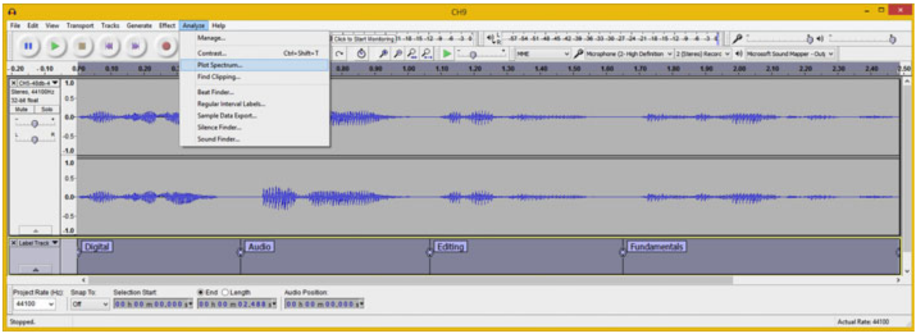

Figure 9-8. *Select the Analyze ➤ Plot Spectrum menu sequence*

As you can see in Figure 9-9, the first thing that I did was visualize the Notch Filter (that you used in Chapter 8) to see if it looks as if I took a notch out of the data sample. As you can clearly see in the right-hand screenshot, there is indeed a clear notch at 4500 Hz. You'll take a look at some of these other settings as well.

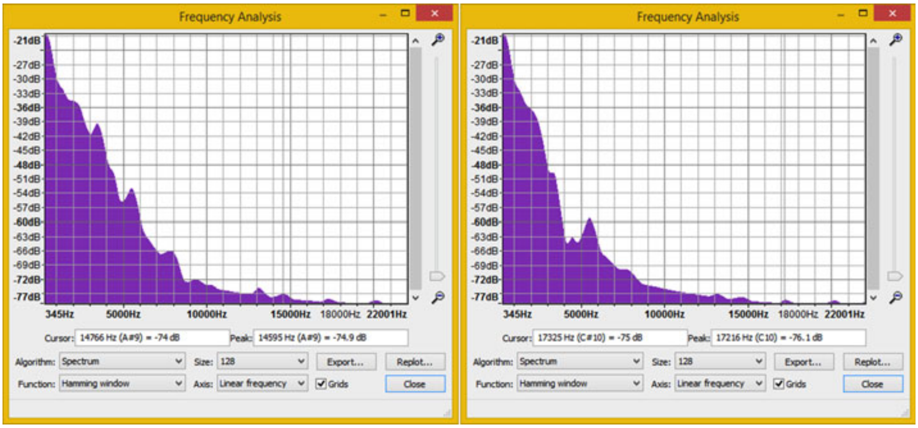

Figure 9-9. *Spectrum Algorithm Size 128, pre/post Notch Filter*

The first thing that you'll look at is how to set the frequency display resolution by using the Frequency Analysis's **Size** drop-down menu. I set it to 256 in the "before" and "after" data samples, so that you can see how it adds more peak data, as the two dialogs show you in Figure 9-10. I'll also show you 512 and 1024 settings later on, so that you can see how these settings visualize your spectrum data.

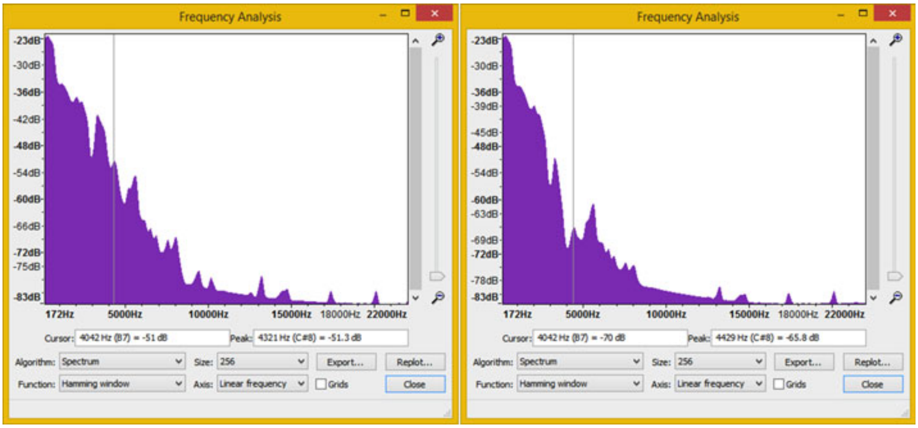

Figure 9-10. *Spectrum Algorithm Size 256, pre/post Notch Filter*

What you have looked at so far is a spectral analysis for your entire data sample; but as you may have surmised, you can also use this tool to analyze subsamples. Let's take a look at that next, as this is an even more useful analysis approach.

Subsample Analysis: Using the Select Tool

Let's select the vocal for Audio, as shown in Figure 9-11. This is made clearer since adding your handy Label Track and the custom label markers that you installed earlier in the chapter. Again, select **Analyze ➤ Plot Spectrum** (see Figure 9-8), which now displays sample data using its algorithms only for the selected data subsample and not for the entire sample. This allows you to only view a subset of the digital audio data that you want to analyze, which also means that there is less data to analyze in the visual graphing function. And, this makes it easier for you to find the audio data that you want to visualize and eventually process. It is analogous to zooming in on an area of a digital image so that you can work only on particular pixels in your digital image. The ability to work with this tool makes it far more powerful.

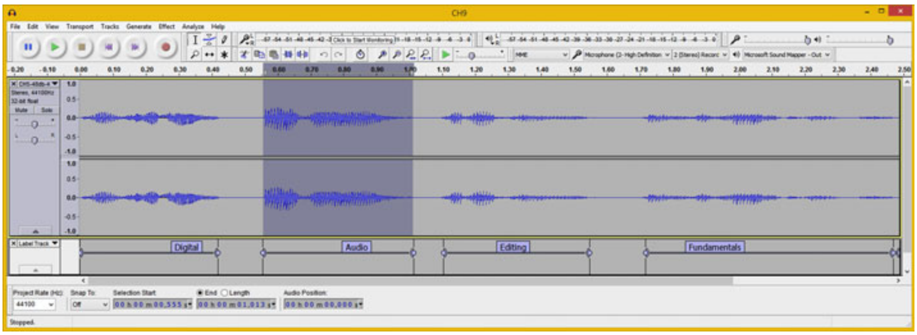

Figure 9-11. *Select vocal for the word audio, above Label Track*

In Figure 9-12, you see that I selected a **Size** of 256 and 512 for the notch filtered (the latest version of the voice-over) data sample to show the frequency sampling resolution difference. The higher sample size shows the notch that you carved with increasing precision.

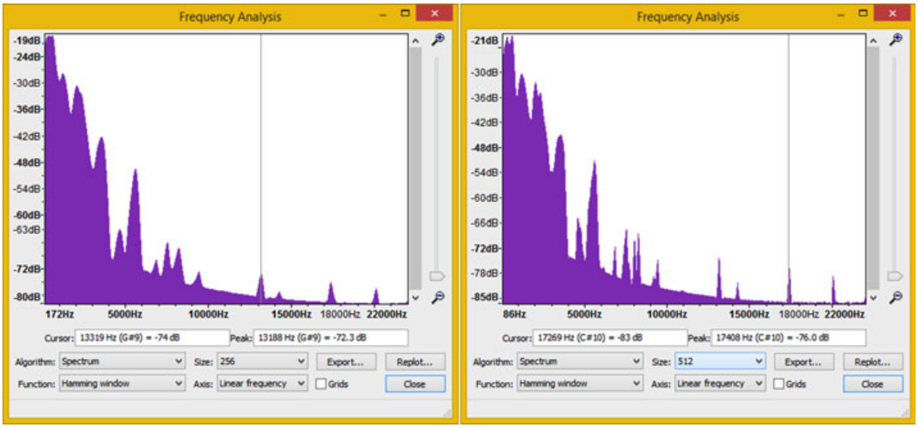

Figure 9-12. *Notch Filter spectral analysis at Size 256 and 512*

It is important to note that the vertical line in your graph should "snap" to the nearest peak as you move your mouse over the data. The **Cursor** and **Peak** readouts provide precise numerical information to use with the Effect menu filters and in their audio effect dialog settings.

Next, let's drill down yet again and use this tool on a data sub-subsample; that is, on just a small portion of a data sample, such as the first part of the word "audio" that you did surgery on in Chapter 7.

Partial Subsample Analysis: Smaller Selections

Select the first portion of the vocal "audio" word sample (see Figure 9-13), and again, invoke **Analyze ➤ Plot Spectrum** to analyze this snippet (or sub-subsample).

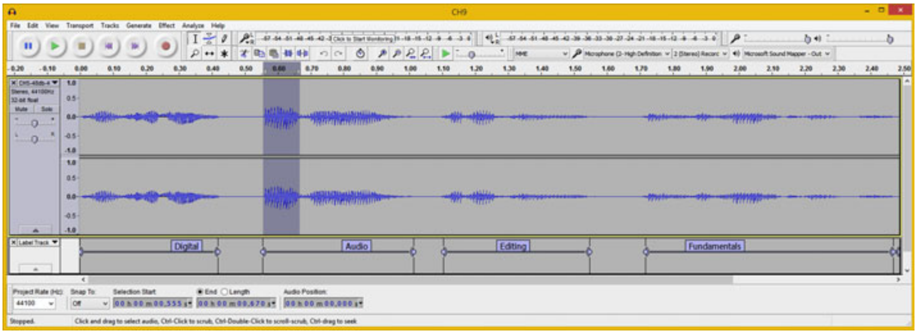

Figure 9-13. *Select an even smaller snippet on front of "audio"*

As shown in the left-hand screenshot in Figure 9-14, you can see the frequency data even more clearly. The fewer data samples that you select, the more clearly you can see the frequency distribution. This is because the data has more room to spread out inside the Frequency Analysis window, as is evident on both sides of Figure 9-14. Notice that in the right-hand screenshot, I changed the **Algorithm** from Spectrum to Standard Autocorrelation.

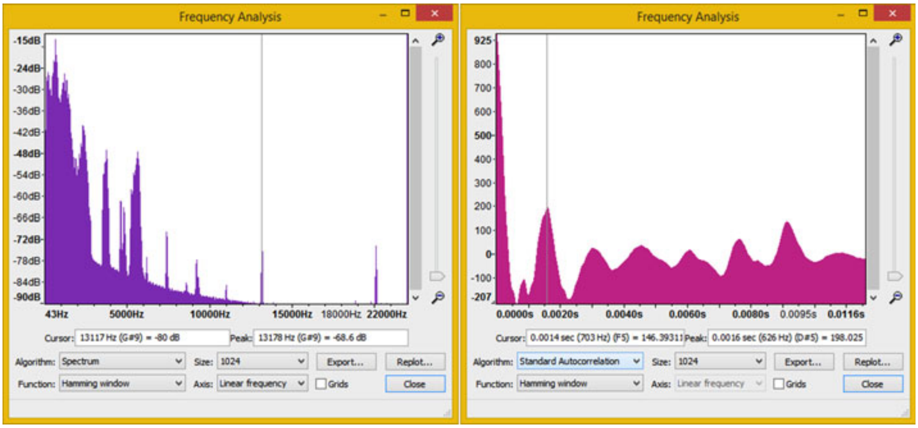

Figure 9-14. *Spectral analysis of first portion of word "audio"*

In case you are wondering what these three different algorithm types do, the **Spectrum** algorithm, which is a default setting, plots the FFT of your audio spectrum data. FFT stands for **Fast Fourier Transform**; it is the algorithm that computes the **Discrete Fourier Transform** (DFT) of any data sequence.

Fast Fourier analysis converts the signal (audio sample data) from its original mathematical domain, usually over time, to a spatial representation; in this case of digital audio waveforms, it is both. This allows you to visualize your data in a different way, which can be quite helpful in assisting you in your workflow decision-making process.