Sonic Complexity: some basic tests

Guy Birkin

Intro

The aim of these tests is to begin an investigation into sonic complexity. The objective is generate a set of audio files based on variations of a few key musical parameters, and to see how a basic measure of complexity responds. In this case, complexity is measured by the information content (size) of digital audio files. The tests investigate how various audio compression algorithms might be useful as measures of sonic complexity. The hope is that understanding how this measure works with simple sounds will prepare the ground for further tests with more complex sounds and, eventually, with music.

Background

The underlying idea is that complexity can be measured in terms of information content.[1] In this case, information content is measured by the size of compressed audio files. I choose this measure of complexity, based on data compression, for a couple of reasons. Firstly, this line of research continues from my doctoral research in visual complexity, which used a similar method of measuring visual complexity with image data compression algorithms.[2] Secondly, the approach of applying information theory to questions of human perception is well established in experimental psychology, particularly in studies of aesthetics.[3] For psychologists,

> the concepts and measures provided by the theory of information provide a quantitative way of getting at some of these questions. The theory provides us with a yardstick for calibrating our stimulus materials and for measuring the performance of our subjects.[4]

One of the things I learned from my earlier research is that data compression is a fairly crude measure of complexity. By this measure, randomness has the highest complexity, but what we perceive and understand as complex is somewhere between order and randomness. The correlation between perceived complexity and data compression / information content only holds at lower end of the scale of complexity. Despite these drawbacks, data compression is a simple, quick and reproducible method. In terms of the quotation above, these tests are a

---

[1] http://www.scholarpedia.org/article/Complexity
[2] https://aestheticcomplexity.wordpress.com/research/phd/
[3] Amongst the first to apply information theory to aesthetics were Abraham Moles – *Information Theory and Esthetic Perception*, (first published in 1958, English translation 1966) and Max Bense – *Information Aesthetic* (1956). Data compression methods were not used until more recently, e.g. Donderi, DC & McFadden, S (2005) 'Compressed file length predicts search time and errors on visual displays', *Displays*, 26 (22): 71–78.
[4] George Miller (1955) 'The Magical Number Seven, Plus or Minus Two', *Psychological Review*, 63 (2): 81–97. Miller used information theory to measure the amount of information that we are able to receive, process, and remember.

kind of calibration exercise. The following analysis looks at how these measures of information content respond to variations in timbre, pitch and loudness.

Method

I made sets of audio files by writing a program in *Mathematica*[5] that generates mono 16-bit 44.1 kHz WAV files. The WAV files were converted using *Foobar2000* music player[6] for open license codecs and *iTunes* for the proprietary Apple file formats. The duration of each file is exactly 1 second (44,100 samples), which is long enough to contain a few cycles of the lowest frequency and short enough to be quick to generate and analyse. The sets cover combinations of the following parameters:

2 types of waveform:

- Sine
- Square

4 amplitude levels:

- 0, -6, -12, -24 dBFS

5 file types:

Lossless:

- FLAC, level 8 (max) compression
- Apple Lossless (Apple encoder), automatic settings

Lossy:

- MP3 (LAME encoder), VBR V0 (highest quality) compression
- AAC (Apple encoder), VBR 127 (highest quality) compression
- OGG Vorbis, q10.0 (highest quality) compression

[5] https://www.wolfram.com/mathematica/
[6] http://www.foobar2000.org/

200 frequencies:

Each set comprises 200 files, covering a range of frequencies from 20 to just under 22,000 Hz. The frequencies are spaced in equal temperament based on an interval ratio of 29:28, which is around 6/10ths the size of a semitone. The full set of frequencies is plotted in Figure 1, which shows that they are equally spaced on a logarithmic scale.

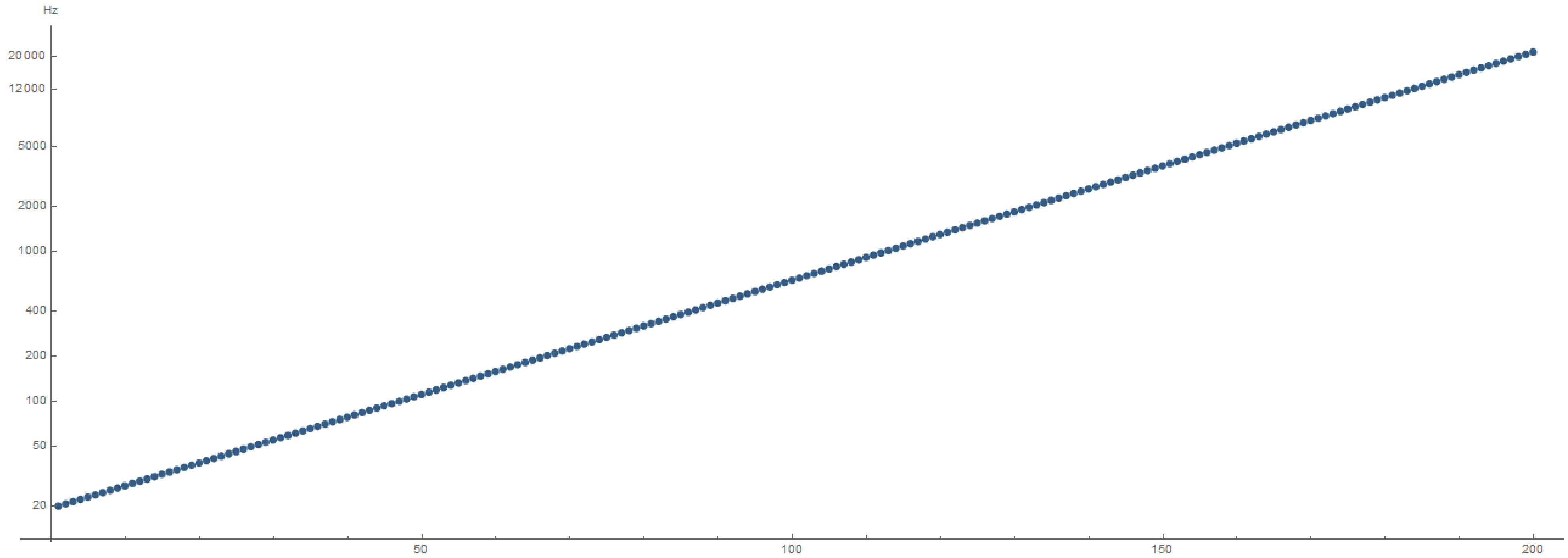

Figure 1: Frequencies of the 200 audio files (Hz, log scale).

Analysis

Because the files are all the same length it makes little difference whether we use file size or bit rate to measure the information content, since they are proportional to each other. For consistency, this analysis sticks with file size as the measure of information content.

Audio Formats: sine wave

Figure 2 plots frequency (Hz) against file size (bytes) for 200 sine waves at 0dB in five formats: FLAC, Apple Lossless, OGG Vorbis, AAC and MP3.

The lossless codecs (FLAC and AL) produce the largest files, and both codecs produce files whose size increases with frequency. There is a notable plateau in the AL files above 8,500Hz, where almost all the files have a file size of 92,344 bytes. This is larger than the size of the corresponding WAV file, which is 88,538 bytes. Perhaps the Apple Lossless codec 'gives up' on compressing audio data with high frequency content (I'll look into it). If a file format doesn't compress the data, then it's of little use as a measure of information content / sonic complexity (for the same reason, uncompressed formats such as WAV can't be used).

The lossy codecs (MP3, AAC and OGG) are quite similar, except that whereas MP3 file size tends to increase slightly with frequency, AAC and OGG decrease slightly.

Figure 3 shows the results for a set at the lower amplitude of -12dB. The lossless codecs have reduced in size more than the lossy ones. Only a couple of the Apple Lossless files have the same large file size of 92,344 bytes, at the two highest frequencies, 20.8 and 21.6 kHz.

Audio Formats: square wave

Figure 4 and Figure 5 plot the results for square wave sound files at 0 and -12 dB. The file sizes are higher on average than the sine wave files, but reach a similar maximum of around 90 kB. All the formats have a less linear pattern than before. The file sizes of the lossless codecs rise more steeply with frequency. They both show a fairly smooth linear increase up to a certain frequency (between 2 and 5 kHz), then the sizes becomes less predictable. In contrast, the lossy codecs tend to peak at lower frequencies.

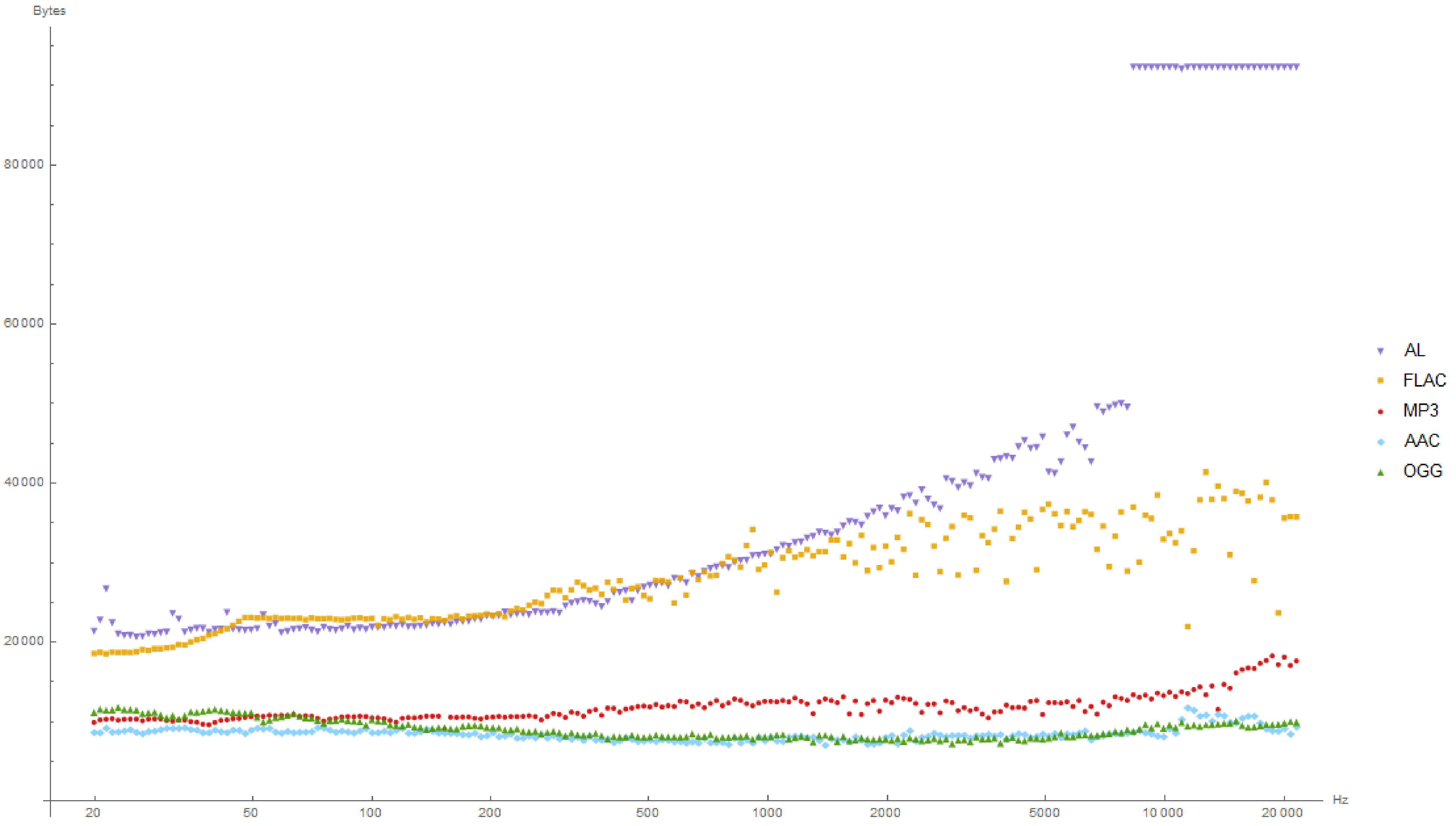

Figure 2: Sine wave, 0 dB (file size × frequency).

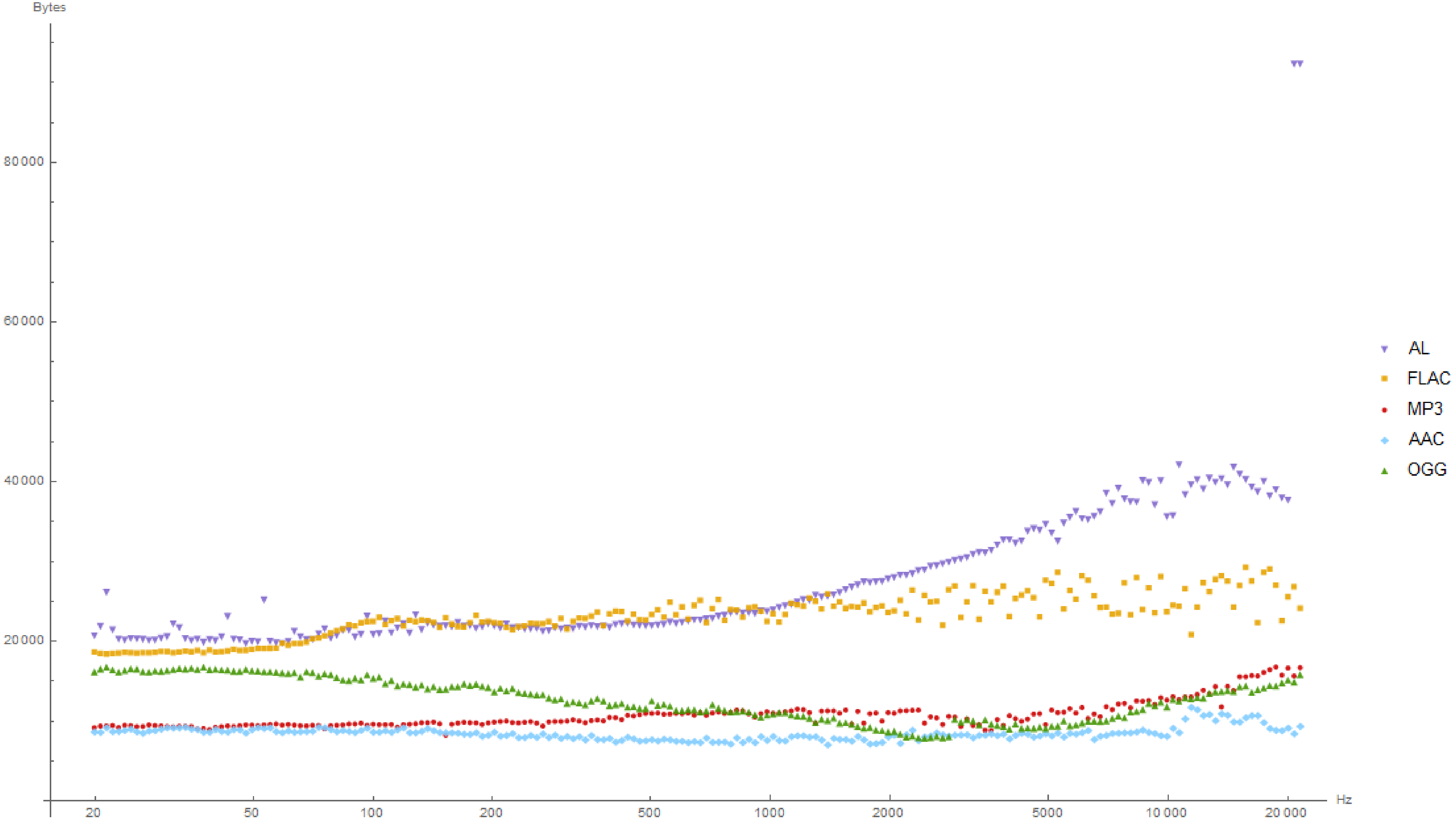

Figure 3: Sine wave, -12 dB (file size × frequency).

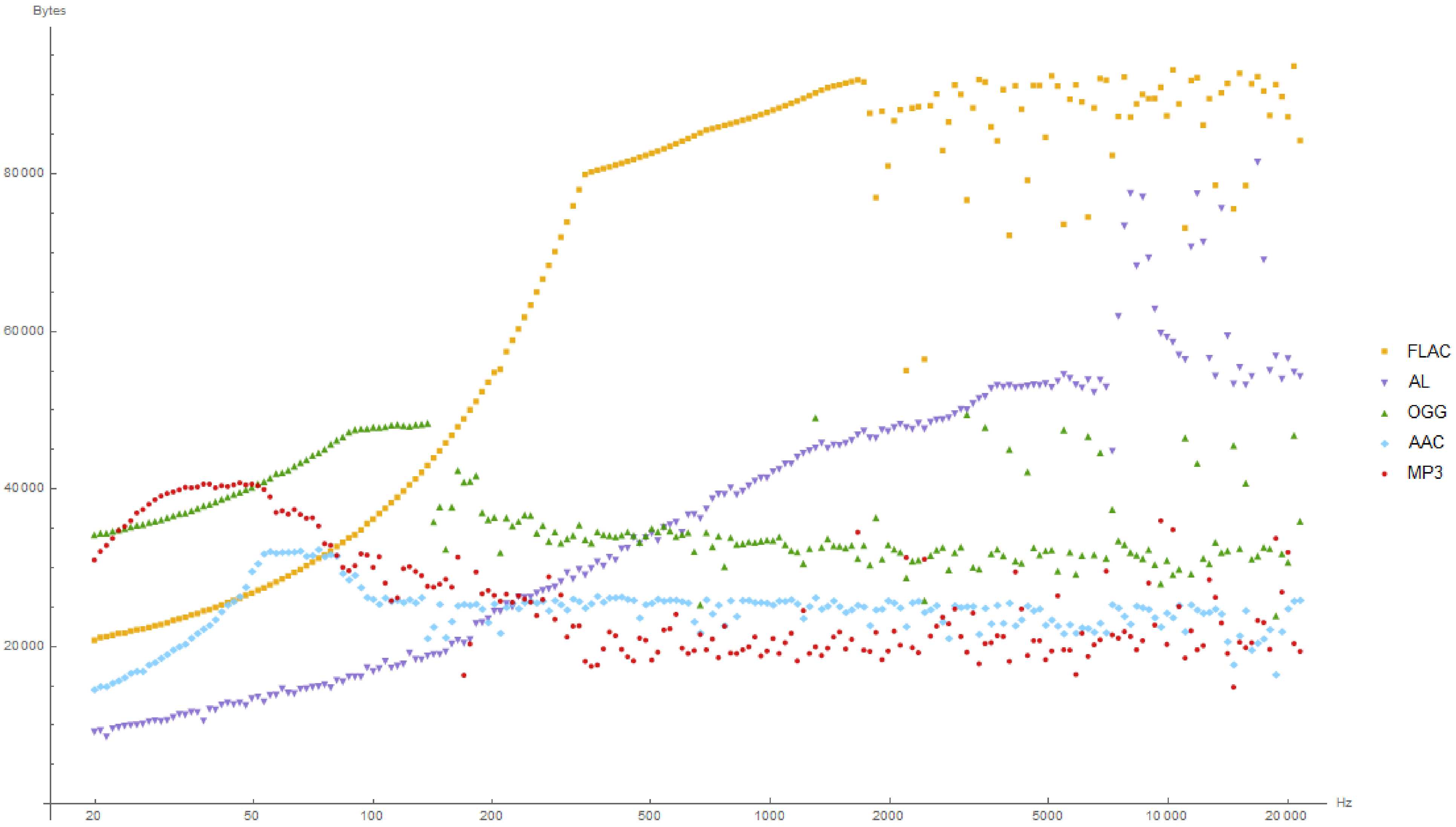

Figure 4: Square wave, 0 dB (file size × frequency).

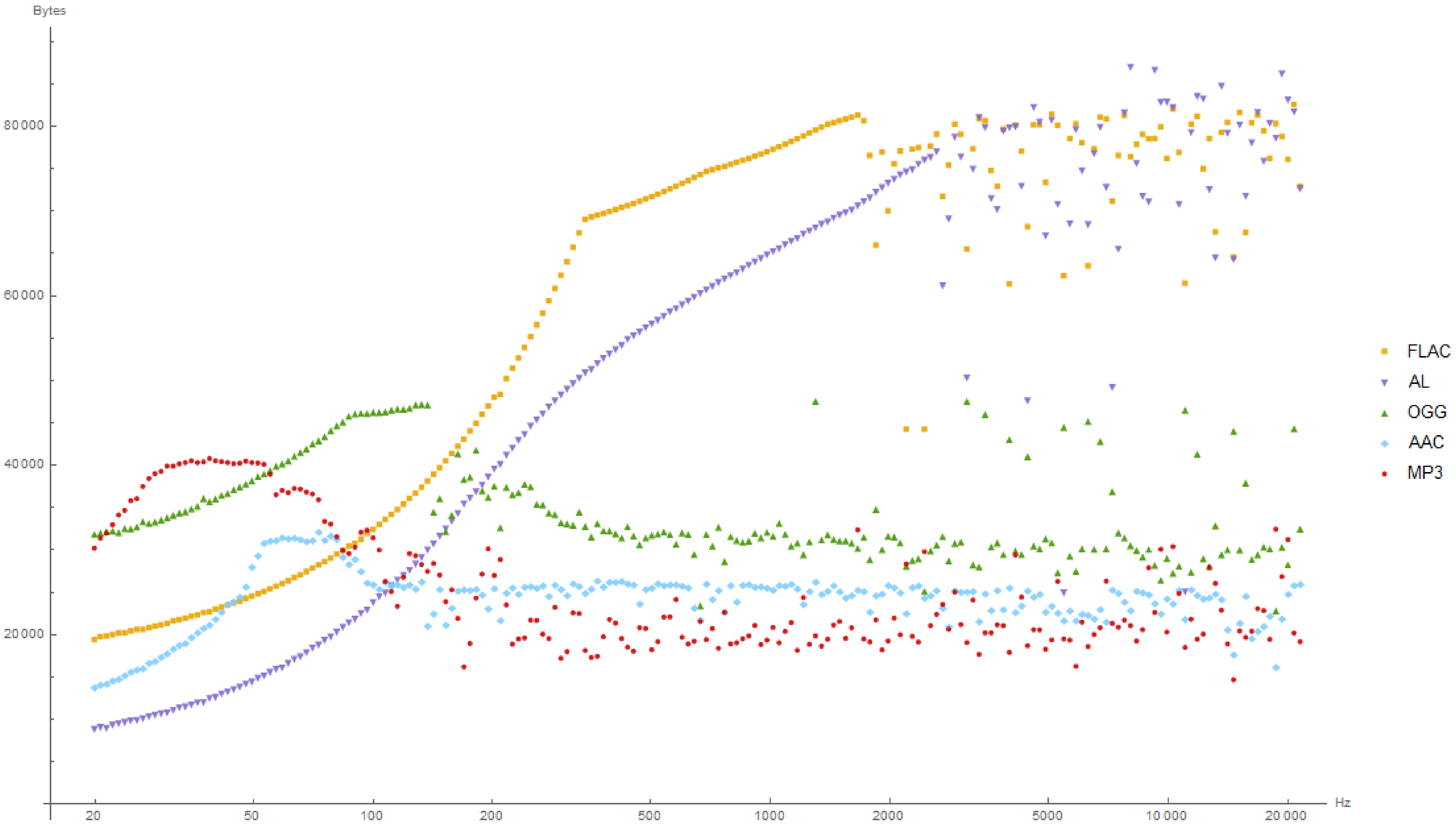

Figure 5: Square wave, -12 dB (file size × frequency).

Amplitude / Loudness

One unexpected finding from the previous graphs is that the 0 dB Apple Lossless files are smaller than the -12 dB ones. This can be seen clearly in Figure 6, which shows four different loudness levels – the 0 dB files are smaller than even those at -24 dB. It could be because the 0 dB files are so loud that it caused the algorithm to modify the audio data in order to avoid clipping.

Figure 7 shows the equivalent results for FLAC files. In this case, there are four sets of data plotted, but it looks like only three because two of the sets (-6 and -12 dB) have almost identical sizes. Similarly, with the sine waves (Figure 8), two of the sets have the same sizes – 0 and -6 dB.

Also, there is a surprising consistency in the variation of file size with loudness, which appears to be proportional to file size. With the square wave set of FLAC files, there is roughly the same size difference between 0 and -12 dB as between -12 and -24 dB. The total sizes of the sets of 200 FLAC files are:

Amplitude:	0 dB	-12 dB	-24 dB
Size:	12.7 MB	11.1 MB	9.5 MB
Size difference:	1.6 MB	1.6 MB	

It’s unclear why there’s such regularity in the sizes of FLAC files, or why it has this particular pattern in relation to frequency, with more variation in the highest frequencies. It’s possibly related to FLAC’s *linear prediction modelling* algorithm, which “exploits the fact that audio data typically has a high degree of sample-to-sample correlation”.[7] Or it could be something to do with the most significant bit corresponding to different loudness levels that’s causing this.

With the square wave FLAC files, the -6 and -12 dB sets have the same sizes, but with sine waves it’s the 0 and -6dB sets. Figure 8 plots three sets of square waves, showing that the 0 dB set is completely obscured by the -6 dB set.

In contrast, the lossy codecs (Figures 9, 10 and 11) produce files of very similar sizes at different amplitudes. All three lossy codecs have a fairly consistent response to frequency only up to around 100 Hz. At higher frequencies, there is more variation in file size. In general, though, louder sounds produce larger files.

[7] https://xiph.org/flac/format.html#scope

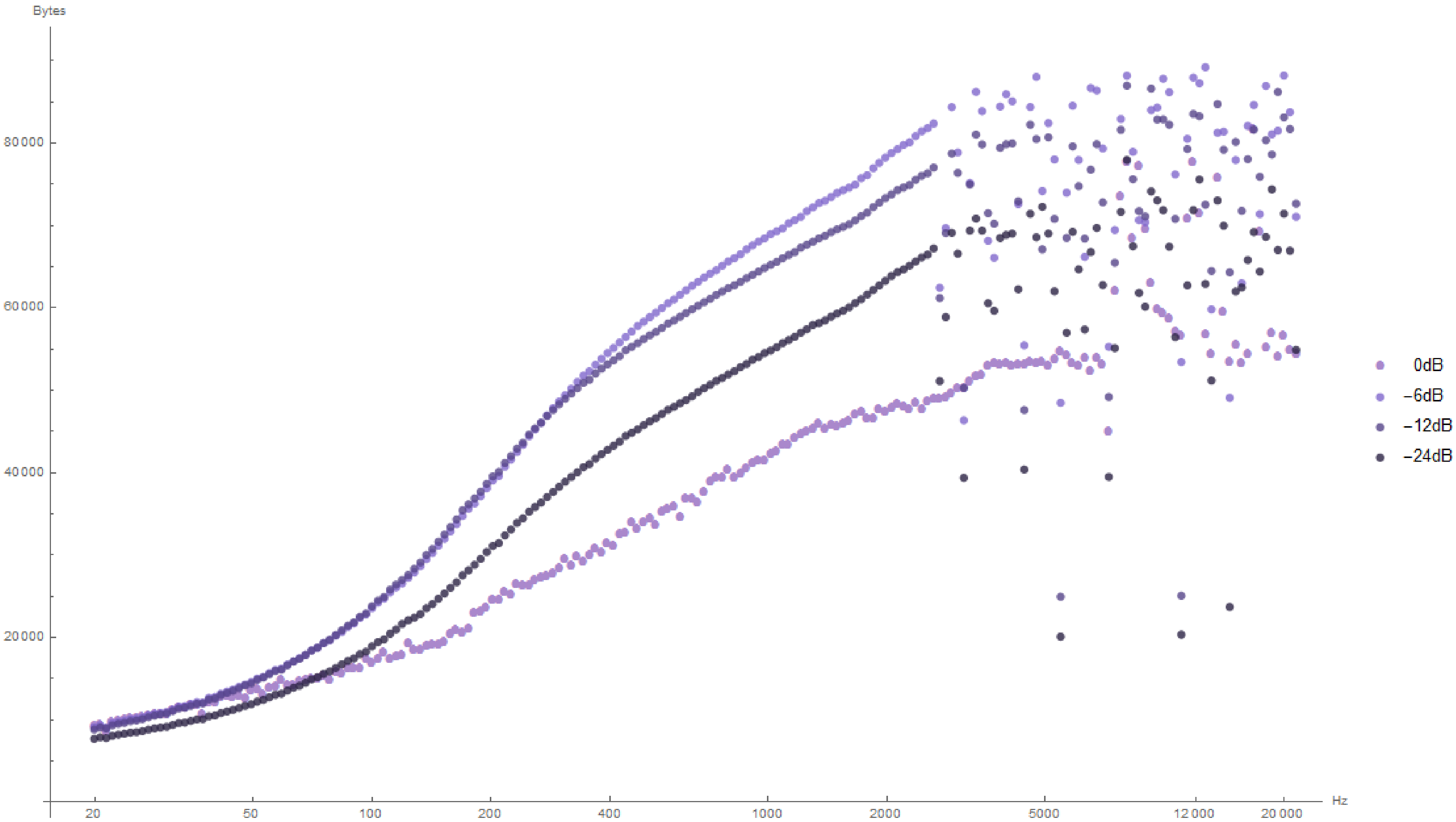

Figure 6: Apple Lossless, square wave, various amplitudes (file size × frequency).

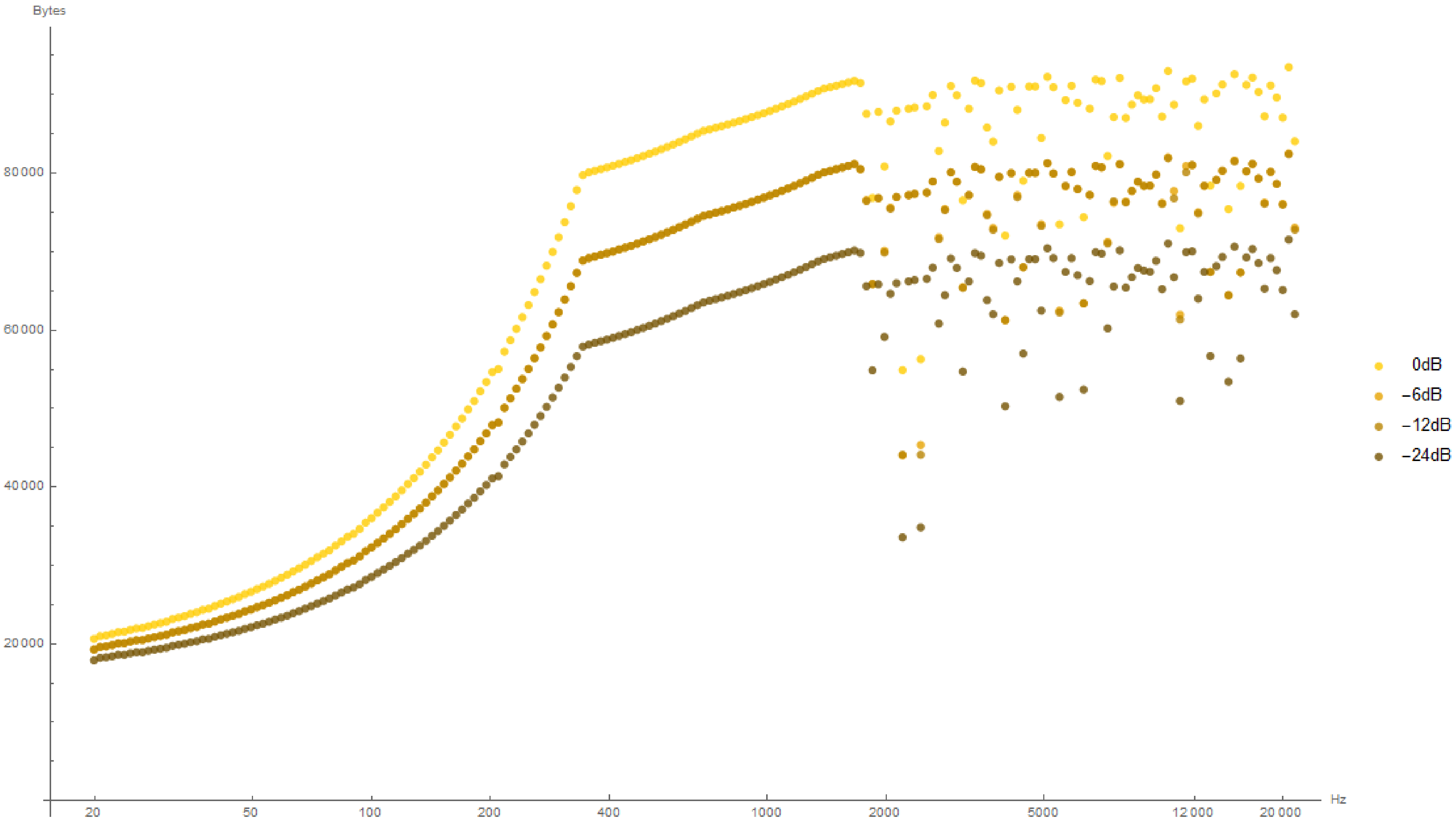

Figure 7: FLAC, square wave, various amplitudes (file size × frequency).

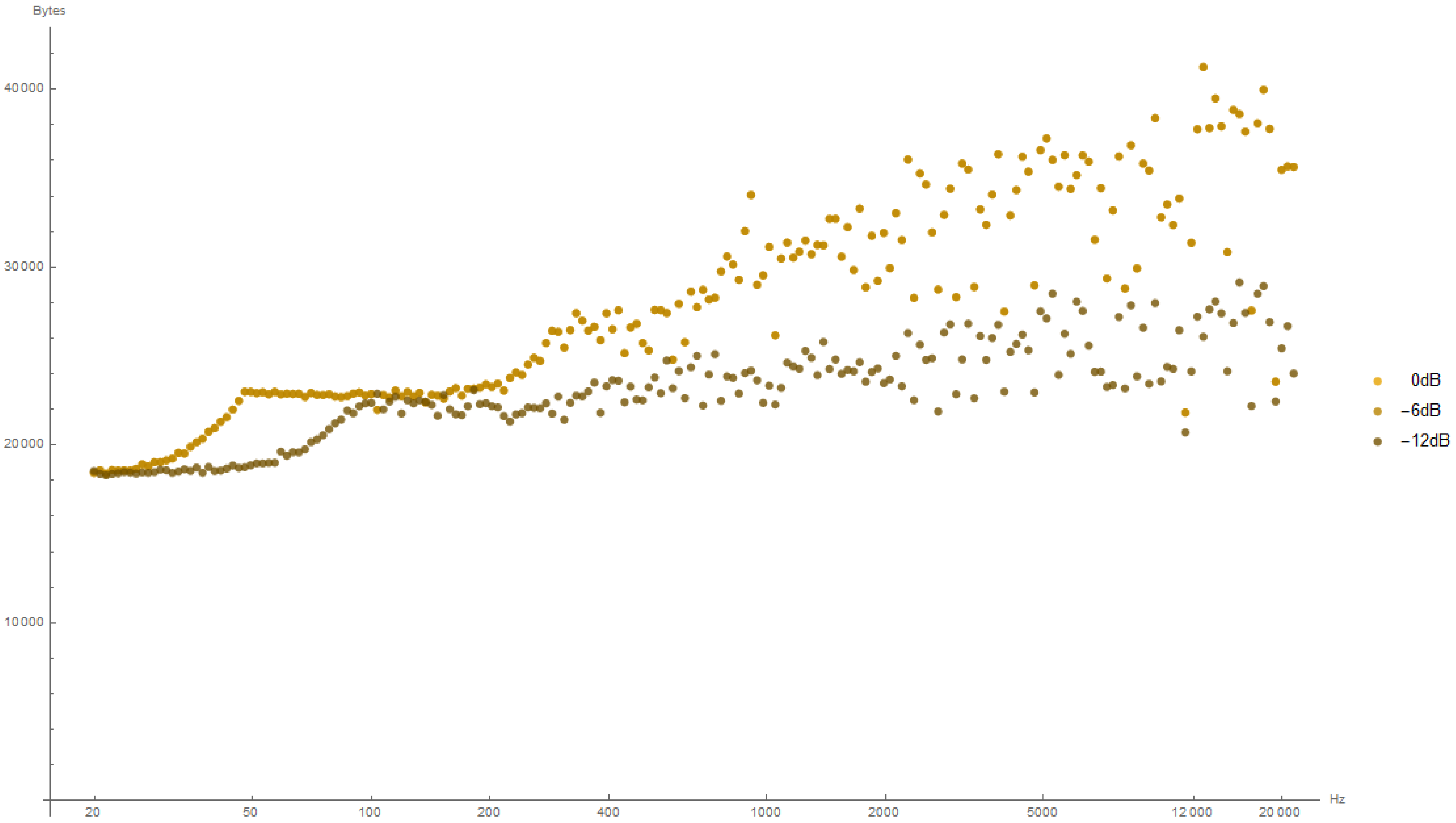

Figure 8: FLAC, sine wave, various amplitudes (file size × frequency).

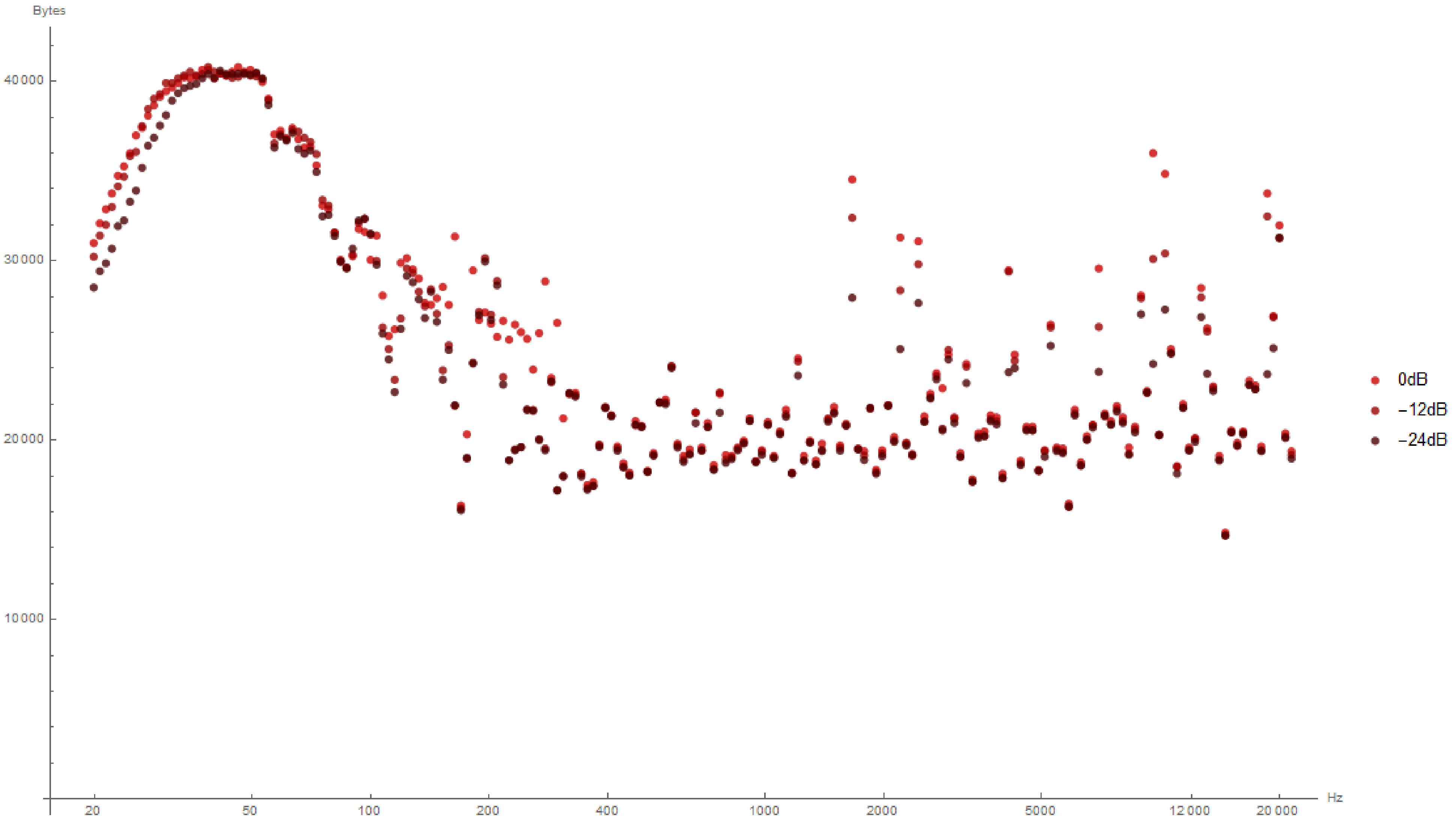

Figure 9: MP3, square wave, various amplitudes (file size × frequency).

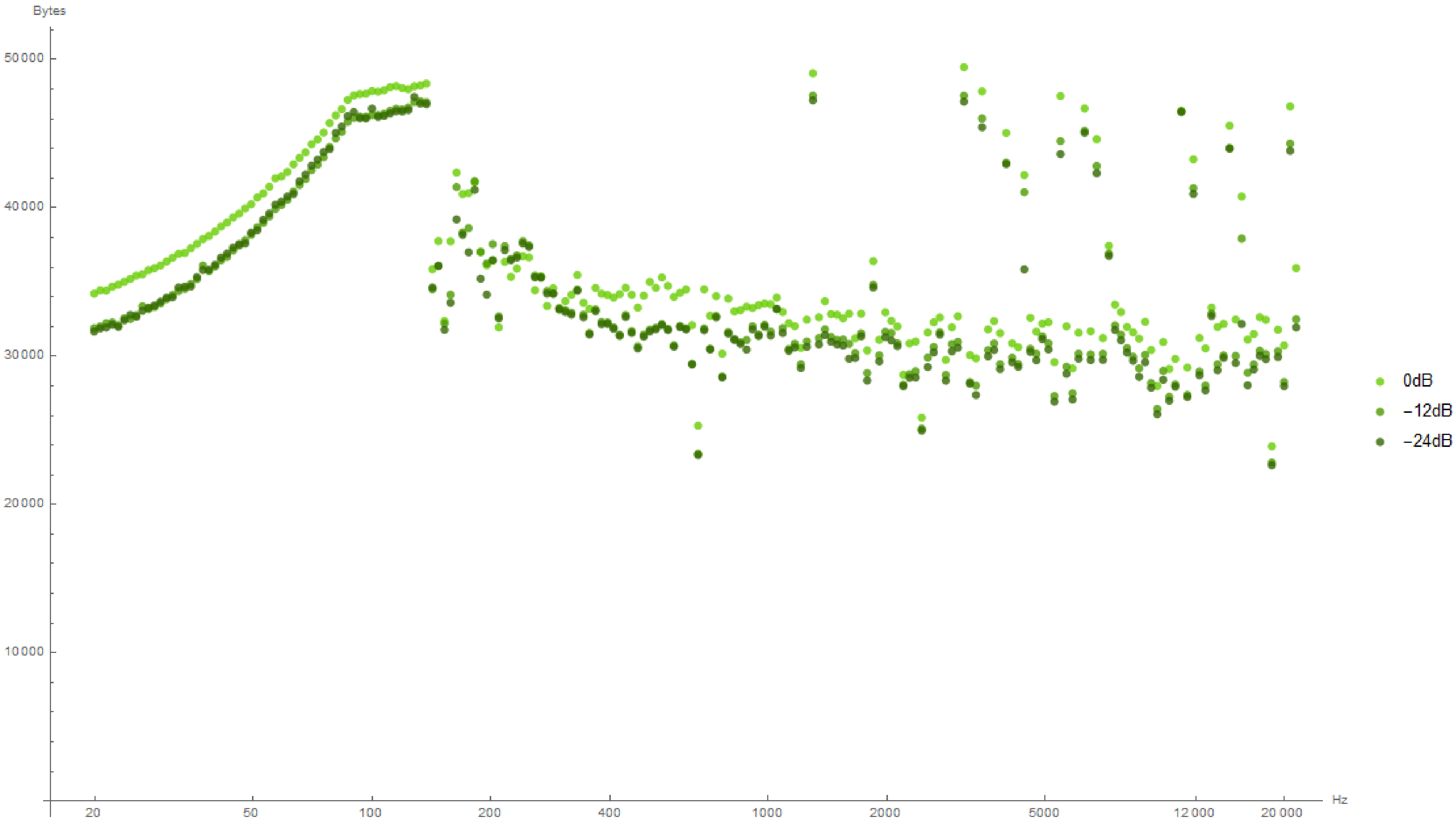

Figure 10: OGG Vorbis, square wave, various amplitudes (file size × frequency).

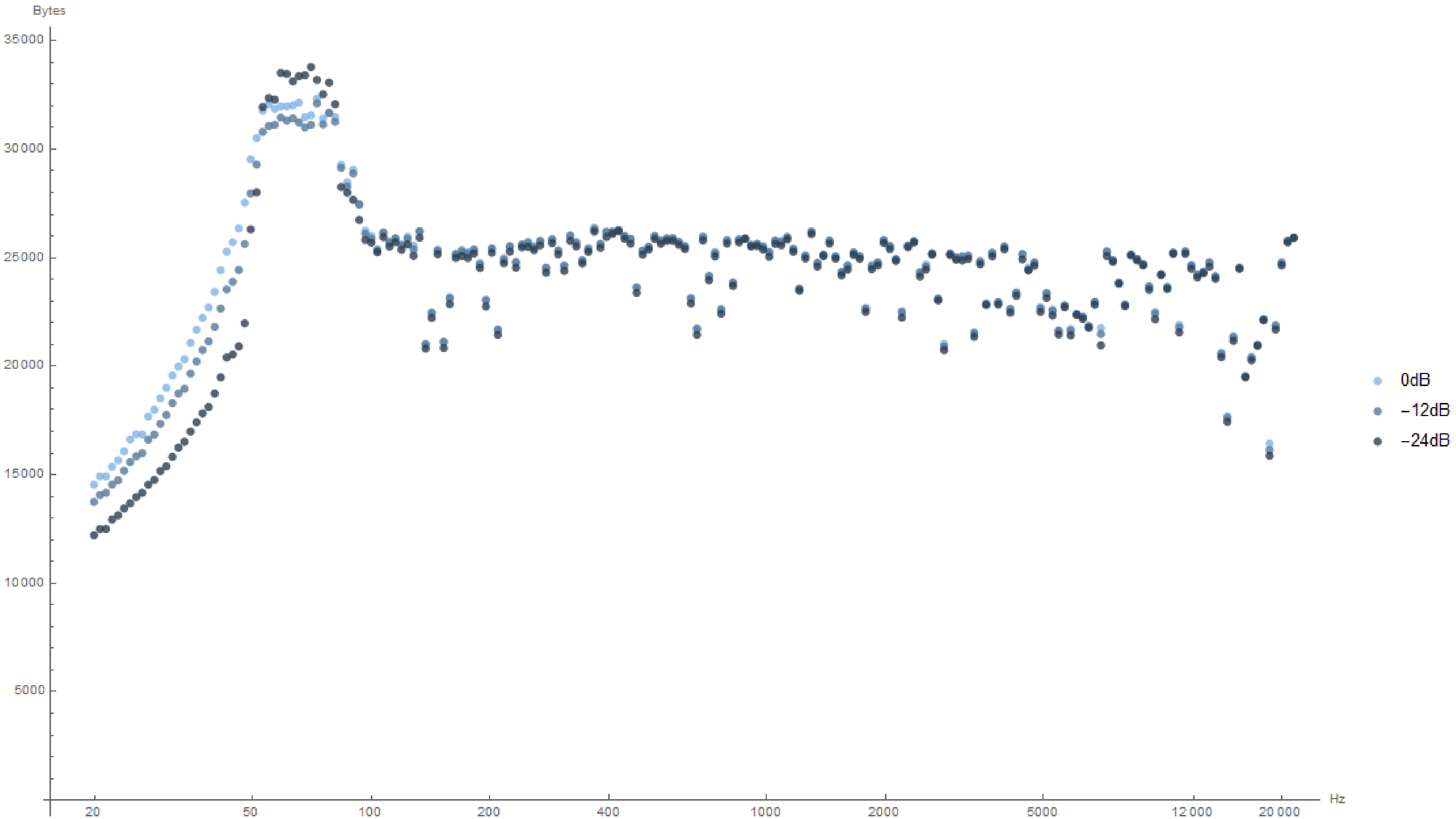

Figure 11: AAC, square wave, various amplitudes (file size × frequency).

Waveform / Timbre

In most cases, the square wave files are larger than the sine wave equivalents (FLAC Figure 12; MP3 Figure 13). By this measure of information content, therefore, a square wave is more complex than a sine wave. In terms of spectral analysis, too, square waves are more complex.

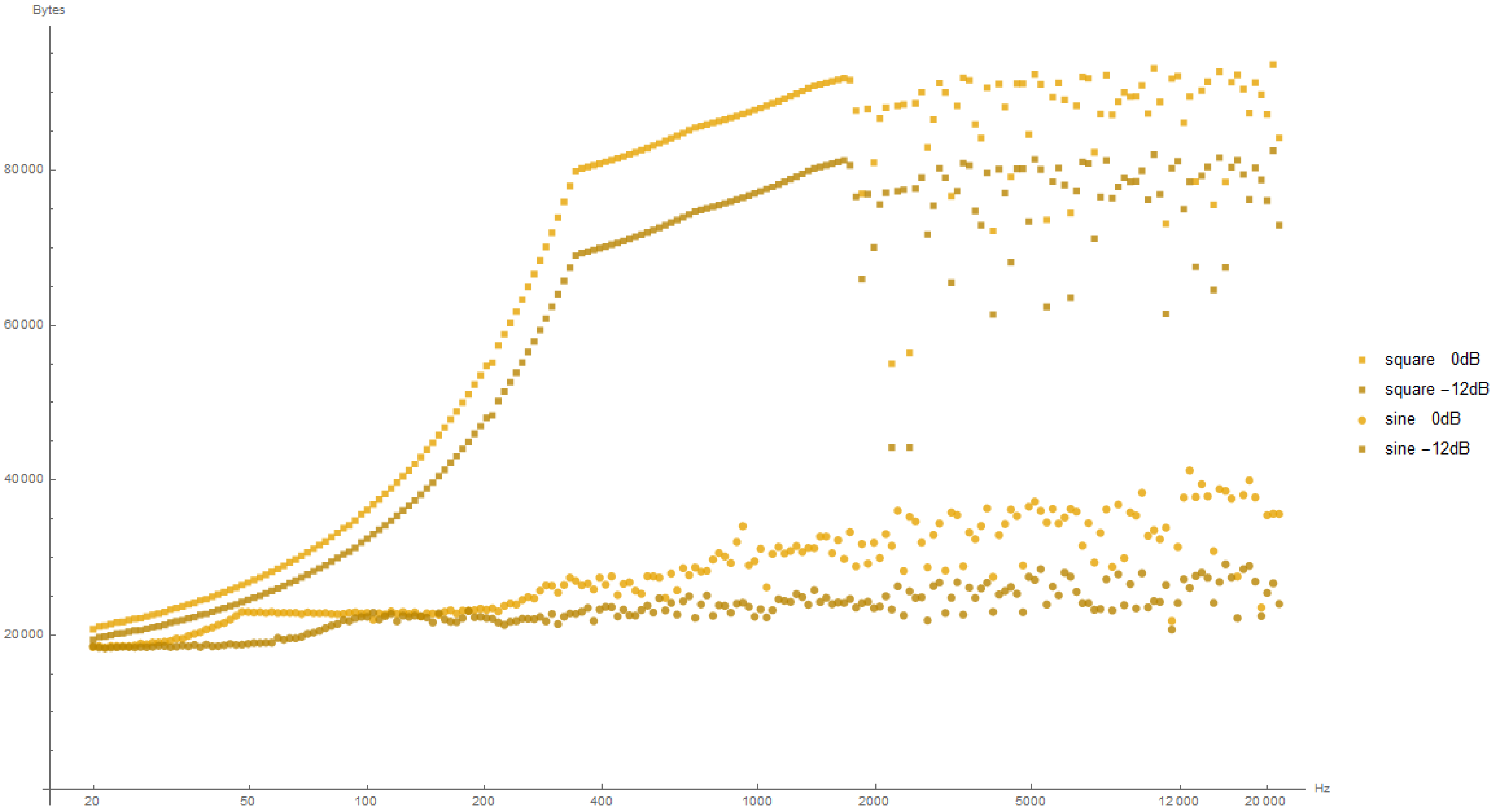

Figure 12: FLAC, square and sine waves, 0 and -12 dB (file size × frequency).

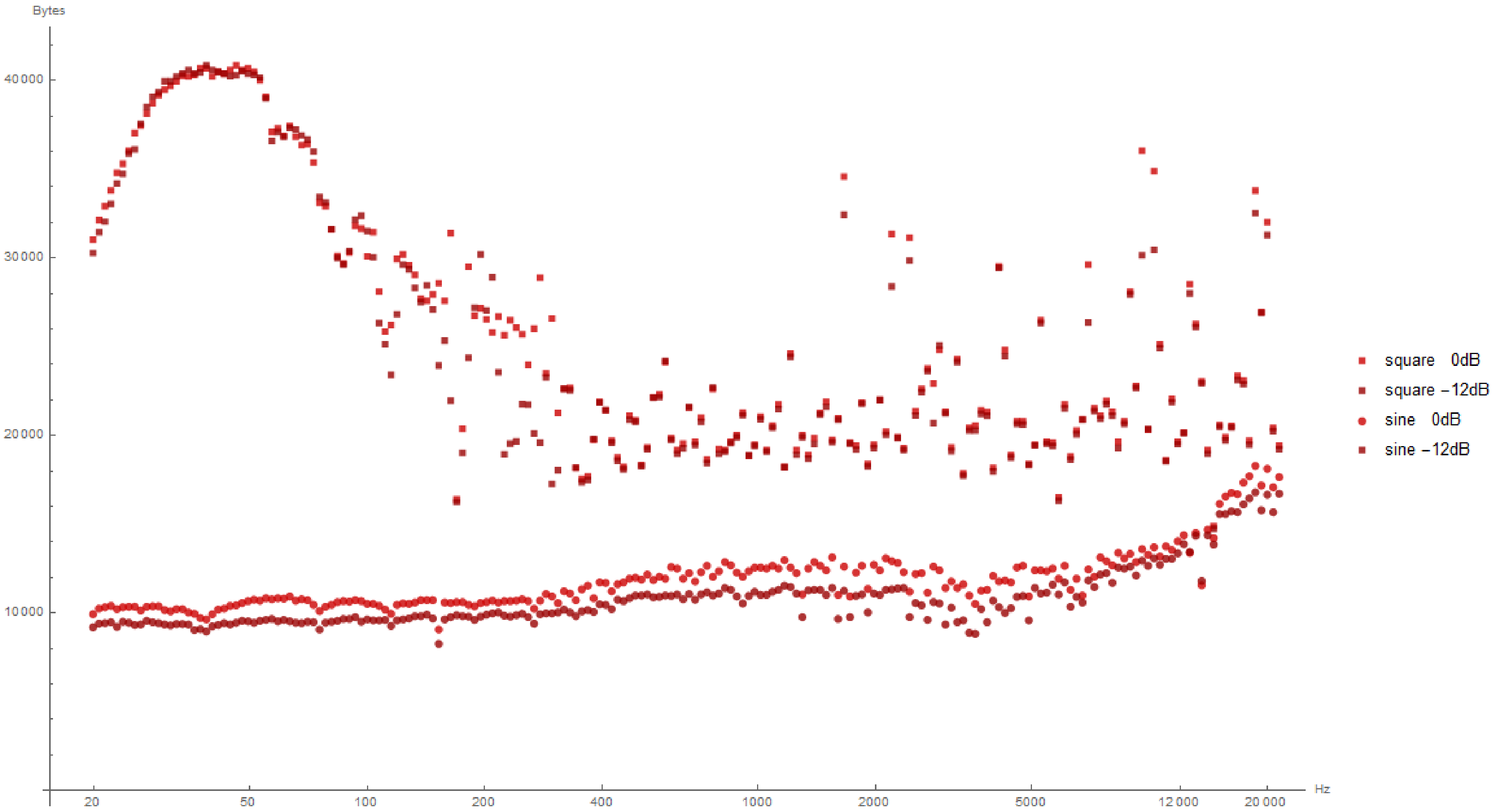

Figure 13: MP3, square and sine waves, 0 and -12 dB (file size × frequency).

Ordering files by size / bit rate

The next few graphs plot frequency of the files when they are sorted in order of size. It's possible to read these graphs like music, since frequency (pitch) is on the vertical axis, like a musical score, and the sequence is from left to right in order of increasing file size. By arranging the actual files with this method and playing them back, we can hear this. The audio that accompanies this document was made this way, using Foobar2000 music player to read the metadata, sort the sets of 200 files, and convert them into a single file – one for each codec. They are the audio equivalent of the graphs below. If the order was perfect, the graph would be a straight line, like Figure 1.

I've used this method as a creative technique in making music – cutting sounds into lots of fragments and re-arranging them according to file size or bit rate. For example, I've used it in projects for the Disquiet Junto[8], in a recent collaborative album with Sun Hammer, *Complexification*,[9] and in a remix that I'm currently working on. When the files are different lengths (unlike the present tests, but more likely when actually making music), sorting them by size or bit rate produces different results.

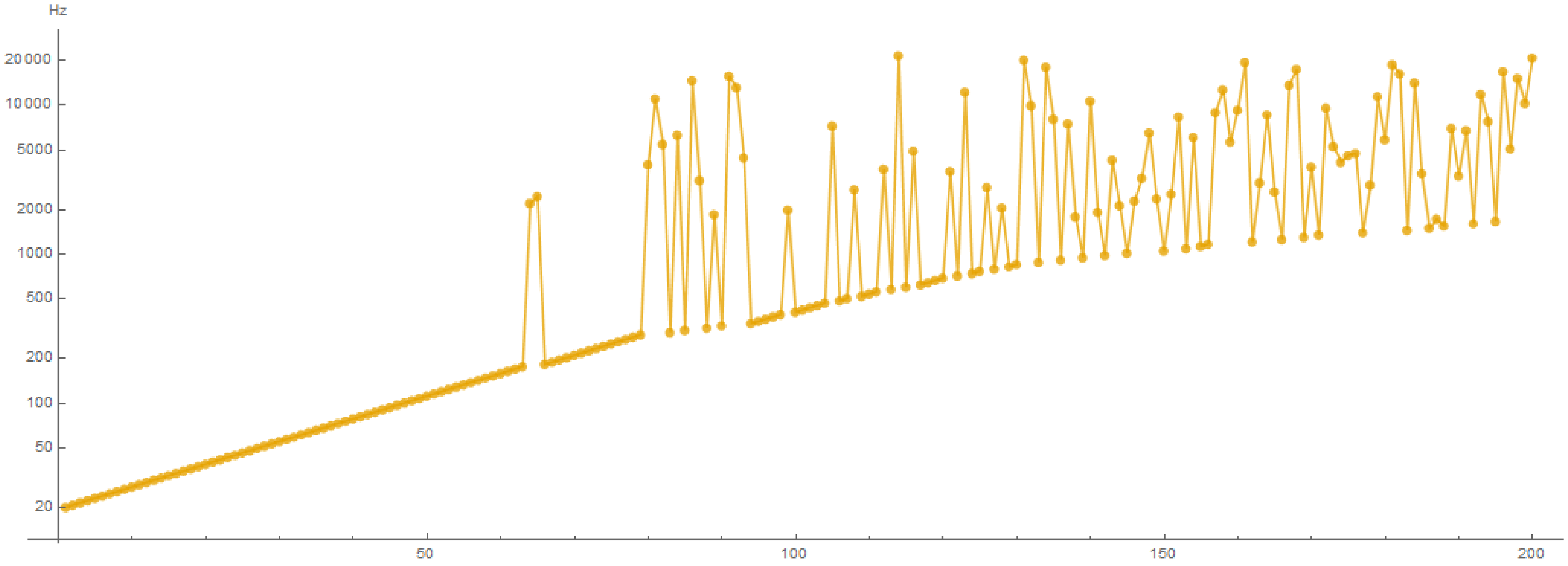

Figure 14: FLAC, square wave, -12 dB (frequency × rank order file size).

[8] http://disquiet.com/2012/01/27/the-disquiet-junto/

[9] http://entracte.co.uk/projects/guy-birkin--sun-hammer-e187/

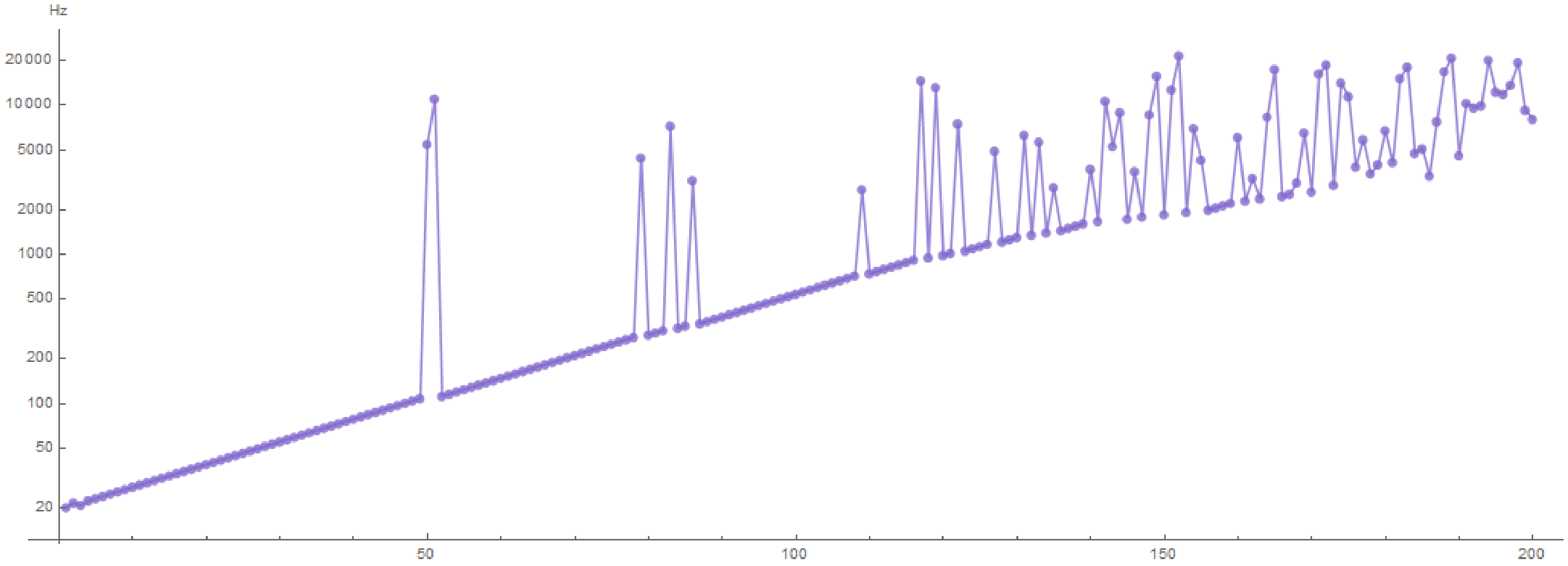

Figure 15: Apple Lossless, square wave, -12 dB (frequency × rank order file size).

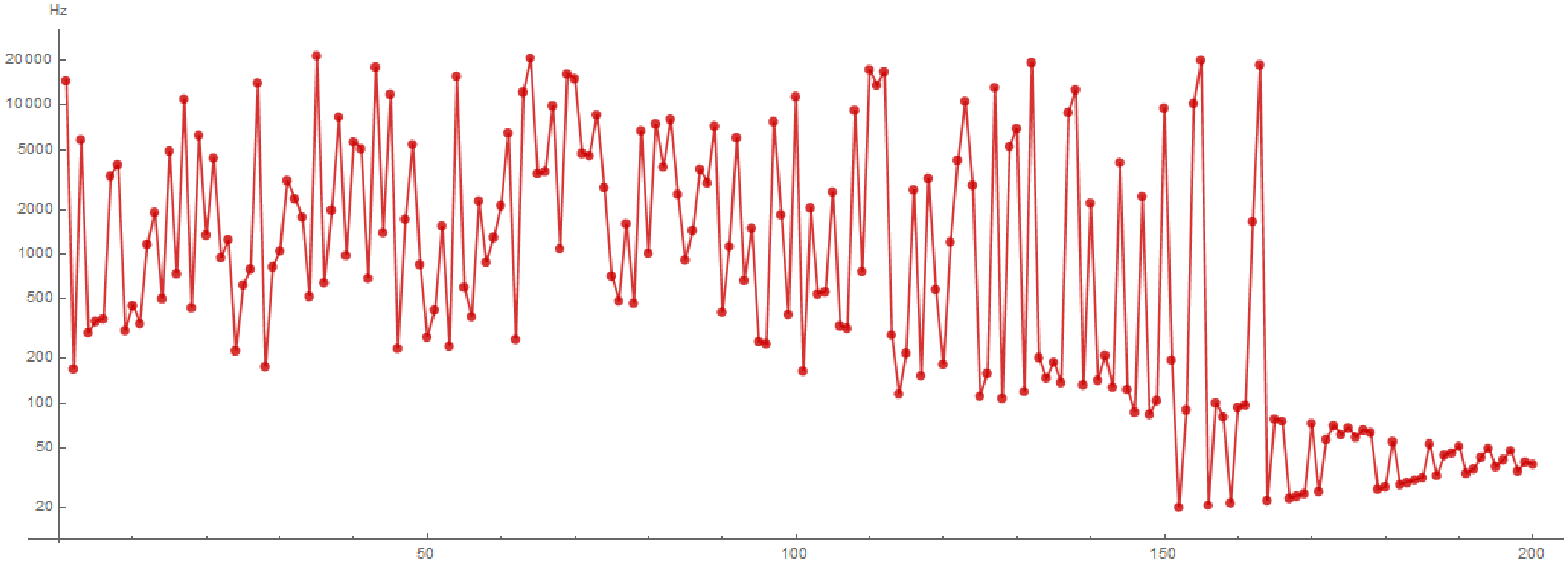

Figure 16: MP3, square wave, -12 dB (frequency × rank order file size).

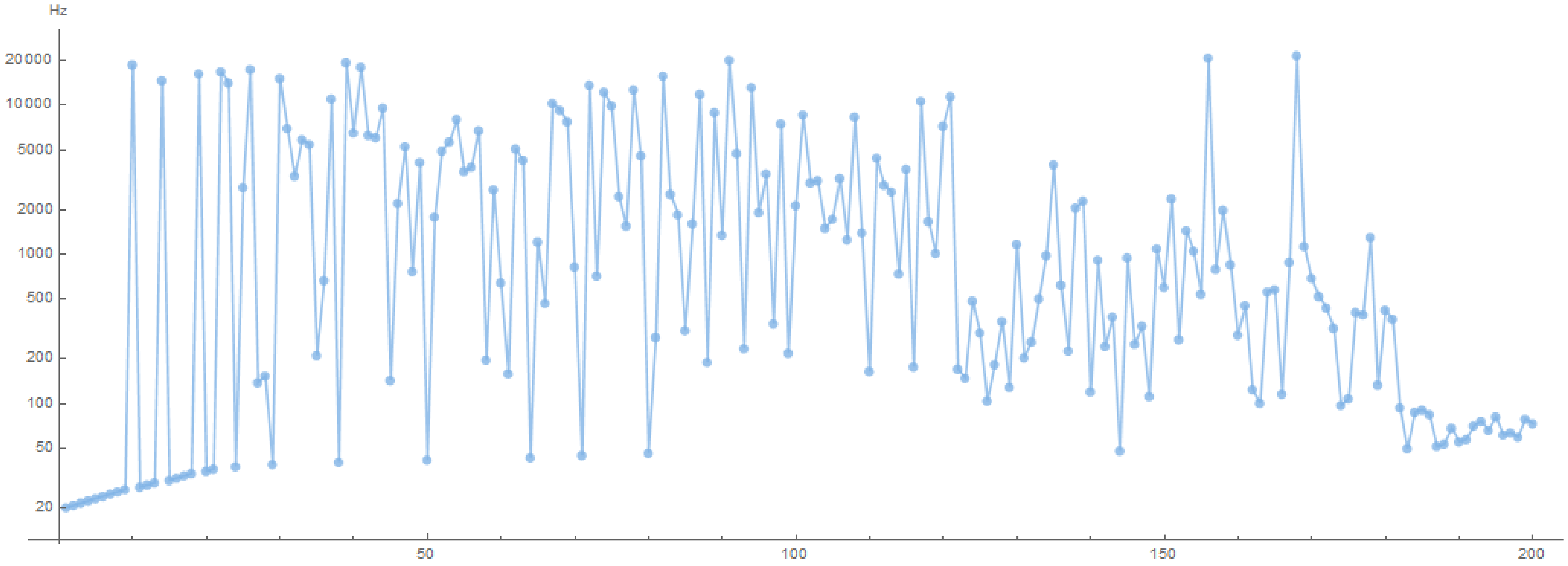

Figure 17: AAC, square wave, -12 dB (frequency × rank order file size).

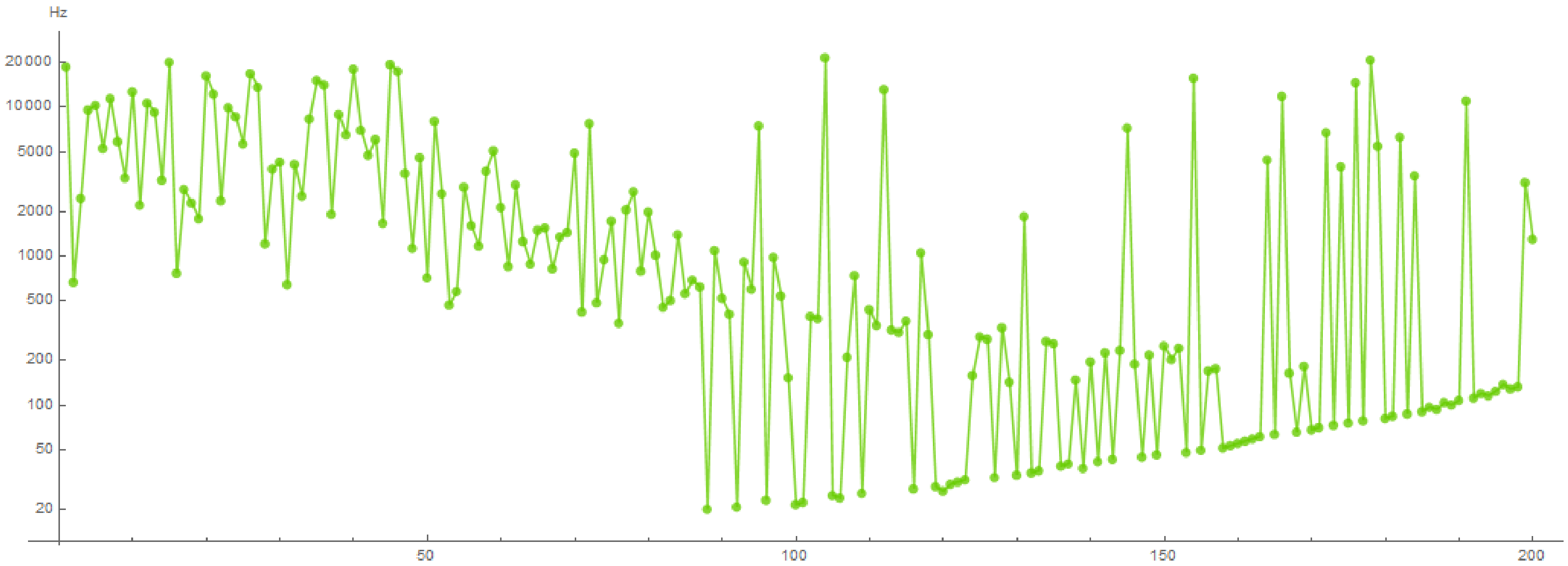

Figure 18: OGG Vorbis, square wave, -12 dB (frequency × rank order file size).

Thoughts

How good are the different audio codecs as measures of sonic complexity?

To some extent, this question depends on how we view the complexity of the set of test sounds. For example, whether two similar sounds at different frequencies are equal in complexity or not. From a perceptual point of view there's little difference – two pitches of the same timbre are equally complex. But from an informational point of view, a higher frequency could be regarded as being more complex, given that it involves a more rapid oscillation – more variation per unit of time. This is the pattern we found in the files sizes of some of these audio codecs.

Over the full range of frequencies, the lossy codecs produce files with little variation in size (like the perceptual point of view), whereas the lossless codecs definitely increase in size with frequency (like the informational view). Which one is 'better' for the purpose of measuring complexity? At this stage, it's impossible to say. The way in which the Apple Lossless codec produces sudden jumps in file size and the way it responds to high amplitude sounds are problematic for measuring complexity – what we need is something that reacts more predictably.

Different waveforms vary in compressibility, and produce larger or smaller files. Square waves make larger files than sine waves, especially at higher frequencies. Square waves are also richer in terms of spectral content, having odd-numbered harmonic overtones, whereas sine waves have none. We can perceive this timbral difference, but that doesn't necessarily correspond with greater perceived complexity.

Next

Future tests will look at:

Duration

Longer files, variety of lengths. Because digital file formats have headers that contribute a small amount to the size of a file, this skews the results slightly. Using longer audio files should reduce the effect of this overhead – testing will establish how much of an effect it has.

Timbres/soundwaves

Noise. Different types – white, pink, brown, blue, grey. We might find some interesting results with noise, because it will reveal more clearly the effects of the various filtering processes in audio codecs. The MP3 codec, for example, uses a *hybrid polyphase filterbank* that reduces the amount of high frequency content, which makes for better compression and smaller files.[10] Lossy codecs like MP3, OGG Vorbis and AAC also use psychoacoustic modelling that involves various filtering processes that save space for the frequency bands that we are most sensitive to.[11] In theory, this might make MP3 a better candidate for measuring sonic complexity in a way that more closely matches human perception.

---

[10] http://wiki.hydrogenaud.io/index.php?title=High-frequency_content_in_MP3s

[11] http://wiki.hydrogenaud.io/index.php?title=MP3#The_psychoacoustic_model

Certainly, a lot of effort has gone into the development of codecs to improve both sound reproduction quality and data compression capabilities. This is useful for the attempt to investigate sonic complexity with data compression.

Stereo sound

Two channels instead of one. Does greater stereo difference produce proportionally larger file sizes?

Time

One of the big differences between these tests of sonic complexity and my earlier research on visual complexity is that sound is time-based. The present tests don't even begin to explore the temporal aspects of sound. Future tests will explore ways of incorporating time into measures of sonic complexity. For example, looking for a measure that tracks changes bit rate over time – a dynamic measure – rather than averaging it out over a whole sound file. This might be achievable if it's possible to inspect the data in the audio files, for example reading the size of each frame or chunk of encoded data in VBR files. With such data we could plot a histogram of frames by size, or plot the block sizes over time.

Guy Birkin

November 2015

https://aestheticcomplexity.wordpress.com/