Grand Valley State University
ScholarWorks@GVSU

Peer Reviewed Articles | Mathematics Department

10-2017

Imitating the Shazam App with Wavelets

Edward Aboufadel
Grand Valley State University, aboufade@gvsu.edu

Follow this and additional works at: https://scholarworks.gvsu.edu/math_articles

 Part of the Mathematics Commons

ScholarWorks Citation

Aboufadel, Edward, "Imitating the Shazam App with Wavelets" (2017). *Peer Reviewed Articles*. 1.
https://scholarworks.gvsu.edu/math_articles/1

This Article is brought to you for free and open access by the Mathematics Department at ScholarWorks@GVSU. It has been accepted for inclusion in Peer Reviewed Articles by an authorized administrator of ScholarWorks@GVSU. For more information, please contact scholarworks@gvsu.edu.

Imitating the *Shazam* App with Wavelets

EDWARD ABOUFADEL
Grand Valley State University
Allendale, Michigan
aboufade@gvsu.edu

Like searching for a needle in a haystack, suppose that we have a large set of signals (finite sequences of numbers) $\{\mathbf{s_1}, \mathbf{s_2}, \mathbf{s_3}, \dots\}$, and a special signal $\mathbf{q}$ that may or may not be in the collection. How can we find signals in the collection that are similar if not identical to $\mathbf{q}$, and *how can we do this quickly?* A solution to this question is the basis of the *Shazam* smartphone app, where a listener captures a short excerpt of a recorded song with the smartphone's microphone, and in a matter of moments the app reports the name of the song and the artist [**12**]. There the "needle" is the excerpt, and the "haystack" is a vast corpus of popular music. The *Shazam* algorithm is powered by Fourier analysis [**15**], and the purpose of this paper is to present a simpler, wavelet-based method that captures the basic process used by the app.

Solutions to this problem are useful in situations where the description of the "needle" might not be precise or may have noise in it, such as the *Shazam* problem, and where there will be frequent searches of the "haystack." For this presentation, we will use a "haystack" of comparable and accessible signals. The Jaeb Center for Health Research has made a large database of continuous glucose monitor (CGM) data available to the public*. The data comes from a recent study of type-1 diabetes (an autoimmune disorder characterized by the destruction of the islet beta cells in the pancreas by the body's own immune system) that involved 451 patients wearing a CGM for 6 or 12 months [**9**]. In Figure 1 we see an example of a "CGM day" from a type-1 diabetic patient: 288 readings of positive integers—one every five minutes.

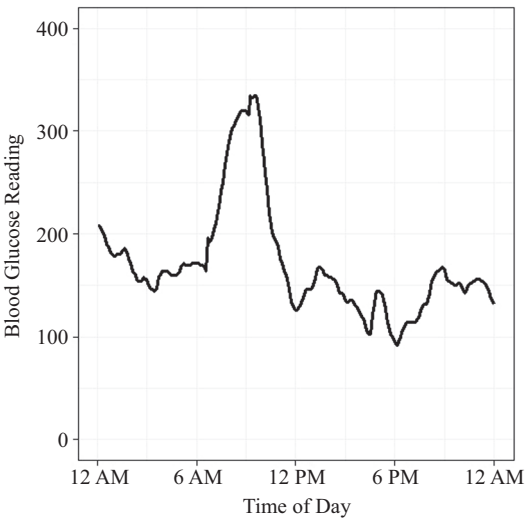

Figure 1 Example of an blood glucose daily chart.

Math. Mag. **90** (2017) 287–295. doi:10.4169/math.mag.90.4.287. © Mathematical Association of America
MSC: Primary 65T60

*http://diabetes.jaeb.org/RT_CGMRCTProtocol.aspx

In the following, we will describe how to take a "needle"—a CGM day labeled **q**—and find similar days in our database. The key idea—which is how the *Shazam* app works—is to create reduced, and practically unique, representations of every CGM day in the database, and then to compare these short representations rather than the original signals. Basically, there is an initial investment of time and computing power to generate short signatures for each signal in the database, and then a search can be performed quickly by comparing the signatures. To create the signatures, we will use a visualization called a *wavelet scalogram*. Wavelets are excellent tools for this type of signal reduction for CGM data, because of the rough and irregular quality of the data.

Wavelet filters

For our signal matching algorithm, wavelet filtering is an important tool. Wavelets have been applied in a wide range of areas, such as video cameras [**5**], Internet worm detection [**6**], and the design of low-power pacemakers [**8**]. Wavelets came to prominence in the late 1980s, and Ingrid Daubechies was a key researcher in their development. For good reason, there are families of wavelet filters that are named after her [**11**]. In our simplified version of the *Shazam* process, we will use the Daubechies-6 wavelet filters. Applying a *filter* to a finite time series yields another finite series. Filtering a time series may identify structures in the signal.

There are two Daubechies-6 filters: a *low-pass filter* which creates a "blur" of the input time series, and a *high-pass filter* which reveals details within the time series. The low-pass filter is a weighted average of entries in the time series, leading to a new and shorter time series that appears similar to the original. Entries in the high-pass filter output are close to 0 when the input series is nearly constant, linear, or quadratic, while other behavior in the input series shows up as larger values (in absolute value) in the high-pass output.

Given a signal $\mathbf{s} = \{s_i\}$, i from 1 to n, we take six entries at a time to compute filter outputs. For the low-pass filter [**14**], the calculation is

$$H_k = h_0 s_{2k} + h_1 s_{2k+1} + h_2 s_{2k+2} + h_3 s_{2k+3} + h_4 s_{2k+4} + h_5 s_{2k+5}, \tag{1}$$

where the filter coefficients $\{h_i\}$ are

$$\begin{aligned} &h_0 \approx 0.2352, h_1 \approx 0.5706, h_2 \approx 0.3252, \\ &h_3 \approx -0.0955, h_4 \approx -0.0604, h_5 \approx 0.02491 \end{aligned} \tag{2}$$

and k varies from 1 to $n/2$. These coefficients were derived by Daubechies so that the underlying wavelet functions satisfy certain properties: compact support, orthogonality, and regularity [**1**]. Filtering in this way creates a new signal that is half the length of the input signal. The high-pass filter calculation is

$$G_k = -h_0 s_{2k-4} + h_1 s_{2k-3} - h_2 s_{2k-2} + h_3 s_{2k-1} - h_4 s_{2k} + h_5 s_{2k+1}. \tag{3}$$

Applying these filters yield two new time series of length $n/2$. The low-pass output is a weighted moving average of entries in the original time series, while the high-pass output captures short-term changes in the time series. We call this the first scale of the analysis.

When using wavelet filters, a *pyramid scheme* is often implemented. A pyramid scheme involves applying the filters over-and-over to the low-pass output from the previous scale. So the second scale consists of two new time series of length $n/4$, the third scale has output length $n/8$, etc. There is a point where the resulting low-pass output is quite short and no further filtering is useful.

Before filtering the CGM signals, we apply a common technique known as "padding" the signals [**13**], which is extending the signal s_i in some way for $i < 1$ and/or $i > 288$. This is necessary for some of the low-pass filter calculations for k near 288 and some of the high-pass calculations for k near 1. Some ways to pad signals might be by adding 0's on the end, or extending the series as if it were periodic. (For example, if the original signal has length 28, then assign $s_{29} = s_1$, $s_{30} = s_2$, etc.) Since a CGM day is 24 hours, the periodic approach makes sense here.

When filtering each CGM day $\mathbf{q}$, the first round yields two signals of length 144. Other rounds produce signals of lengths 72, 36, and 18, and the fifth produces length 9. At this point, we no longer have an even number of entries and the low-pass output of length 9 is quite short, so we end the scheme, leaving us with low- and high-pass outputs on five different levels. In Figure 2 we see two examples of CGM days and their corresponding scale-5 "blurs."

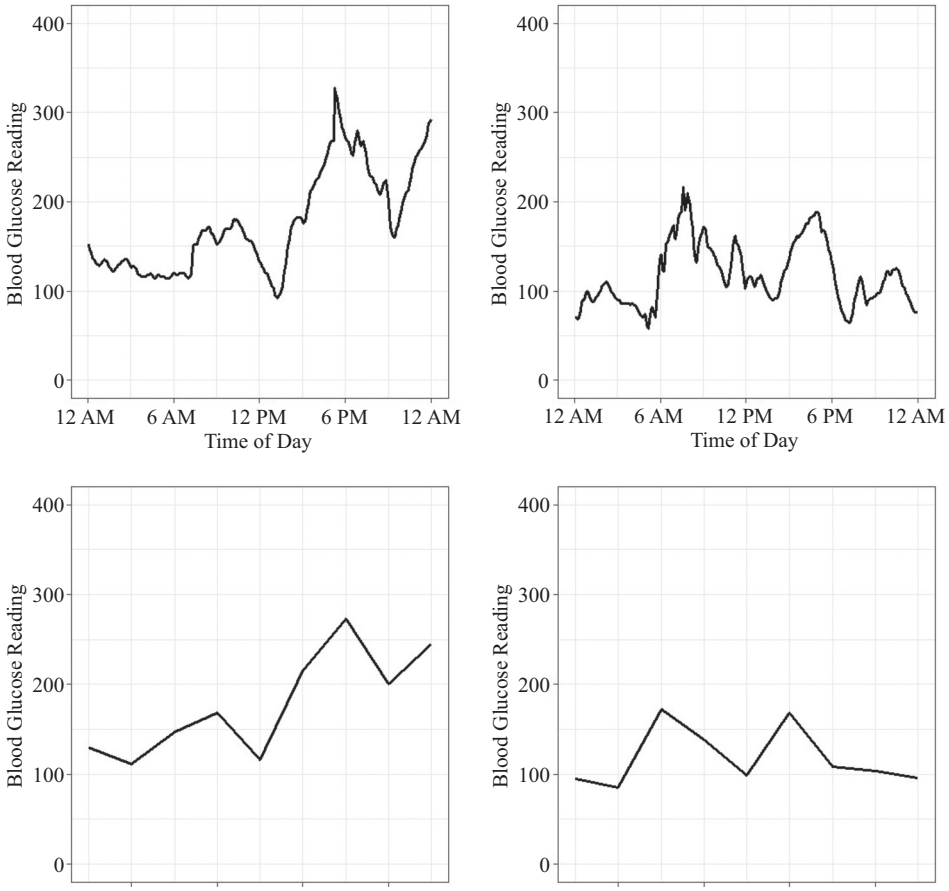

Figure 2 CGM days and scale-5 wavelet blurs.

Creating wavelet scalograms and signatures

If speed was not important, we could compare the original signals using classic distance formulas. For example, for any pair of signals of equal length $\mathbf{q} = \{q_i\}$ and $\mathbf{s} = \{s_i\}$, $i = 1, \ldots, n$, there are several distances, or norms, that we can calculate between these signals, such as the 1-norm, the 2-norm, or the infinity norm (or sup norm):

$$\|\mathbf{q}-\mathbf{s}\|_1 = \sum_{i=1}^{n} |q_i - s_i|,$$

$$\|\mathbf{q}-\mathbf{s}\|_2 = \sqrt{\sum_{i=1}^{n} (q_i - s_i)^2},$$

$$\|\mathbf{q}-\mathbf{s}\|_\infty = \max_{1\le i\le n} |q_i - s_i|.$$

However, to perform a signal match *quickly* is important, and for larger n and a large database of signals, these calculations can take considerable time. In this section we will describe how to reduce the original CGM signals to shorter signatures that can be compared instead.

A *wavelet scalogram* is a visual representation of the high-pass filter output that highlights the most dramatic changes in the signal. For our method, for each CGM day we create a scalogram as follows: We start with the five scales of analysis that are calculated from a CGM signal, giving us 279 high-pass outputs (from scales of length 144, 72, 36, 18, and 9). After identifying the 16 largest and 16 smallest (most negative) entries. we replace those largest entries with the tag $+1$, the smallest entries with the tag -1, and the remaining 263 entries with the tag 0. This is a lossy process (it cannot be reversed) and it is an example of *percentile thresholding*, which is often applied in signal processing to filter output.

After thresholding, the scalogram is created by using our time of day as the x-axis and scale as the y-axis. At each scale we populate a row of rectangles, one for each high-pass output entry, and color the rectangles black for a top-16 entry (a tag of $+1$), gray for the bottom-16 entries (a tag of -1), and white for the rest of the entries. When comparing the scalogram to the original time series, one can see that, in general, black and gray areas of the scalogram correspond to places of interesting behavior in the time series. Black areas usually correspond to where blood glucose is rising, while gray is where blood glucose is falling, and we might get some black or gray rectangles on the left or right end due to the periodic padding of the signal. The scale indicates the length of time of the behavior, with the larger scale capturing behavior over longer periods of time. Two examples of scalograms can be found in Figure 3.

From each scalogram we can create a 71-item vector for each day that we will call a *signature*. The first nine entries will be the nine low-pass output values on the fifth scale, each rounded to the nearest integer. The remaining entries are the tags from the fifth, fourth, and then third scales. In Figure 3, the left example is* 269-36827, and here is the signature for this CGM day, divided up as it was created:

[67, 79, 133, 63, 92, 165, 191, 148, 155;

$-1, 1, -1, 0, 1, 1, -1, 1, 1$;

$0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, -1, -1, 0, -1, 1, 1, 0$;

$0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0,$

$-1, 0, -1, 0, -1, 0]$.

In this way we can represent each CGM signal $\mathbf{q}$ (of length 288) by a shorter, structured signature $\tilde{\mathbf{q}}$ of nine positive integers and 68 tags from the set $\{-1, 0, 1\}$. Once these signatures are calculated, we will use them instead of the original signals to compare CGM days.

*given by patient number in the Jaeb database and "Excel Day," where January 1, 1900 at 12 midnight is "1."

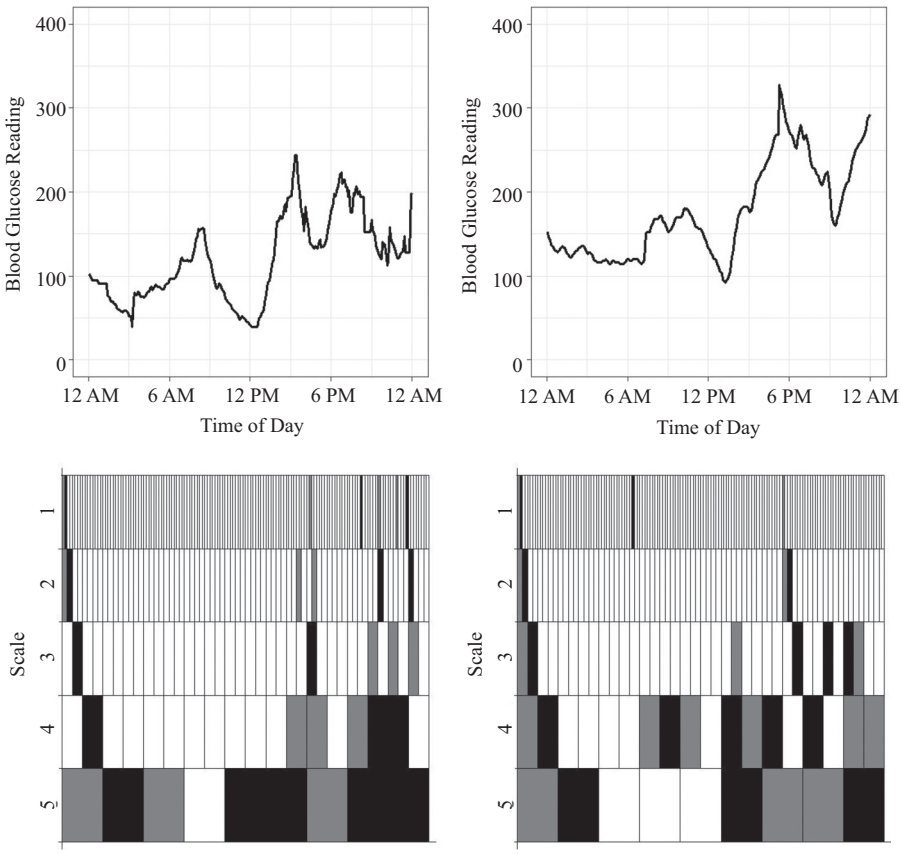

Figure 3 CGM days and scalograms.

Matching CGM days

In this section we will define a measure of similarity S between two signatures $\tilde{\mathbf{q}}$ and $\tilde{\mathbf{s}}$. We will consider these signatures (and the corresponding CGM days) to be "similar" if the measure $S(\tilde{\mathbf{q}}, \tilde{\mathbf{s}})$ is sufficiently small. Starting with a test signal, we will be satisfied with matches that twist and turn in ways like the test signal, and which possess glucose values that are close to the test signal's values during most periods of the day. Figure 4 contains a few examples of what we have in mind, with days that are similar to test day 337-36931.

Our measure S will be defined by calculating a "penalty" that comes from comparing the "high" parts of the signatures, and then a separate penalty by comparing the "low" parts. The "high" parts of the signatures can be used to find signals with coincidental twists and turns. The basic idea is that the more the -1 and $+1$ tags match in the two signatures, the more the signals will have similar shape. We will quantify this by calculating penalties when the tags do not match, giving greater weight to the entries in the fifth scale and fourth scale (the two bottom rows of the spectogram). For signatures $\tilde{\mathbf{q}}$ and $\tilde{\mathbf{s}}$, we define $h(\tilde{\mathbf{q}}, \tilde{\mathbf{s}})$ as follows: for each pair of corresponding entries in the "high" parts of the signatures, we assign the following penalties:

	-1	0	1
-1	0	6	12
0	6	3	6
1	12	6	0

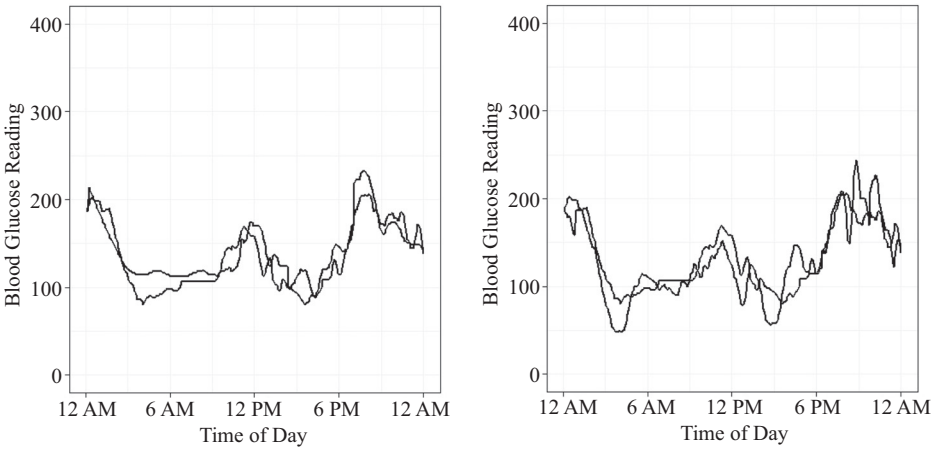

Figure 4 Matched CGM days with 337-36931.

Additionally, for the fourth scale, we double the penalties, and for the the fifth scale, they are quadrupled. Then all penalties are added together to compute h. Because of the uncertainty of the 0 tag, there are penalties whenever it appears, but there is a smaller penalty for a pair of 0's.

If we try using h alone as our similarity measure, matches of a test signal will have comparable ups-and-downs, but can be quite distant from each other vertically. For instance, a CGM signal and a second signal created by adding 100 to the first signal would match perfectly, but this is not what we have in mind for similarity. For this reason, the "low" parts of the signatures must also be taken into account.

A natural approach to compare "low" entries would be to use vector space norms such as the 1-norm, the 2-norm, or the infinity norm that were mentioned above. But each of these three norms are flawed when used to compare signatures. To decide how to use the "low" entries, a discussion of what we will desire to identify as "similar" is necessary.

If two CGM days appeared to follow the same history for most of the 24 hours and diverged for an hour or two, but not in an extreme way, then that would be acceptable as "similar." Consequently, if two signals agreed more or less on seven or eight of their nine low entries, while on the other entries the differences were not extreme, and h for this pair was small, then we would want our similarity measure between these two days to be small. Also, for type-1 diabetes, all differences between CGM signals are not the same. First of all, any reading above 240 is considered a "high," and the target of the type-1 diabetic patient and any caregiver is to maintain blood glucose numbers between 70 and 140 mg/dL. It is then the case that a difference of 10 between 75 and 65 is the difference between "in range" and "low," and this difference is just as, if not more important than, the difference between 65 and 55. In turn, both of these differences are much more important than between 310 and 300.

Consequently, if we think of using these norms to define penalties between CGM days that are "not similar," the use of any one of these norms is problematic. For both the "2-norm" and the "infinity norm," there is a high penalty for isolated corresponding entries that are significantly different. So for two signals that agree for nearly their whole length but differ significantly in one small period of time, the calculation of either of these norms can yield relatively large values. The "1-norm" will take into account all nine differences between the two "low" signatures, but unless we weight the individual differences based on the value of "low" entries, we will either over-penalize or under-penalize certain differences.

Rather than modify the "1-norm," we propose using a combination of these three norms as part of our similarity measure, favoring pairs of CGM days that are assessed

low penalties from all three norms. Of the nine differences between the "low" entries, pairs of signals where the largest of the nine differences are relatively small, and the average of the nine differences are also relatively small, will have a similarity measure which is relatively small. With this in mind, we define the following similarity measure between two different CGM signatures $\tilde{\mathbf{q}}$ and $\tilde{\mathbf{s}}$:

$$S(\tilde{\mathbf{q}}, \tilde{\mathbf{s}}) = 0.01\|\tilde{\mathbf{q}} - \tilde{\mathbf{s}}\|_1 + 0.02\|\tilde{\mathbf{q}} - \tilde{\mathbf{s}}\|_2 + 0.04\|\tilde{\mathbf{q}} - \tilde{\mathbf{s}}\|_\infty + 0.01h(\tilde{\mathbf{q}}, \tilde{\mathbf{s}}), \qquad (4)$$

where the first three norms are calculated on the first nine entries of the signatures, and h applies to the rest of the entries. We also set $S(\tilde{\mathbf{q}}, \tilde{\mathbf{q}}) = 0$, for all $\tilde{\mathbf{q}}$, since $h(\tilde{\mathbf{q}}, \tilde{\mathbf{q}}) \neq 0$. Small coefficients are included in each part of the formula for S to give our measure of similarity a reasonable size when applied to CGM data, and to help with potential future computer calculations (e.g., dealing with a matrix of similarity measures). The coefficients were determined through trial-and-error by visually studying the matches that were identified for selected test signals.

We now have the pieces needed to create a system for matching CGM signals: First, there is a one-time calculation of signatures for all signals in the database. Once this investment in calculation is complete, it is then relatively quick to find matches for our test signal by using the signatures and S. It is this speed of matching that is the rationale for our proposed method. It would be simpler, but significantly longer in time, to just compare all the original signals using one of the three norms above, or some combination of those norms.

Figure 5 shows a second set of examples of matching a test day, in this case day 59-36688 in the Jaeb database. Similar days are identified by values of S less than 10.

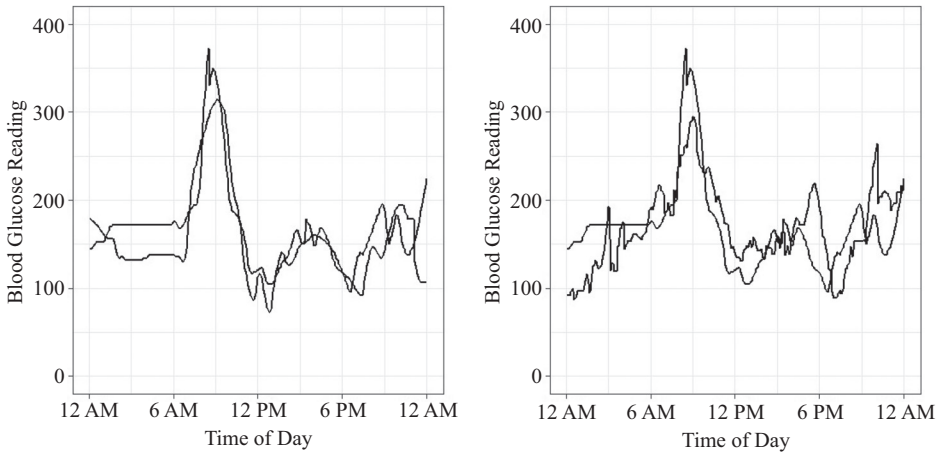

Figure 5 Matched CGM days with 59-36688.

Further discussion

How might this method of matching CGM days be used in practice with patients, caregivers, and doctors? Currently a patient can use CGM data to fine-tune short-term blood glucose management: determining at any time if extra insulin is needed to "correct a high," or if a "fix" of extra carbohydrates is required for low blood sugar. For long-term management, a blood test known as A1C* is done every three months, providing a measure of average blood glucose over the past six months. However, the A1C test doesn't capture issues of range and volatility of CGM readings.

*glycated hemoglobin is known as HbA1C, or just A1C.

One idea is to use the similarity measure S to compare the days of a patient with "exemplary" days—near-perfect glucose days where all CGM readings are between 70 and 140 and there is very little change from one reading to the next. With such a comparison, a patient and treatment team can analyze how close the patient's days are to these "exemplary" days, and this may shed light on issues of range and volatility. The more that is understood about a patient's blood glucose dynamics, the better that variables such as medication, food, and activity can be adjusted [**4, 10**].

Specifically, it might be enlightening to find days with large S values compared to "exemplary" days. An analysis like this can be revealing for some patients, such as patient 484 in the Jaeb database. For this patient, the reported A1C in the Jaeb data files was 6.6%, a very good reading, and it corresponds to an average blood glucose of around 150. However, a study of the patient's CGM data using our similarity measure S shows fewer exemplary days than would be expected. Graphs of individual days, such as 484-37036 and 484-37171 (see Figure 6), suggest that for this patient, there are many highs and lows of short duration that an A1C score would not detect. With this understanding, treatment plans could be adjusted to address these issues of volatility and range.

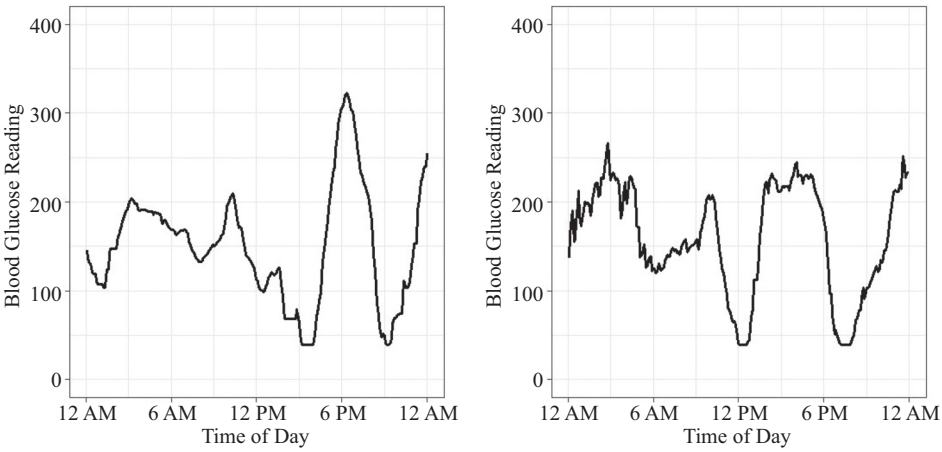

Figure 6 Patient 484: Two days with significant highs and lows, but a good average.

Returning to the *Shazam* app, the methods used to match an audio clip to a song in its database are similar to what has been described in this paper, but more complicated. For instance, Fourier analysis is used instead of wavelet filters, which leads to the creation of *spectograms* rather than scalograms. The signature/similarity measured calculations also involve finding peak values in the spectrogram, but in a significantly different way. Also, modifications are needed to take into account that only a short segment of a song is the test signal, rather than the whole song. Details can be found in [**15**]. The scalograms that we created are based on standard wavelet scalograms [**2**] and the use of wavelet scalograms with audio signals is demonstrated in [**3**]. That paper also incorporates *quantization*, which is a variant on the use of tags that is described in this paper.

Finally, the *Shazam* algorithm and the methods in this paper are part of the field of data science, which has seen massive growth in the past few years. Advances have been fueled by the growing availability of large data sets as sensors and storage devices have gotten smaller and cheaper, and more legacy data is made freely available on the Internet. New innovations are anticipated over the next 10–20 years [**7**], and there are many parts of upper-level collegiate mathematics that can open doors to this discipline.

Appendix: Preprocessing the CGM data

To create our set of CGM signals to search through, we only used the days in the Jaeb database which had at least 209 out of 288 possible CGM readings. This led to 42,799 days coming from 346 patients. For the days that were used, when CGM readings were missing, the previous known reading was used as a reasonable replacement. So the database that we used consisted of 42,799 days, each with 288 CGM positive integer readings.

Acknowledgment The author would like to thank Walter Stromquist, former editor of *Mathematics Magazine*, for his advice to revise early drafts of this paper. The author is also indebted to anonymous reviewers for careful and, at times, challenging critiques of earlier drafts of the manuscript.

REFERENCES

1. E. Aboufadel, S. Schlicker, *Discovering Wavelets*. Wiley, New York, 1999.
2. P. Addison, *The Illustrated Wavelet Transform Handbook*. Institute of Physics Publishing, Philadelphia, PA 2002, doi:10.1201/9781420033397.
3. S. Baluja, M. Covell, Content fingerprinting using wavelets. *3rd European Conference on Visual Media Production CVMP 2006*, 198–207, doi:10.1049/cp:20061964.
4. R. Beaser, *Joslin's Diabetes Deskbook: A Guide for Primary Care Providers.* Second Ed. Joslin Diabetes Center, Boston, MA, 2010.
5. F. Caimi, D. Kocak, F. Dalgleish, J. Watson, Underwater imaging and optics: Recent advances, *Oceans 2008* (2008) 1–9, doi:10.1109/OCEANS.2008.5152118.
6. B. Chen, B.-X. Fang, X.-C. Yun, Wavelet analysis based worm attack early detection, *Int. J. Comp. Sci. Eng. Sys.* **1** no. 3 (2007) 175–186.
7. A. Gandomi, M. Haider, Beyond the hype: Big data concepts, methods, and analytics, *Int. J. Inform. Manag.* **35** no. 2 (2015) 137–144, doi:10.1016/j.ijinfomgt.2014.10.007.
8. S. Haddad, W. Serdijn, *Ultra Low-Power Biomedical Signal Processing: An Analog Wavelet Filter Approach for Pacemakers*. Springer, New York, 2009, doi:10.1007/978-1-4020-9073-8.
9. JDRF CGM Study Group, JDRF randomized clinical trial to assess the efficacy of real-time continuous glucose monitoring in the management of type 1 diabetes: Research design and methods, *Diabetes Technol. Therapeutics* **10** no. 4. (2008) 310–321.
10. JDRF Research Team, JDRF Experts Weigh in on Better Metrics for Blood-Glucose Control, http://jdrf.org/blog/2013/jdrf-experts-weigh-in-on-better/.
11. W. Lawton, Applications of complex valued wavelet transforms to subband decomposition, *Signal Proc.* **41** no. 12 (1993) 3566–3568.
12. J. Medeiros, How Shazam pivoted its business to help US television broadcasters, *Wired UK* **19** no. 10 (2011).
13. J. Trygg, S. Wold, PLS regression on wavelet compressed NIR spectra, *Chemometrics Intell. Lab. Syst.* **42** no. 1–2 (1998) 209–220, doi:10.1016/S0169-7439(98)00013-6.
14. P. VanFleet, *Discrete Wavelet Transforms: An Elementary Approach with Applications*. Wiley-Interscience, New York, 2008.
15. A. Li-Chun Wang, An industrial-strength audio search algorithm, in *Proceedings of the International Conference on Music Information Retrieval ISMIR* (2000) 7–13, doi:10.1109/IITAW.2009.110.

Summary. With the *Shazam* smartphone app, a listener captures a short excerpt of a recorded song with the smartphone's microphone, and in a matter of moments the app reports the name of the song and the artist. Fourier analysis is a key mathematical tool that powers the app. In this paper, we describe a wavelet-based method that captures the basic process used by the Shazam app to search a database of number sequences (signals) to find those that are similar to a test signal. We will describe our implementation with a different source of signals: continuous glucose monitor data from the management of type-1 diabetes.

EDWARD ABOUFADEL (MR Author ID: 613192) is a Professor of Mathematics at Grand Valley State University, and an Assistant Vice President for Academic Affairs. He is an applied mathematician who has worked with students and other collaborators on projects such as detecting potholes with smartphones, using wavelets to read CAPTCHAs, and designing objects for 3D printing.