



CS/NEUR125 Brains, Minds, and Machines

Lab 10: Neural synchronization: power and coherence spectrum

Due: Wednesday, April 19

In this lab, you'll learn about basic "spectral analysis" tools for describing neural synchronization, which has been proposed to be involved in distinct ways in multiple cognitive processes like attention, working memory, and "binding" sensory features into unified perceptions. As we'll see in our Journal Club paper on "Object-based Attention," rhythmic synchronization may enhance the communication between distant brain areas that need to coordinate--in this case the coordination is for the purpose of attending to a particular stimulus type, but neural coherence may be a more general mechanism by which the brain flexibly routes information flow between different brain modules. The spectral analysis tools we'll explore in this lab give us a way to quantify neural synchronization so that we can systematically investigate how it may or may not relate to attention, perception, or any forms of cognition that require coordination of large groups of distributed neurons.

To begin, create a copy of this Google document, modify the title of the copy to include your partner names, and share the copy between partners, as you did in previous labs. The questions for you to answer as you progress through the lab are [shown in blue](#).

[Start MATLAB](#) on your lab Mac and use **Fetch** to download the folder named **spec_Lab** from the CS file server to the Desktop on your Mac. You will find this folder inside the cs125/download folder in your individual account on the CS server. For this lab, set the **Current Folder** in MATLAB to the **spec_Lab** folder.

To motivate learning about ways to describe neural synchronization, notice in Figure 1 below how the EEG recording ("electroencephalogram") looks different in different states of consciousness. When the EEG trace shows a regular **oscillation**, like the *alpha waves* in the drowsy condition, or the large-amplitude *delta waves* of deep sleep, we can estimate the frequency of the oscillation by counting how many peaks occur in a one-second time period. In other words, the frequency in Hertz is the number of peaks per second, or cycles per second.

But to be able to describe more subtle differences, like the difference between the awake example and the stage 1 sleep example, we need a more quantitative way to describe the EEG traces. And similarly, our peak-counting approach (above) won't work on more complicated signals that involve more than just a regular oscillation with a single frequency. For example in the waking example in Figure 1 we can see that sometimes the trace wiggles up and down quickly (a higher frequency) and sometimes it wiggles more slowly (a lower frequency). Similarly, if you look at the "K-complex" pattern in the stage 2 sleep example trace, you can see a big slow wave with a smaller faster wiggle "riding" inside the big wave--that pattern involves more than one frequency at the same time. So, our initial reason for learning about the power spectrum (below) is to have a more precise and quantitative way to describe neural signals like these EEG traces.

Awake – low voltage – random, fast



Drowsy – 8 to 12 cps – alpha waves



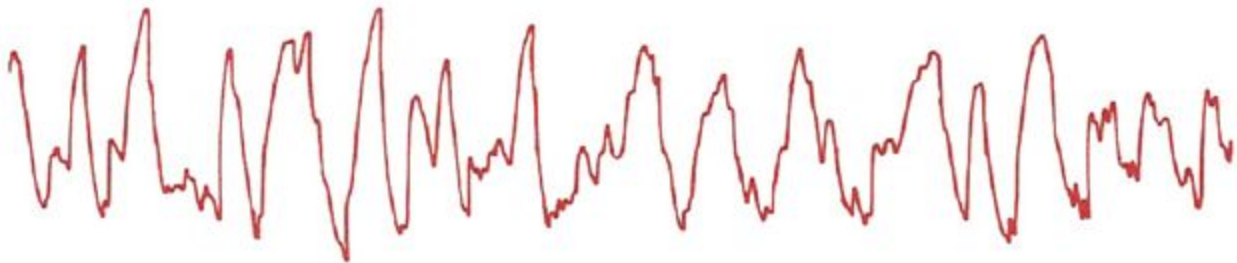
Stage 1 – 3 to 7 cps – theta waves



Stage 2 – 12 to 14 cps – sleep spindles and K complexes



Delta Sleep – 1/2 to 2 cps – delta waves $>75 \mu$ V



REM Sleep – low voltage – random, fast with sawtooth waves



Figure 1: EEG examples recorded in different states of waking consciousness and sleep.
https://www.researchgate.net/profile/Todd_Swick/publication/7522244/figure/fig1/AS:277776197275664@1443238414431/Fig-1-EEG-patterns-from-wakefulness-into-drowsiness-and-then-into-stages-1-through-4.png

But what does this have to do with neural synchronization? The short answer is that any movement of the EEG signal requires the coordinated--i.e. synchronized--electrical activity of many neurons near the recording electrode. So when we see a regular oscillation of the EEG signal like the alpha waves in the drowsy example below, that means that many neurons are turning "on" and "off" together, at a frequency around 10 Hz. One could even say that the EEG signal measures local neural synchronization.

I. Power spectrum, spectrogram, and coherence spectrum

Joseph Fourier proved that any time varying signal can be written as a sum of sine waves of different frequencies¹. A **power spectrum** is a plot or a mathematical function that quantifies *how much of each frequency*² is present in a mixed signal like the EEG traces in Figure 1. So if a signal is almost a perfect sine wave, with frequency f , we would expect its power spectrum function to have a peak at that frequency (on the x-axis), and low values at all other frequencies. The height of that peak at f , measured on the y-axis in terms of "power," is proportional to the *amplitude* of the sine wave, *squared*. In other words, a larger amplitude sine wave is reflected as a larger power at that sine wave's frequency--i.e. a taller peak at frequency f .

Let's see this for ourselves by making a sine wave in MATLAB and plotting its power spectrum.

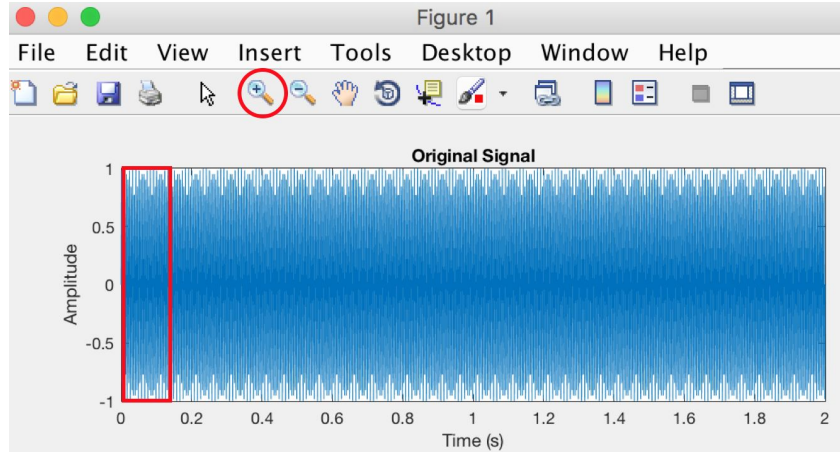
powerSpec.m: Run this script to plot a sine wave and play it as sound. Note the power spectrum plot with its horizontal frequency axis and power plotted on the vertical axis. Again, this is a way to show how much of each frequency is in the sine wave signal you made. Since it is completely made up of one frequency we see a peak at that frequency and no power at any other frequency.

In the subplot showing the signal, zoom in on the window of time from 0 to 0.1 seconds of the horizontal axis and count peaks of the oscillation to verify the frequency. To zoom in, first click on the magnifying-glass icon in the menu bar at the top of the figure window (circled in red in the figure below), move your mouse to the upper left corner of the signal plot, drag your mouse to create a rectangular box surrounding the initial part of the sine-wave signal (also highlighted in red below), and then release the mouse. Click the magnifying-glass icon again to complete the zoom. You can click on the hand (pan) icon to shift the enlarged signal to the left or right.

Q1. Find the frequency specified in the code. Based on that frequency, how many peaks do you expect to see between 0 and 0.1 seconds? How many do you actually count?

¹ Granted, to get a *perfect* reconstruction of an arbitrary signal you might have to include an *infinite* number of different sine waves in your sum. But a good approximation using a finite number of sine waves can still be very useful.

² An **amplitude spectrum** is almost the same thing as a power spectrum. Both plot peaks at the dominant frequencies in a time-varying signal; the power is the amplitude squared.



- In this example we are using the function “pwelch” to compute the power spectrum. The simplest way to use this function to plot the power spectrum of a signal called sig, sampled at a sampling frequency of fs, is:

```
pwelch(sig,[],[],[],fs)
```

Now add a spectrogram plot by uncommenting the relevant lines in the m-file. A **spectrogram** is a way to display a signal’s power spectrum over time, so that one can see how the contribution of different frequencies changes over time. The color (z-axis) on the spectrogram corresponds to the y-axis of the power spectrum.

- The spectrogram function can be called using the following syntax:

```
[s,f,t,p] = spectrogram(sig>window,noverlap,[],fs);
```

In the outputs, p is the spectrogram we are plotting, f is its frequency axis, and t is its time axis (s is another version of the spectrogram). Window sets how long our sliding-time-window should be (in samples) and noverlap is the number of samples of overlap between adjacent windows.

When you run the script make sure to arrange the figure windows so that you can see both of them.

Find the variable that sets the frequency of the sine wave and run the script for frequency 4 Hz, 44 Hz, 220 Hz, and 440 Hz. For a low frequency you can see all the peaks and troughs in the signal in the plot, but you can’t really hear it as a sound. On the other hand when the frequency gets too high you can’t see the signal without zooming in on a shorter period of time, but you can hear the signal as a sound. (At the *sampling rate*, fs = 1000 Hz, set in the code, you can’t use a frequency greater than or equal to 500 Hz.) *What do you observe about the relationship between the 220 and 440 Hz frequencies and their corresponding sounds?*

Q2. What happens to the plots and the sound if you change the amplitude of the signal wave?

Add multiple frequencies to observe how the signal, power spectrum, and spectrogram changes. Try this both for low frequencies (to see the signal) and high frequencies (to hear the signal).

Q3. Paste an excerpt from your modified `powerSpec.m` script into this document to show how you created a second sine wave and analyzed and plotted the combined signal. Notice how the power spectrum and spectrogram represents your changes to the signal.

Q4. Now instead create a signal which consists of a sine wave followed by a sine wave of a different frequency, and note how the power spectrum and spectrogram change. Of these two types of plots, which one does NOT show whether the two different frequencies happened simultaneously as opposed to one after the other? *Hint: instead of adding sine waves of two frequencies, this time you want the first part of your signal vector to hold a sine wave of one frequency, and the second (later) part of the signal vector to hold a sine wave of a different frequency.*

Spectrograms of sounds are known as **sonograms**. Use the `isignal` variable in the `powerSpec.m` file script to view the signals, power spectra, and spectrogram for segments of three different natural sounds: the Hallelujah chorus from Handel's Messiah, human speech, and bird song. Notice where the frequencies of the neural signals in Figure 1 are on the frequency axis, as compared to the frequencies of audible sounds.

Q5. For each sound example, where are the peak frequencies in the power spectrum and spectrogram?

Q6. For the bird song example, add a line of code so that you only play, analyze, and plot a single "cheep" from the over-all song. Paste that line into this document to show us how you did it. *Hint: if you want the part of the song `y_bird` that starts at 1.3 seconds, you need to multiply 1.3 seconds times the sampling frequency to determine which element of the vector `y_bird` correspond to 1.3 seconds.*

When we have two signals, say from different parts of the brain, the **coherence spectrum** gives us a way to quantify how *coherent*, or *synchronized*, they are with each other *at each frequency*. More precisely, "coherent" is a looser-than-usual meaning of "synchronized," because two sine waves with a constant offset between their peaks are just as *coherent* as two sine waves whose peaks line up perfectly with zero "phase lag." So another way of defining coherence is as a "consistent phase relationship" between two signals, at a particular frequency.

Like a power spectrum, a coherence spectrum has frequency on its horizontal axis, because it represents how coherent the two signals are *at each possible frequency* of oscillation. The vertical axis represents the magnitude of the coherence at each frequency as a **value between zero and one (with zero meaning no consistent phase relation and one meaning perfect coherence)**.

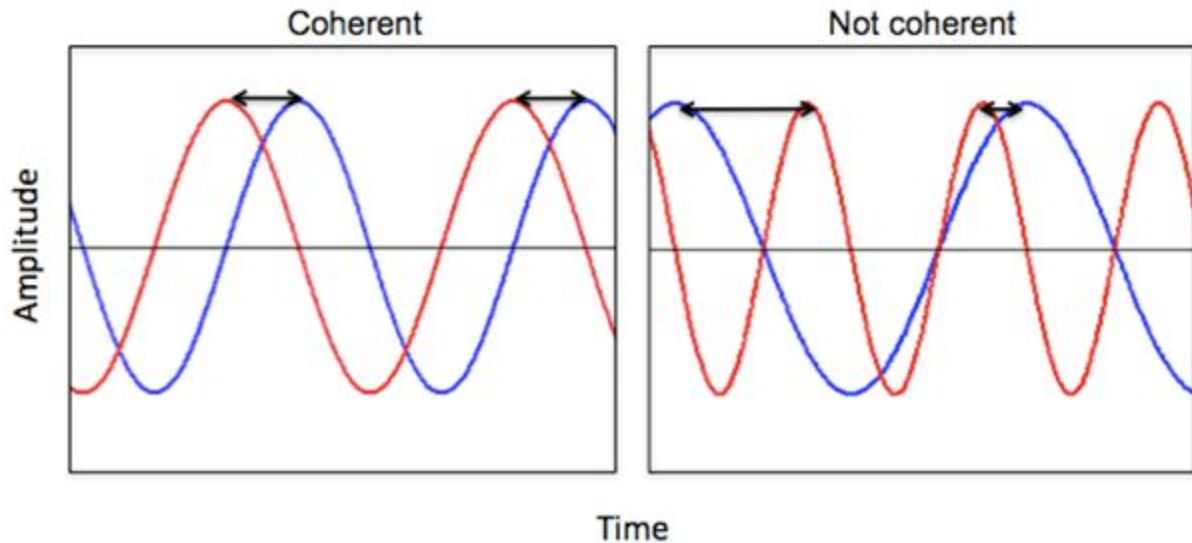


Figure 2: Example of coherent and non-coherent simple waveforms. Coherence reflects the consistency of signal timings from two different areas of the brain. It is believed to reflect communication between distant areas of the brain. Like spectral power, coherence is generally specified within a certain frequency range.

`coherenceTest.m`: Run the script to create two sine waves of equal frequency each “embedded” in different “noise,” plot the two signals, their power spectra and the coherence spectrum representing the timing relationships between the two signals. Note the peak in the coherence spectrum at the main frequency of oscillation present in both signals. *Zoom in on 0.1 seconds of the signals to verify the (nearly) constant phase relationship (peak-to-peak relationship) between the two signals.*

- Here’s how we’re calling the coherence function `mscohere` in our script:

```
mscohere(um,un,[],[],[],fs)
```

where `um` and `un` are the two signals, and `fs` is their sampling frequency in Hz.

Now change the frequency of one of the signals by multiplying its frequency by a factor $F = 1 + \epsilon$ with $\epsilon < 1$.

Q7. What happens to the coherence spectrum? What is the largest epsilon still showing a noticeable peak in the coherence spectrum? This illustrates how it is possible for two signals to be incoherent, or to go in and out of coherence, even if the two signals have very similar frequency content.

Q8. Suppose you compute the coherence between a signal and its “reverse” (i.e. the same signal, backwards in time)? What would the coherence spectrum look like in this case? Modify the code to test your prediction.

Conclusions from part 1

- A signal's **power spectrum** shows how much of each frequency is in it. A **spectrogram** displays how the signal's power spectrum changes over time. Both of these are calculated from a single signal, so if they are calculated from a neural signal (like one EEG or MEG channel), high power at a particular frequency corresponds to **local oscillatory neural synchronization**.
- A **coherence spectrum** represents how synchronized two signals are, *at each frequency*. In neural terms, if the two signals are from different brain areas, a high coherence implies **long-range oscillatory neural synchronization**.

II. Local and long-range neural synchrony in a behaving rat brain

In this part of the lab we'll apply our spectral analysis tools to some real neural data recorded from behaving rats in the Wiest Lab, to see how the spectral analysis provides additional ways to characterize the data. In particular, the spectrogram will show us the presence of local oscillatory synchrony at specific frequencies, and the coherence spectrum will tell us how synchronized the signals in different areas are with each other. As we'll see in our Journal Club paper on "Object-based Attention," rhythmic synchronization may enhance the communication between distant brain areas that need to coordinate--in this case the coordination is for the purpose of attending to a particular stimulus type, but neural coherence may be a more general mechanism by which the brain routes information flow between different brain modules.

During the recording session that we will look at, the rat was performing a simple **auditory detection task** that we called "lick on beep." The rat was thirsty and sitting in front of a computer-controlled water spout. Every once in a while the computer would randomly play a beep, and if the rat licked within 3 seconds after the beep it could get a few drops of water. When the beep sounds and the rat licks, we call that a **hit** trial, and when the beep sounds but the rat does not lick for whatever reason, we call that a **miss** trial. As neuroscientists studying attention and perception, it is natural to ask if some brain states are more "sensitive" or "responsive" to sensory cues than others.

ratSpec.m: Run this script to load and scroll through local field potential signals recorded from two electrodes implanted in frontal and parietal cortex of a rat while the animal performs a simple auditory detection task. When the figures come up for the first selected trial, arrange the two figure windows for convenient viewing. Notice the panel of Figure 1 that shows the two signal LFPs with the frontal cortex signal on top in red and the parietal signal in blue below. This parallels the frontal spectrogram shown in the top panel of Figure 2 and the parietal spectrogram in the bottom panel. In each of these panels the zero on the horizontal time axis marks the time when the beep sounded.

Meanwhile the bottom panels of Figure 1 show the spectrum of coherence between the two areas for the 1-second periods before and after the beep. The title shows the trial number and whether this trial was a hit or miss--whether the rat licked after this beep.

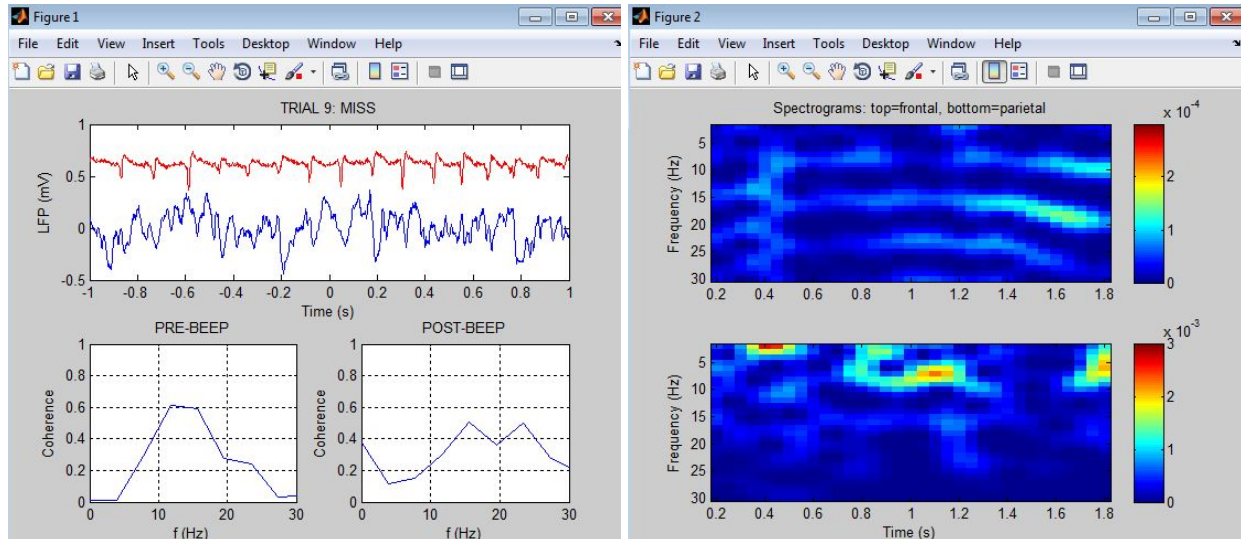


Figure 3: A rhythm we'll call mu, appearing mostly distinctly in the frontal LFP (red trace and top spectrogram).

Notice the spectral features of the first selected trial (trial 9), which illustrates a rhythmic brain state that appears spontaneously in rats when they are bored or sleepy. In this trial (and in this session more generally) this brain state appears most distinctly in the frontal LFP (the red trace and top spectrogram), but it also appears in the parietal (blue) LFP. On our first pass through the 40 selected trials, try to note which trials are examples of this rhythmic brain state--let's call it **mu**³ for short--and how many of those mu trials are hits vs. misses.

Questions 9 and 10 below ask you about this mu oscillation, but before considering those let's look at the second selected trial. If you press any key except 'b' (for back) or 'q' (for quit), you'll see the second selected trial (trial 13). This one exemplifies a different oscillatory state that appears in some trials, in the parietal LFP (the blue bottom trace and spectrogram) when the frontal LFP is not in the mu state we focused on above. Because this oscillations appears in the frequency range associated with theta oscillations (roughly 4-9 Hz), let's call this parietal rhythmic state **theta**. Read questions 9-11 below and then use the trial table handout to step through all 40 selected trials and note whether each trial showed any mu or theta rhythm, and whether each trial was a hit or a miss. Then use the data you collected in your table to answer questions 9-11.

Q9. How would you briefly describe the frontal and parietal spectrogram pattern in trial 9? Which trials would you say appear to show the same general state as trial 9? How consistent is the "mu" pattern you observed in the first trial (9) with the spectral pattern in the other trials you identified as mu? Does the coherence spectrum show any consistent pattern in this brain state?

Q10. How many of the mu trials you listed in the previous question were hits and how many were misses? If you observe this brain state on a new trial before knowing whether the trial was a hit or a miss, which would you predict?

³ Some people call this brain state in rats "7-12 Hz high-voltage rhythmic spikes" or "spike-wave discharges." I studied this brain state to learn how to record neural activity in rats (Wiest & Nicolelis, 2003).

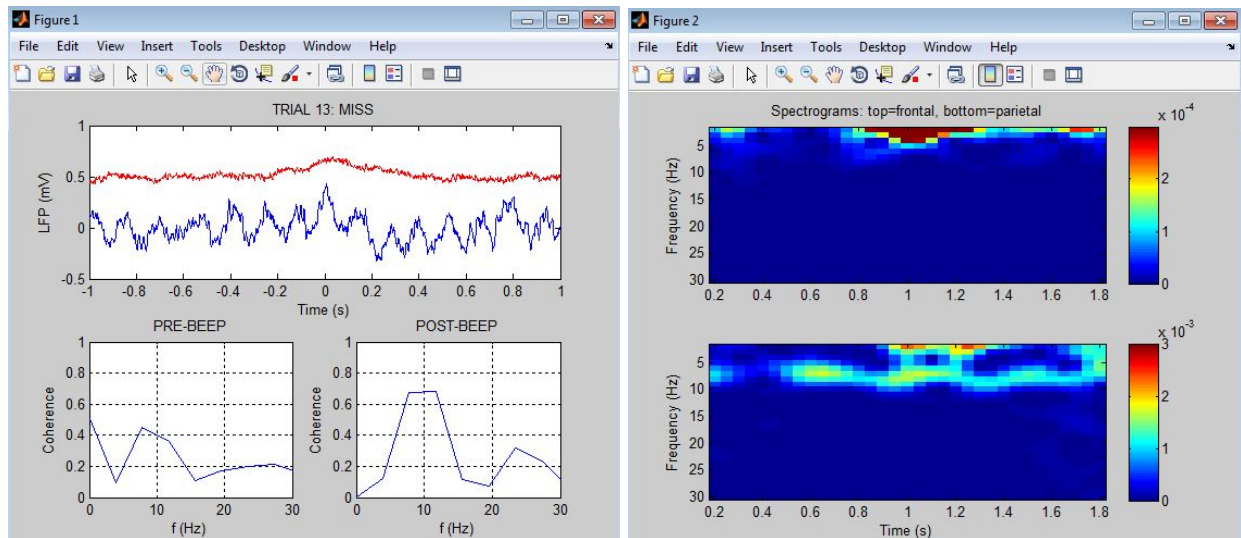


Figure 4: A rhythm we'll call theta, appearing in the parietal LFP (blue trace and bottom spectrogram).

Q11. What is the main frequency of the parietal theta oscillation? Follow a similar procedure as in the previous questions above, to estimate the probability of a hit given the presence of this brain wave in a particular trial.

Now try to define a (fairly simple) set of criteria for predicting whether a given trial will be a hit or a miss--whether the rat will lick--based on the spectral analysis results plotted in the spectrograms and pre- and post-beep coherence spectra. You may want to step through the trials one or two more times to make notes on what spectral features seem to be associated with hits or misses. When you formulate your hit-predicting criteria, try to be specific in terms of the scales in the figures. For example you might say "if the frontal power at 20 Hz goes to 1.5 (green) or above at any time during a trial, we'll predict it's a hit, but if the pre-beep coherence is higher than 0.4 at any frequency below 10 Hz then we'll predict a miss." Try to formulate a fairly simple set of criteria in no more than 2 or 3 sentences.

Q12. Type your criteria into this document, then alter your ratSpec script so that the title of Figure 1 no longer displays whether each trial is a hit or a miss trial. Rerun the script and use your stated criteria to predict whether each trial is a hit or a miss based on your spectral criteria, and keep track of your prediction for each trial. Evaluate the performance of this set of criteria and report your percent trials correctly predicted here. How does your algorithm's performance compare to the expected performance from blind guessing?

Q13. What would be a better way to predict the animal's behavioral response (hit or miss) based the spectral "signature" of these two LFP signals on each trial? *Hint: what computational approach did Tom Mitchell and colleagues use to find fMRI "signatures" of the meanings of nouns?*