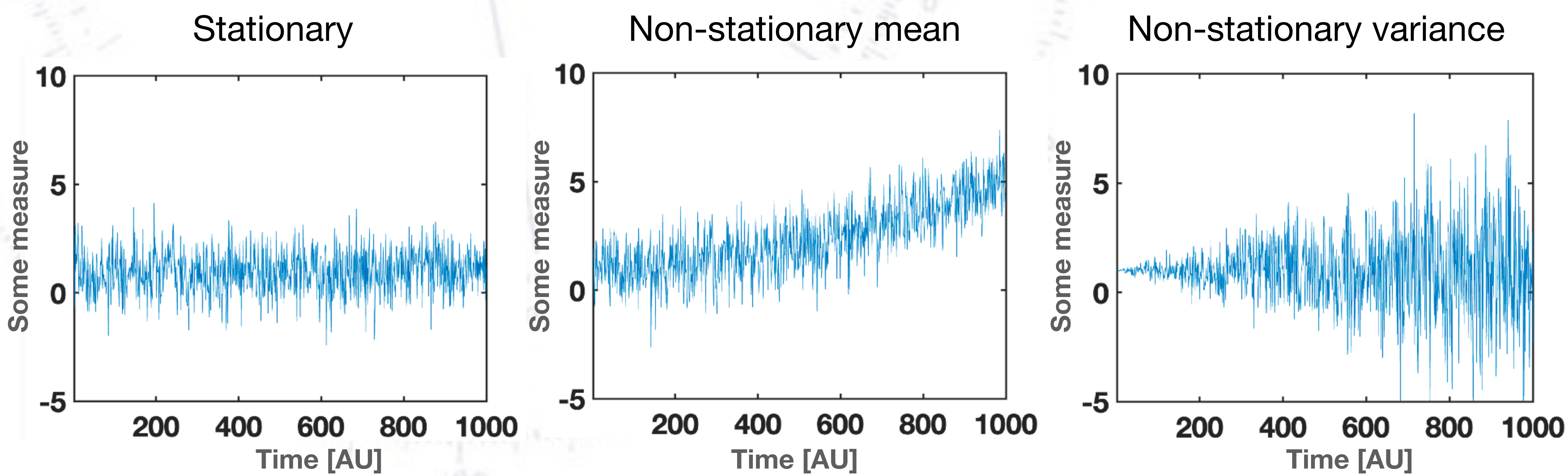
It is highly encouraged that you work on your own with the tools and try to apply them on some data you have at hand.

Stationarity

A time series is defined to be strongly stationary if all points on average are independent of the time of measurement.

Typically we will use the definition of weakly stationary processes. Here the first moment of a signal should be constant and the variance non-infinite.

Stationarity is of fundamental importance to many of the methods we are going to apply.



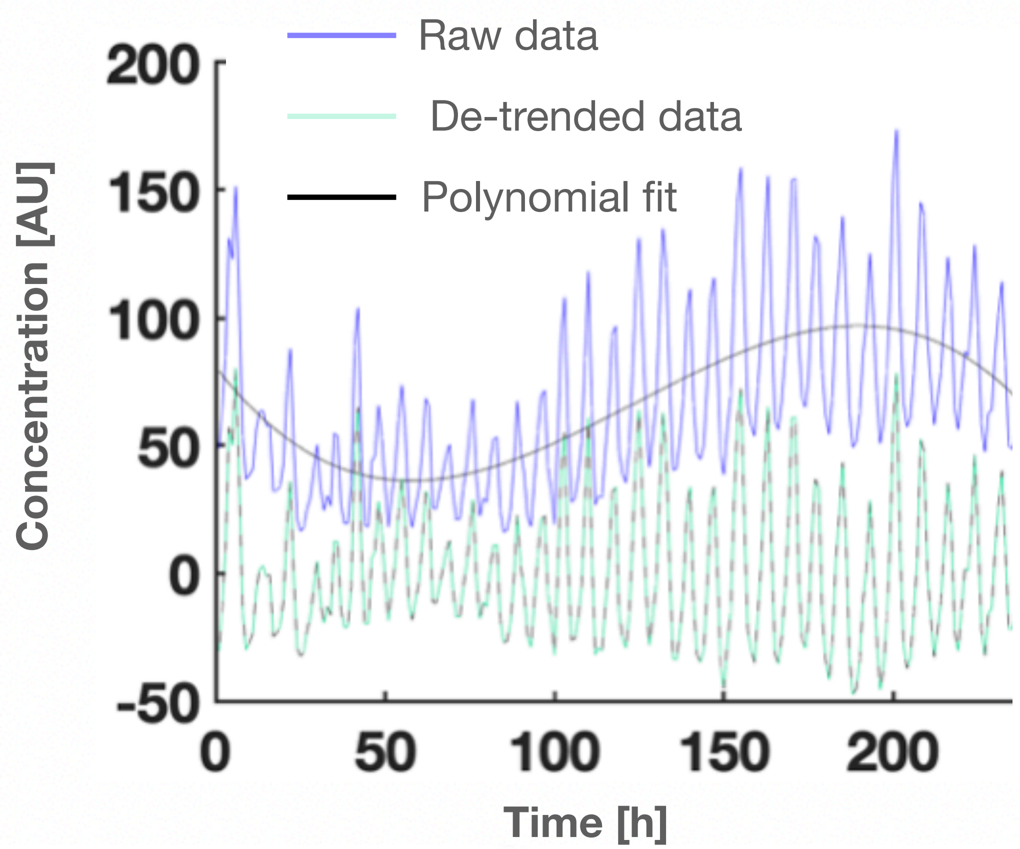
De-trending algorithms

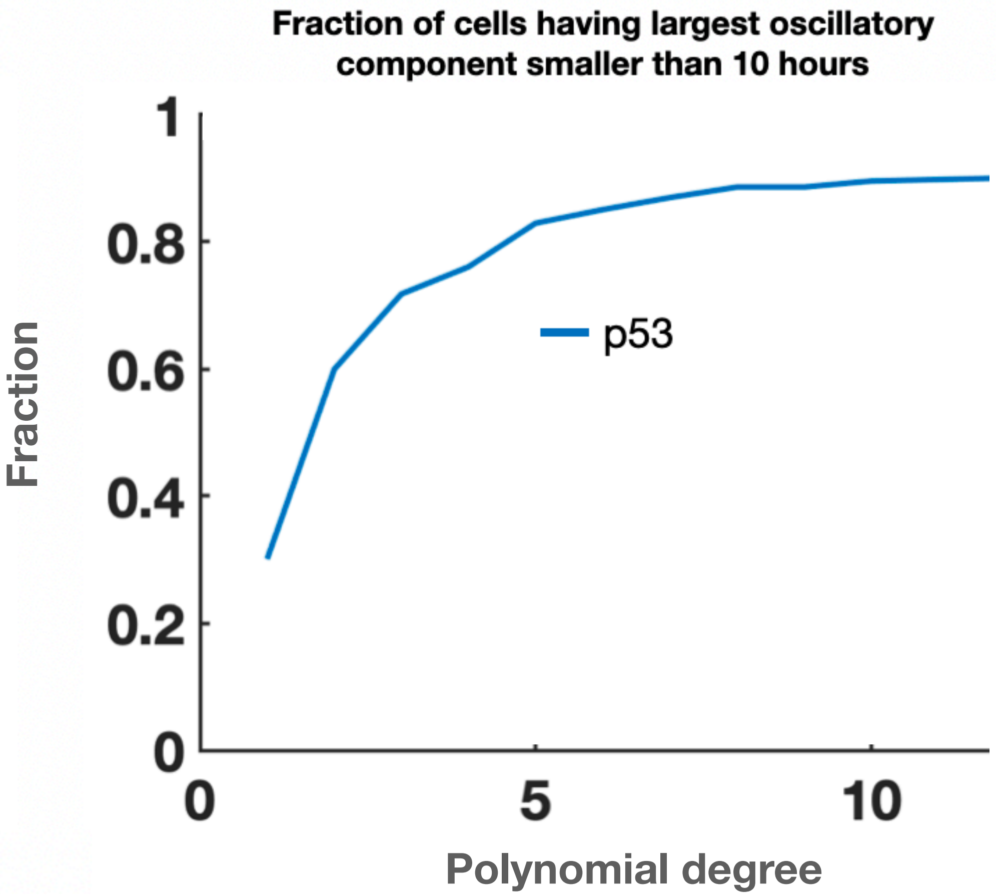
Typically we would like a process to be stationary. This can be done by applying different kinds of filters.

Of particular importance is the Polynomial filter

$$\hat{\vec{\theta}} = (\tilde{\mathbf{C}}\mathbf{V}^{-1}\mathbf{C})^{-1}\tilde{\mathbf{C}}\mathbf{V}^{-1}\mathbf{y}$$

$$\mathbf{C} = \begin{pmatrix} 1 & x_1 & x_i^2 \\ 1 & x_2 & x_2^2 \\ \vdots & \vdots & \vdots \\ 1 & x_n & x_n^2 \end{pmatrix} \qquad \mathbf{V} = \begin{pmatrix} \sigma_1^2 & 0 & \cdots & 0 \\ 0 & \sigma_2^2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sigma_n^2 \end{pmatrix}$$



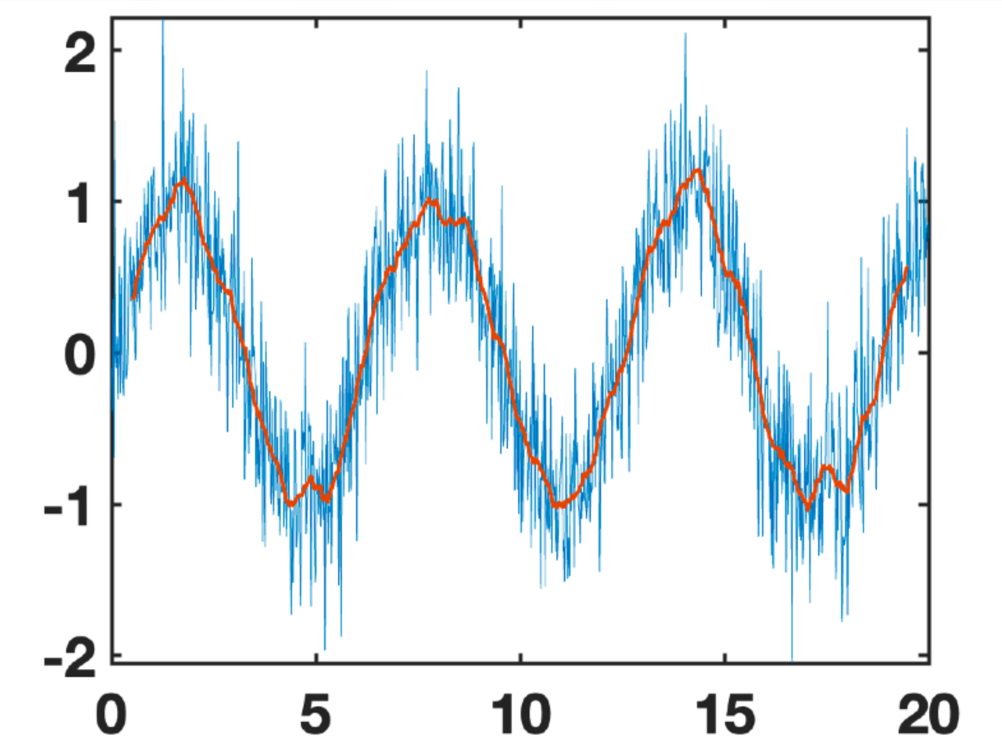


De-noising algorithms

Another aspect of the time series analysis is the subtraction of stochastic noise.

Many tools exist and for simplicity we will introduce one of them: the Salitzky-Golay filter.

Here we simply take a given point, and then perform a polynomial filtering in +/- m data points.



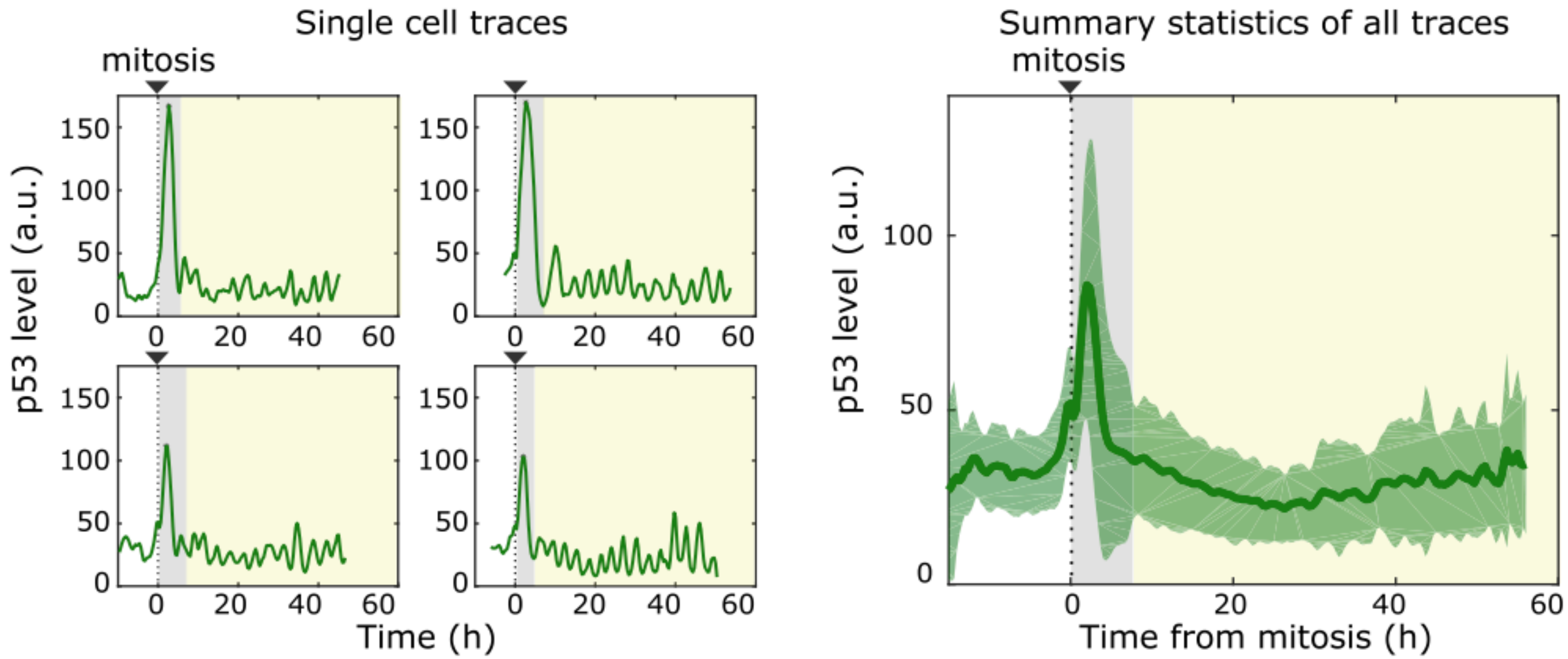
This is done directly through our matrix notation

And as the implemented poly fit, where we typically choose a polynomium of order 3.

Fourier analysis - example

From this inspiration, we can directly apply the Fourier analysis.
All respectable software has this coded, and typically we apply the FFT - Fast Fourier Transform - that estimates the coefficients much faster.

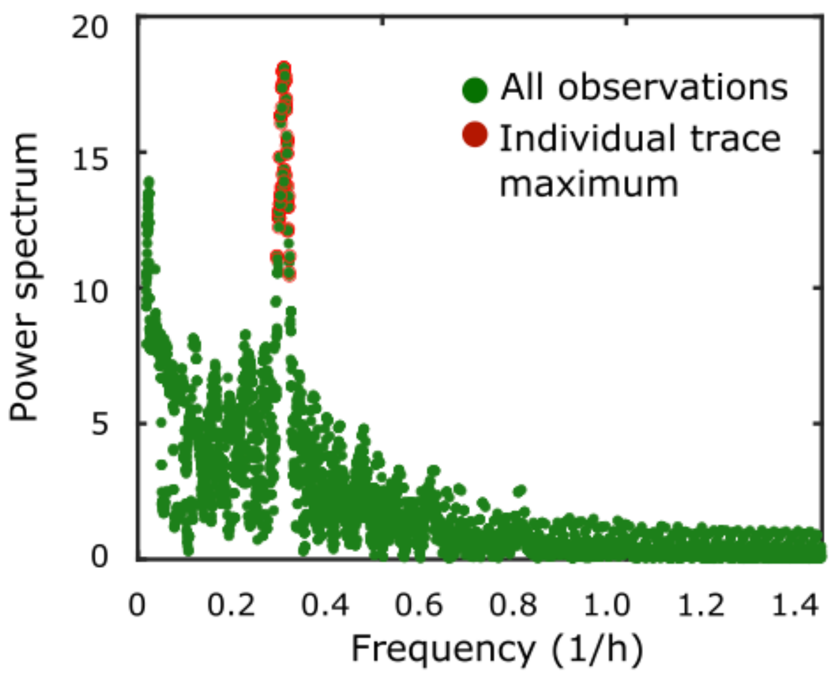
Is we have a dataset of say 100 cells, we can perform an FFT analysis on each separate dataset - NOT their means.

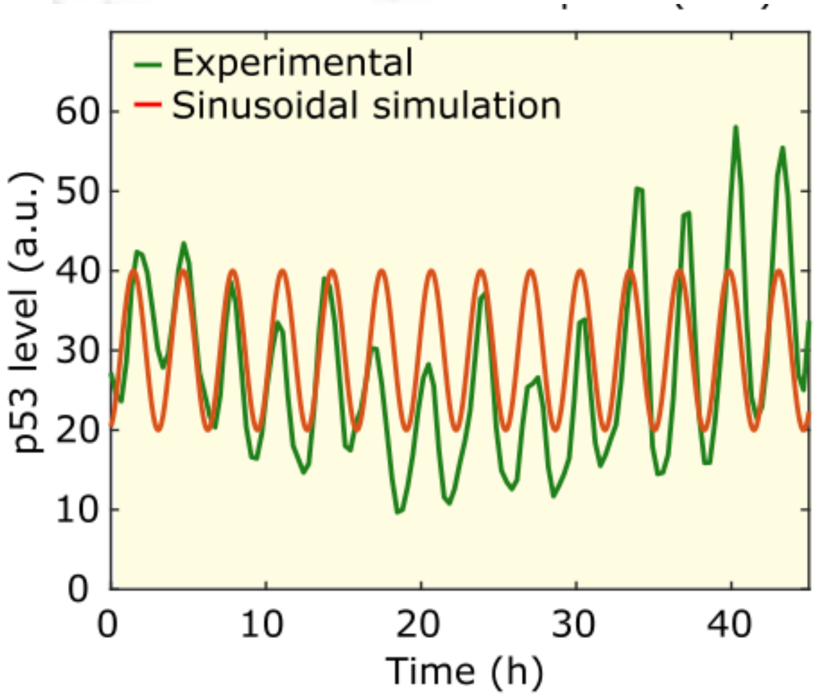


Fourier analysis - example

From this inspiration, we can directly apply the Fourier analysis.
All respectable software has this coded, and typically we apply the FFT - Fast Fourier Transform - that estimates the coefficients much faster.

Is we have a dataset of say 100 cells, we can perform an FFT analysis on each separate dataset - NOT their means.





Fourier analysis

The majority of people have applied the Fourier analysis, but what is it actually?

To give direct insight, we assume to have a time series that will be periodic, we call this f(x). We try to fit it with a periodic function S(x).

$$S_n(x) = A_0 + A_1 cos(x) + ... + A_n cos(nx) + B_1 sin(x) + ... + B_n sin(nx)$$

To find the best fit, we define an error function (ε(x)) and calculate the least square fit (analytically)

$$f(x) = S_n(x) + \epsilon(x) \qquad M = \frac{1}{2\pi}\int_{-\pi}^{\pi} \epsilon(x)^2 dx$$

Taking the derivative of M, with respect to each coefficient, we find:

$$\frac{\partial M}{\partial A_k} = 0 \Rightarrow A_k^* = \frac{1}{\pi}\int_{-\pi}^{\pi} f(x) cos(kx) dx \qquad \frac{\partial M}{\partial B_k} = 0 \Rightarrow B_k^* = \frac{1}{\pi}\int_{-\pi}^{\pi} f(x) sin(kx) dx$$

Which are actually the Fourier coefficients. So it’s simply an analytical way to derive the best oscillatory fit.

Fourier analysis

The majority of … s it actually?

To give direct in … periodic, we call this f(x). We …

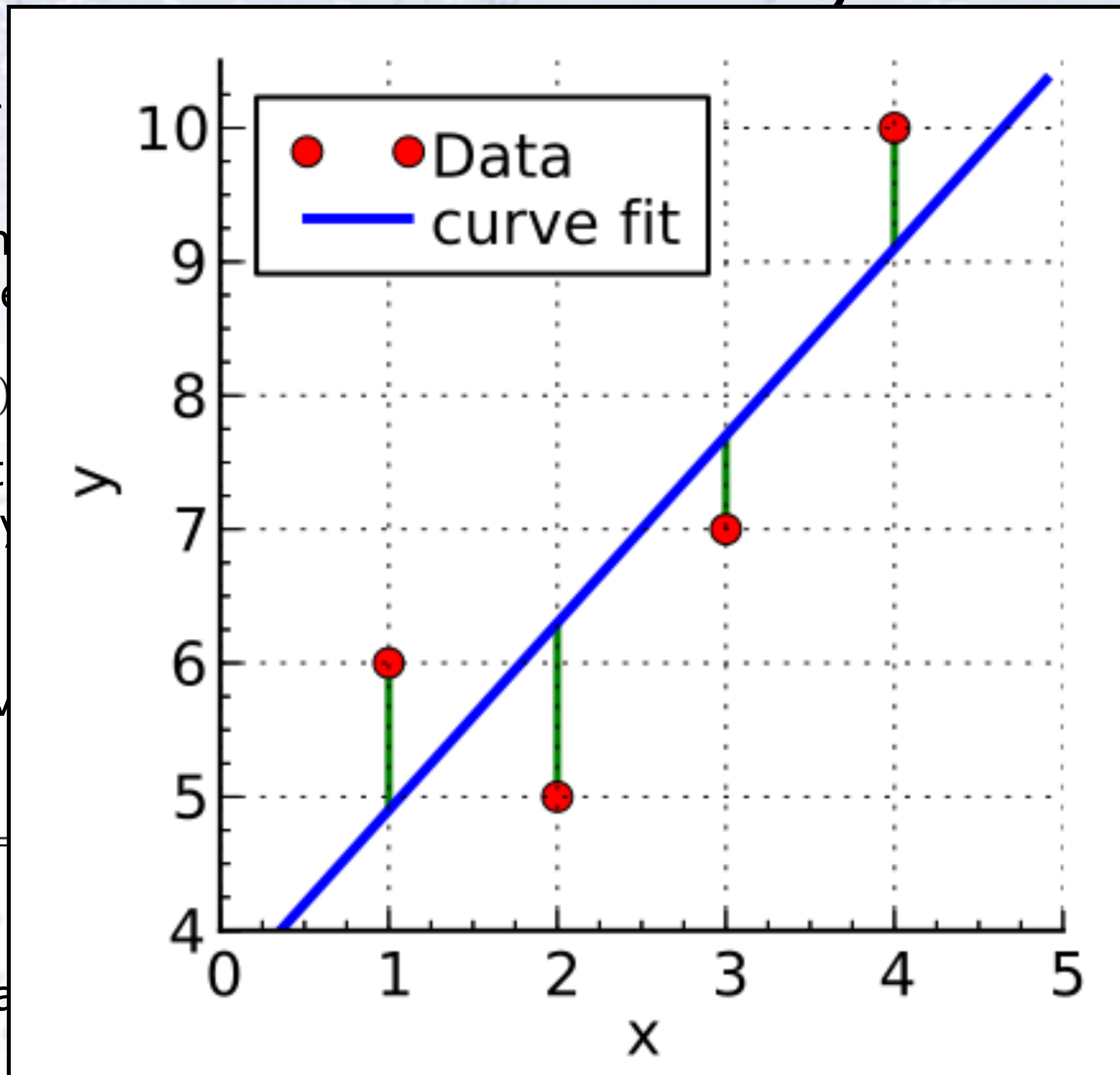
$S_n(x)$ …

To find the best … the least square fit (analy…

Taking the deriv…

$\frac{\partial M}{\partial A_k} =$ … $(x)sin(kx)dx$

Which are actua… …cal way to derive the best …



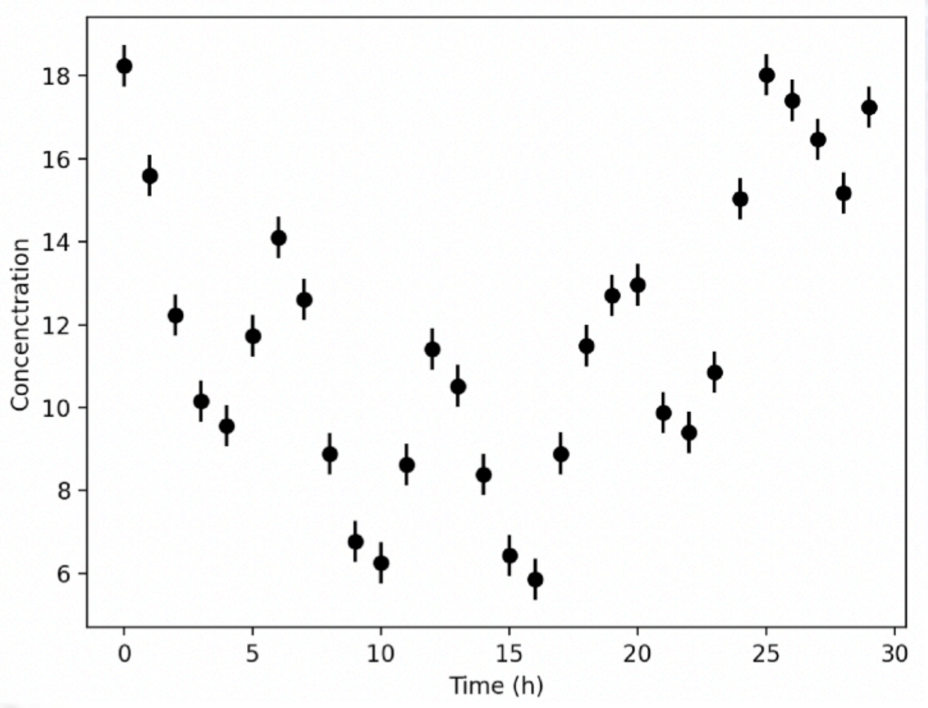
Simple an analytical Chi-square result!!!

Why not just do a least square fit directly?

Many of you might think: With large computers why can't I just plug everything into a fitting tool and obtain parameters?

Assume we have a data series and we want to estimate the oscillations:

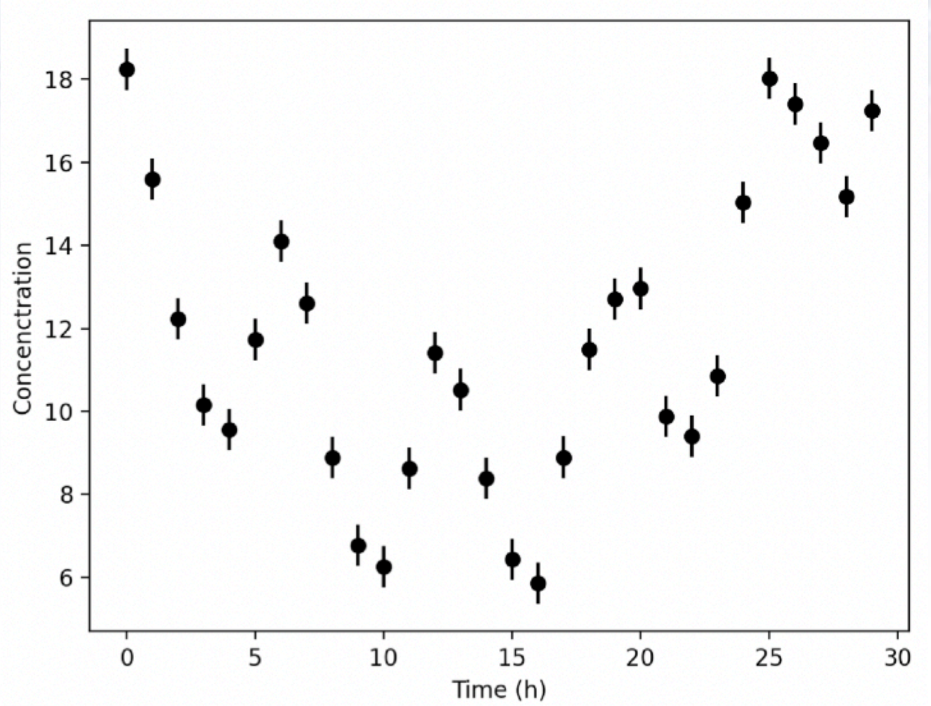
$$f(x) = c_1 + c_2 x + c_3 x^2 + c_4 cos(c_5 x)$$



Lets now assume that we know almost all parameters, except one with a factor 10 uncertainty:

Why not just do a least square fit directly?

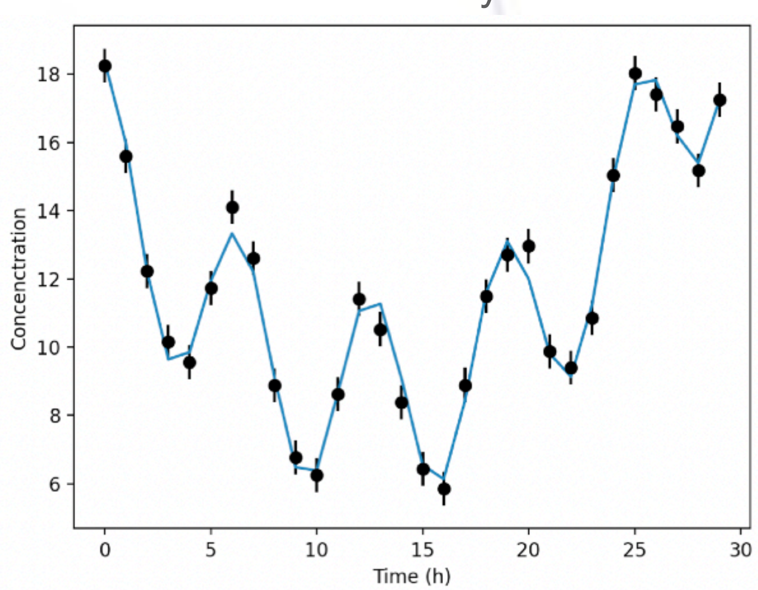
Many of you might think: With large computers why can't I just plug everything into a fitting tool and obtain parameters?

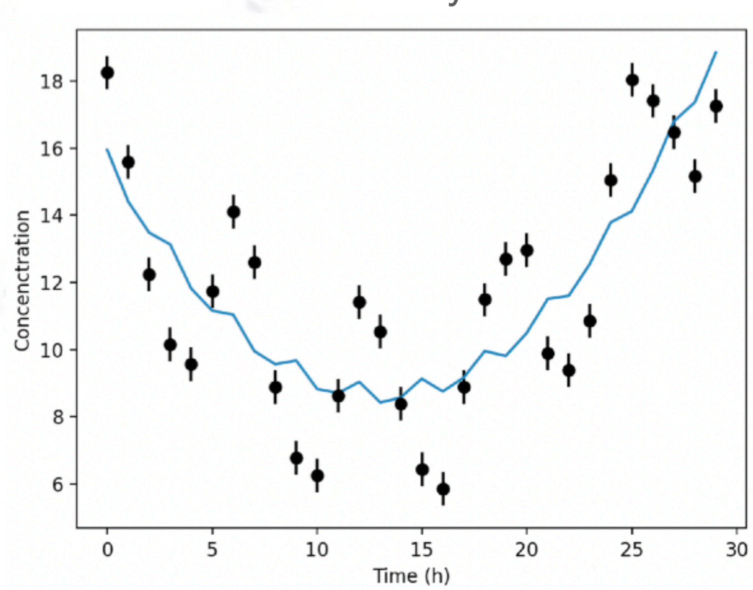


Assume we have a data series and we want to estimate the oscillations:

$$f(x) = c_1 + c_2x + c_3x^2 + c_4cos(c_5x)$$

Lets now assume that we know almost all parameters, except one with a factor 10 uncertainty:





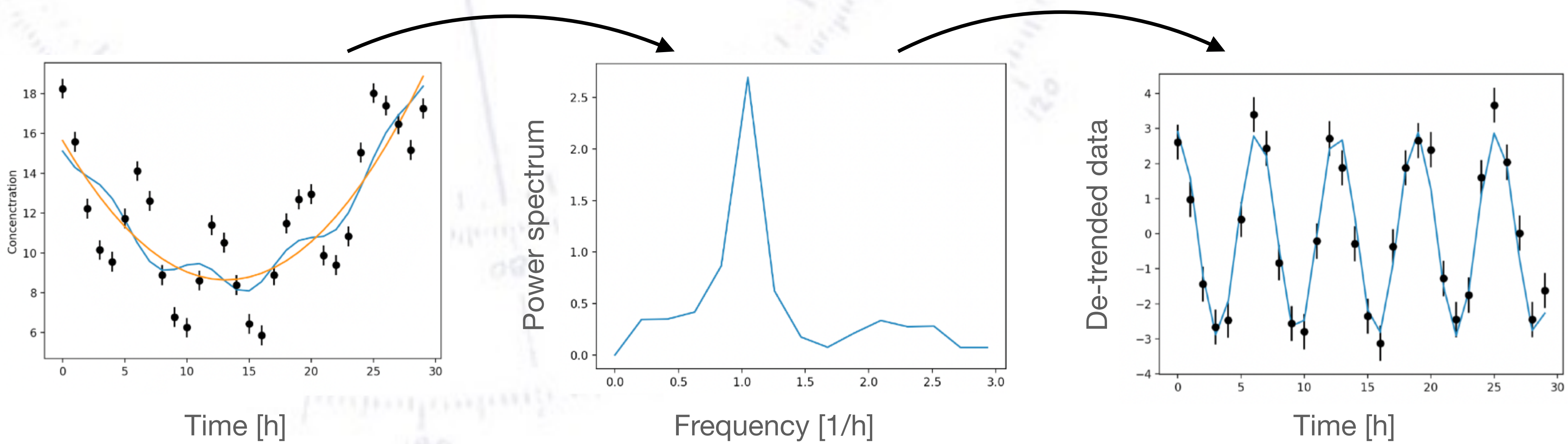
Why not just do a least square fit directly?

Well, if we are interested in the oscillations, why not just apply the Fourier transform in the first place?

Since the data has a mean that is non-constant, and thereby not stationary, the underlying dynamics will dominate the Fourier spectrum.

Therefore we need to subtract this first - and this is done using a polynomial fit.

Finally we always test if the peak of the Fourier spectrum corresponds to the oscillations we are observing.



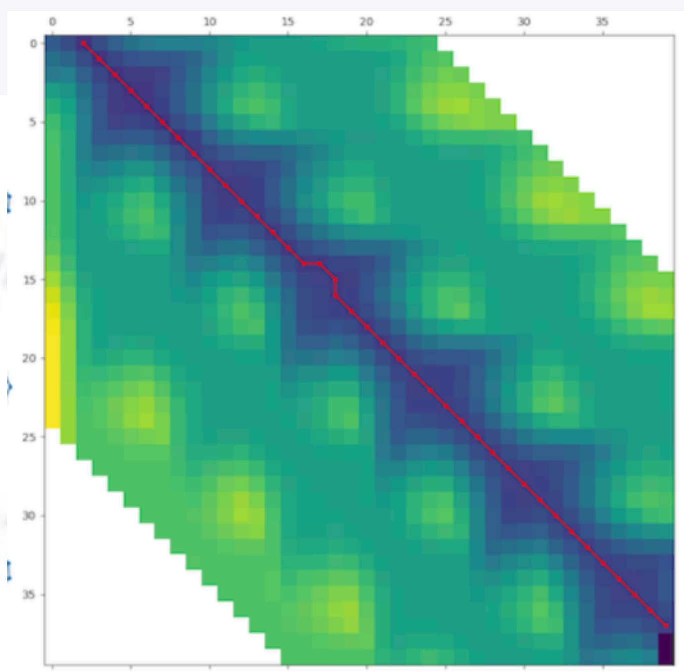
Dynamics Time Warping (DWT)

A very typical problem is the comparing of two data series that definitely seem similar

However comparing their similarities directly will not give any good result.

Instead we compare them using Dynamics Time Warping, where we consider the distances between the value of the points.

```
for i := 1 to n
    for j := 1 to m
        cost := d(s[i], t[j])
        DTW[i, j] := cost + minimum(DTW[i-1, j  ],    // insertion
                                    DTW[i  , j-1],    // deletion
                                    DTW[i-1, j-1])    // match
```

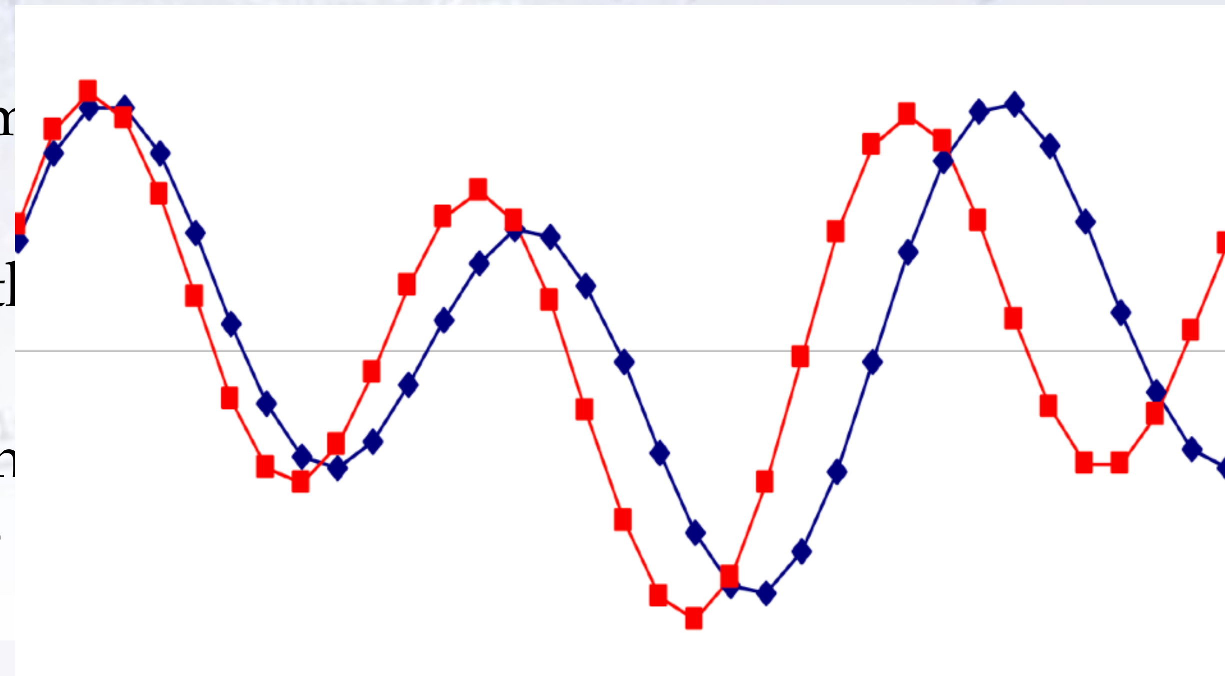


Dynamics Time Warping (DWT)

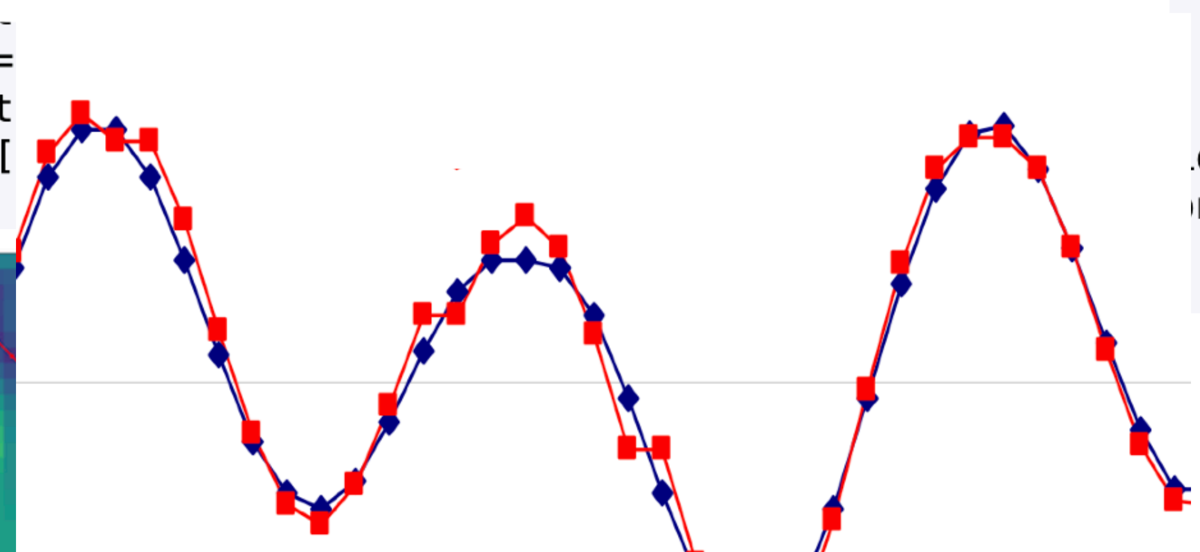
A very typical problem ... tely seem similar

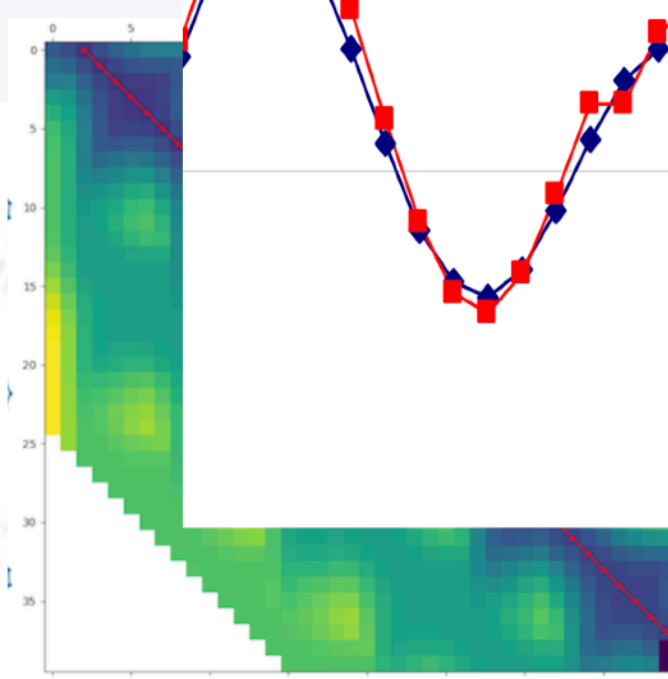
However comparing t ... result.

Instead we compare th ... consider the distances between the



```
for i := ...
    for j := ...
        cost ...
        DTW[ ...                                         ...on
                                                          ...n
```





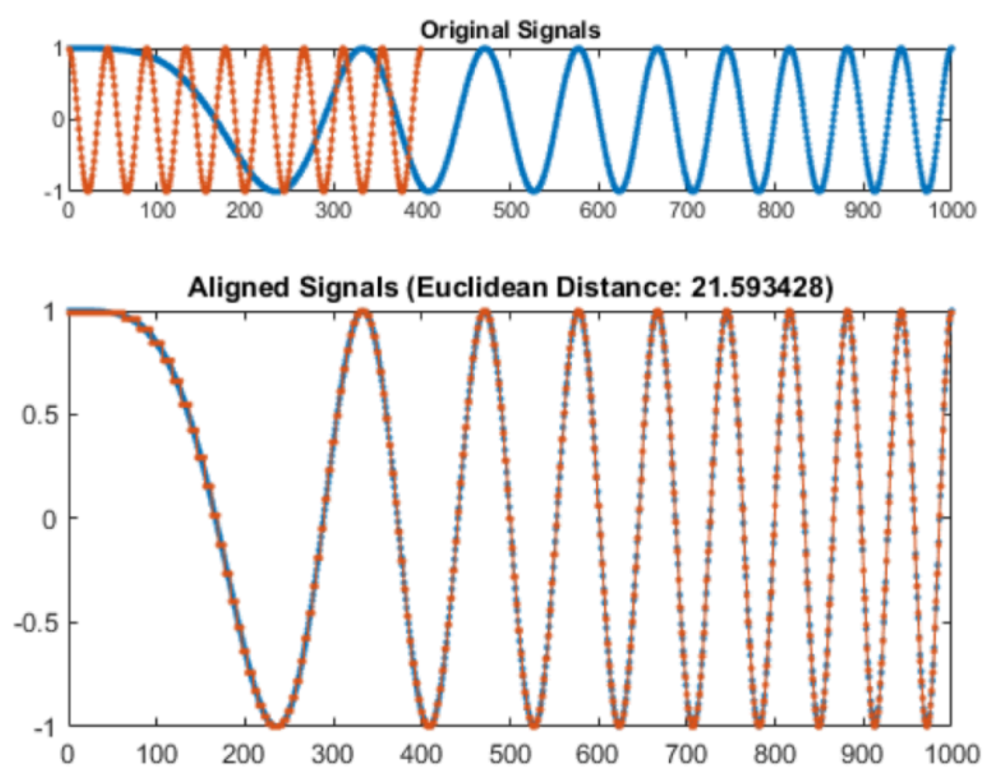
Dynamics Time Warping (DWT)

This corresponds to creating a matrix calculating all distances between points.

Now in order to produce the optimal alignment, we now generate a path in this “distance” space.

Here we have some rules, for instance that each component can only jump 1 step in the warping path. This ensures continuity of the path.

Softwares are available for optimising these calculations for instance in DTW in Python



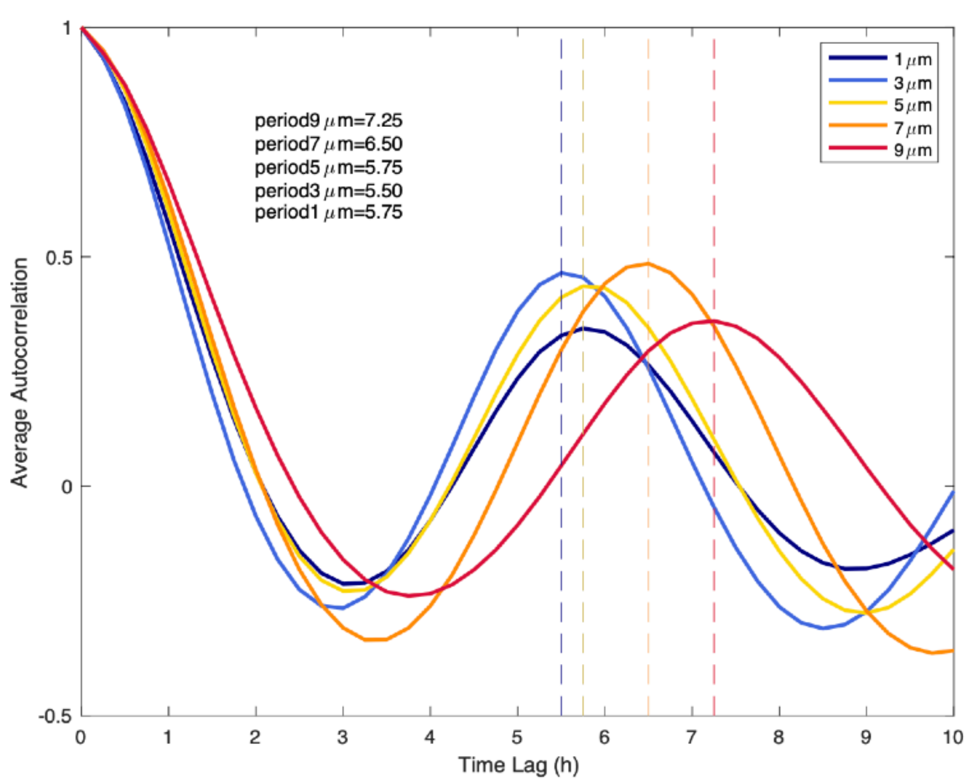
Autocorrelation

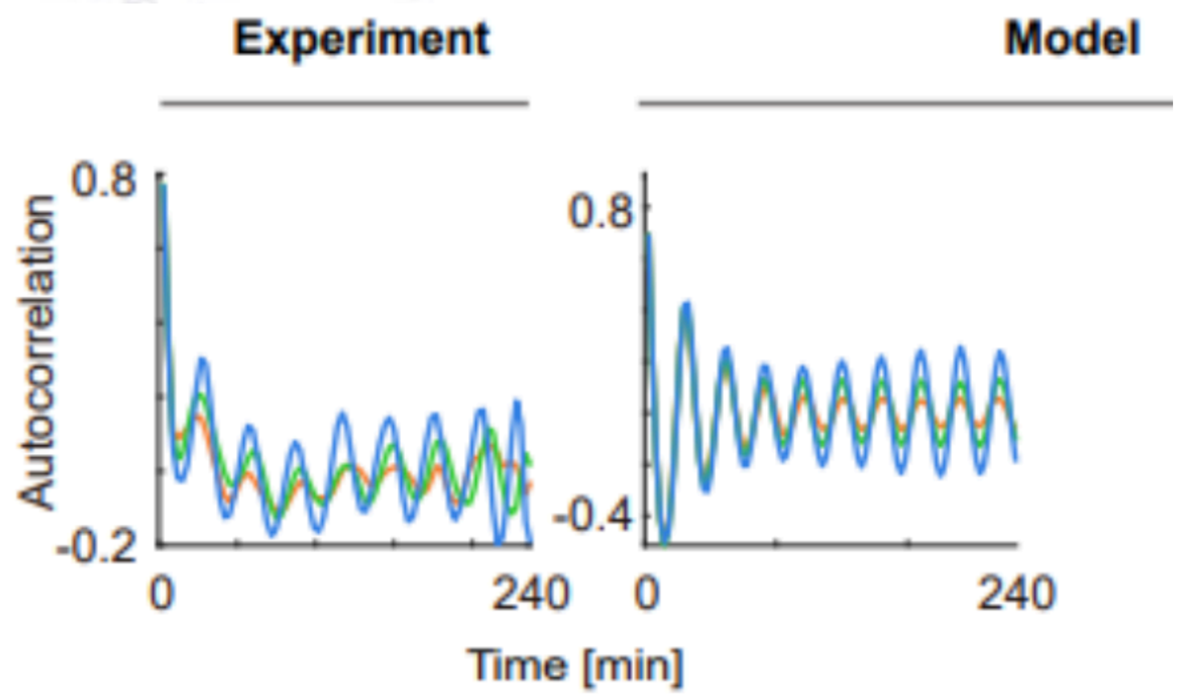
Lets say we want to compute the periodicity of some signal.

We can directly see at which points data is correlated, by calculating the Pearson correlation coefficient at future time points.

By doing this and increasing the time addition we evaluate correlation in, we obtain an autocorrelation.

We can do this across different conditions and compare the resulting periodicity.



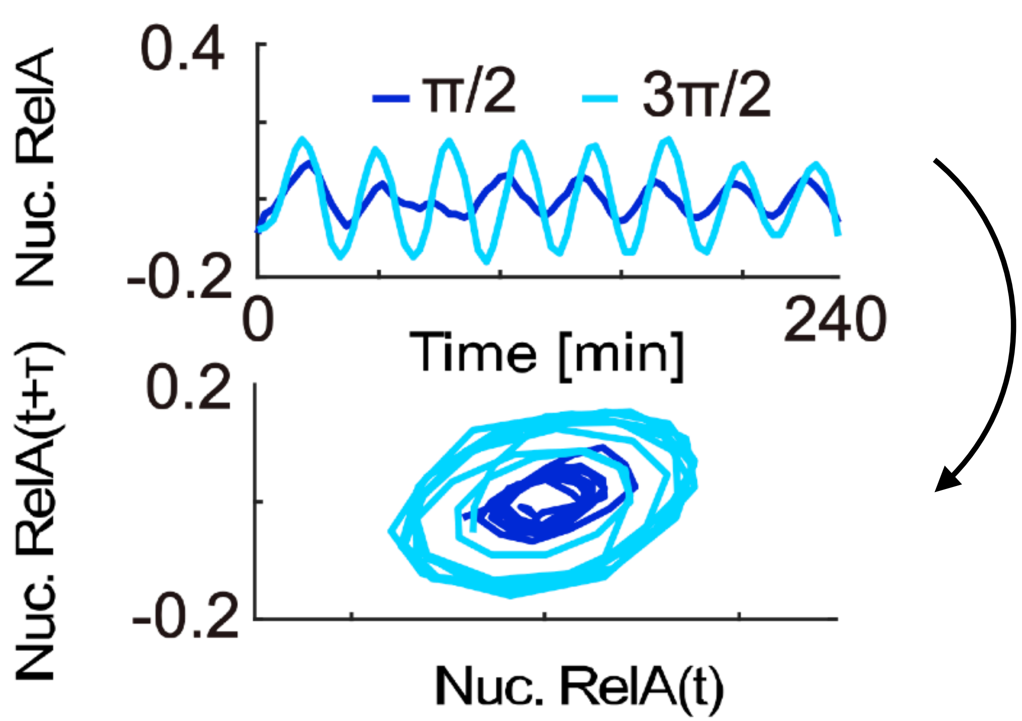


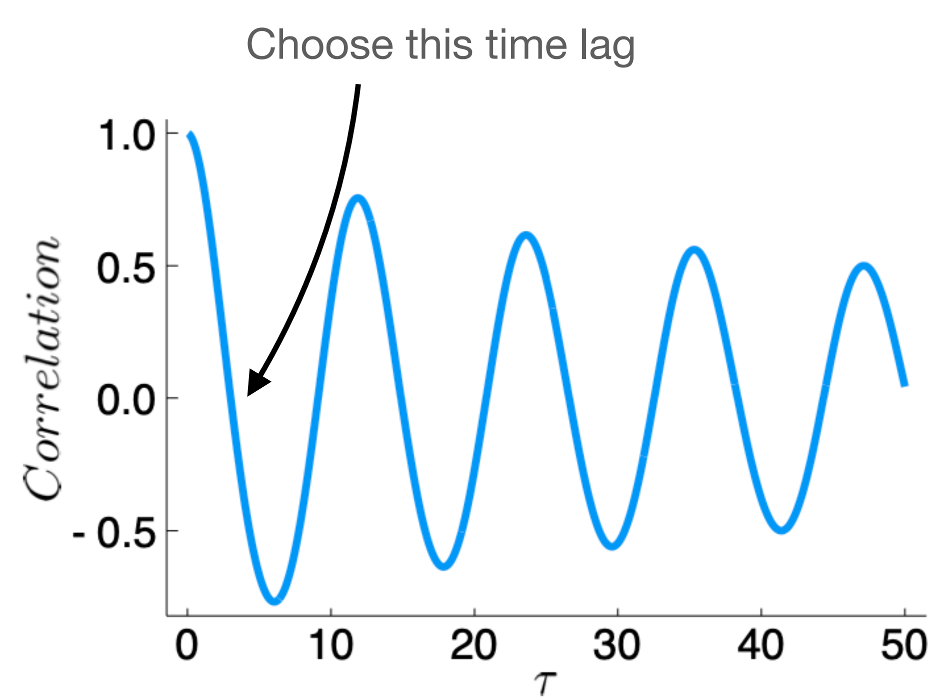
Time embedding

Suppose we have a time series, but we would like to obtain information about the geometry of the phase space.

Then we can apply a technique known as time embedding. Here we basically take a time series from x(1:end-L+1) and x(L:end) - and this can be extended to any dimension.

Building on a result known as "Taken's theorem", the embedded attractor will have the same properties as the underlying network in many dimensions.





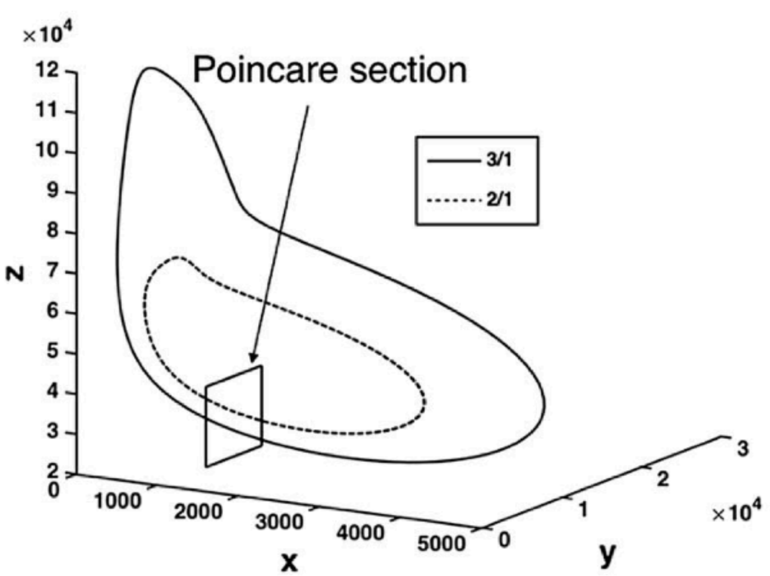
Poincare section

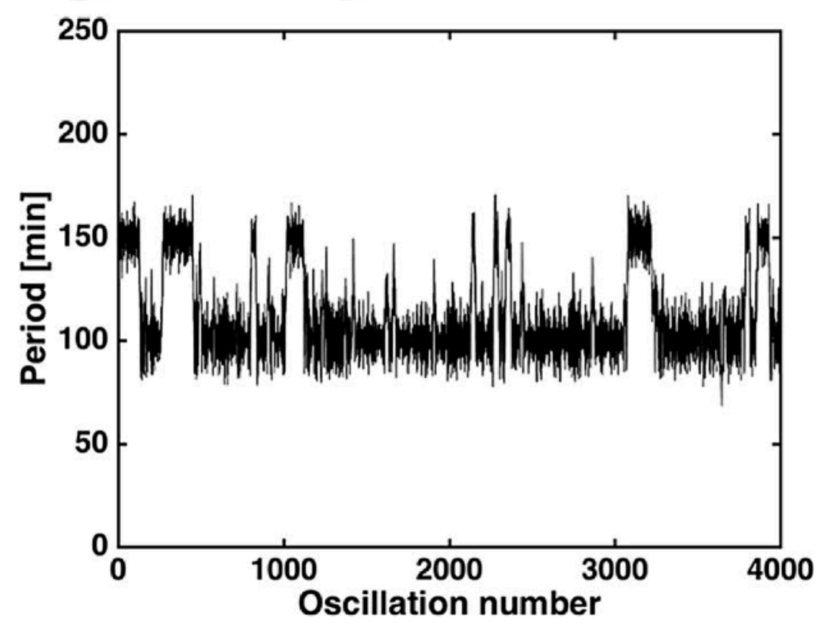
A Poincare section is has dimension n-1, where n is the dimensionality of the data.

If for instance the data is in a 3 dimensional phase space, the Poincare section is a plane that is situated at some location (after your choice) where the trajectory passes through.

In this way, one can study recurrent dynamics, and investigate this in a discrete time series.

This can for instance be applied when studying the motion between two limit cycles, where the transitions might occur in the middle of a rotation.





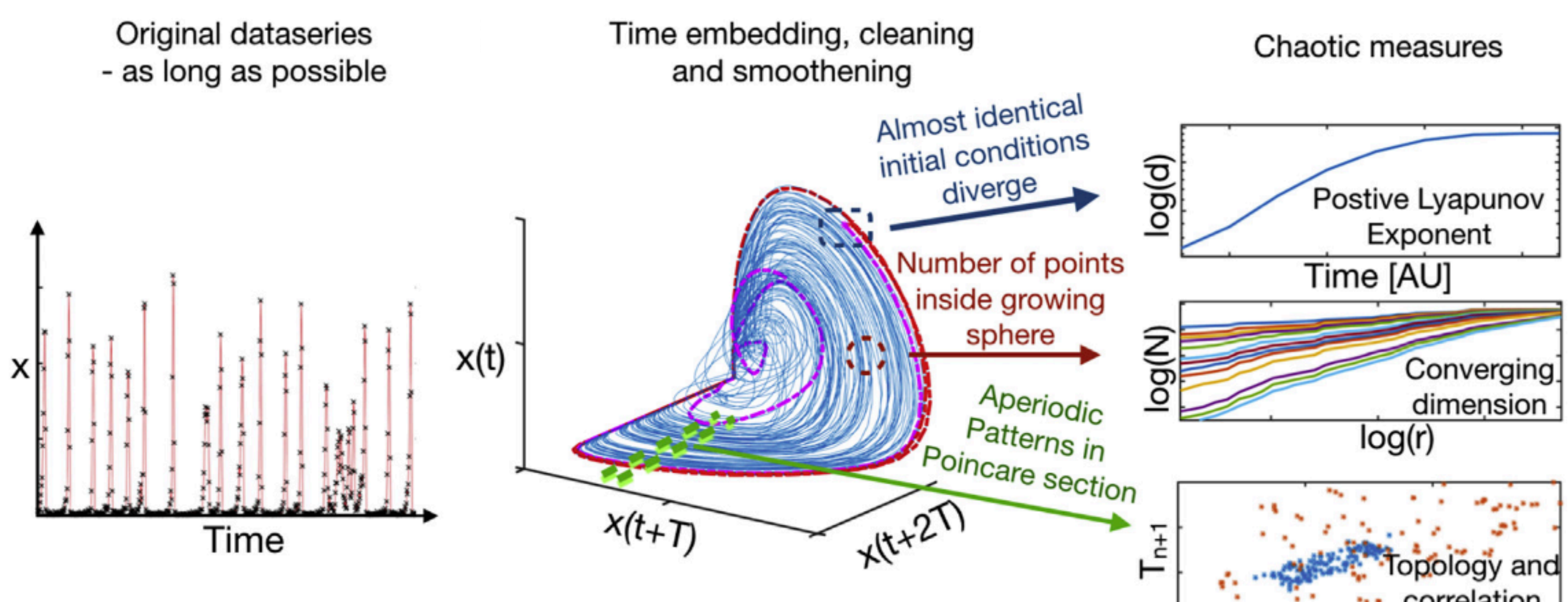
Challenge: Chaotic dynamics

When we look for dynamics, we typically investigate the possibility of oscillations.

However more complex dynamics exist that is related to the features of oscillations.

In particular we want to mention the possibility of chaotic dynamics. This is defined by the fact that two trajectories, starting infinitely close to each other diverge in time.

This is difficult to access, when we cannot perform two experiments starting from exactly same conditions - however methods exist to estimate this.



Summing up

In this brief lecture we have covered some fundamental elements in data analysis of time series:

1) Investigate stationarity - possibly append polynomial fitting

2) Investigate oscillatory components - apply Fourier analysis

3) Consider if de-noising of data could strengthen the analysis

4) When comparing oscillatory signals - align the signals using dynamic time warping

5) Investigate the phase space of the signal using time embedding