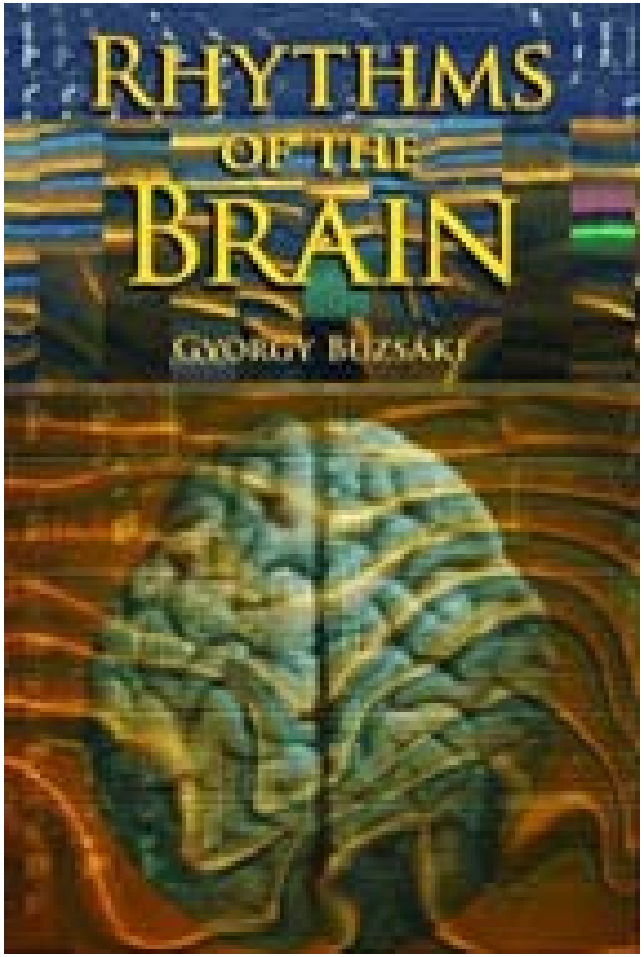
# Computational Neuroscience [Tutorial]

# Objective of the Tutorial

- “Oscillations” is an *hot* topic in neuroscience
- The importance of the role played by oscillations has been made clear by several studies
- However the exact role of oscillations in stimulus coding is still not well understood: which are the important rhythms? Which neural phenomenon corresponds to which frequency?

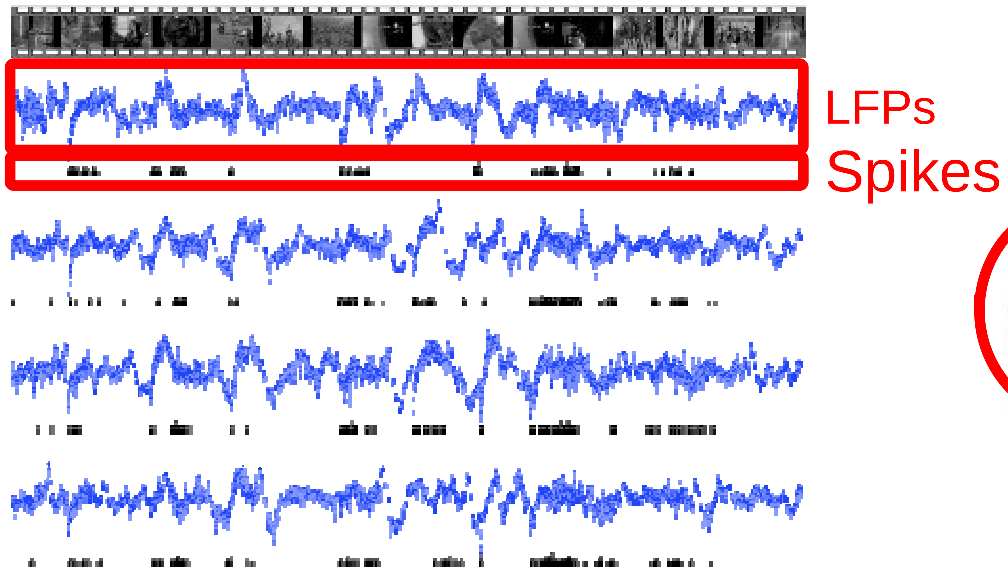
# Objective of the Tutorial

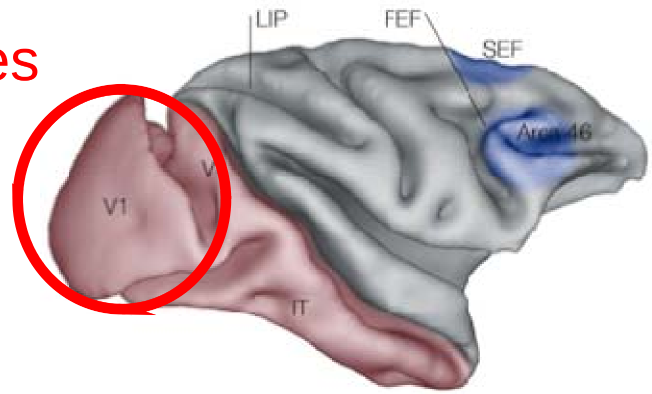


# Objective of the Tutorial

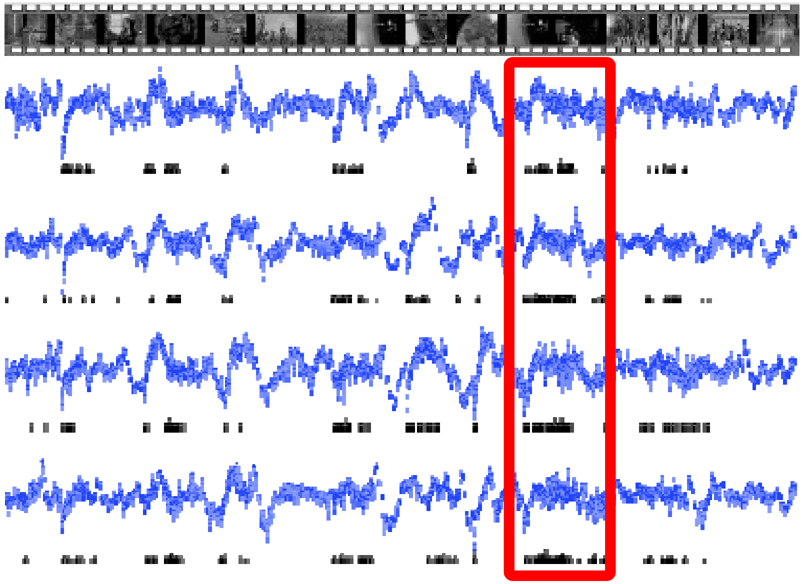
- In this tutorial we will focus on the analysis presented in *Belitski et al. Journal of Neuroscience 2009*
- The aim of this work was to understand which are the most important rhythm in the cortex from a decoding point of view
- The goal is to provide the fundamental tools for to reproducing the same type of analysis on your own / on your data

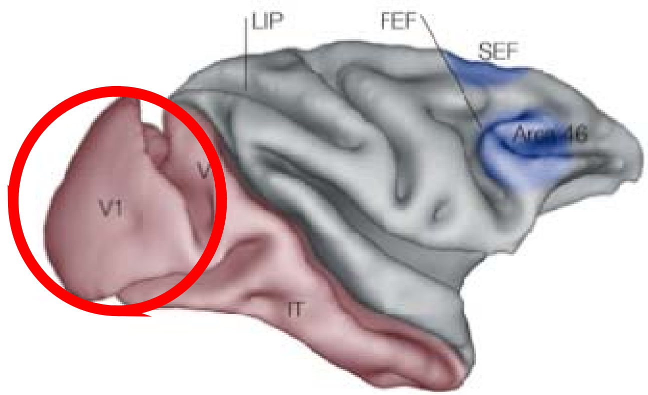
# Objective of the Tutorial



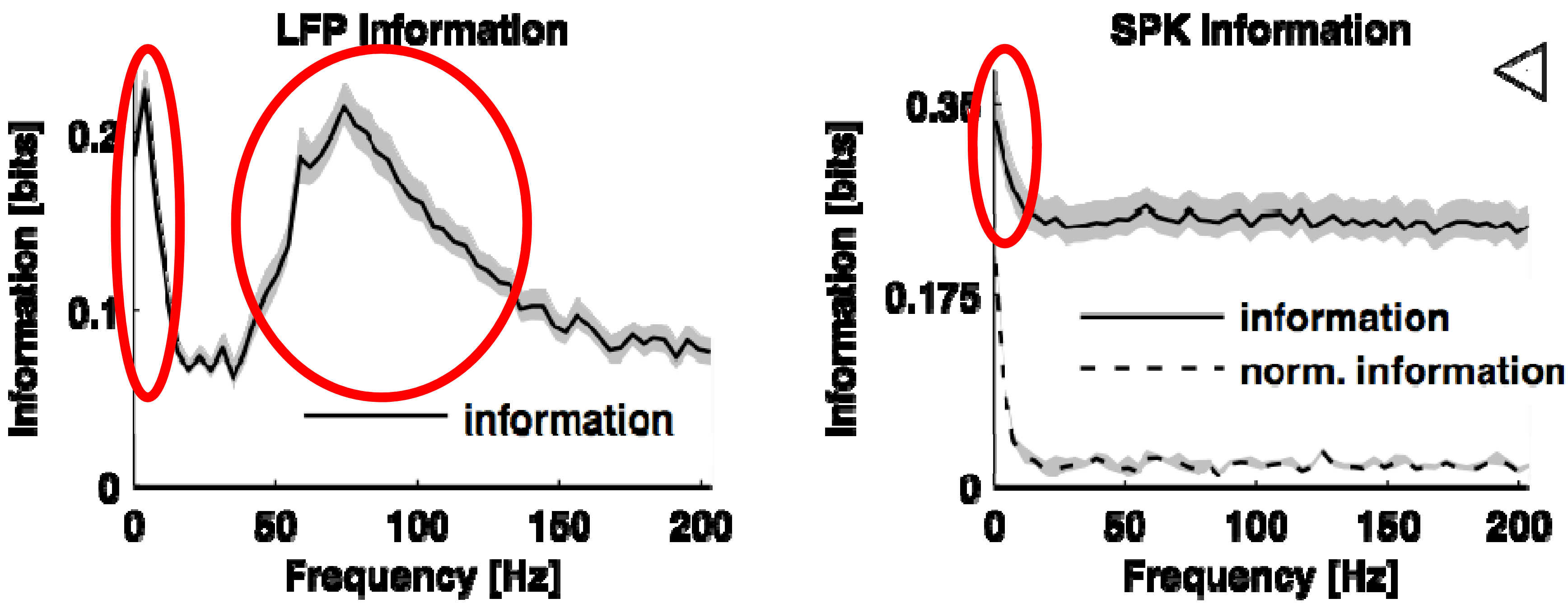



# Objective of the Tutorial





# Objective of the Tutorial



# Organization of the tutorial

- **PART I** - Introduction  to Fourier-based spectral analysis techniques
- **PART II** - Introduction to the bias problem and to the information breakdown toolbox
- **PART III**: how to perform the single-frequency information analysis

# During this tutorial we'll try to…

- Use as little math as possible
- Give a look to some simple Matlab example scripts

= thing are getting technical !!!

# PART I – INTRODUCTION TO FOURIER SPECTRAL ANALYSIS TECHNIQUES

# The Underpinnings of the Fourier Transform

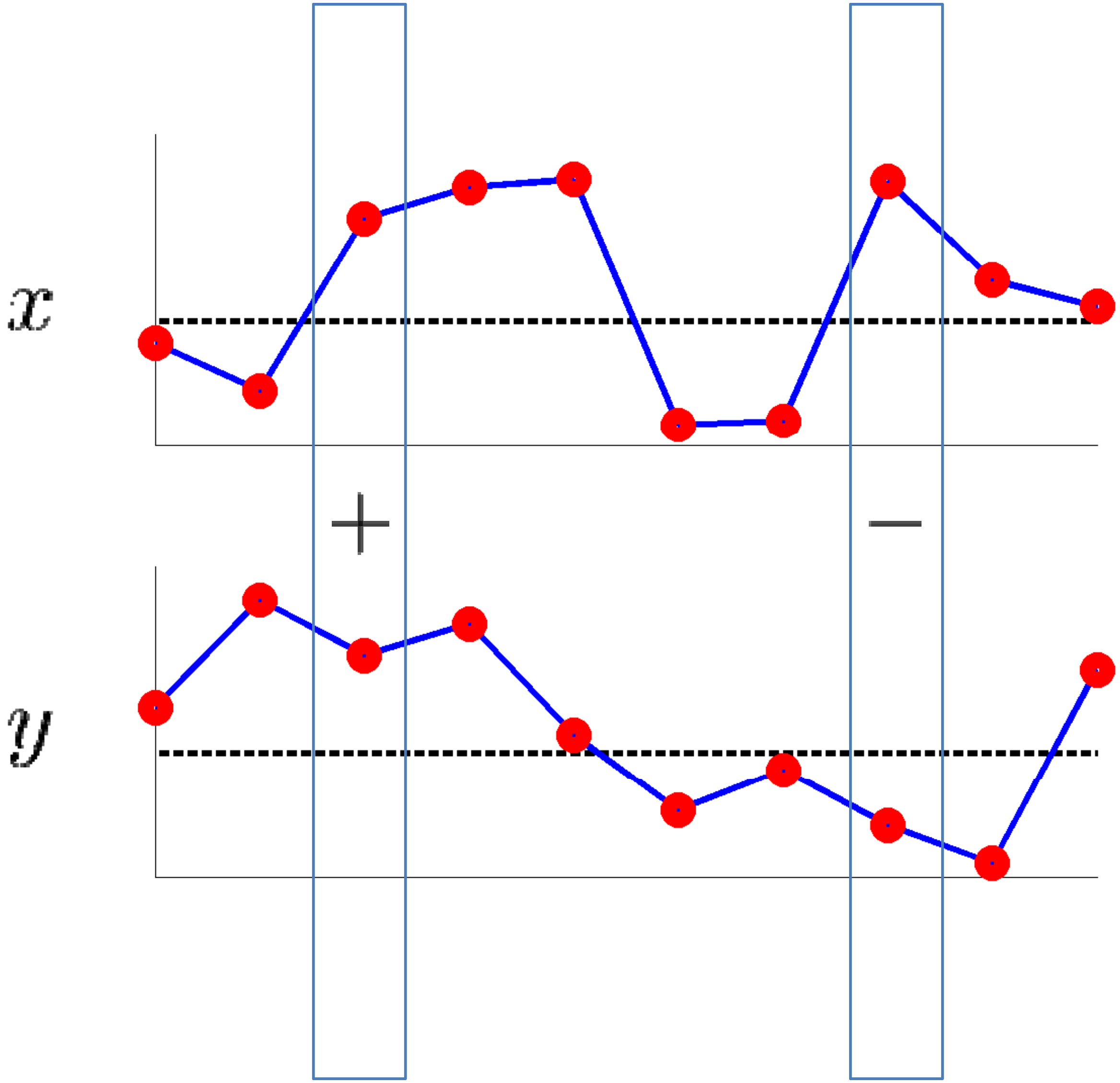
- **THEOREM:**
  - *any* time-series can be expressed as a discrete weighted sum of oscillatory functions: sines and cosines
  - If the time-series is $N$-points long then we only need to consider $N/2$ distinct frequencies
  - These frequencies are equally-spaced in the range 0Hz to $Fs/2$, $Fs$ being the sampling frequency

# Correlation

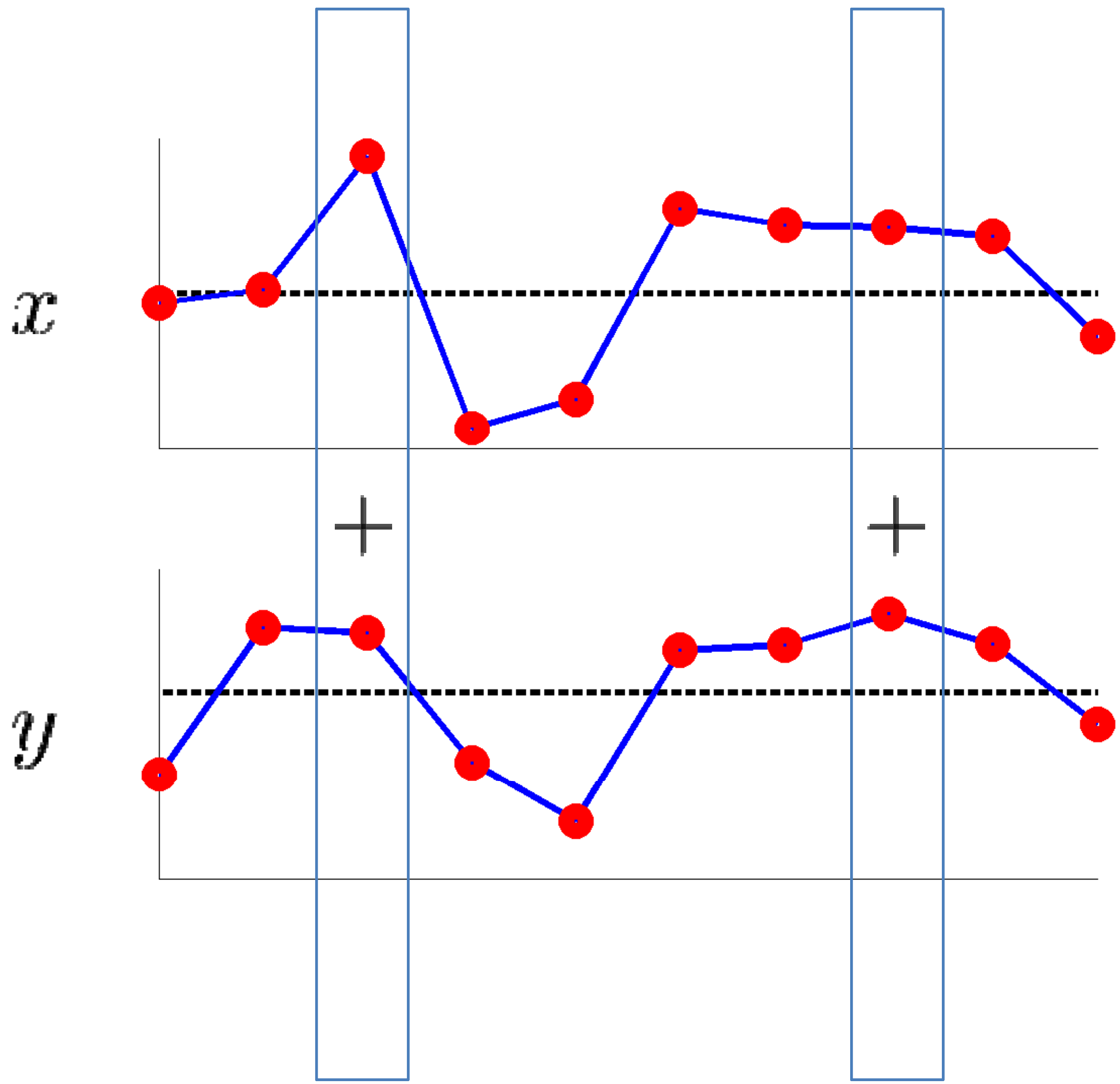
- How do we extract the coefficient for each sine and cosine?
- In general, how can we measure the similarity between two (zero-mean) time series?

$$\frac{1}{N}\sum_{i=1}^{N} x[i] \cdot y[i]$$

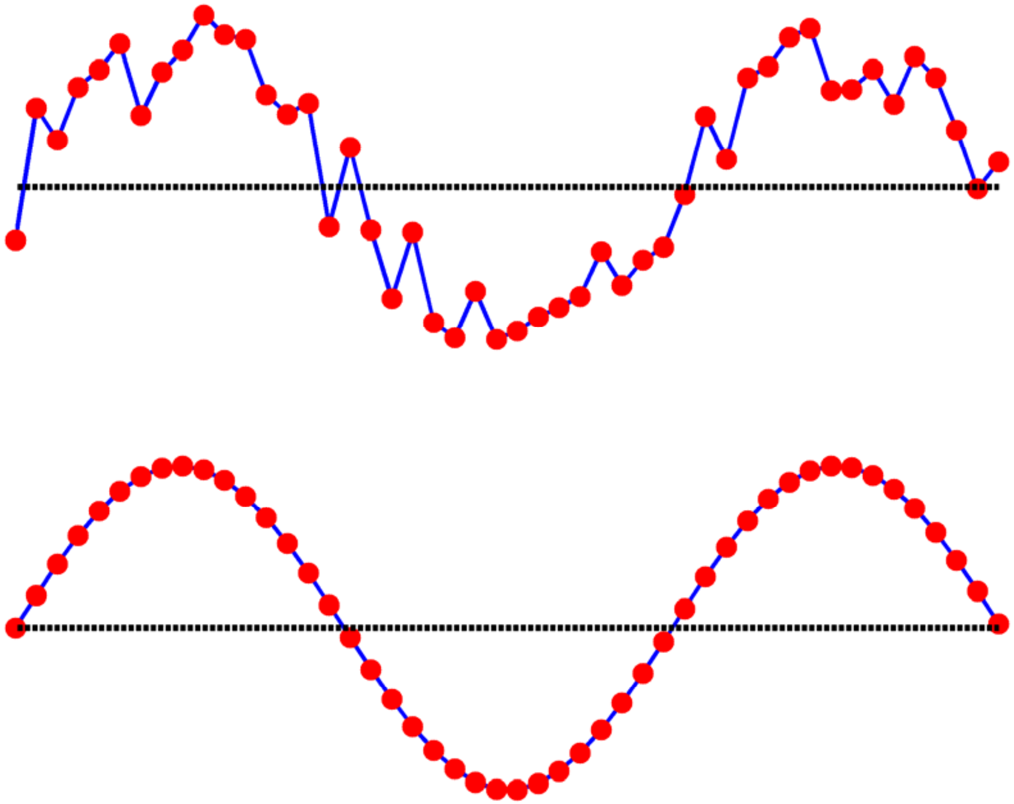
# Correlation



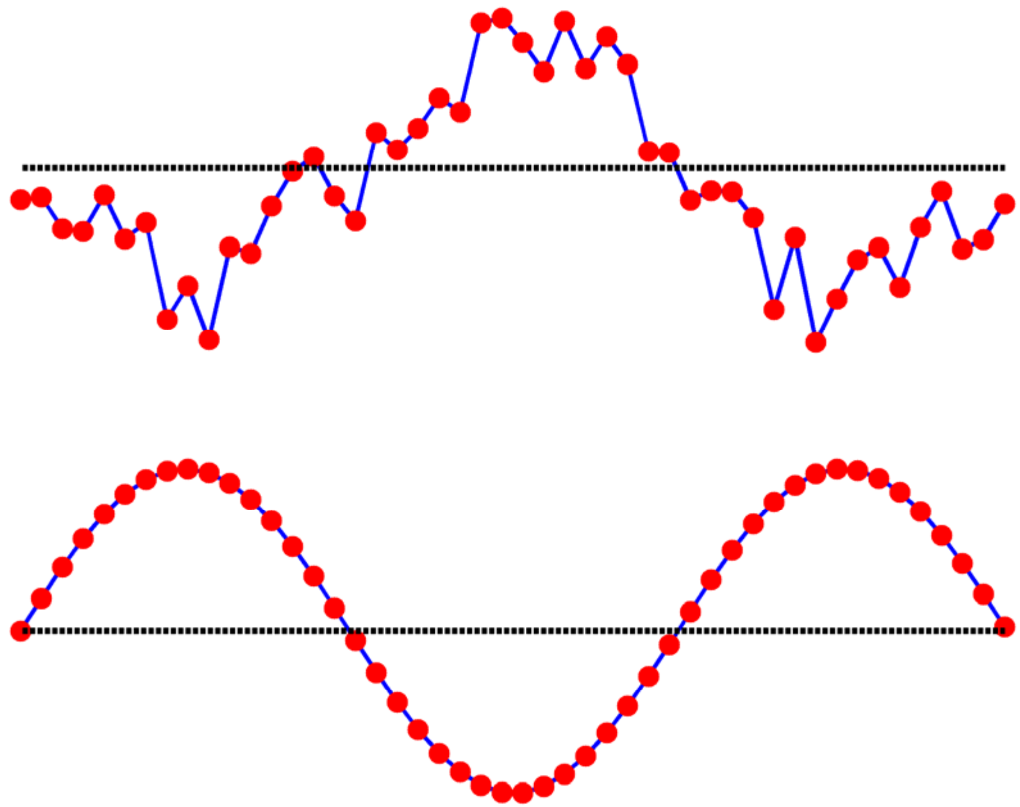
# Correlation



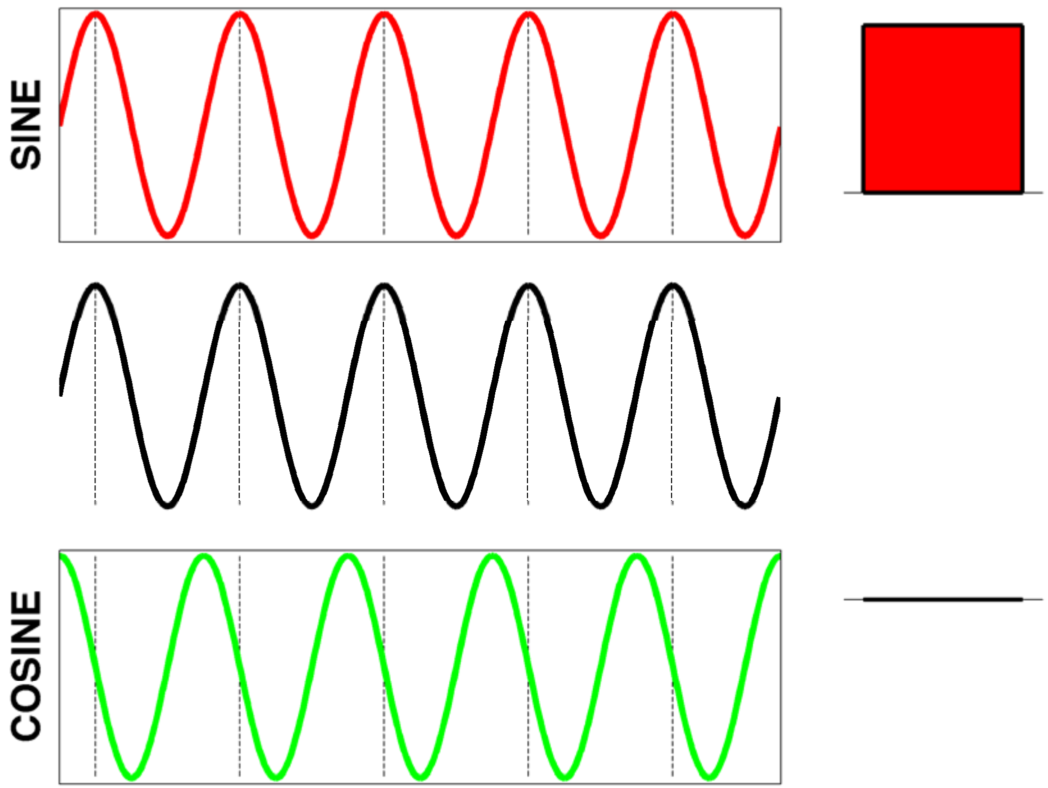
# Correlation



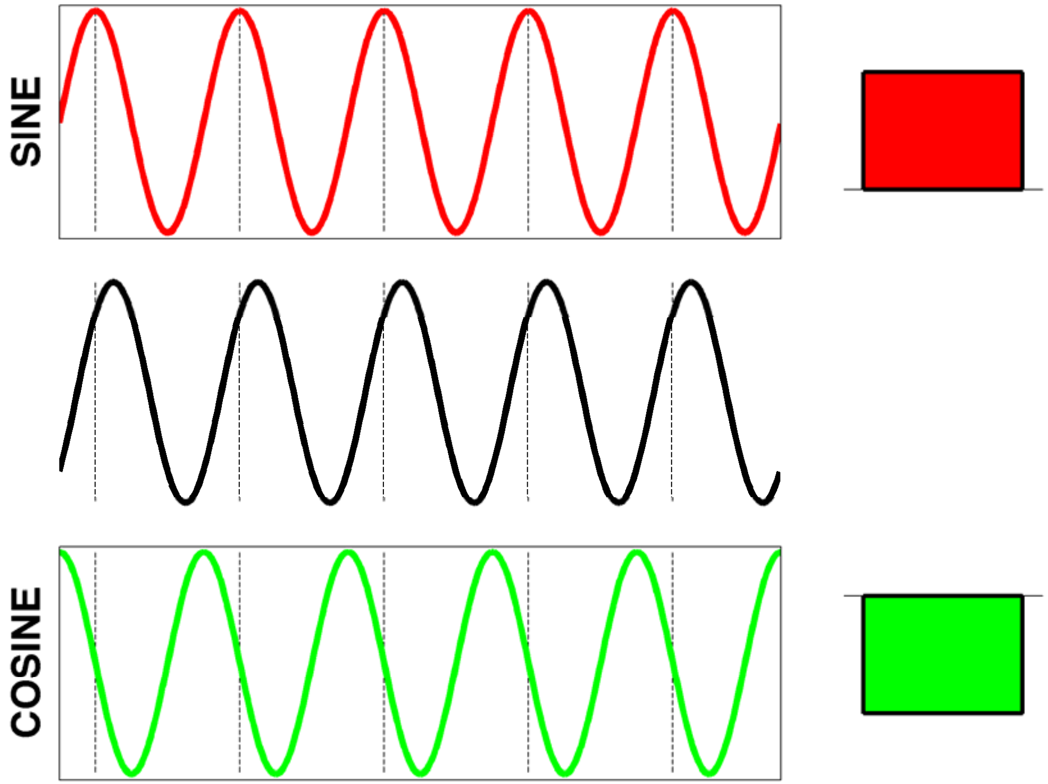
# But what if we are in THIS situation?



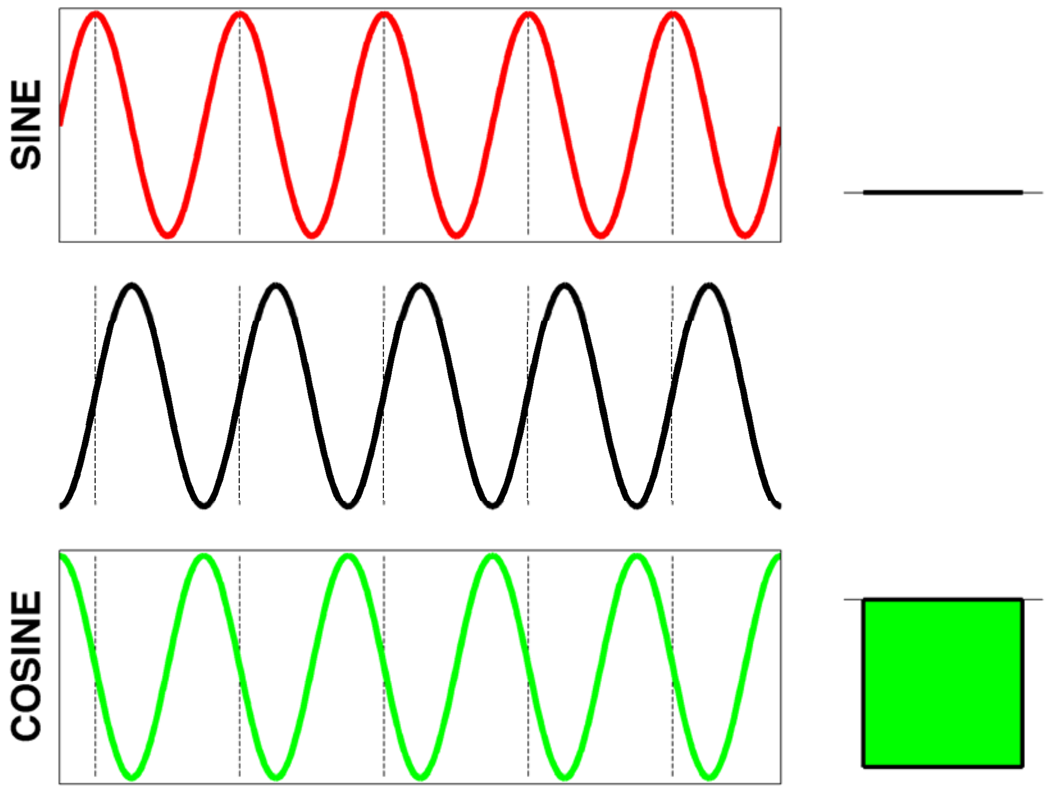
# Untangling the phase problem



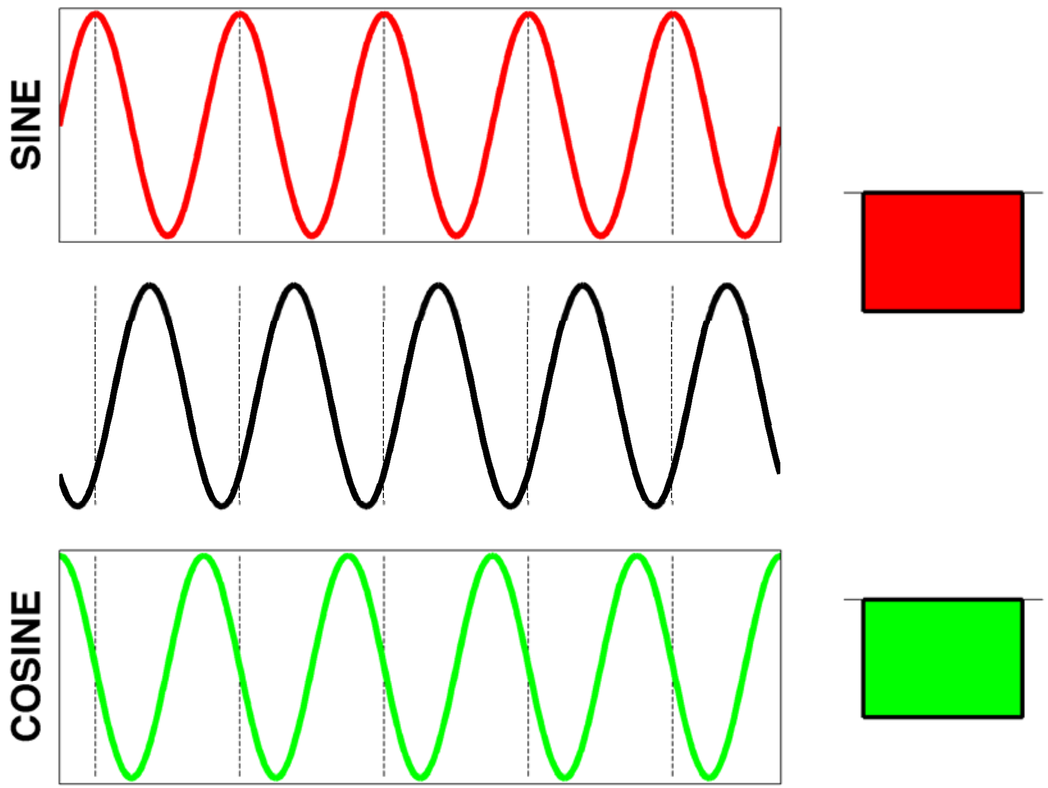
# Untangling the phase problem



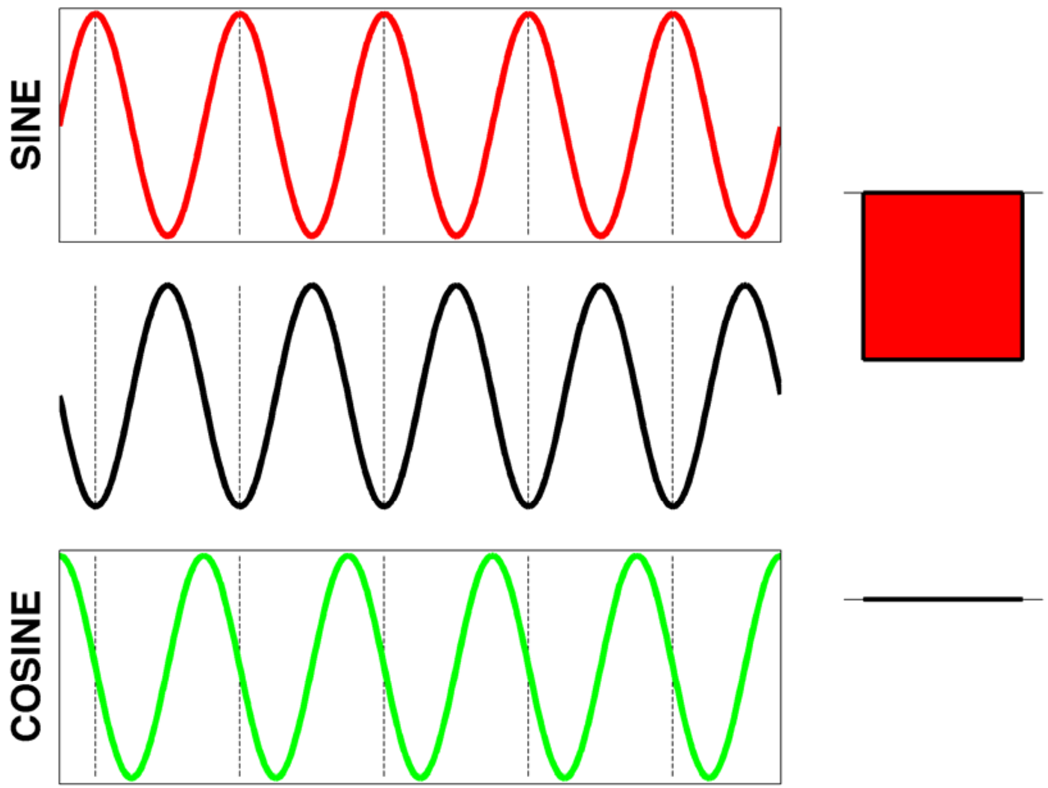
# Untangling the phase problem



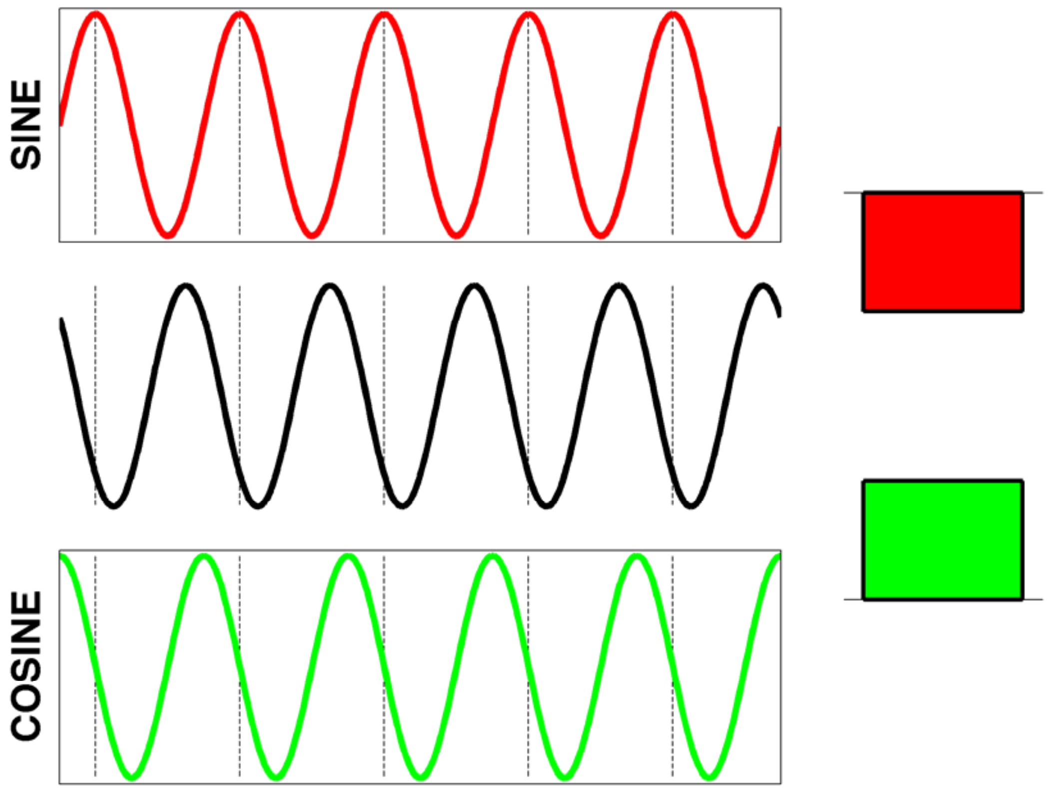
# Untangling the phase problem



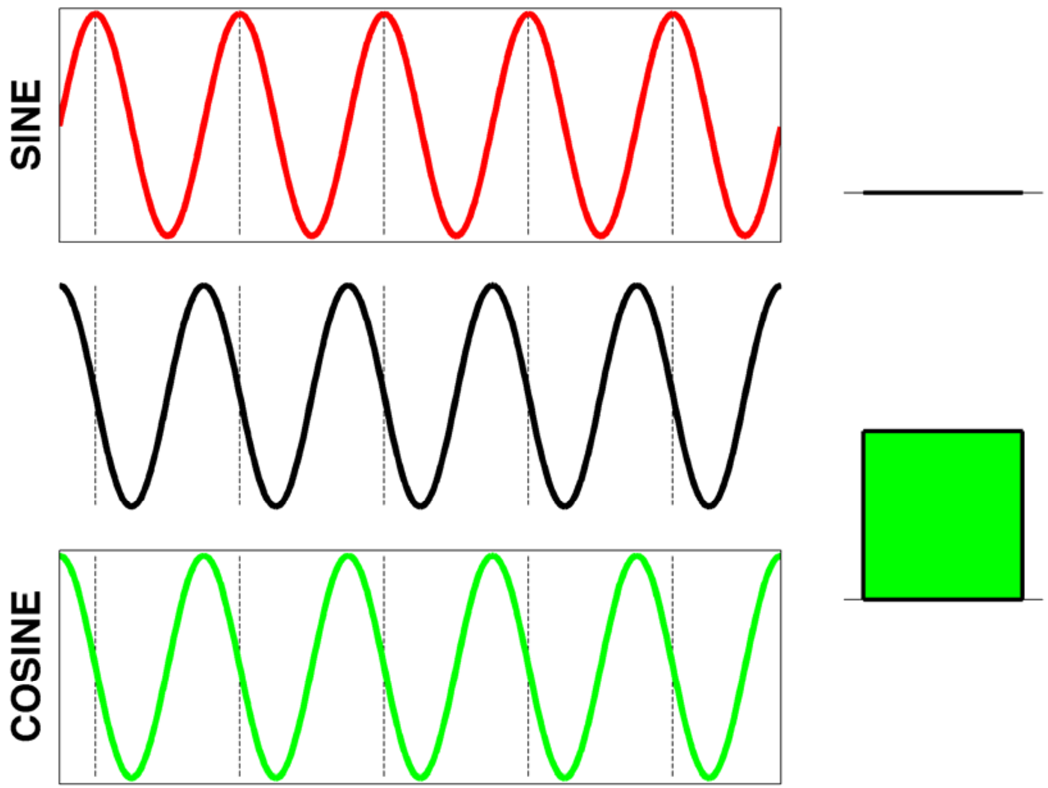
# Untangling the phase problem



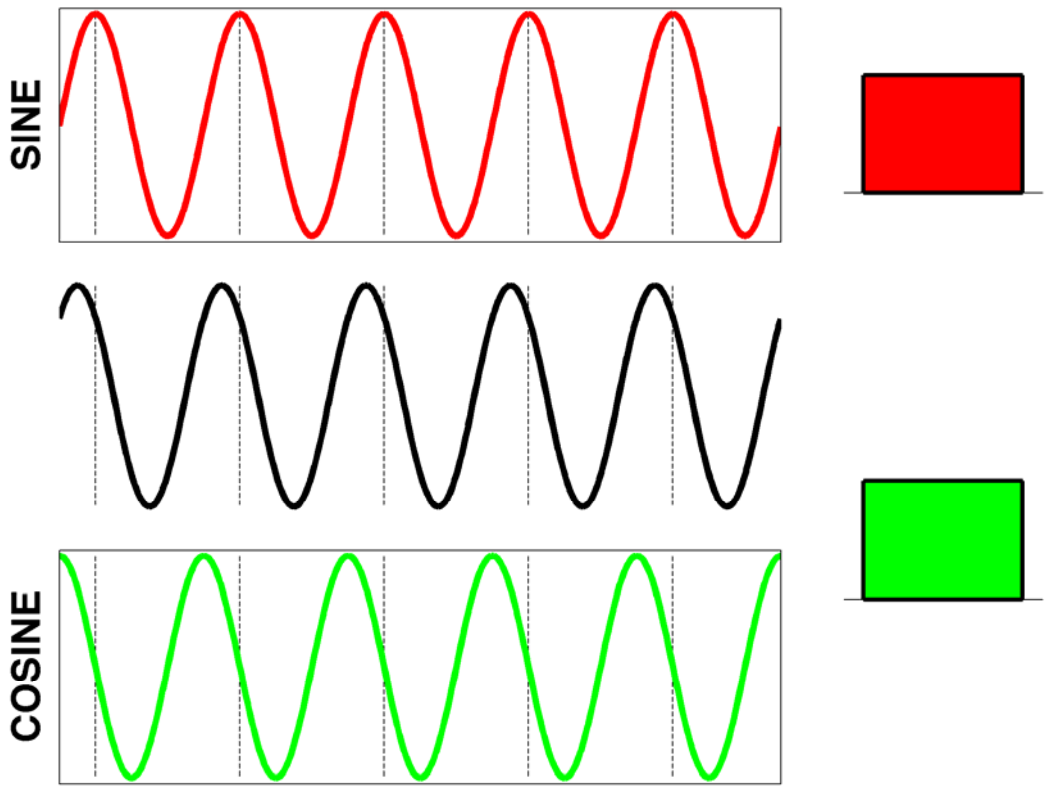
# Untangling the phase problem



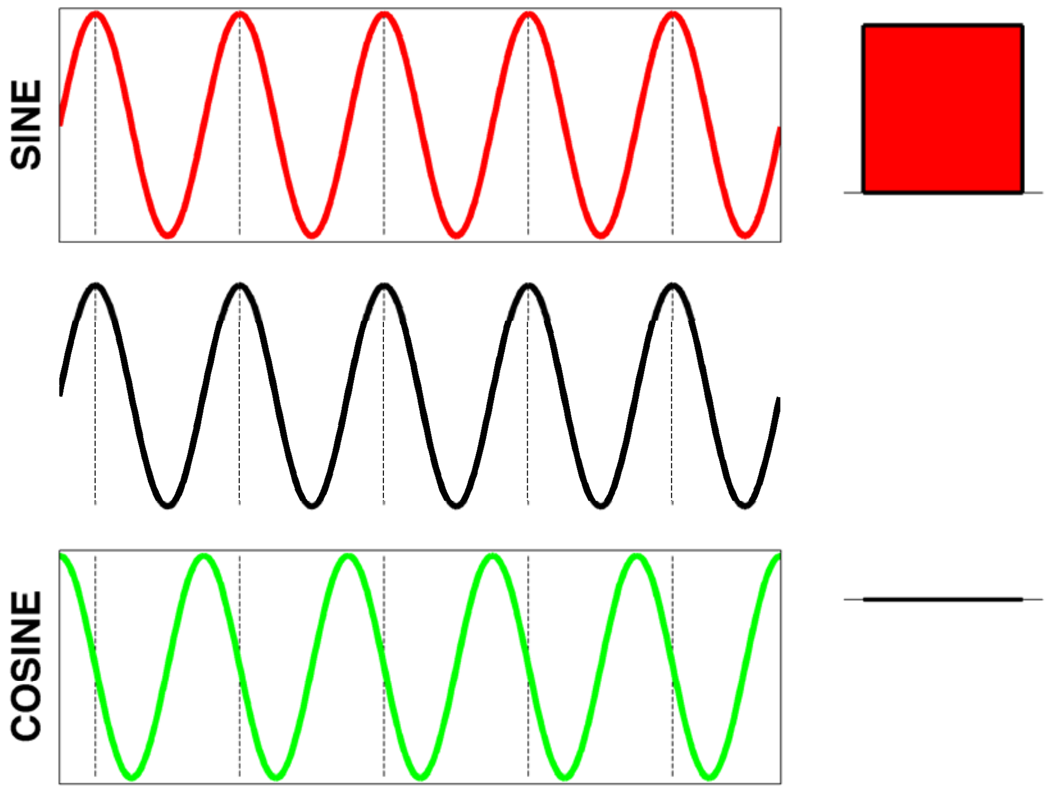
# Untangling the phase problem



# Untangling the phase problem

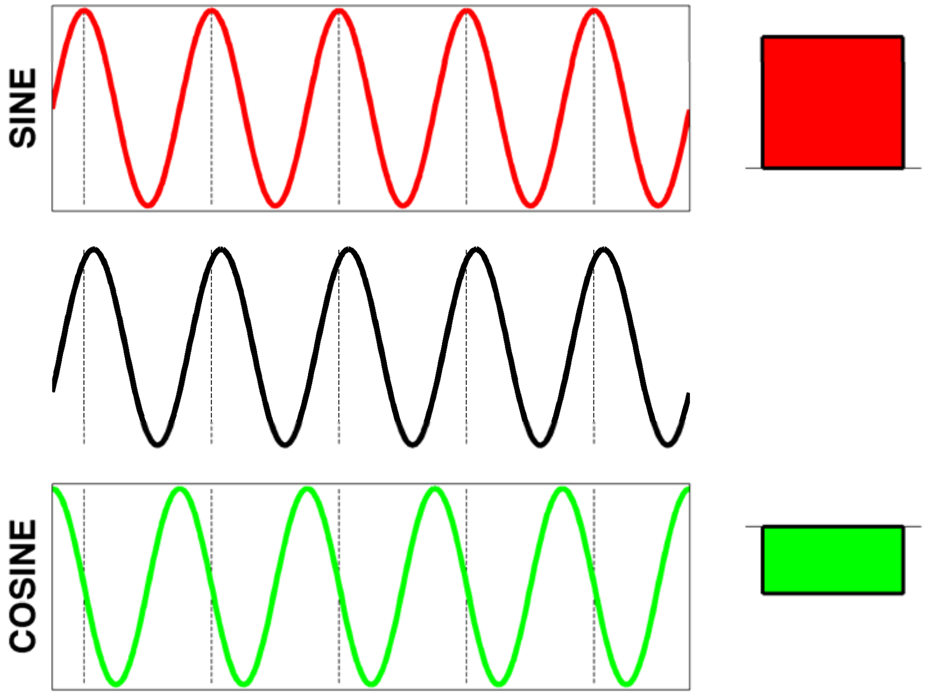


# Untangling the phase problem



# Untangling the phase problem

- **SOLUTION:** for a given frequency *f*, we simply need to compute the correlation between our time series and both a sine and a cosine wave of frequency *f*.



# The FFT

- How to keep the two correlation values for a given frequency well organized? Well, store the two values into the real and imaginary part of a complex number.
- Matlab does this very efficiently through the function **`fft`**

# Periodogram

- How much of a given frequency is present in my time-series?
- **SOLUTION:** Take the two coefficients corresponding to a given frequency *f*, square them and sum them and you'll obtain the power at frequency *f*

# The periodogram in Matlab

- Let **`x`** be our time series (a Matlab array)

```
N = length(x);
X = fft(x);
S = abs(X(1:N/2)).^2;
```

- Matlab **`periodogram`** function.

# Incorrect statements

- "*The periodogram* ***is*** *the spectrum of my time-series*" (FALSE! it **can** be an **estimate** of the spectrum if the process is stationary)
- "*The periodogram allows me to compute the spectrum by `pretending' that my time-series is periodical*". (FALSE! If the process is periodic than it provides the true spectrum otherwise it's just an estimator)

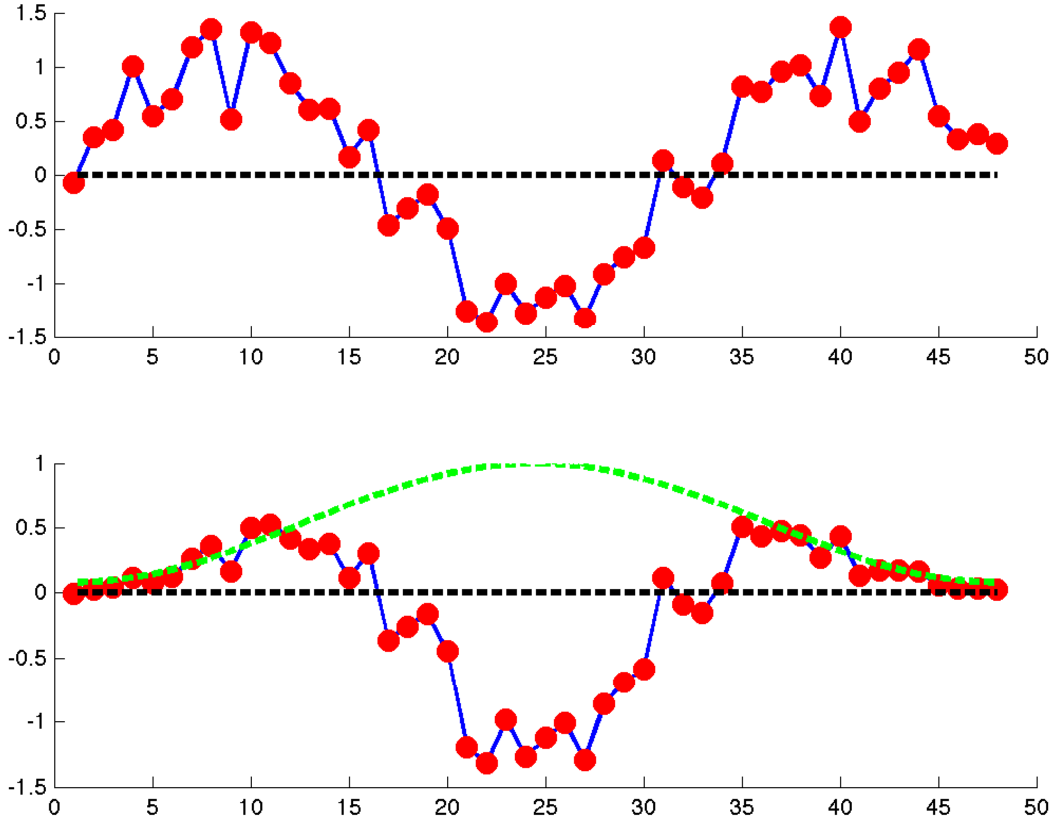
# Problems with the periodogram

- BIAS PROBLEM: estimates done using the periodogram of spectra with high-dynamic ranges can be heavily biased.
- VARIANCE: the variance of the periodogram spectral estimates
  - is very large (as large as the value we wish to estimate)
  - does not decrease when we increase the number of points in your time analysis (*i.e.* you collect more data but the estimate remains as bad as it was before!).

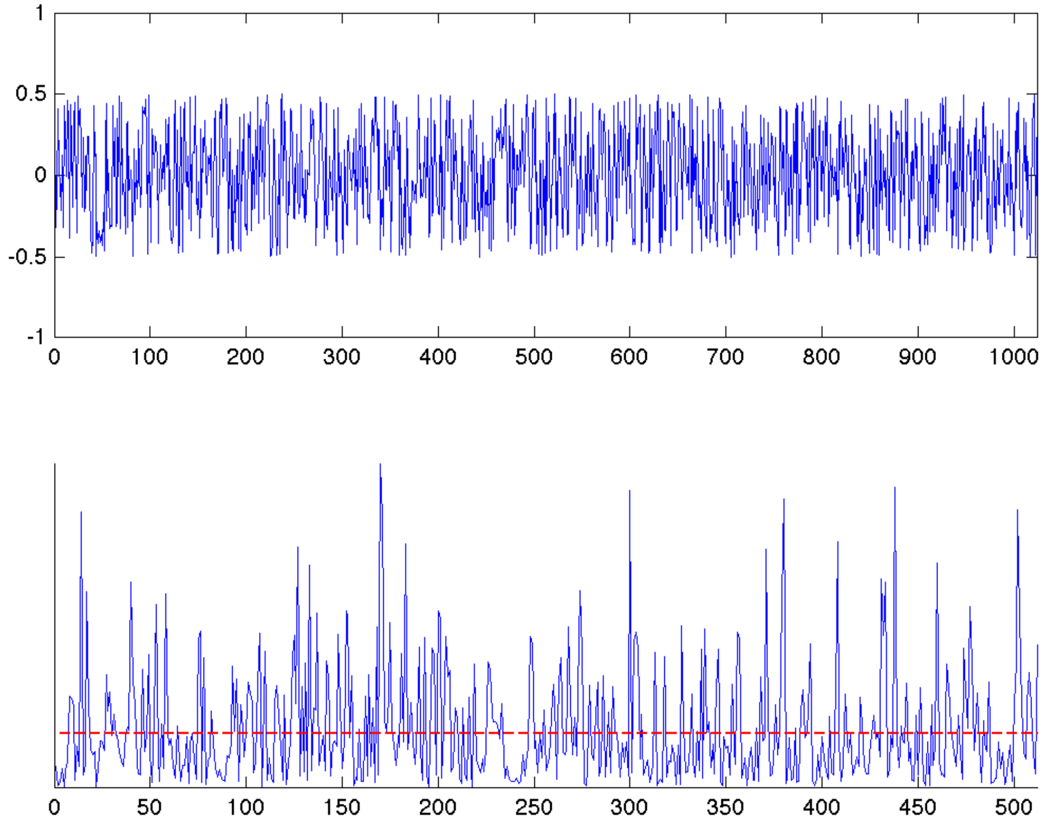
# The bias problem - Tapering

- The bias in the periodogram arises because we are trying to estimate the spectrum with a very high resolution (each of the N/2 Fourier frequencies). We are stretching our data to the limit…
- To reduce the bias we need to sacrifice spectral resolution, this is done by multiplying the time series with a taper (e.g. the Hamming window). Unfortunately, by doing so we are throwing away some information in our data.

# The bias problem - Tapering

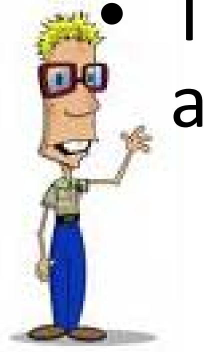



# The variance problem




# The variance problem

- With the periodogram, when we add a data point (*N* -> *N*+1) we do not use this additional information to improve the estimates in the N/2 frequencies we were already considering. Instead, we use this new information to try estimate the spectrum at a *new* additional frequency. Thus adding points cannot possibly decrease the variance of our estimates.
- The only solution is to try to “average” … but average on what?



# Smoothing the periodogram [1-2]

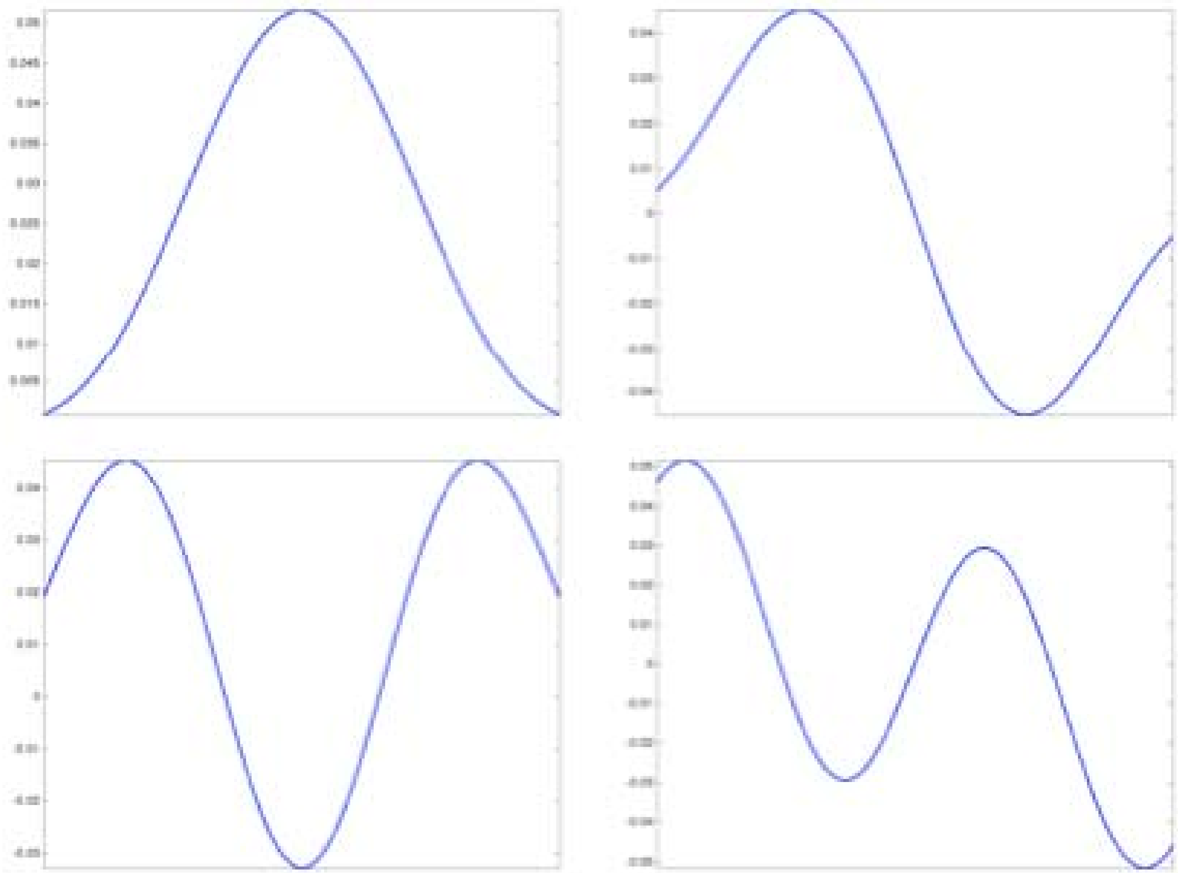
- One possibility is to smooth the spectrum in the frequency domain (*lag-window* estimator)
- Another possibility is to
  – cut the time-series into shorter segments,
  – compute the periodogram on the shorter (tapered) sequences
  – average all these estimates together.

This is what is done in the Welch's (very popular, see Matlab's **`pwelch`** function) and Bartlett's estimators.

# Smoothing the periodogram [3]

- A third (very elegant) possibility is to try to recover some of the information lost when tapering using several different tapers each of extract a different piece of information from the data which is lost when using the other tapers.
- This is the so called *multi-taper* estimator (Matlab's **`pmtm`** function)

# Multitaper



# References

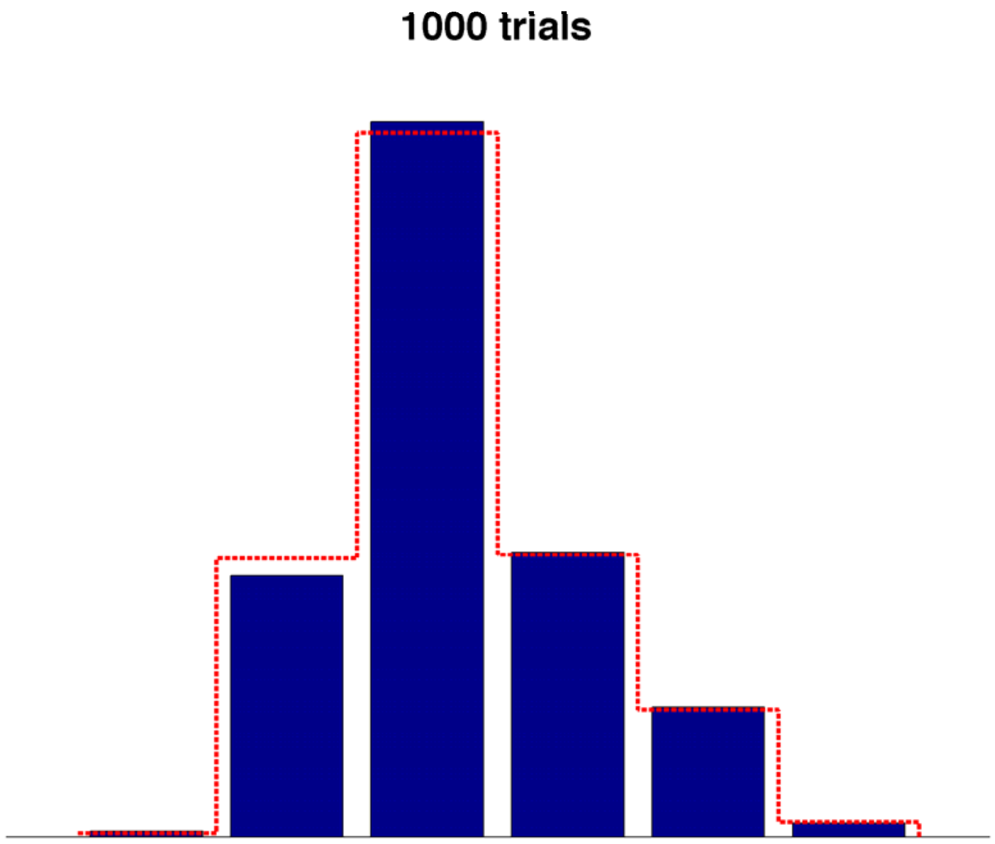

- Percivald and Walden, *Spectral Analysis for Physical Applications*
- Proakis, Manolakis, *Digital Signal Processing*

# PART II – ESTIMATING MUTUAL INFORMATION

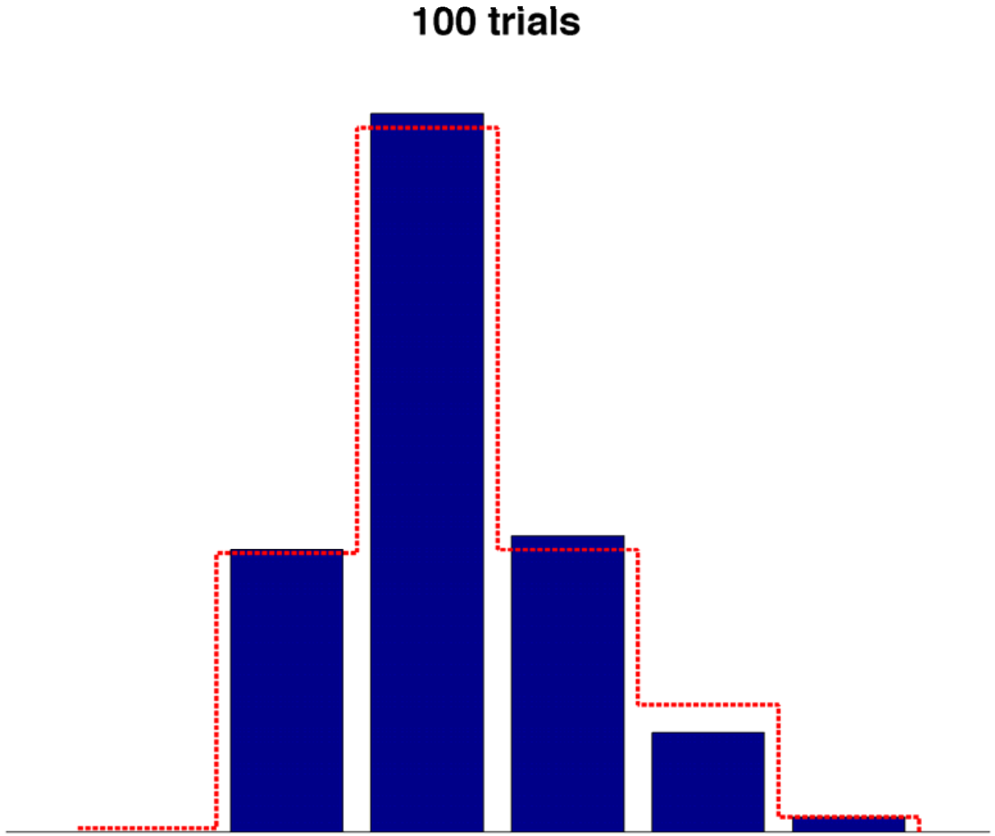
# Estimating mutual-information

- We want to estimate the mutual information between the movie scene and spectral power
- In general to estimate mutual information we need to estimate the probability distributions.

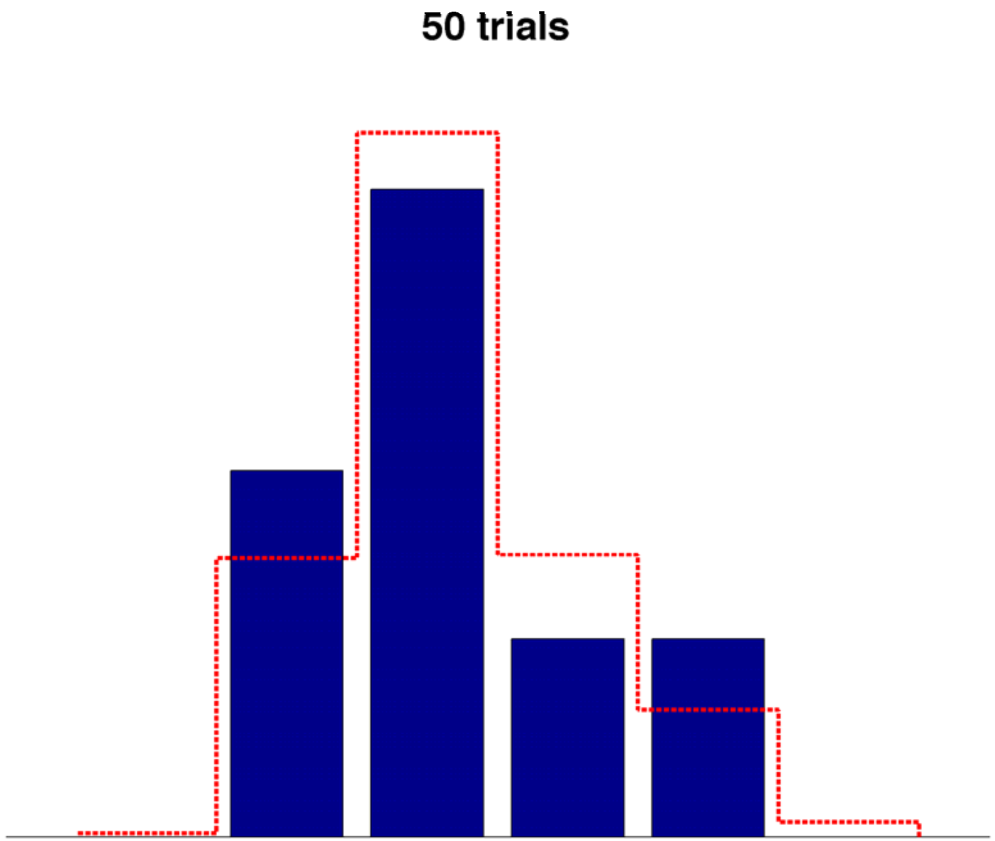
# The bias problem



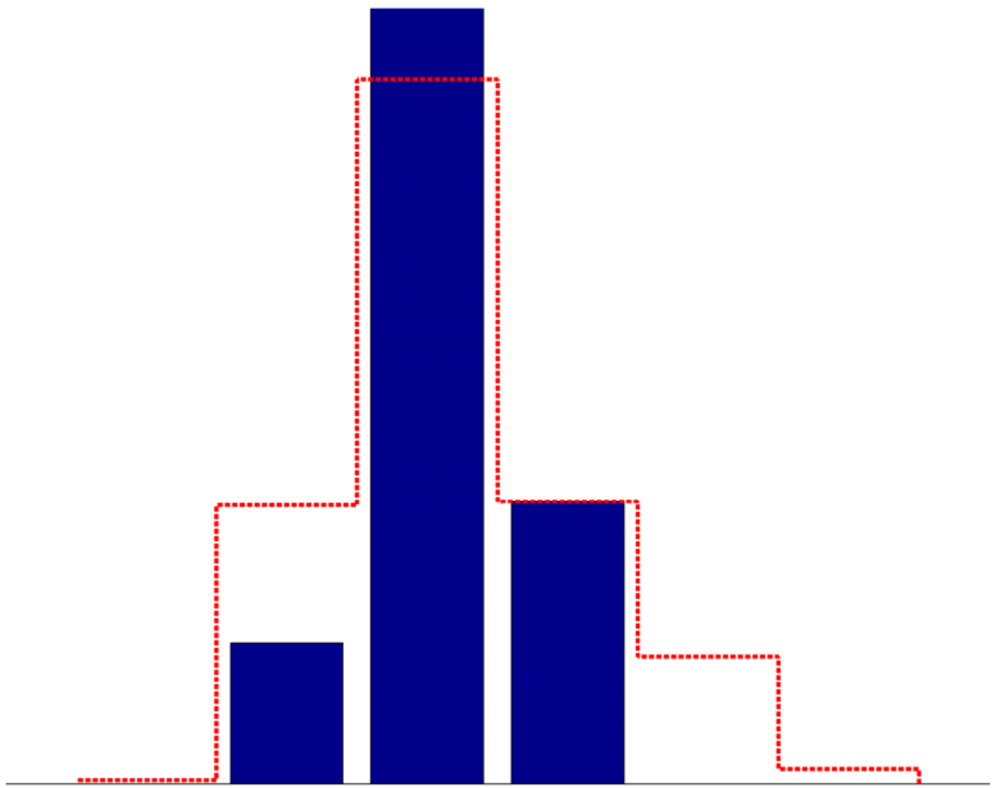
# The bias problem



# The bias problem



# The bias problem

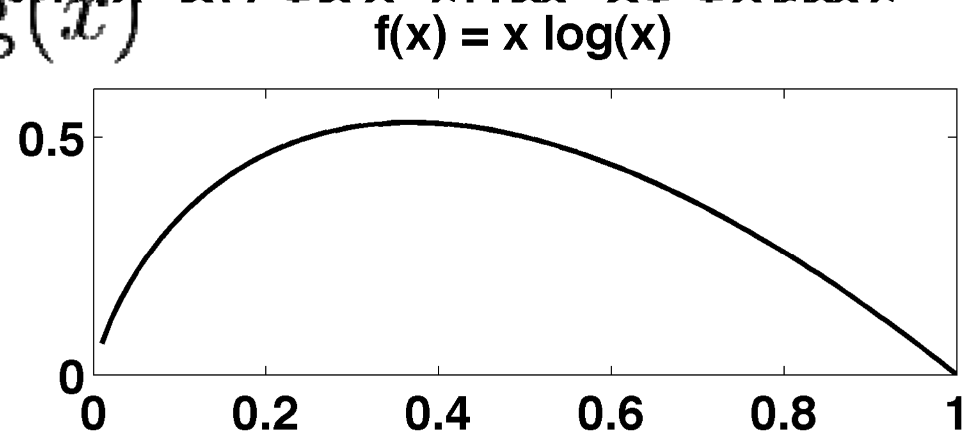


# The bias problem

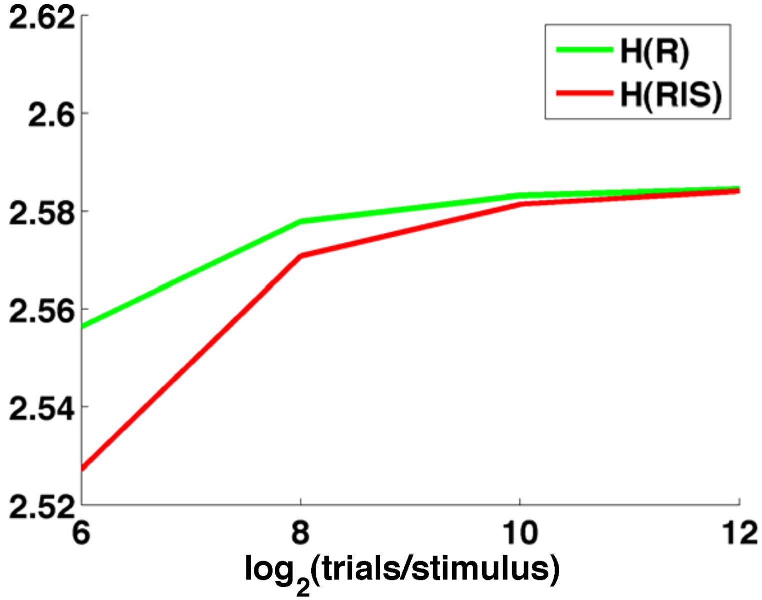
- How does this affect the estimates in mutual information? Well, mutual information is given by the difference between two entropies

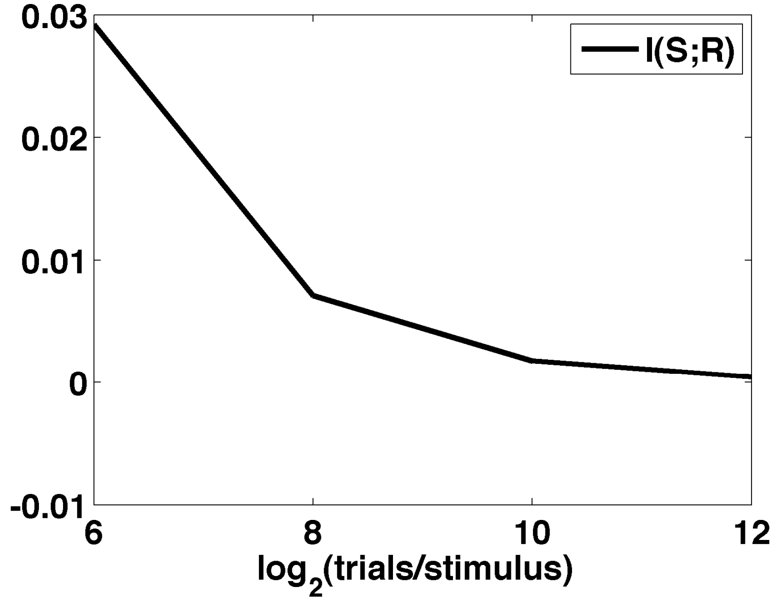
$$I(S;R) = H(R) - H(R|S)$$

and each entropy is given by the sum of terms of the form $-x\log(x)$



# The bias problem



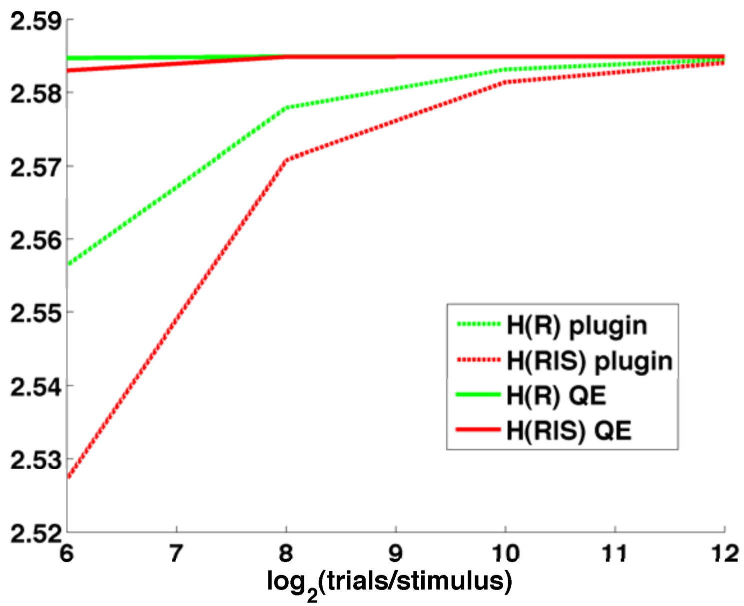



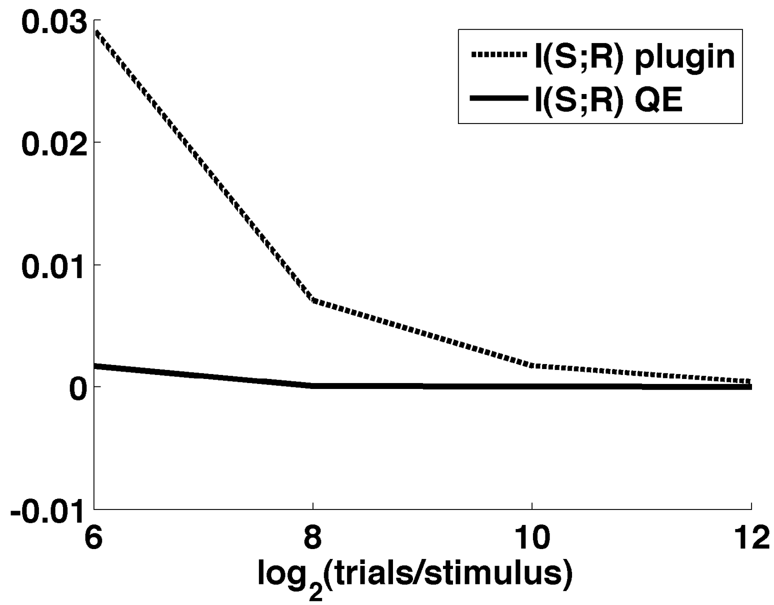
# Quadratic Extrapolation

- Divide the available trials into blocks of N/2 , N/4 …
- Compute average information for N , N/2 , N/4 ... Data and fit the dependence of I on N to the above quadratic expression
- Estimate the true (N=∞) value from the best-fit expression

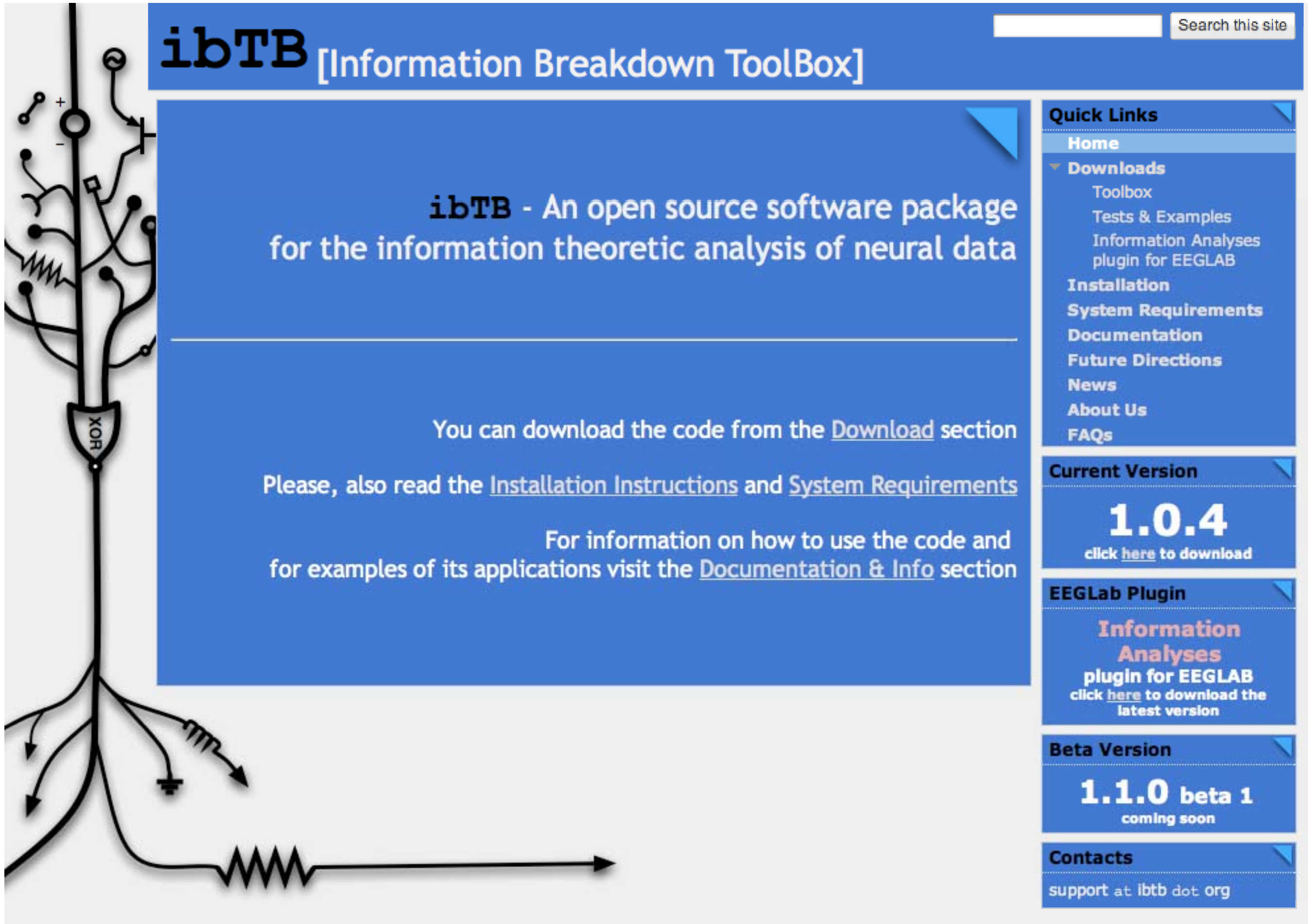
$$I_N = I_{true} + \frac{C_1}{N} + \frac{C_2}{2N^2} + \cdots$$

# Quadratic Extrapolation



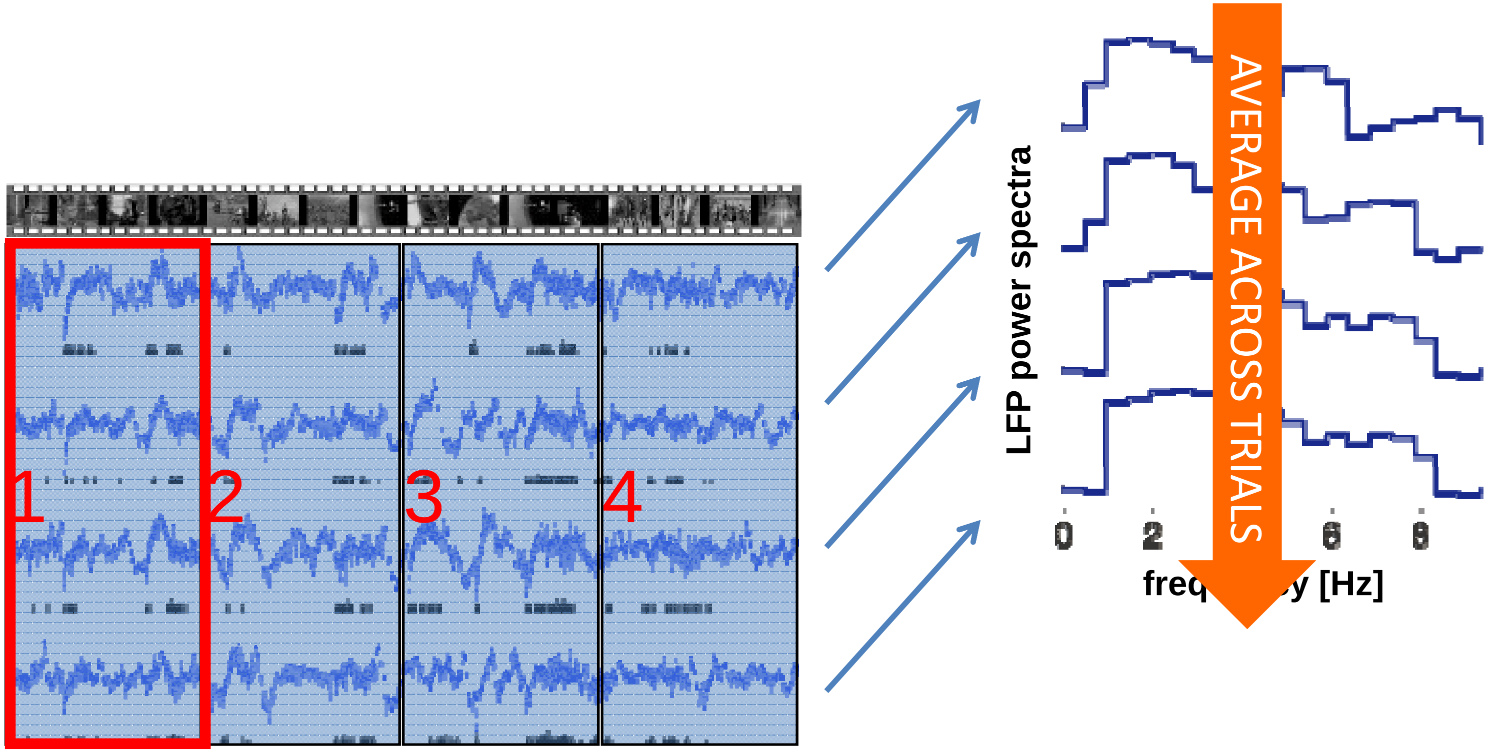
# Toolbox



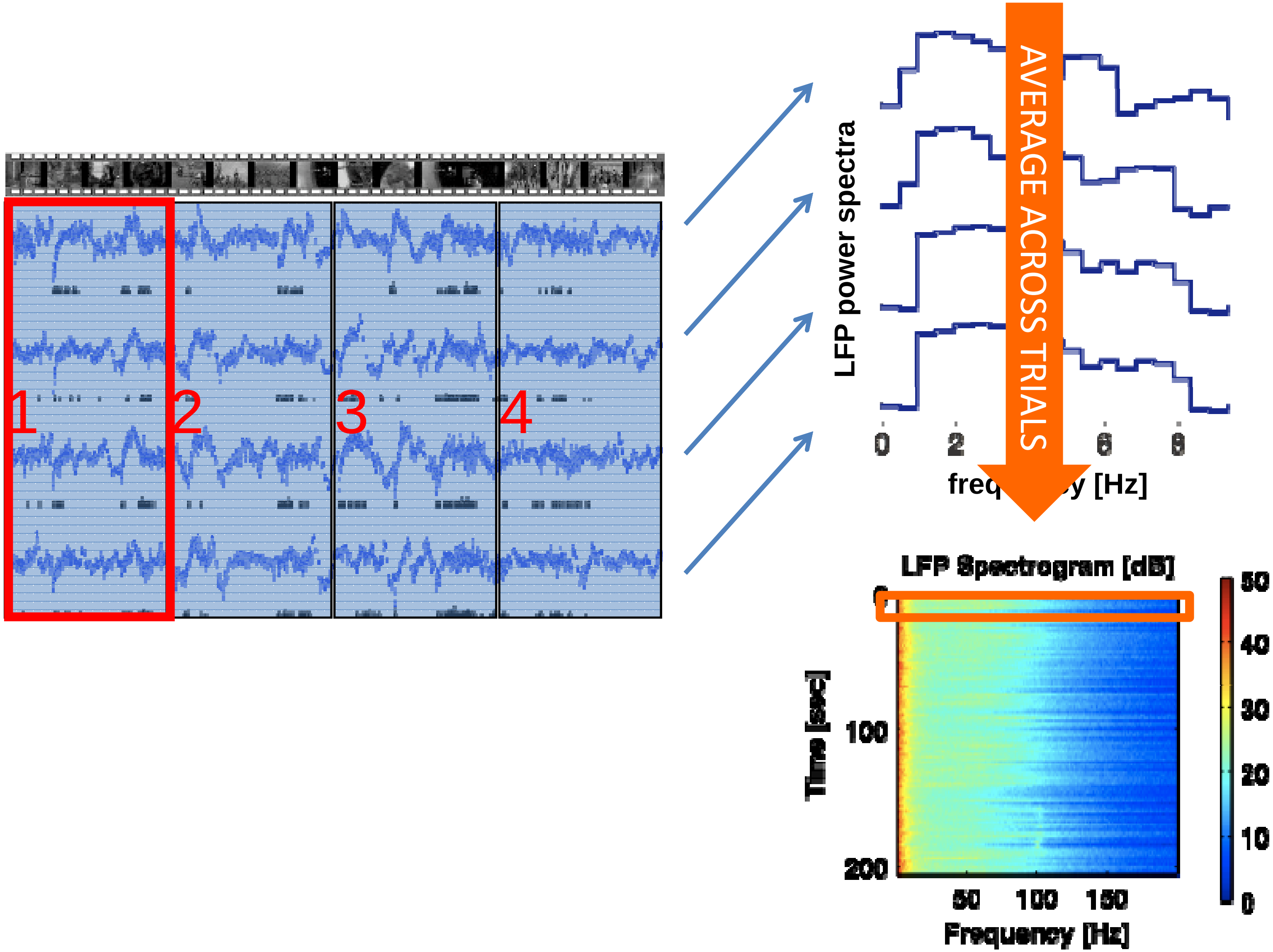

www.ibtb.org

# PART III – HOW TO PERFORM THE SINGLE FREQUENCY INFORMATION ANALYSIS

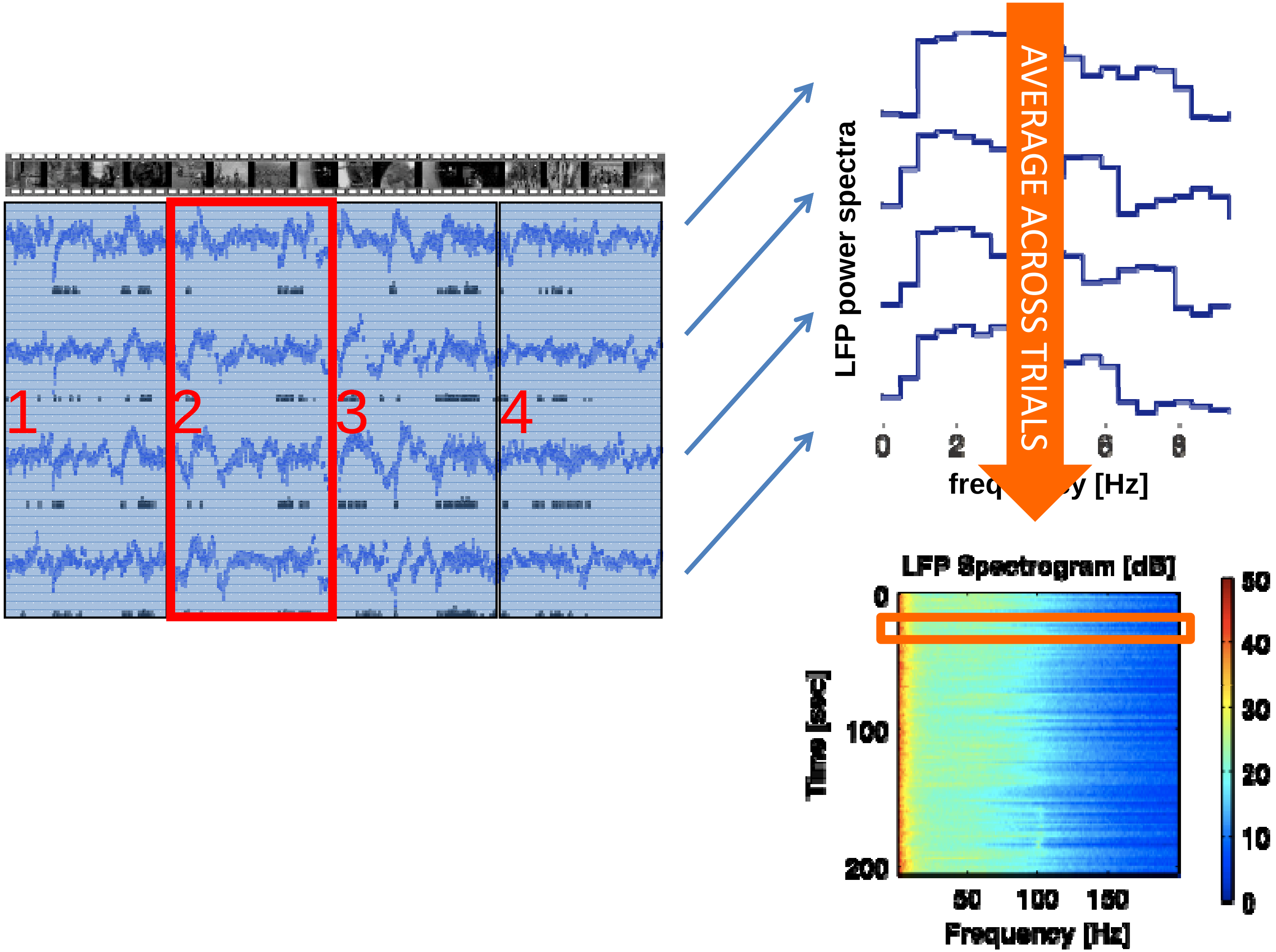
# LFP spectral analysis



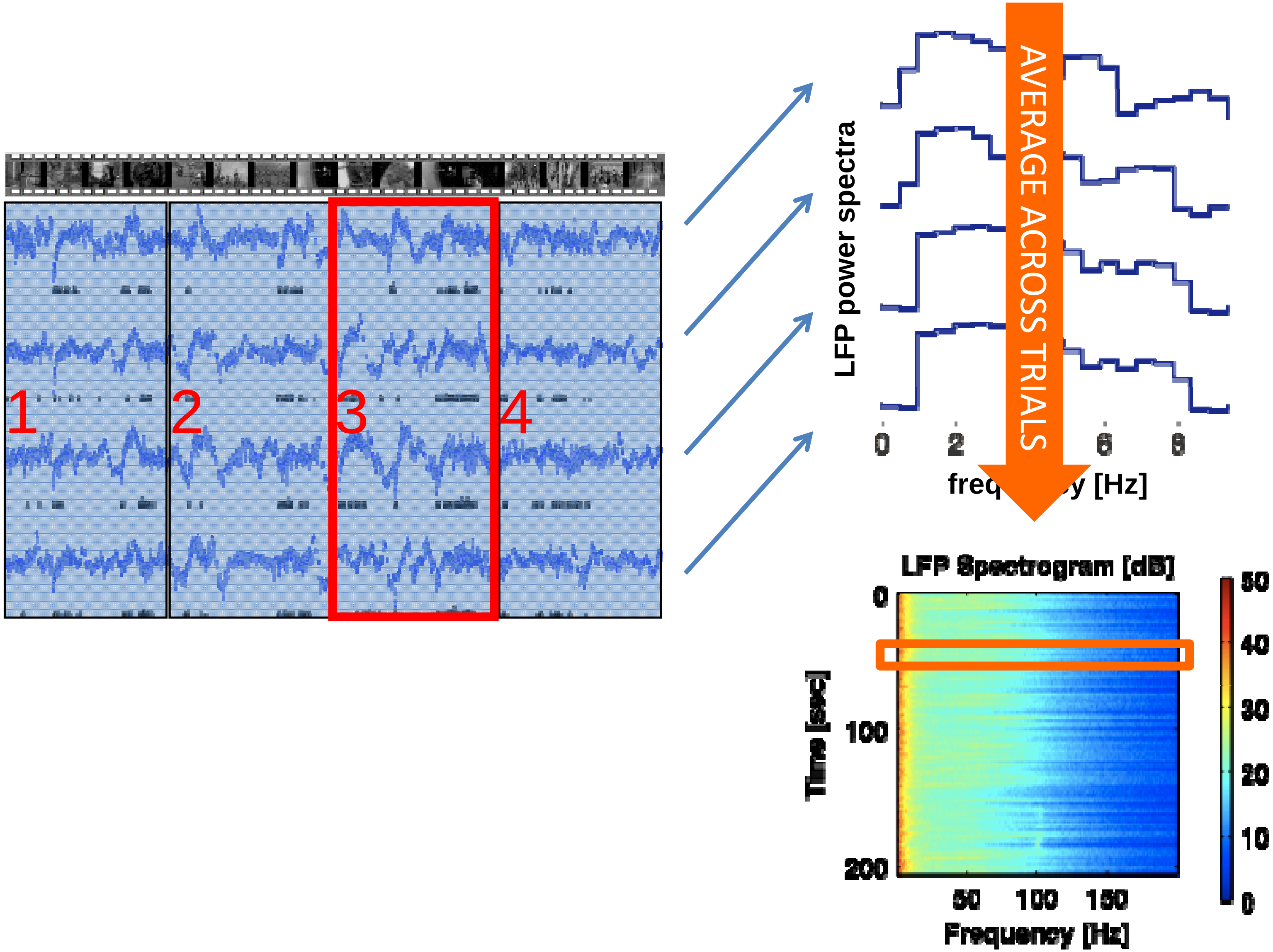
# LFP spectral analysis



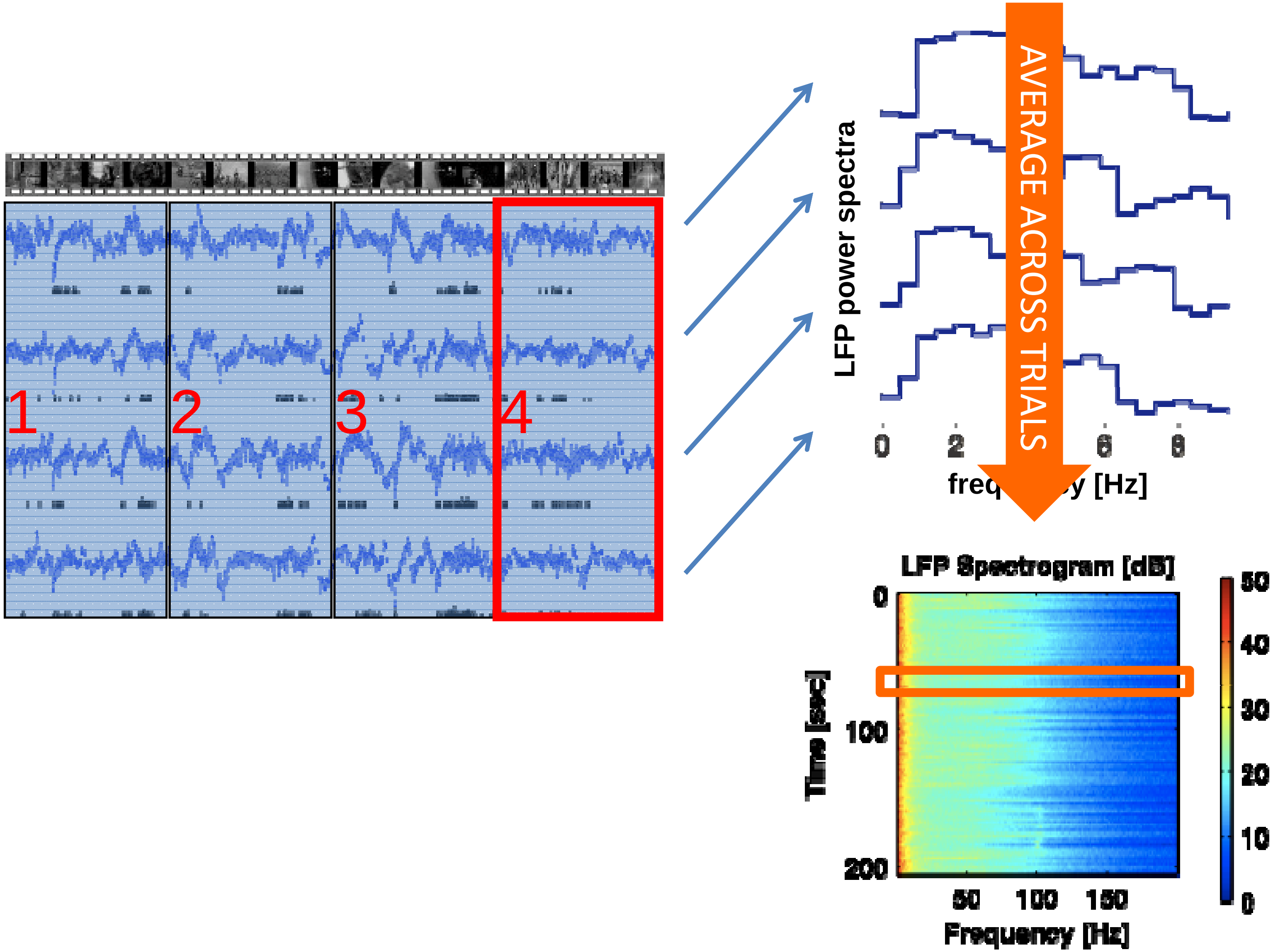
# LFP spectral analysis



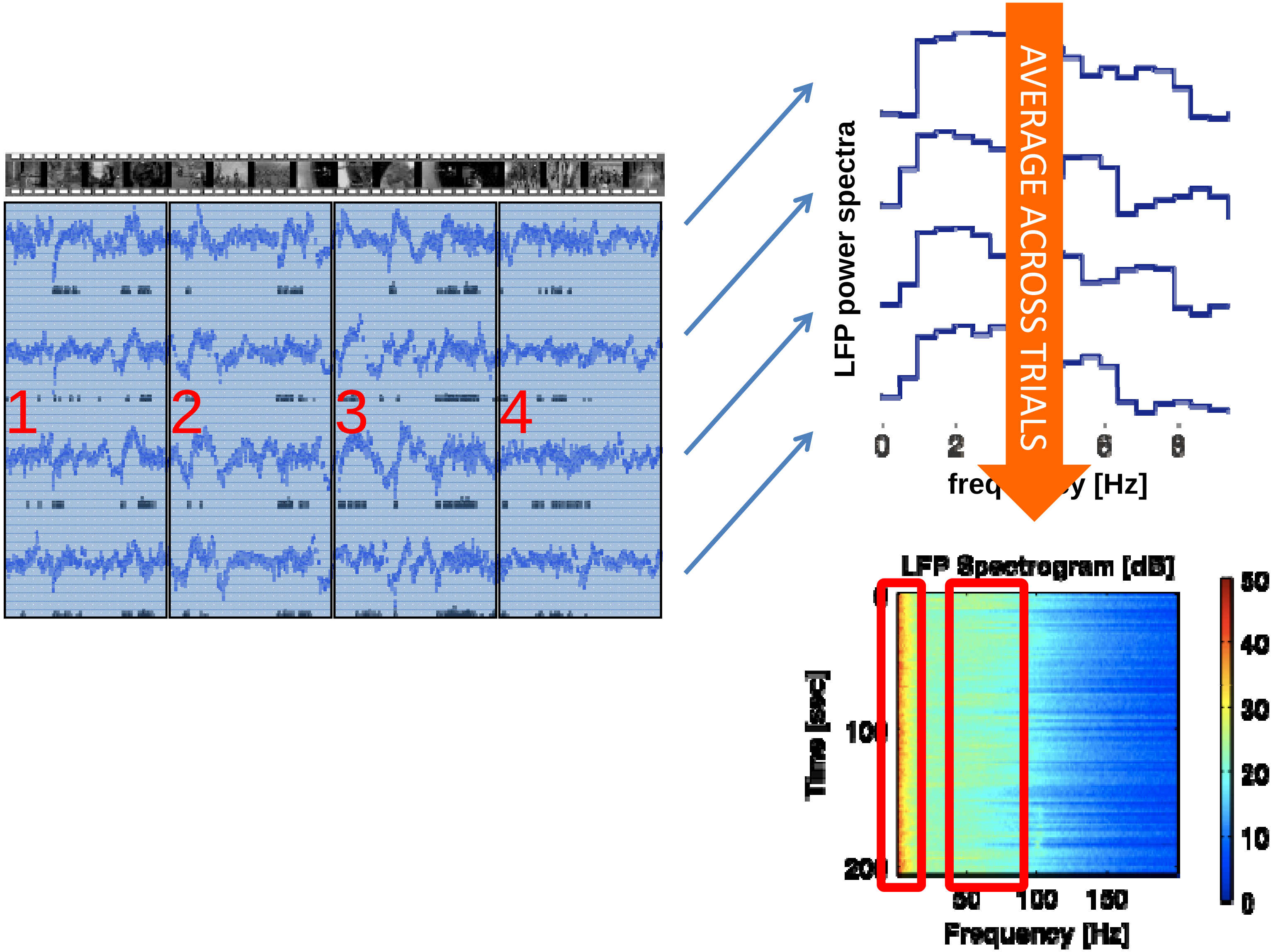
# LFP spectral analysis



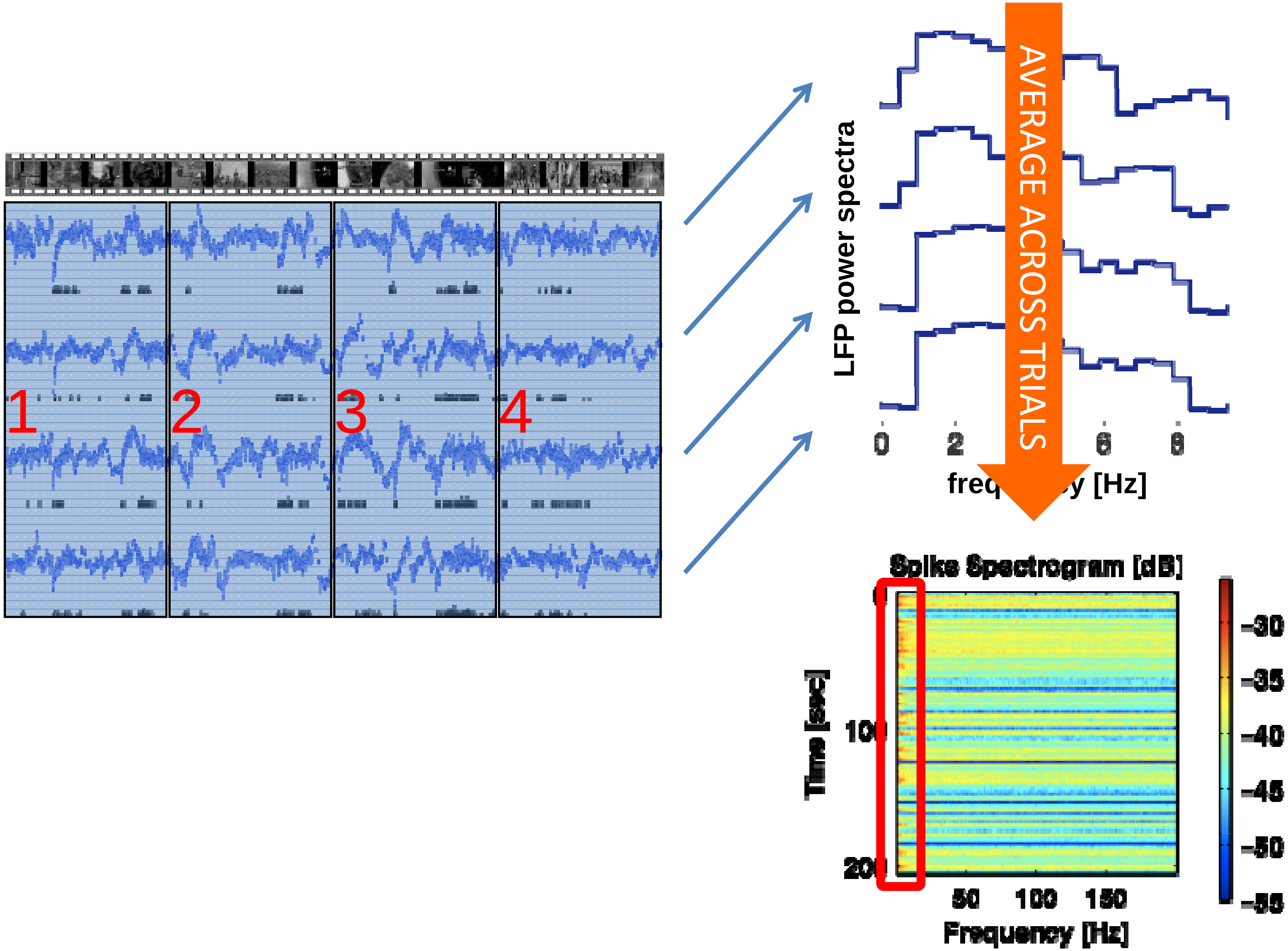
# LFP spectral analysis



# LFP spectral analysis



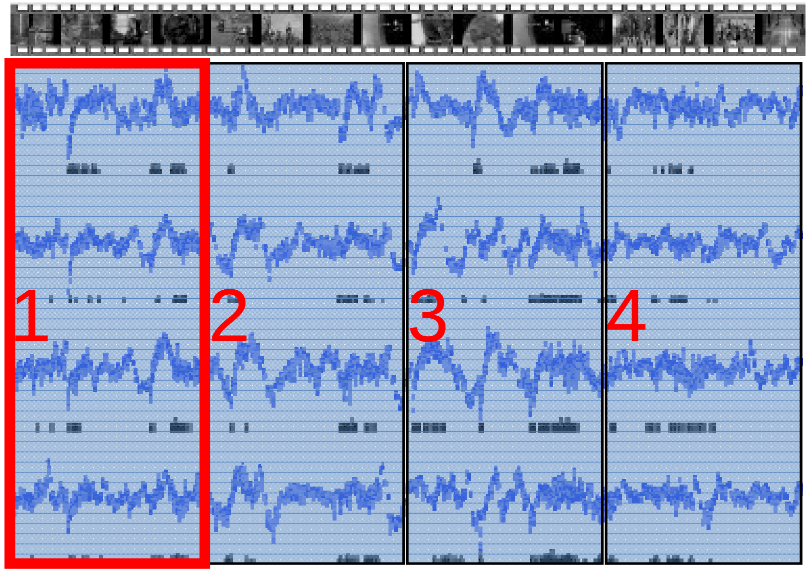
# Spike spectral analysis

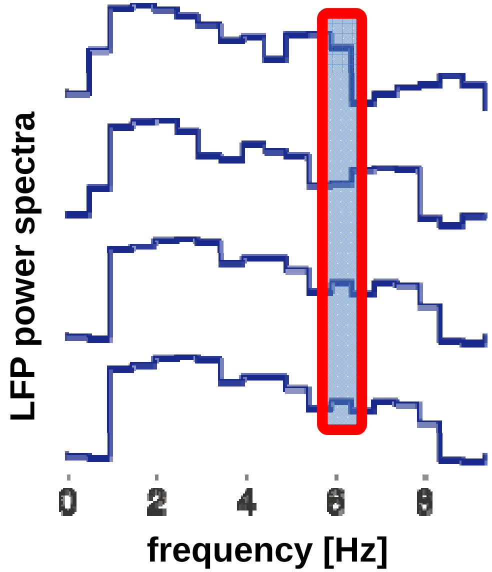


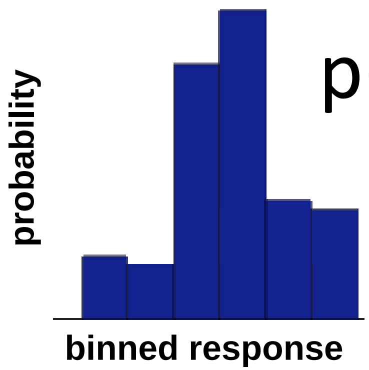
# Conclusion about the spectral investigation

- At least for some frequencies, power really appears to convey information about which window is presented.

# Taking the trials into account

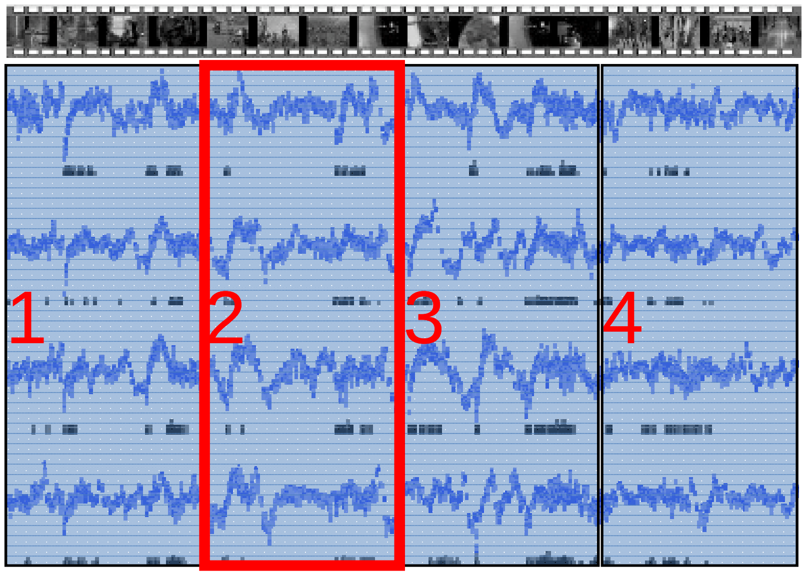


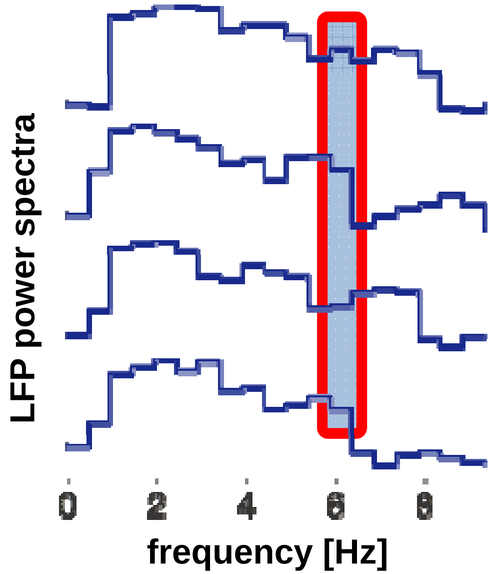



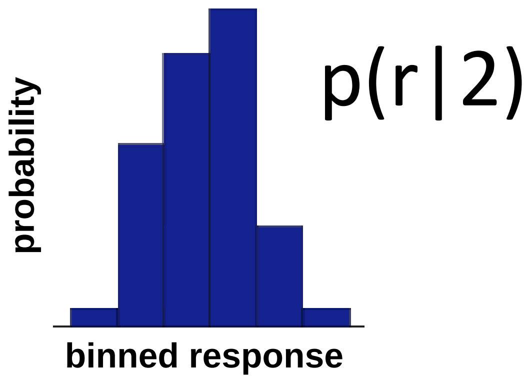




$p(r|1)$

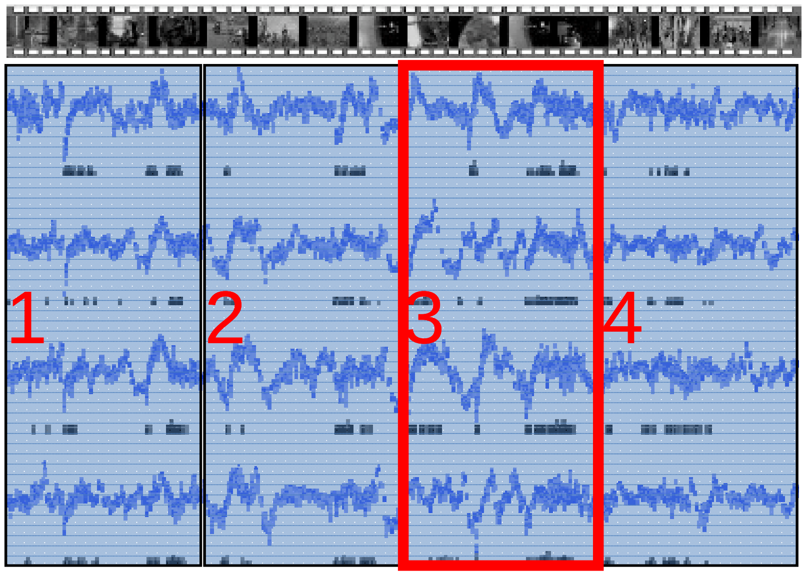
# Taking the trials into account

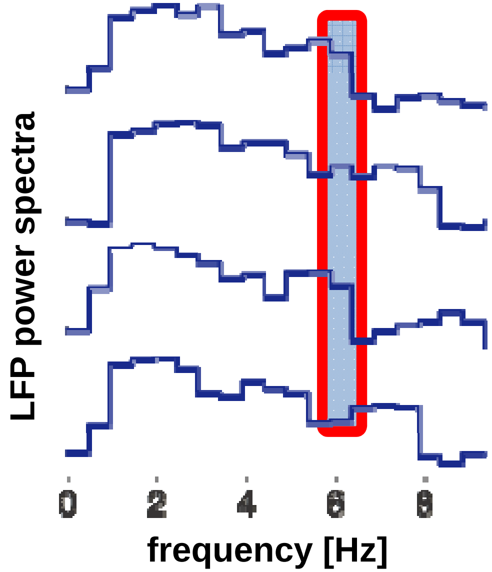


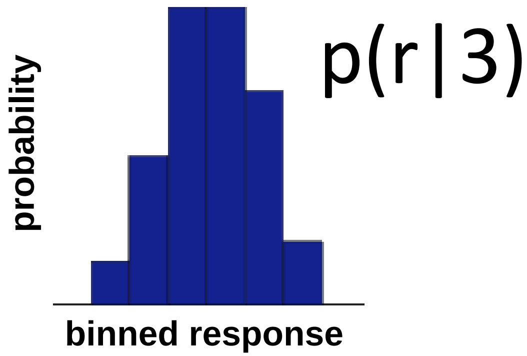



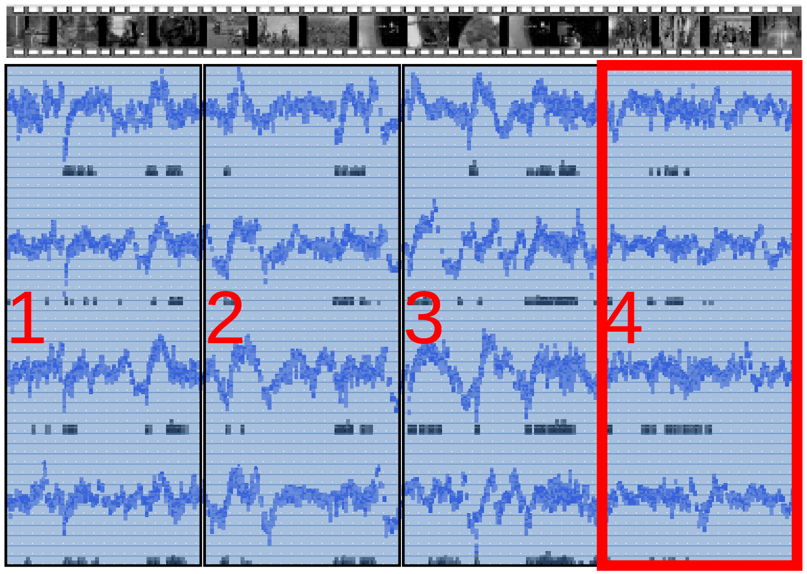
# Taking the trials into account

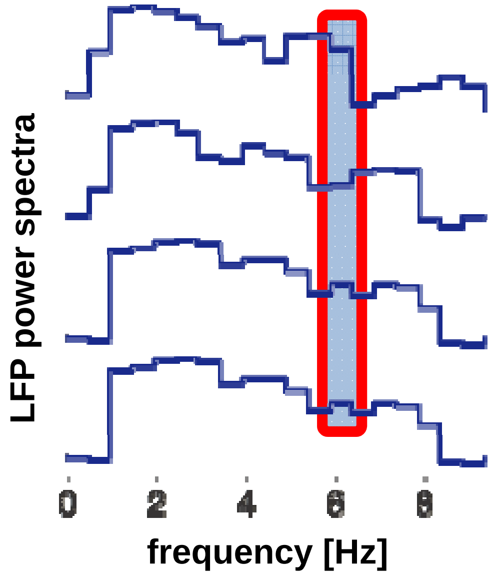


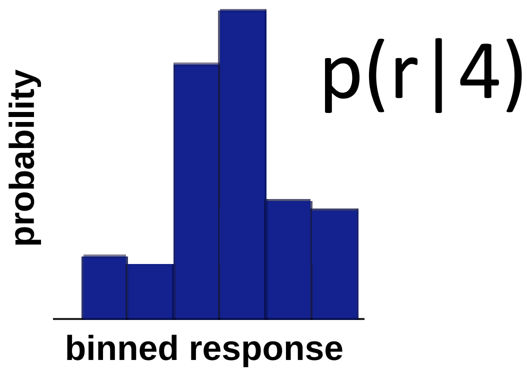



# Taking the trials into account



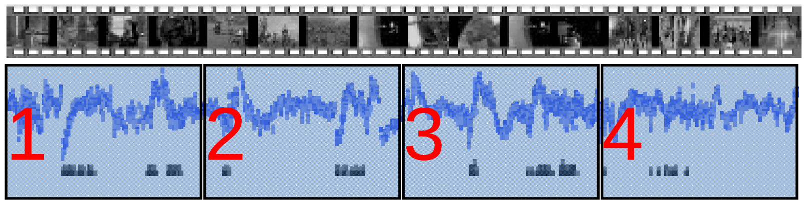



- But then we have everything we need we need to perform an information analysis for the power at each frequency!

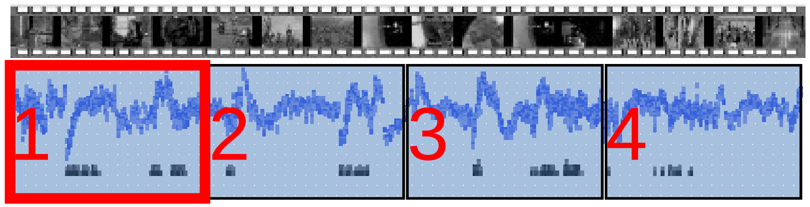
# Performing the frequency information analysis

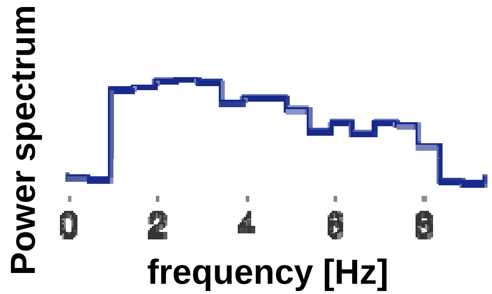
- **STEP 1:**
  - Break each recorded sequence (trial) down into windows of chosen length (e.g. 2 seconds);
  - At this point each window constitutes a "stimulus"



# Performing the frequency information analysis

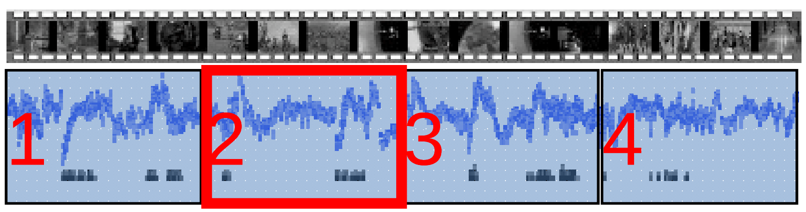
- **STEP 2:**
  - For each window perform spectral analysis using the spectral estimation technique

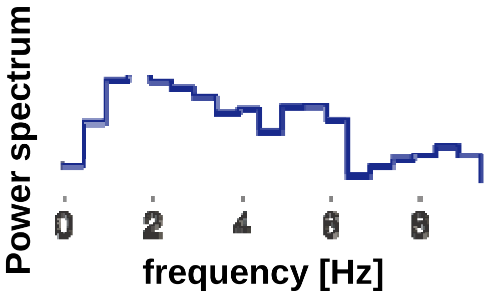


# Performing the frequency information analysis

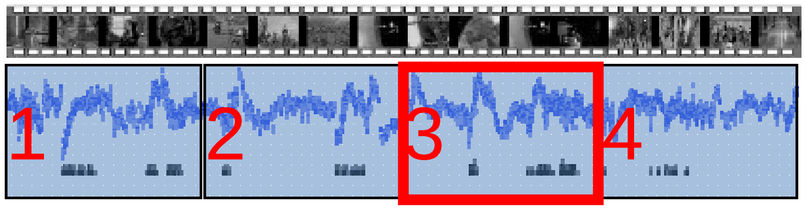
- **STEP 2:**
  - For each window perform spectral analysis using the spectral estimation technique

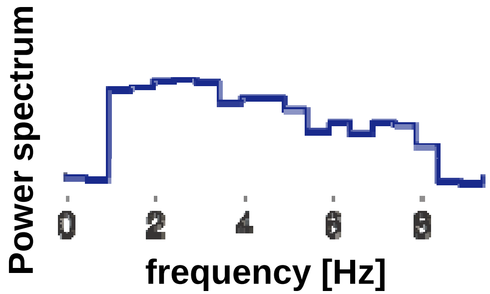


# Performing the frequency information analysis

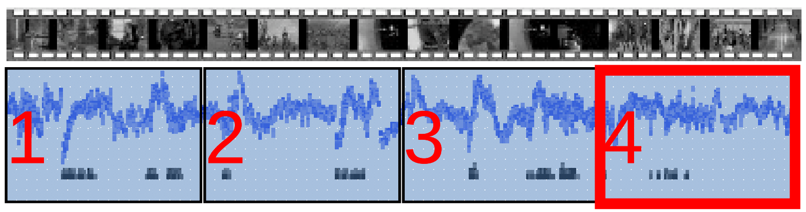
- **STEP 2:**
  - For each window perform spectral analysis using the spectral estimation technique

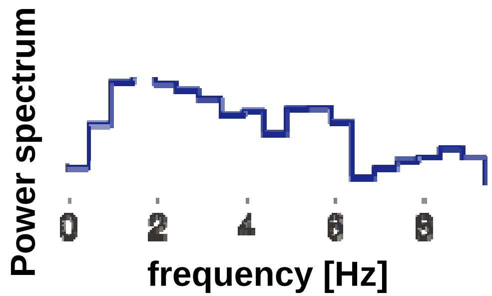


# Performing the frequency information analysis

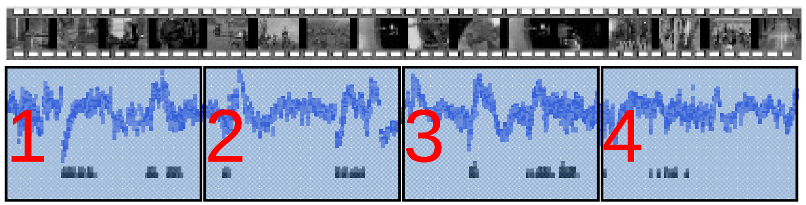
- **STEP 2:**
  - For each window perform spectral analysis using the spectral estimation technique

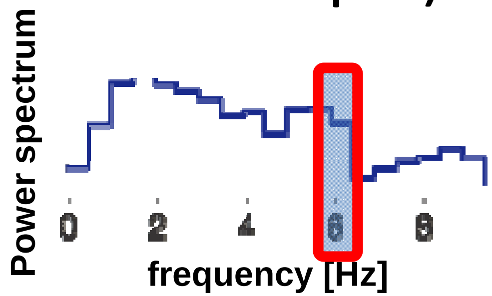


# Performing the frequency information analysis

- **STEP 3:**
  - Fix a frequency: the power values at that frequency represent the responses to the stimuli (the distinct windows)
  - Feed these responses to the toolbox to compute information (use a bias correction technique).



# Performing the frequency information analysis

- **STEP 4:**
  - Repeat for all available frequencies