Analysis methods

The basis for our approach is the STFT, a useful standard DSP tool. First we dice up an audio sample into short time slices of a few milliseconds each. Then we apply a window function (e.g. Hamming window) to each slice to reduce aliasing effects. Finally we apply an FFT to get the frequency spectrum of each time slice. The choice of overlap between time slices determines the temporal resolution. This process yields a view of how the sample's frequency content changes in time, which is plotted in a spectrogram. Judicious selection of various frequency bands in the spectrogram lets us distinguish one instrument's note events from other instruments. In this way we extract the rhythm and the Swing.

See figure 1 for a typical spectrogram image. The musical sample is the intro for *It Don't Mean a Thing (if it ain't got that Swing)* recorded by Duke Ellington and Louis Armstrong (1962). The first 5 seconds of the 19 second sample contains a series of thin yellow/red spikes which are from the hi-hat cymbal (close-up in figure 1a). The remaining 14 seconds are dominated by Armstrong's trumpet solo (no vocals in this example). In the lowest part of figure 1a there is a dense concentration of red which is the piano and bass. Close-up of figure 1a (figure 1b) shows more details of the low frequencies. For analyzing the timing details, we choose a high frequency band to get the hi-hat cymbal note events, and several low frequency bands which contain the piano and bass. We want to identify musical note events for each instrument, and to extract the relative timing details so that we can specify the rhythm directly from the recording, rather than approaching from the perspective of sheet music.

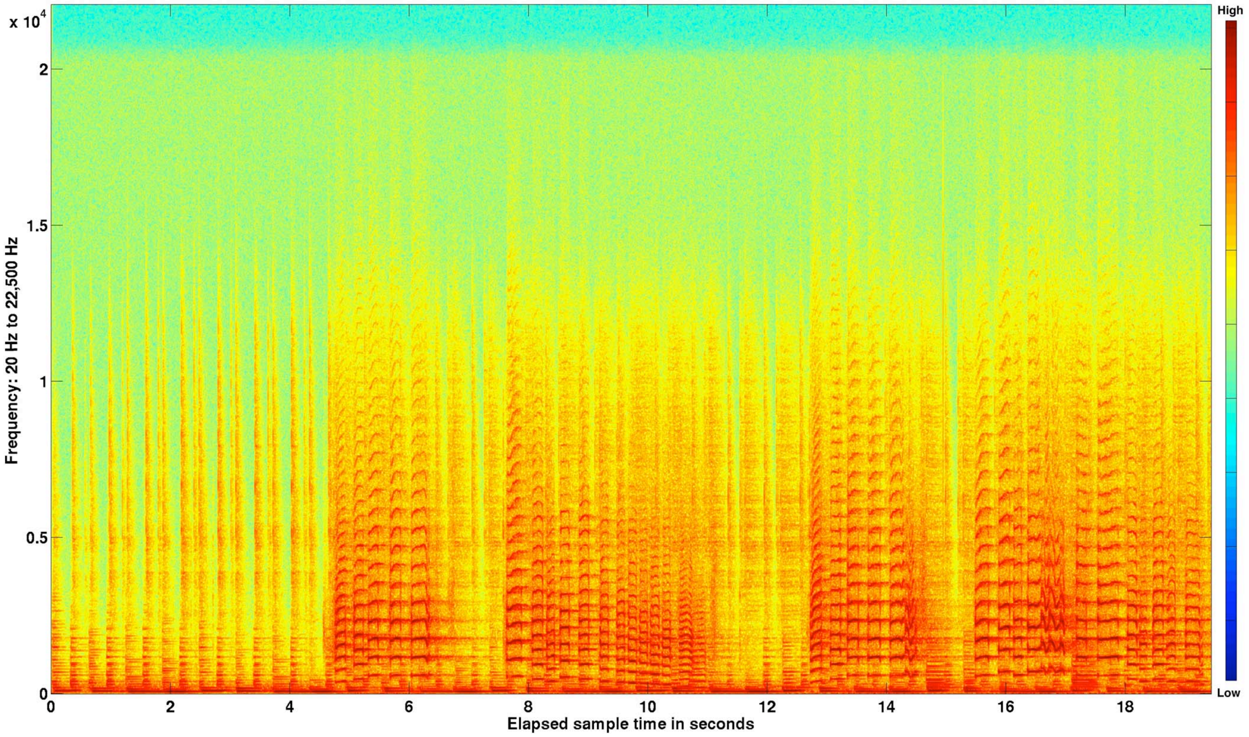

Figure 1 Spectrogram of intro for ***It Don't Mean a Thing (if it ain't got that Swing).***

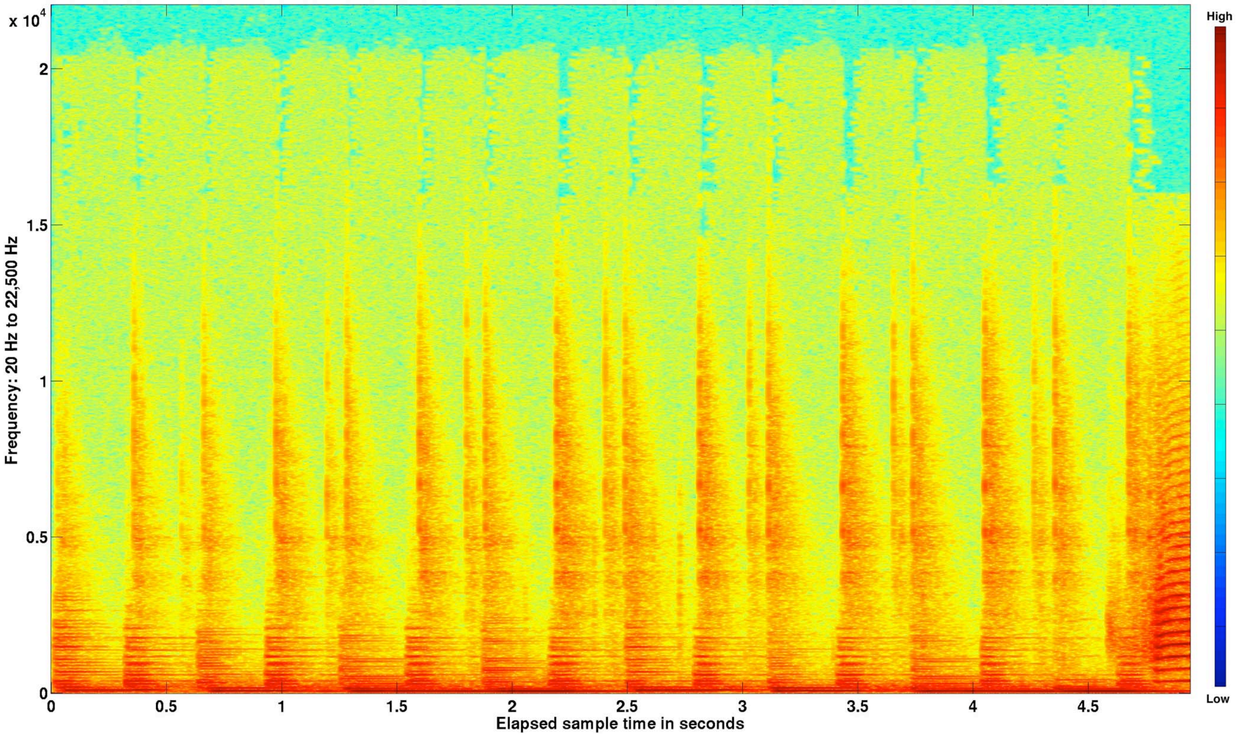

Figure 1a Closeup of hi-hat cymbal at the beginning of the intro.

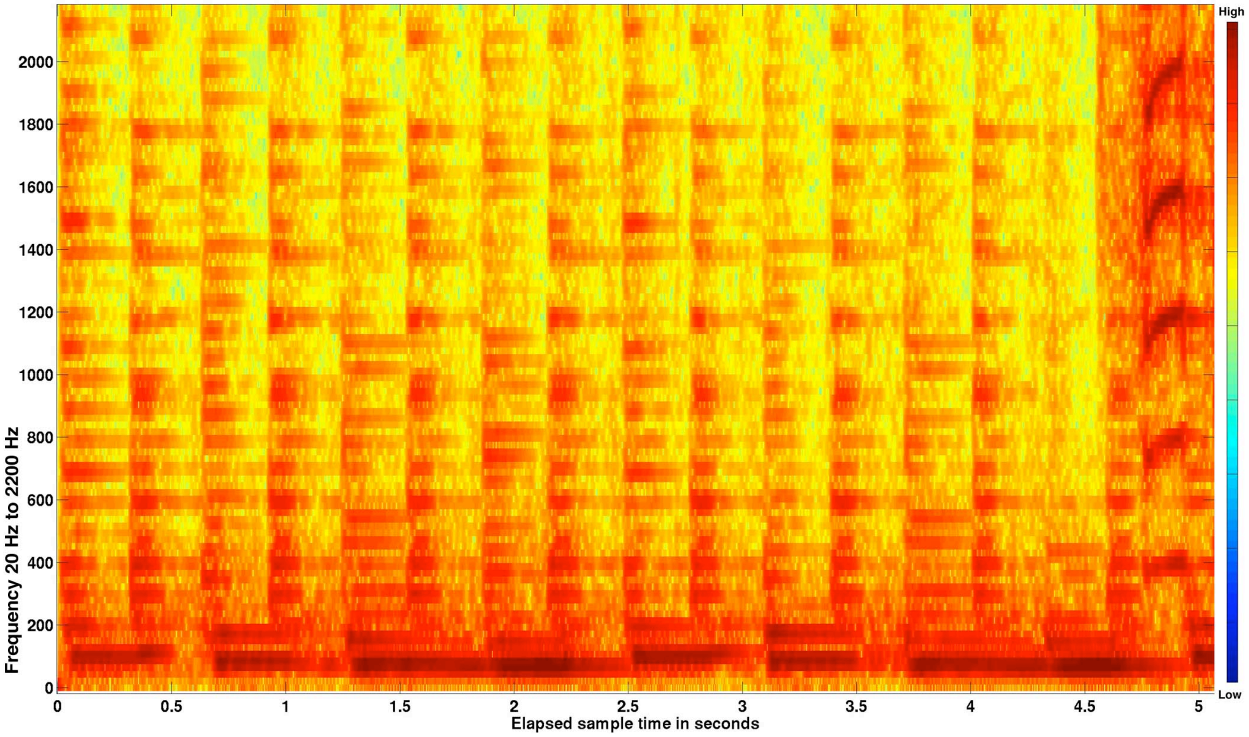

Figure 1b Closeup of low frequencies showing details of piano and bass.

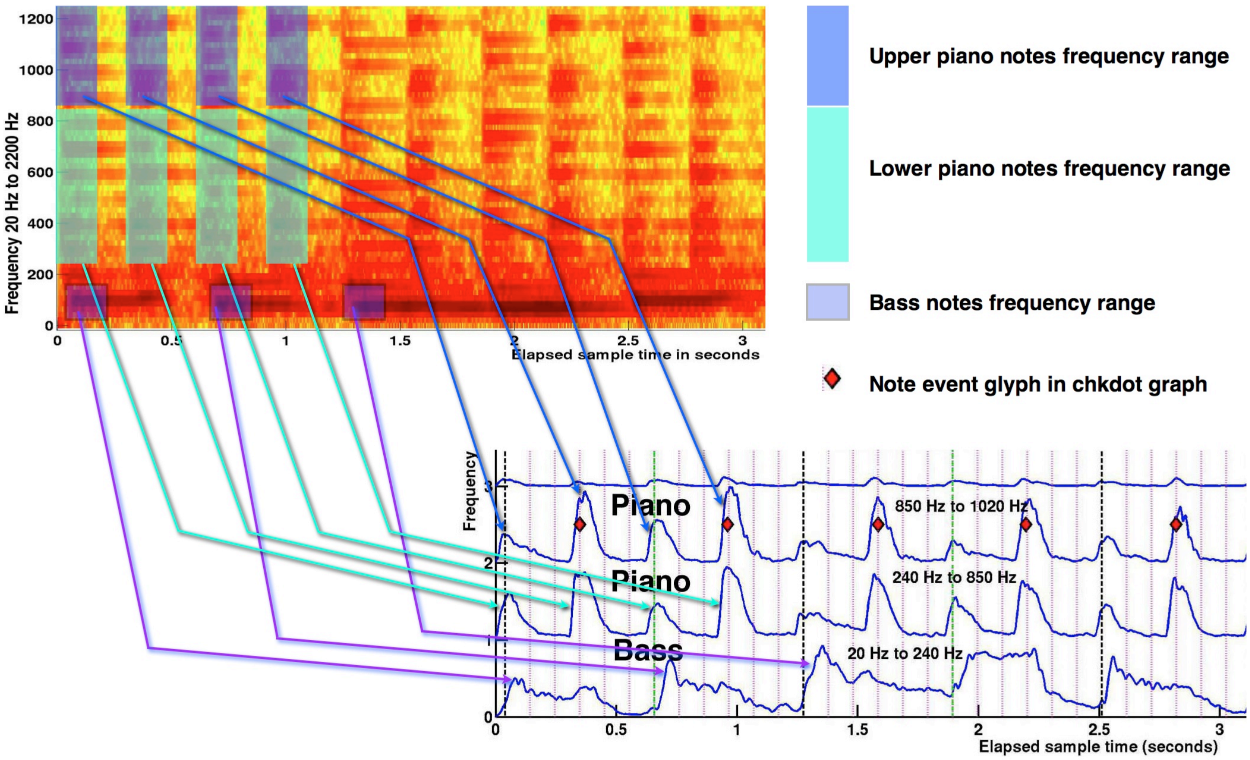

Figure 1c The mapping process from spectrogram to time series graphs.

After selecting sensible frequency bands in the spectrogram, we want to create time series graphs of the changing power (loudness) of each frequency band. Figure 1c shows the process of making time series power graphs from the spectrogram. To do this, we add the values for each separate time slice in each frequency band. The total sum of a time slice in a frequency band becomes the Y value of the time series power graph for that frequency band at that point in time. The X value for the graph is the elapsed time (in the original recording) that corresponds to the time slice in the spectrogram. Values are indicated by color in the spectrogram, each color being simply a number in the spectral data set (specifically, the amplitude coefficient of the Fourier component for a frequency). The distinction between power and amplitude is not important for this process, since the overall shape of the graph will be about the same for both. We will be looking for power peaks in the graphs since it is reasonable to view the "note event" as the loudest time point of the audio power graph.

The number of values in a time/frequency tile (one time slice for a frequency range) is typically between a few dozen and a few hundred, depending on the size of frequency range. The set of summed totals in each frequency band for all time slices is used to create the time series graph for that frequency range. There are the same number of points in the time series as there are time slices in the spectrogram, which simplifies time comparisons between the different types of plots. The several time series that we generate from the chosen set of frequency bands are then stack plotted from low to high frequency in the `chkdot` diagram – `chkdot` is the MATLAB script we developed for this work (everything needs a name). The `chkdot` diagram (figure 2) shows time aligned musical events for all frequency bands, as played by different instruments.

Our code searches the `chkdot` waveforms for peaks representing the note events (peak power or loudness in the frequency band). The time locations of these note events are extracted automatically by simple logic that chooses the point where the graph first turns back downwards immediately following a sharp vertical rise above some sensible threshold level. This is adequate but not ideal since in real music there are numerous artifacts that may befuddle the idea that a note event is clear, sharp and precise. The notes played by the hi-hat in figure 2 are ideal, but the notes played by the piano are not. You can see in the second and third graphs from the bottom that there is a collection of two, three or four small ripples at the top of some of the peaks. These are events caused by two, three or four fingers hitting the piano keys, but which are not precisely synchronized. Meticulous listening to the original recording can reveal the multiplicity of key note events in this frequency range. For convenience we chose the first event in such a cluster. The question of what the musician's intention was, or whether our choice of note event time location is identical to the *perceived* time location by a listener calls for further research. In this article, we are focussed on characterizing rhythmic timing and we believe our choice, while slightly arbitrary and ambiguous in some cases, is nonetheless reasonable for the current context.

We mark the musical meter and subdivision in a straightforward way on the `chkdot` plots. Figure 2 shows the breakdown for the hi-hat cymbal and piano/bass parts in *It Don't Mean a Thing*. The note events are marked by red diamonds which are placed along the (invisible) line in the center of the horizontal band that contains the time series graph. In figure 2 we have one set of note events for low frequencies (850 to 1020 Hz) and a second set of note events for high frequencies (7500 Hz to 22,000 Hz). Notice that some note events are sharp and distinct (upper waveform: the hi-hat cymbal) while other time series waveforms may have many jagged sections where the precise time of a note event may be ambiguous (bottom three time series: piano upper, piano lower, and bass).

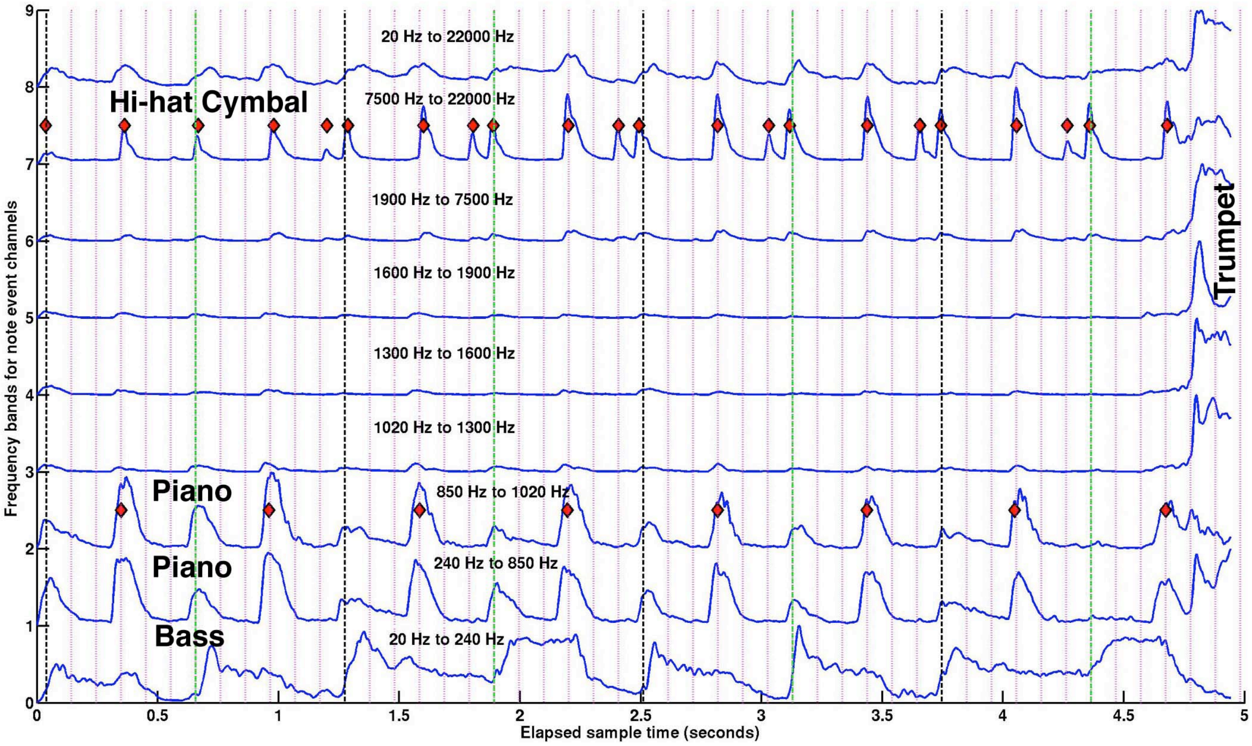

Figure 2 Multi-frequency time series graph showing separation of instruments by frequency.

The less distinct waveforms, especially the bass, are spread out more in time than the sharp events, indicating that the attack envelope of the sound is slower for these events. These note events actual time locations may also be somewhat imprecise. Often, the piano sound and bass may obscure each other. Separating these overlapping note events would need more sophisticated signal processing techniques than we currently use. Nonetheless, we can fairly easily identify enough note events to specify the rhythmic timing details. These details reveal where the Swing lives in all of this technical complexity.

To mark the musical subdivision, we first select note events that represent a *pulse* to use for the basic beat in the musical sample, such as the downbeat in a musical measure. We can subdivide this main beat in any convenient way, depending on the rhythm we want to measure.

For example, triplet subdivision is a common timing feature in Swing, so we choose to subdivide the main beat by six. This gives us a backbeat (divide by 2) and triplets in the same scheme.

While triplets can be marked in sheet music, the standard subdivision of MB notation is by factors of 2. This is one reason why notating Swing music is somewhat difficult: triplets do not fit naturally into a subdivide by 2 metaphor. We show later that Swing can also contain subdivisions which are neither factors of 2, nor accurately described by triplet subdivision. Our approach avoids the limitations of subdivision that are inherent to MB notation.

The actual note events in the recording are used to determine the musical meter and subdivision of the beat in the `chkdot` diagram. Essentially the reverse of playing a tune by reading sheet music, we extract note information from the recording which could be used to *generate* sheet music. The pulse in figure 2 is marked by green and black vertical lines, which correspond to the downbeat of the measure in MB notation (a two measure phrase, one green and one black). We subdivide each musical measure by six, looking for triplet notes (the classic Jazz Swing pattern) and mark this subdivision by using six pink lines in the `chkdot` diagram. The pink line exactly in the middle between a black and green line represents the time location of the *backbeat* of the rhythm. Thus we see that the piano/bass peaks are on the downbeat and backbeat, with diamond markers on certain backbeats in the third time series up from the bottom of the chart. We use these events to mark the pulse. The hi-hat cymbal note events (in the time series at the top) occur on the downbeat, backbeat and triplet pickup to the downbeat and backbeat. The triplet timing is indicated by note events on a pink line just ahead of a black or green line.

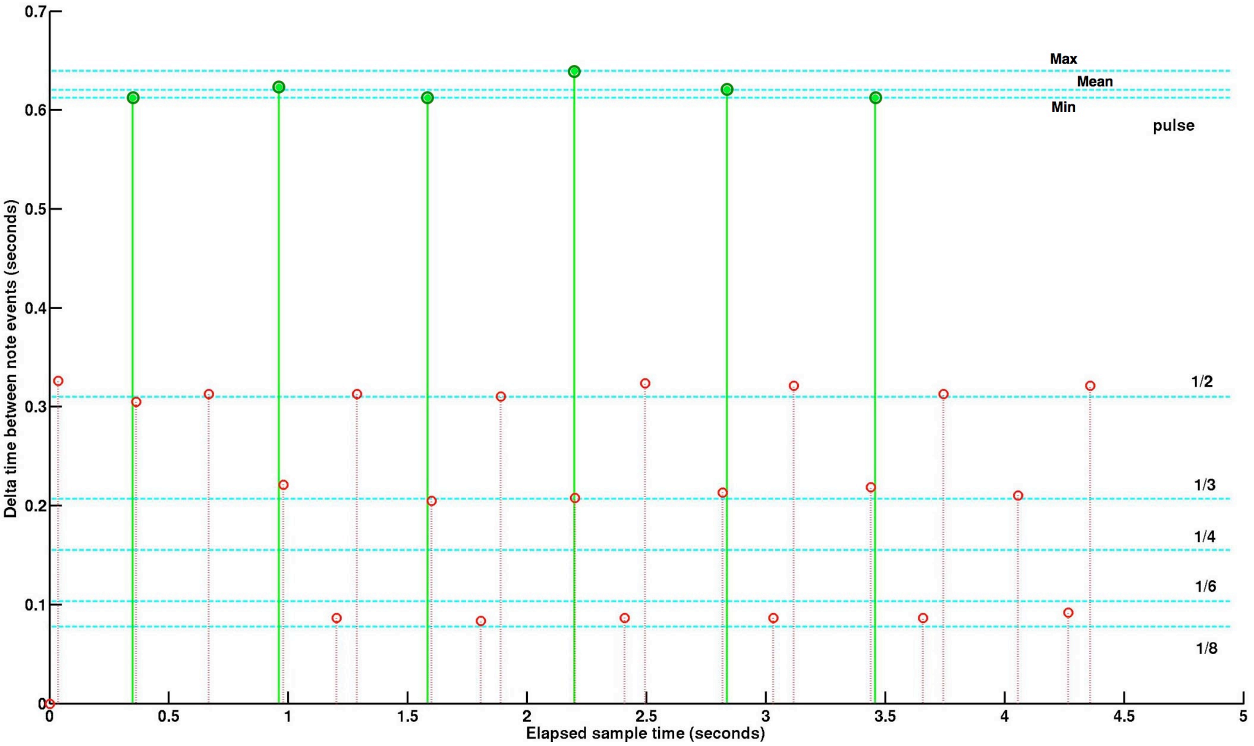

Figure 2a `Diffdot` *plot showing time differences between related note events. Piano notes are at the top marking the pulse, and hi-hat notes are in the lower half, showing backbeat and swung notes.*

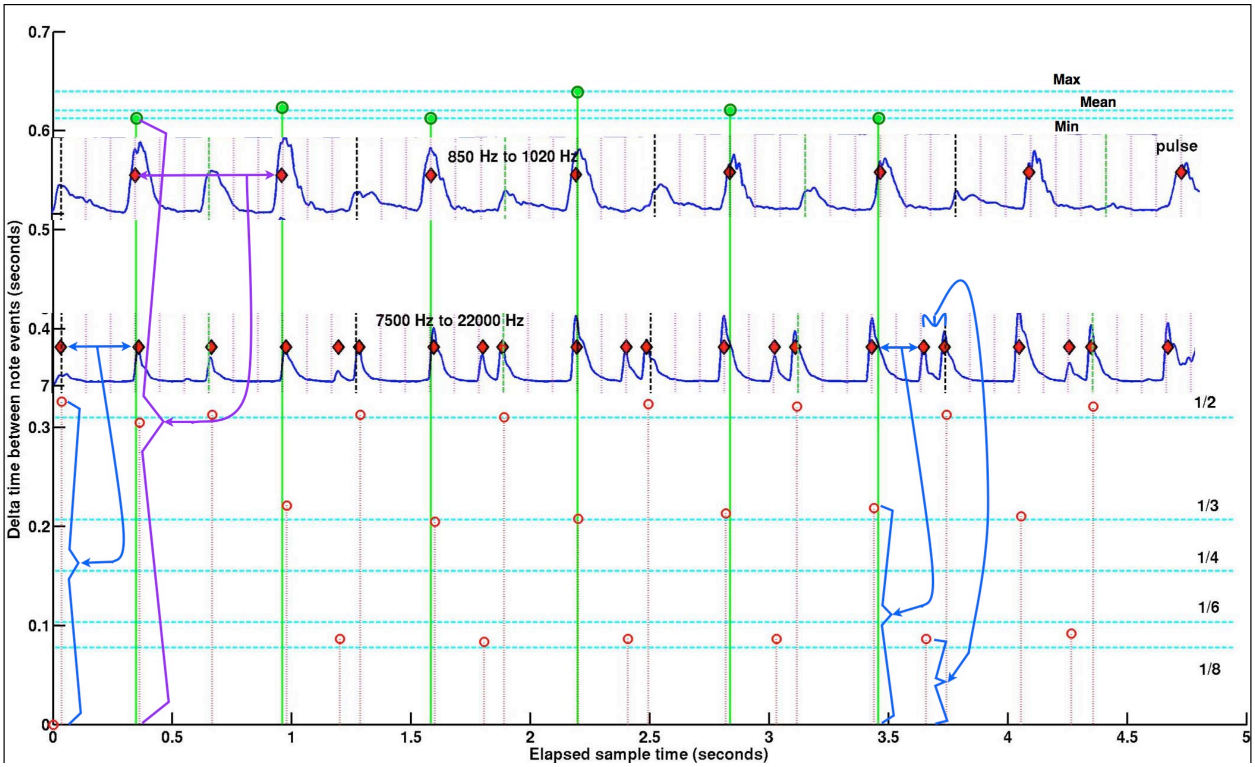

Figure 2b The process of mapping note events in the time series to note timing format in `diffdot`.

We use the extracted time locations of the note events in a new type of plot (`diffdot`, figure 2a) that shows the time *differences* (delta time) between notes. This time delta corresponds to the length of a musical note in MB notation: 1/4 note, 1/2 note, 1/8 note etc. The pulse is used for the master time clock (whole note), and a time delta with length 1/2 of the pulse would be a half note in MB notation, 1/4 of the pulse length would be a quarter note and so on. Because we can subdivide the beat by any number that makes sense for a musical sample, we easily accommodate triplets (divide by 3) or any other note time duration. Since the pulse note event timings have some variation, we notate the minimum, maximum and mean (average) of the time differences, and use the mean value as our canonical pulse time, used to subdivide the beat.

Figure 2b shows the mapping process from `chkdot` to `diffdot` plots. We have superimposed the two time series in fig 2a which were marked for note events over the `diffdot` plot for the same time range. The elapsed time on the X axis is the same for both forms. A red diamond on the `chkdot` plot maps to a circle on the `diffdot` plot, red circles for the hi-hat, and green circles for the piano. The X position of matching diamond/circle pairs is the same. The Y position of the circles indicates the time from that note event until the next note event in the set. Thus longer notes, such as the pulse, are at the top of the `diffdot` plot, and shorter notes are in the lower half of the plot.

In figure 2a and 2b, the red dots are the hi-hat note events. Notice the first three red circles are (fairly) evenly timed on the backbeat (1/2) of the pulse. These three time deltas corresponds to the first *four* diamonds in the corresponding time series graph. After four note events, the hi-hat starts to play triplet notes, clearly visible on the pink subdivision lines in the `chkdot` diagram, and transferred to the `diffdot` diagram onto the 1/3 and 1/6 (lying between 1/6 and 1/8, really). These note events on the 1/3 and 1/6 lines of figure 2a are the time deltas between the swung notes in figure 2, and the beats immediately before and after: i.e. 1/2 - 1/3 = 1/6. The slight imprecision of the note timings in this example indicate a somewhat loose rhythmic style for this recording. Later we will analyze a recording which has a very tight rhythmic style. This is another aspect of the music performance that can be read directly from the `diffdot diagram`.

Note events are essentially transferred one for one from the `chkdot` to the `diffdot` plots. Since `diffdot` shows the differences between note event times in `chkdot`, then there is one less event in the `diffdot` diagram than is in the `chkdot` diagram. `Chkdot` plots are more intuitive to read since they are a direct parallel of standard musical notation. `Diffdot` plots may need some more careful inspection, but even if you don't understand them perfectly (analytically), you can still easily read them by looking at the spatial patterns to get an intuitive sense of how the Swing works.

In addition to the time differences between note events, the `diffdot` plots can also show the *variations* in time locations of repetitive musical events extracted from the `chkdot` plots, such as pulse, backbeat and swung notes. This is not a feature which can be written in MB notation, so far as we know. The `diffdot` plots also clearly show how on some beats, the two instruments are not precisely synchronized: in some cases, the hi-hat plays slightly before the piano note event, and in other cases, the reverse is true. This can be read directly by looking to see whether the green line is to left or right of the red line for that particular time location. Only the beat in the center of the graph is exactly synchronized.

Please be clear that the `chkdot` diagrams are a direct representation of standard MB subdivision and counting, albeit with more fine grained timing information included, whereas the `diffdot` plots are a novel view of the same information, essentially looking at first difference form of the original timing information.

To process each musical sample into a spectrogram, we use a short audio clip that is typically ten to twenty seconds long. These are edited to be played with seamless looping, such as in Quicktime player, in order for us to listen to the rhythm very carefully for extended periods of time. While this is not strictly needed for the analysis, we found that it greatly enhanced both our enjoyment and understanding of the rhythms. Our experience is that anomalies as short as five or ten milliseconds are sufficient to be perceptible as a break in the rhythmic flow. This is distinguished from editing artifacts such as may cause an unnatural transition in the audio waveform, like a click or pop. For these reasons we always edit at zero crossing points in the audio waveform. This is not always sufficient to avoid all artifacts which can be perceived either explicitly or intuitively by a well trained human ear.

Choice of frequency resolution and STFT window overlap is constant for each processing run, but may differ for different samples. Sometimes we processed a single sample repeatedly, using several different choices of parameters. These results provide an interesting insight into the Heisenberg Uncertainty aspect of the time/frequency tradeoff that is inherent to Fourier analysis. We hope in the future to provide a thorough empirical report on this topic, which is not obvious merely by learning the mathematical theory. We have found that 2048 point FFT and three to ten millisecond time slice overlap are well suited to many samples. In some cases we used a time resolution as short as 0.5 milliseconds. Visual inspection of the spectrogram allows us to choose the frequency bands most likely to distinguish musical notes played by various instruments. Sets of the (possibly overlapping) frequency bands are summed to obtain time series plots of the audio power in the several bands.