Spectral Processing Lecture

Chronx: chronux.org

Download and follow instructions to set up.

1) Question: how do we examine continuous data:
   a) Local Field Potentials
   b) Intracellular data
2) Collecting the data
   a) All continuous signals must be sampled
   b) Examples:
      i) Important parameters:
         (1) Sampling frequency *f*
            (a) Number of samples per second
         (2) Filtering
            (a) Frequencies that are present in the signal
               (i) High pass frequency (hpf)
               (ii) Low pass frequency (lpf)
         (3) Nyquist Frequency
            (a) 2 * lpf
         (4) Potential pitfall: Aliasing
            (a) Example:

c) Reconstructing the data:
 i) Convolve with sin(x) / x function
  (1) Matlab: INTERPFQ

3) Properties of the data
 a) Fundamental trade-off between time and frequency resolution
 b) Fourier transform

$$\hat{f}(\xi) = \int_{-\infty}^{\infty} f(t) e^{-2\pi i t \xi} dt$$

 i) where t is time

ξ is frequency

$e^{i\omega} = \cos(\omega) + i\sin(\omega)$

 ii) Basic idea:
  (1) Take a periodic signal and break down in different components.
   (a) Each component has an amplitude and a phase
    (i) Complex representation
  (2) Assumption of stationarity
   (a) The individual components of the signal are stable over a particular window.

 c) Power spectrum
  (1) Assessment of magnitude of signal without reference to phase
  (2) Matlab: Spectrum.method
   (a) Start with Thompson mutlitaper method
  (3) Examples:

 d) Spectrogram
  i) Power spectrum in small windows:
   (1) Matlab: spectrogram
   (2) Chronux: mtspectrogramc
    (a) [S,t,f,Serr]=mtspecgramc(data,movingwin,params)
  ii) Useful when you want to look at only certain parts of the data.

4) Filtering the data
 a) Examine one frequency band.
 b) All smoothing and binning is filtering

c) Two types of filters

- i) FIR
  - (1) “finite impulse response”
  - (2) Function that is convolved with signal to suppress frequencies outside a specified range
  - (3) Many parameters
  - (4) Example:
- ii) IIR
  - (1) “infinite impulse response”
  - (2) Filter that depends on feedback
  - (3) Few parameters
- iii) General form and Matlab code:

FILTER One-dimensional digital filter.
Y = FILTER(B,A,X) filters the data in vector X with the
filter described by vectors A and B to create the filtered
data Y. The filter is a "Direct Form II Transposed"
implementation of the standard difference equation:

a(1)*y(n) = b(1)*x(n) + b(2)*x(n-1) + ... + b(nb+1)*x(n-nb)
- a(2)*y(n-1) - ... - a(na+1)*y(n-na)

  - (1) For FIR, a(2:na+1) = 0
    - (a) No feedback
  - (2) For IIR a(2:na+1) is not always 0
    - (a) Feedback

d) Types of filters

- i) Smoothing
- ii) High pass
- iii) Low pass
- iv) Band pass

e) Generating filters

- i) Smoothing filter
  - (1) Gaussian convolved with signal
    - (a) FFT of Gaussian is Gaussian
      - (i) Modulates power of frequencies in original signal

(ii) Examples

ii) Matlab: fdatool
  (1) Select type of filter and parameters
    (a) E.g. bandpass, FIR, equiripple
  (2) Specify order (number of parameters)
    (a) Usually best to do minimum order for FIR
  (3) Specify frequency bands for filter and attenuation of stop band or bands
    (a) The closer the stop and pass bands, the bigger the filter needs to be
  (4) Examine frequency response
    (a) Measured in dB (decibels)
      (i) Wikipedia: When referring to measurements of power or intensity, a ratio can be expressed in decibels by evaluating ten times the base-10 logarithm of the ratio of the measured quantity to the reference level. Thus, if L represents the ratio of a power value P1 to another power value P0, then LdB represents that ratio expressed in decibels and is calculated using the formula:

$$L_{DB} = 10\log_{10}\left(\frac{P_1}{P_0}\right)$$

      (ii) where P_1 is the level you are interested in
      P_0 is a reference level

      3 decibels = 2 x larger
      10 decibels = 10 x larger
      20 decibels = 100 x larger

f) Filter trade-offs:
  i) Frequency specificity vs. time specificity
    (1) Narrow frequency windows require longer filters
    (2) Filter length is related to temporal specificity

g) Effect of filtering:
  i) All filters have both amplitude profile and a phase profile.
  ii) Filtering results in an frequency dependent alteration of phase
  iii) To avoid, use Matlab command FILTFILT which runs the data through the filter forwards and backwards
    (1) Creates phase offset in forward run
    (2) Reverses in backward run.

h) Hilbert transform
  i) Extracts amplitude envelope and phase of a signal.
  ii) Matlab: HILBERT(x)
    (1) Returns h = real (R) + imaginary (I) components of signal.
    (2) Amplitude = sqrt(R^2 + I^2) = ABS(h)
    (3) Phase = atan2(I,R) = ANGLE(h)

iii) Example:

5) Comparing multiple signals
   a) Coherence
      i) Measures extent to which two frequency bands are in phase
         (1) Matlab [Cxy,F] = MSCOHERE(X,Y,WINDOW,NOVERLAP,NFFT,Fs) returns the magnitude squared coherence computed as a function of physical frequency (Hz). Fs is the sampling frequency specified in Hz. If Fs is empty, it defaults to 1Hz.X and Y are the two signals.

         (2) Chronux: [C,phi,S12,S1,S2,t,f,confC,phistd,Cerr]=cohgramc(data1,data2,movingwin,params)
            (a) Moving window coherence across frequencies