An Introduction to Field Analysis Techniques: The Power Spectrum and Coherence

Mark A. Kramer, PhD

Department of Mathematics and Statistics
Boston University
Boston, Massachusetts

Introduction

As large data sets (e.g., multisensor, high-density recordings) become more prevalent in neuroscience, analysis routines to characterize these data become more essential. Neuronal field data often exhibit rhythms, and spectral analysis techniques provide tools to characterize these rhythms and succinctly summarize important features in these large data sets. In this chapter, we provide a hands-on, nontechnical introduction to some of the spectral analysis material presented in this Short Course. This brief review necessarily provides a limited description of spectral analysis; excellent references exist with many more details (Priestley, 1983). Instead, we focus on case study data available for download at http://math.bu.edu/people/mak/sfn-2013/. Embedded within this chapter is MATLAB code; the reader is encouraged to explore these data and methods on his or her own.

Field Analysis Techniques Step by Step

Introduce single-sensor data: visualization

To start, we focus our analysis on a single field recording. This recording may represent an electro-encephalographic (EEG), magnetoencephalographic (MEG), electrocorticographic (ECoG), or local field potential (LFP) observation. We collect $T = 2$ s of data (sampling frequency $f_0 = 500$ Hz) from a single sensor (Fig. 1A). In this figure, the voltage trace appears as a continuous curve. However, closer inspection reveals that these data consist of discrete points in time (asterisks in Fig. 1B). The spacing between these points is small: In this case, $\Delta = 2$ ms, which corresponds to the reciprocal of the sampling frequency. Visual inspection of Figure 1B suggests rhythmic activity with a

period of ~15 ms. To characterize the rhythms beyond visual inspection, we compute the power spectrum (Fig. 1C). In the next sections, we will introduce the notion of the power spectrum, provide intuition for the method, define important quantities of interest, and introduce the notion of tapering.

Power spectrum defined

There exist many techniques to characterize field data (Pereda et al., 2005; Greenblatt et al., 2012). Here, we compute the power spectrum of the data using a well-established technique: the Fourier transform. To summarize, the "power spectrum" is the magnitude squared of the Fourier transform of the data. The power spectrum indicates the amplitude of rhythmic activity in the data as a function of frequency. Many subtleties exist in computing and interpreting the power spectrum, some of which we will explore here. In doing so, we will strengthen our intuition and our ability to deal with future, unforeseen circumstances in other data sets.

Power spectrum: computation and implementation

We start by presenting the formula and MATLAB code to compute the power spectrum. Throughout the rest of this chapter, we will focus on aspects of this computation in more detail. The power spectrum ($S_{xx,j}$) of a signal x is defined as follows:

$$S_{xx,j} = (2\Delta^2 / T) \, X_j X_j^* ,$$

which is the product of the Fourier transform of x at frequency f_j (X_j) with its complex conjugate (X_j^*), scaled by the sampling interval (Δ) squared and the total duration of the recording (T). Notice the units of the power spectrum are (in this case): $(\mu V)^2/Hz$.

1A

1B

1C
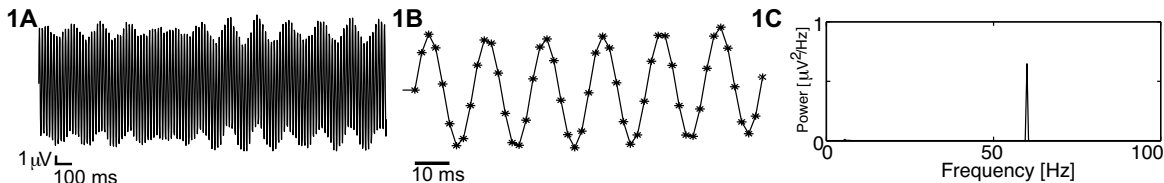
$1\,\mu V$ 100 ms

10 ms

Figure 1. *A,* $T = 2$ s of collected data (sampling frequency $f_0 = 500$ Hz) from a single sensor. The voltage trace appears as a continuous curve. *B,* Closer inspection reveals that these data consist of discrete time points (asterisks). The spacing between these points is small: $\Delta = 2$ ms, corresponding to the reciprocal of the sampling frequency. Activity with a period of ~15 ms is apparent. *C,* Plot of the power spectrum, which displays the power as a function of frequency.

Computing the power spectrum in MATLAB and plotting the results require only a few lines of code:

```
xf = fft(x);                    %1. Compute the Fourier transform
                                   of x.

Sxx = 2*dt^2/T * xf.*conj(xf);  %2. Compute the power spectrum.

Sxx = Sxx(1:length(x)/2+1);     %3. Ignore negative frequencies.

df = 1/max(T);                  %4. Determine the frequency
                                   resolution.

fNQ=1/dt/2;                     %5. Determine the Nyquist
                                   frequency.

faxis = (0:df:fNQ);             %6. Construct the frequency axis.

plot(faxis, Sxx)                %7. Plot power versus frequency.

xlim([0 100])                   %8. Select frequency range.

xlabel('Frequency [Hz]');       %9. Label axes.
ylabel('Power')
```

The results of this computation are plotted in Figure 1C. Notice the large peak in power at 60 Hz. This peak agrees with our visual inspection of the EEG data (Fig. 1B), in which a dominant rhythm at 60 Hz can be approximated. In subsequent sections, we will explore some subtleties of the power spectrum and strengthen our intuition for this measure.

Power spectrum: intuition

The power spectrum is proportional to the squared Fourier transform of the data. We may think of the Fourier transform as "comparing" the data x to sinusoids oscillating at difference frequencies f_j. When the data and sinusoids "match," the power at frequency f_j is large, whereas when the data and sinusoids do not match, the power at frequency f_j is small. To illustrate this principle, we consider an example in which the data are a perfect cosine function with frequency 10 Hz (Fig. 2A, gray). Choosing $f_j = 4$ Hz, we construct another cosine function (Fig. 2A, red) oscillating at 4 Hz. To calculate the power in the data at 4 Hz, we multiply the data (Fig. 2A, gray) by the sinusoid (Fig. 2A, red) at each point in time, then sum the

result. This point-by-point multiplication is plotted in Figure 2B. Notice that the product alternates between positive and negative values over time. Therefore, when we sum the product (i.e., when we sum the red curve in Fig. 2B over time), we expect a value near zero. In this case, the sinusoid at frequency $f_j = 4$ Hz does not align with the data, and the power at this frequency is nearly zero.

Now consider the case in which we choose a cosine function at frequency $f_j = 10$ Hz. With this choice of f_j, the data and the cosine function align perfectly (Fig. 2C). The product of the cosine function and the data is always nonnegative (Fig. 2D); therefore, the summation is a large positive number, and the power in the data at frequency $f_j = 10$ Hz is also large. In this sense, the power spectrum reveals the dominant frequencies that "match" the data.

Important quantities: frequency resolution and Nyquist frequency

Two important quantities to consider when computing the power spectrum are as follows:

1. The frequency resolution, $df = 1/T$, is the reciprocal of the total recording duration.

2. The Nyquist frequency, $f_{NQ} = f_0/2 = 1/(2 \Delta)$, is half of the sampling frequency f_0.

For the data considered here, the total recording duration is 2 s ($T = 2$ s), so the frequency resolution $df = 1/(2$ s$) = 0.5$ Hz. We can therefore resolve frequency differences of 0.5 Hz, but no smaller. To improve the frequency resolution (i.e., make df smaller), we must increase the duration of recording (i.e., make T bigger). The sampling frequency f_0 is 500 Hz, so $f_{NQ} = 500/2$ Hz $= 250$ Hz. We can therefore observe frequencies up to 250 Hz, but no higher. To increase the highest frequency observable, we must increase the sampling frequency.
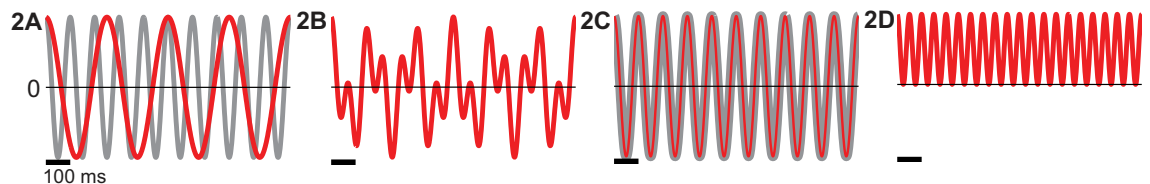
Figure 2. Example intuition for computing the power spectrum. *A,* The data consist of a perfect cosine function with frequency 10 Hz (gray). We choose $f_j = 4$ Hz, a cosine function (red) that oscillates at 4 Hz. *B,* Plotted point-by-point multiplication for the two curves in *A*. The product alternates between positive and negative values over time. *C,* We choose another cosine function (red) at frequency $f_j = 10$ Hz, which aligns perfectly with the data (gray). *D,* The product of this cosine function and the data is always nonnegative. Calibration: *A–D,* 100 ms.

MATLAB relates the indices of vector S_{xx} (line 2 of MATLAB code) to the frequencies as shown in Figure 3. Because the field data are real (i.e., the observed data have zero imaginary components), the negative frequencies are redundant. We therefore ignore the second half of the frequency axis (line 3 of MATLAB code) and define a frequency axis in MATLAB that spans 0 to f_{NQ} in steps of df (Fig. 3).

The impact of aliasing

The Nyquist frequency is the highest frequency we can hope to observe in the data. To illustrate this fact, we consider a simple example data set: a sinusoid that oscillates at some frequency f_s. We do not observe these true data. Instead, we observe a sampling of these data that depends on our sampling interval Δ. If we sample the data at a high rate, $f_0 \gg f_s$, then we can accurately reconstruct the underlying data (Fig. 4A) given only the discrete samples. However, if we sample

the data at a lower rate, such that $f_0 < 2f_s$, the sampling produces an oscillation occurring at a different, lower frequency (Fig. 4B). This phenomenon of a true high-frequency signal appearing as a low-frequency signal upon sampling is known as "aliasing."

The decibel scale

Often, weak rhythms of interest remain hidden from visual inspection because of large peaks at other frequencies in the power spectrum. One visualization technique to emphasize lower-amplitude rhythms is to change the scale of the power spectrum to decibels. The decibel is a logarithmic scale and is easily computed in MATLAB (Fig. 5A).

The default rectangular taper

By doing nothing, we automatically apply a rectangular taper to the data (Fig. 5B, red). The rectangular taper multiplies the observed data by 1 and

3

Index	1	2	3	4	...	N/2-1	N/2	N/2+1	N/2+2	N/2+3	...	N-1	N
Frequency	0	df	2df	3df	...	f_{NQ}- 2df	f_{NQ}- df	f_{NQ}	-(f_{NQ}- df)	-(f_{NQ}- 2df)	...	-2df	-df

Ignore

Figure 3. Relationship between the indices of vector S_{xx} to the frequencies. Because the field data are real, the negative frequencies are redundant.

Figure 4. Illustration of aliasing. *A,* A sinusoid that oscillates at frequency f_s (black) with sampling (green) at a high rate, $f_0 \gg f_s$. *B,* Sampling (red) of the data at a lower rate, $f_0 < 2f_s$, produces an oscillation at a different, lower frequency, i.e., "aliasing."

Figure 5. *A,* The power spectrum of the data in Fig. *A* plotted on a decibel scale emphasizes lower-amplitude rhythms. *B,* A rectangular taper (red) applied to the data that multiplies the observed data by 1 and the unobserved data by 0. The data continue perpetually, although only a small interval (lower trace) is observed.

the unobserved data by 0. We can think of the value 1 as representing the time period when our recording device is operational; the data continue "forever" (Fig. 5B, upper trace), but we observe only a small interval (Fig. 5B, lower trace). The rectangular taper makes explicit our knowledge about the observed data (in this case, the 2 s interval) and our ignorance about the unobserved data, which are assigned the value zero. By computing the power spectrum of the 2 s of data, we actually compute the power spectrum as the product of two functions: the observed data and the rectangular taper. The impact on the power spectrum is the emergence of "side lobes"—regions of increased power that surround spectral peaks. These side lobes can potentially mask important, lower-power activity (Fig. 5A).

Impact of the Hanning taper

The Hanning taper acts to smooth the sharp edges of the rectangular taper. The Hanning taper gradually increases from zero, reaches a maximum of 1 at the center of the taper, then gradually decreases to zero (Fig. 6A, blue). Notice that data at the edges of the taper become dramatically reduced (Fig. 6A, lower). The power spectrum (Fig. 6B) possesses two main differences: (1) The peaks are wider when using the Hanning taper compared with the rectangular taper (Fig. 5A), and (2) the side lobes are reduced when using the Hanning taper compared with the rectangular taper. These two features illustrate

the tradeoff between the two window choices. By accepting wider central peaks, we reduce the power in the side lobes. To compute and apply the Hanning window in MATLAB, we must replace line 1 of the MATLAB code with the following:

```
>> xf = fft(hann(length(x)).*x);    %1. Apply Hanning taper to x, then
                                         compute Fourier transform.
```

Note that two peaks at low frequency become more apparent after applying the Hanning taper (compare Fig. 5A with 6A at frequencies <20 Hz). Modern approaches to tapering include the multitaper method (Thomson, 1982; Bokil et al., 2010).

A measure of association: coherence

Thus far, we have focused on field data recorded from a single sensor. However, brain recordings often consist of multiple sensors, and recent advances in recording technology promise observations of brain activity from many sensors simultaneously (Viventi et al., 2011). How do we make sense of these large, simultaneous, multivariate recordings? Many techniques exist (Pereda et al., 2005; Greenblatt et al., 2012). Here, we focus on field data and consider time series recorded simultaneously from two sensors during a task. To characterize these data, we compute the coherence, which has many applications in neuroscience (Engel et al., 2001).
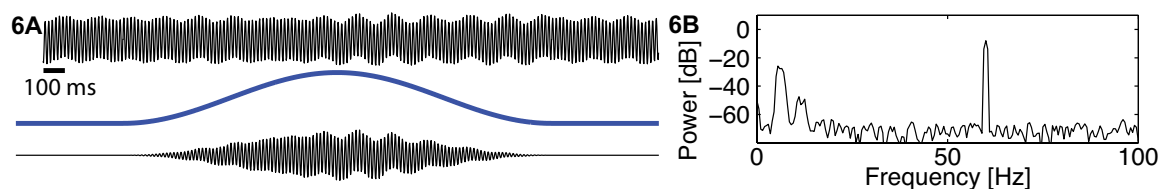

Figure 6. *A,* The Hanning taper (blue) smoothes the sharp edges of the rectangular taper, going from zero, up to 1, and back down to zero. Data at the edges of the taper become dramatically reduced (lower trace). *B,* The power spectrum using the Hanning taper possesses wider peaks and reduced side lobes compared with the rectangular taper (Fig. 5*A*).
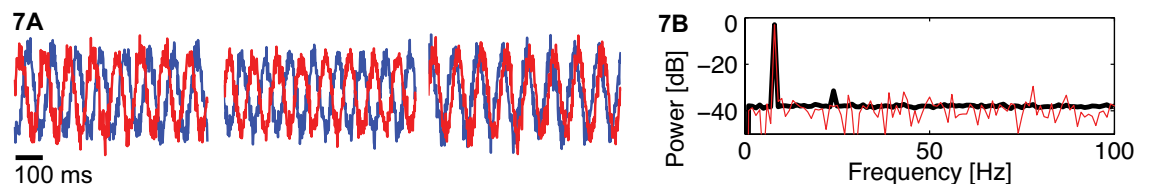

Figure 7. *A,* Example data visualized from the first three trials for both sensors (red and blue), suggesting rhythmic activity. *B,* The trial-averaged power spectrum (black). Compared with the power spectrum from a single trial (red), the variability of the power is greatly reduced. A large peak in power is seen at 8 Hz and a smaller peak at 24 Hz.

Visualization and trial-averaged power spectrum

The data consist of 100 trials, each of 1 s duration, recorded simultaneously from two sensors. To start, we visualize the data in the first three trials for both sensors (Fig. 7A, red and blue). The results suggest rhythmic activity. To further characterize the rhythmic activity, we compute the trial-averaged power spectrum for a single sensor (Fig. 7B, black). Because we possess multiple trials, and we assume that each trial represents an instantiation of the same underlying process, we average the power spectra across trials to compute the trial-averaged power spectrum. Compared with the power spectrum from a single trial (Fig. 7B, red), the variability of the power is greatly reduced. By reducing the variability in this way, interesting structure in the data becomes more apparent. In this case, we observe a large peak in power at 8 Hz and a smaller peak at 24 Hz.

Coherence defined

To assess the association between activity recorded at the two sensors (which we label x and y), we compute the coherence. To do so, we first define the cross-spectrum (Fig. 8A). Compared with the expression for power

(discussed earlier in Power Spectrum: Computation and Implementation), we replace X_j^* with $Y_{j,k}^*$. That is, we replace the Fourier transform of x with the Fourier transform of y. Notice that we also include the trial index (subscript k), sum the product of $X_{j,k}$ and $Y_{j,k}^*$ over all trials, and then divide by the total number of trials K. Using polar coordinates, we may write this expression in a slightly different way (Fig. 8B). Here, $A_{j,k}$ ($B_{j,k}$) is the radius at frequency index j and trial index k for the signal x_k (y_k), and $\Phi_{j,k}$ is the phase difference between the two signals at frequency index j and trial index k. At last, we define the coherence $\kappa_{xy,j}$ in Figure 8C; the symbol $< S_{xy,j} >$ indicates the magnitude of the trial-averaged cross-spectrum. In other words, the coherence is the magnitude of the trial-averaged cross-spectrum between the two signals at frequency index j, divided by the magnitude of the trial-averaged power spectrum of each signal at frequency index j. We can evaluate this expression by replacing the trial-averaged spectrum terms with the corresponding expressions in polar coordinates (Fig. 8D).

Coherence: intuition

To gain intuition for the behavior of $\kappa_{xy,j}$, we make the simplifying assumption that the amplitude at each

8A
$$< S_{xy,j} > = \frac{2\Delta^2}{T} \frac{1}{K} \sum_{k=1}^{K} X_{j,k} Y_{j,k}^*$$

8B
$$= \frac{2\Delta^2}{T} \frac{1}{K} \sum_{k=1}^{K} A_{j,k} B_{j,k} \exp(i\Phi_{j,k})$$

8C
$$\kappa_{xy,j} = \frac{|< S_{xy,j} >|}{\sqrt{< S_{xx,j} >} \sqrt{< S_{yy,j} >}}$$

8D
$$= \frac{|\sum_{k=1}^{K} A_{j,k} B_{j,k} \exp(i\Phi_{j,k})|}{\sqrt{\sum_{k=1}^{K} A_{j,k}^2} \sqrt{\sum_{m=1}^{K} B_{j,m}^2}}$$

8E *(assume identical amplitudes)*
$$= \frac{1}{K} |\sum_{k=1}^{K} \exp(i\Phi_{j,k})|$$

Figure 8. Mathematical expressions for the coherence between activity recorded at two sensors (*x* and *y*). *A*, Equation for the cross-spectrum. B, Equation for the cross-spectrum in polar coordinates. *C,* Equation for the coherence. *D,* Equation for the coherence in polar coordinates. *E,* Simplified expression for the coherence in which the amplitudes in the two signals are assumed identical for all trials.
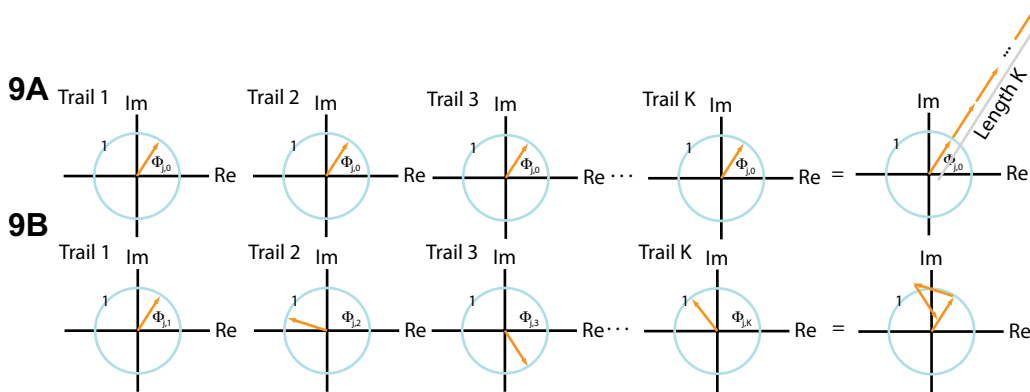

Figure 9. The vector in the complex plane with radius 1, defined by the expression exp(*i* $\Phi_{j,0}$), across trials for two scenarios. *A,* In the first scenario, the vector points in the same direction for each trial (first and middle columns). Summing up these vectors end to end produces a long vector in the complex plane that terminates far from the origin (last column). *B,* In the second scenario, the phase difference $\Phi_{j,k}$ can assume any value between 0 and 2π for each trial. The vectors point in different directions from trial to trial, and the sum of these vectors remains near the origin.

frequency index j is identical for both signals and all trials; that is, $A_{j,k} = B_{j,k} = C_j$. Then the expression for coherence (Fig. 8D) becomes Figure 8E. In this case, the coherence simplifies to an expression that involves only the phase difference between the two signals averaged across trials; the amplitudes in the numerator and denominator have canceled out.

We now interpret this simplified expression in two scenarios. First, we assume that at some frequency index j, the two signals possess a constant phase difference across trials. We therefore replace $\Phi_{j,k}$ with $\Phi_{j,0}$ because the phase difference is the same for all trials k. Now consider the expression: $\exp(i\,\Phi_{j,0})$. This defines a point in the complex plane with radius 1, which we can think of as a vector leaving from the origin at angle $\Phi_{j,0}$ to the real axis (Fig. 9A, first column). To compute the coherence $\kappa_{xy,j}$ requires the summation of these terms across trials (Fig. 8E). This defines a sum of vectors in the complex plane, each of radius 1. Because the phase difference is the same for each trial, this vector points in the same direction for each trial (Fig. 9A, middle columns). By summing up these vectors end to end, we produce a long vector in the complex plane that terminates far from the origin (Fig. 9A, last column). The coherence (Fig. 8E) is this vector length, divided by K, so we conclude in this case that $\kappa_{xy,j} = 1$, which indicates strong coherence between the two signals. The strong coherence results from the constant phase relationship between the two signals across trials at frequency index j.

As a second scenario, consider another frequency index j in which the two signals have a random phase difference over trials. In this case, the phase difference ($\Phi_{j,k}$) can assume any value between 0 and 2π for each trial. To visualize this, we examine the phase differences in the complex plane (Fig. 9B); the vectors point in different directions from trial to trial. Because the coherence (Fig. 8E) is this vector length, divided by K, we conclude in this case that $\kappa_{xy,j} \approx 0$, which indicates no coherence between the two signals. The weak coherence results from the random phase relationship over trials between the two signals at this frequency index.

To summarize, coherence is a measure of the relationship between x and y at the same frequency. The coherence ranges between 0 and 1, $0 \leq \kappa_{xy,j} \leq 1$, in which 0 indicates no coherence between signals x and y at frequency index j, and 1 indicates strong coherence between signals x and y at frequency index j. The coherence is a measure of the phase consistency between signals at frequency index j across trials. We note that, because the coherence requires the Fourier transform, the issues of frequency resolution, Nyquist frequency, and tapering (Bokil et al., 2010) are identical to those described for the power spectrum: The frequency resolution of the coherence is $1/T$, and the Nyquist frequency is half of the sampling frequency, just as before.

Coherence: computation and interpretation

There are a variety of alternatives for computing coherence. Here we compute the coherence by implementing the mathematical expressions in MATLAB:

```
K = size(x,1);                    %Define the number of trials.

N = size(x,2);                    %Define the number of indices
                                    per trial.

dt = t(2)-t(1);                   %Define the sampling interval.
T = t(end);                       %Define the duration of data.
Sxx = zeros(K,N);                 %Create variables to save the
                                    spectra.

Syy = zeros(K,N);

Sxy = zeros(K,N);

for k=1:K                         %Compute the spectra for each
                                    trial.

  Sxx(k,:) = 2*dt^2/T * fft(x(k,:)) .* conj(fft(x(k,:)));

  Syy(k,:) = 2*dt^2/T * fft(y(k,:)) .* conj(fft(y(k,:)));

  Sxy(k,:) = 2*dt^2/T * fft(x(k,:)) .* conj(fft(y(k,:)));

end
Sxx = Sxx(:,1:N/2+1);             %Ignore negative frequencies.
Syy = Syy(:,1:N/2+1);
Sxy = Sxy(:,1:N/2+1);
Sxx = mean(Sxx,1);                %Average the spectra across
                                    trials.

Syy = mean(Syy,1);

Sxy = mean(Sxy,1);
cohr = abs(Sxy) ./ (sqrt(Sxx) .* sqrt(Syy));
                                  %Compute the coherence.

df = 1/max(T);                    %Determine the frequency
                                    resolution.

fNQ = 1/ dt / 2;                  %Determine the Nyquist
                                    frequency.

faxis = (0:df:fNQ);               %Construct frequency axis.
plot(faxis, real(cohr));          %Plot the results
xlim([0 50]); ylim([0 1])         %Set the axes limits
xlabel('Frequency [Hz]')          %Label axes.
ylabel('Coherence [ ]')
```
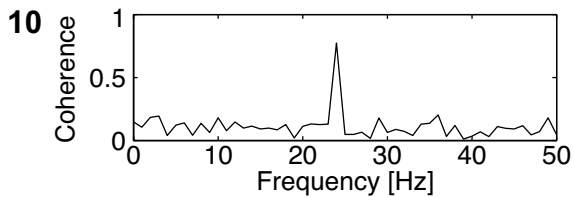
10

Figure 10. The coherence between *x* and *y* is strong at 24 Hz and weak at all other frequencies.

We plot the coherence between *x* and y in Figure 10. We find in this case strong coherence at 24 Hz and weak coherence at all other frequencies. Comparing the power spectrum (Fig. *7B*) with the coherence (Fig. *10*), we find that the dominant rhythm (8 Hz) is not coherent between the two sensors, whereas the weaker rhythm (24 Hz) is coherent.

Conclusion

This chapter provides a brief introduction to the power spectrum and coherence. As "big data" become increasingly common in neuroscience, computational tools to assess interesting structures within time series, as well as relationships between simultaneously recorded time series, will become more critical.

References

Bokil H, Andrews P, Kulkarni JE, Mehta S, Mitra PP (2010) Chronux: a platform for analyzing neural signals. J Neurosci Methods 192:146–151.

Engel AK, Fries P, Singer W (2001) Dynamic predictions: oscillations and synchrony in top-down processing. Nat Rev Neurosci 2:704–716.

Greenblatt RE, Pfliegera ME, Ossadtchic AE (2012) Connectivity measures applied to human brain electrophysiological data. J Neurosci Methods 207:1–16.

Pereda E, Quiroga RQ, Bhattacharya J (2005) Nonlinear multivariate analysis of neurophysiological signals. Prog Neurobiol 77:1–37.

Priestley MB (1983) Spectral analysis and time series. Vols. 1 & 2. Probability and mathematical statistics (Birmbaum ZW, Lukacs E, eds). San Diego, CA: Academic Press.

Thomson DJ (1982). Spectrum estimation and harmonic analysis. Proc IEEE 70:1055–1096.

Viventi J, Kim DH, Vigeland L, Frechette ES, Blanco JA, Kim YS, Avrin AE, Tiruvadi VR, Hwang SW, Vanleer AC, Wulsin DF, Davis K, Gelber CE, Palmer L, Van der Spiegel J, Wu J, Xiao J, Huang Y, Contreras D, Rogers JA, et al. (2011) Flexible, foldable, actively multiplexed, high-density electrode array for mapping brain activity *in vivo*. Nat Neurosci 14:1599–1605.