

On the Reliability of Quake-Catcher Network Earthquake Detections

by Battalgazi Yildirim, Elizabeth S. Cochran, Angela Chung, Carl M. Christensen, and Jesse F. Lawrence

INTRODUCTION

Over the past two decades, there have been several initiatives to create volunteer-based seismic networks. The Personal Seismic Network, proposed around 1990, used a short-period seismograph to record earthquake waveforms using existing phone lines (Cranswick and Banfill, 1990; Cranswick *et al.*, 1993). NetQuakes (Luetgert *et al.*, 2010) deploys triaxial Micro-Electromechanical Systems (MEMS) sensors in private homes, businesses, and public buildings where there is an Internet connection. Other seismic networks using a dense array of low-cost MEMS sensors are the Community Seismic Network (Clayton *et al.*, 2012; Kohler *et al.*, 2013) and the Home Seismometer Network (Horiuchi *et al.*, 2009). One main advantage of combining low-cost MEMS sensors and existing Internet connection in public and private buildings over the traditional networks is the reduction in installation and maintenance costs (Koide *et al.*, 2006). In doing so, it is possible to create a dense seismic network for a fraction of the cost of traditional seismic networks (D'Alessandro and D'Anna, 2013; D'Alessandro, 2014; D'Alessandro *et al.*, 2014).

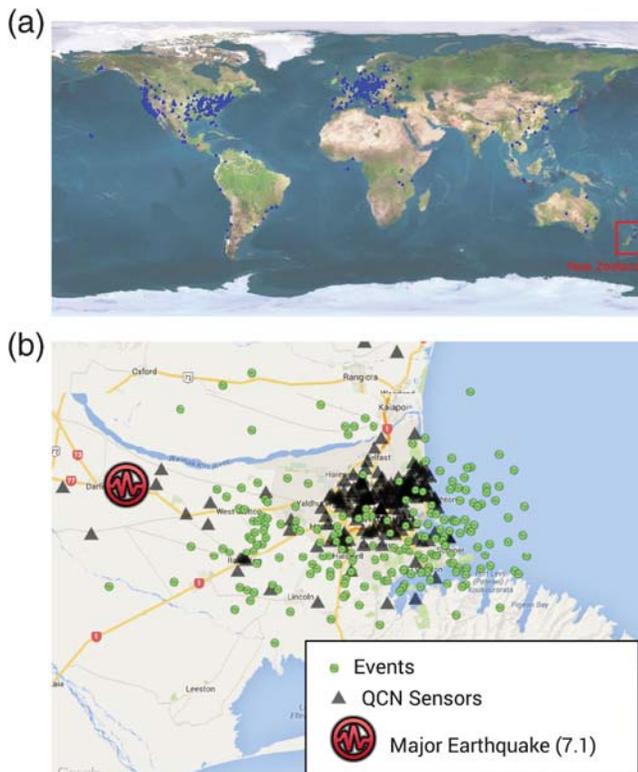
A rapidly deployable and highly mobile seismic network can collect enormous volumes of data at high spatial density during an aftershock sequence following major earthquakes (Naito *et al.*, 2013). Although the low-cost seismic networks described above were primarily designed to detect and characterize earthquakes, the networks have also been used for other purposes such as to monitor building health in Kohler *et al.* (2013). These types of low-cost networks may also have other potential applications such as detecting landslides (Azzam *et al.*, 2011) and locating explosions (Taylor *et al.*, 2011).

The Quake-Catcher Network (QCN) is another variant of a cyber-social seismic network, which has been operating since 2008. Cochran, Lawrence, Christensen, and Chung (2009) and Cochran, Lawrence, Christensen, and Jakka (2009) describe the implementation that uses a client software phase-picking algorithm that is based on a ratio of short-term average and long-term average of the signal (Earle and Shearer, 1994), and based on the communication between the sensor data and the server that is provided by Berkeley Open Infrastructure for Network Computing (BOINC) (Anderson and Kubiawicz, 2002; Anderson, 2004; Christensen *et al.*, 2005; Anderson *et al.*, 2006).

QCN has thousands of sensors deployed around the world (see Fig. 1a) that are installed and hosted by volunteers. In addition, several targeted rapid aftershock deployments have been initiated by QCN including the 2010 M_w 8.8 Maule, Chile, earthquake (Chung *et al.*, 2011) and the M_w 7.1 Darfield, New Zealand, earthquake (Cochran *et al.*, 2012; Lawrence *et al.*, 2014). Following the 3 September 2010, M_w 7.1 Darfield earthquake, the QCN real-time detection system was quickly implemented and began to detect earthquakes beginning on 25 September 2010 (Lawrence *et al.*, 2014). One hundred and ninety-two QCN sensors were installed in and around Christchurch, New Zealand, to detect aftershocks (see Fig. 1b). The real-time system was initiated to rapidly detect, locate, and characterize events recorded by QCN sensors. The rapid earthquake detection system has been improved incrementally since then. Because the system was only designed to work in real time, the improvements in the code could only be tested by events that occurred after the changes had been implemented.

The QCN detection algorithm associates a given trigger to all other triggers occurring within 90 s and 200 km of the original trigger. The algorithm considers a trigger as a probable event if more than five triggers are correlated. The code then performs a grid search for the event origin time and location. Given the event time and location, we calculate the travel-time misfit for each station and the average travel-time misfit for the array (see Fig. 2). If the average misfit is less than 2.0 s, the code considers the event to be a likely earthquake.

Here, we describe results using a recently developed feature of the QCN system that supports retrospective runs (offline) of past data. The ability to simulate past events allows us to test and verify event detection algorithms and supports more rapid algorithm development. The QCN system has archived sensor data since 2009; therefore, we can potentially identify new earthquakes recorded by the network prior to September 2010, when the real-time detection algorithms were implemented. The objective of this study is to describe the retrospective simulator, and use the simulator to optimize the set of parameters used to identify events by validating the results against published earthquake catalogs. The goal of this study is to determine (1) whether a network of low-resolution sensors operating in noisy environments can provide useful earthquake detections, (2) the parameters that have the largest impact on detections, and (3) the optimal set of detection



▲ **Figure 1.** (a) The triggers received from the Quake-Catcher Network (QCN) sensors (blue triangles) are given around the world. (b) The QCN sensor distribution and the QCN true-positive events (the locations are GNS catalog locations) around Christchurch and New Zealand following the Darfield earthquake (3 September 2010).

parameters for the array operating near Christchurch, New Zealand.

The QCN retrospective simulator uses simulated time steps (ideally every second) to emulate the real-time system. Each time step simulates a real-time single second in an accelerated way (approximately 1000 times faster). The simulator loads the triggers from the 200 s prior to the simulated time. The earthquake detection software is then modified, and we test the new parameters by simulating past sensor data. The detection algorithm used here is designed for very rapid event detection and characterization of earthquakes for rapid event awareness and/or earthquake early warning; thus, the method is different from that employed for routine event detection by most seismic networks. The optimal parameters we determine here are tuned for lower resolution sensors deployed in noisy environments, so some tuning of the detection parameters may be needed if more traditional seismic data are used.

We first present improvements to the software and describe the retrospective (offline) simulator in the [Simulation Algorithm and Improvements in the QCN Software](#) section. In the [Verification of the New Code](#) section, we apply the simulator to past data recorded between 25 September 2010 to 1 August 2013 to verify the offline code, including improve-

ments to the code. In the [Validation with NEIC and GNS Catalogs](#) section, we compare the origin time and locations of events from the real-time system and the offline system to GNS Science's GeoNet ([Geonet, 2010–2013](#)) and National Earthquake Information Center ([NEIC, 2010–2013](#)) catalogs. [Retrospective Runs](#) section presents new earthquakes recovered by simulating data before 25 September 2010, prior to the implementation of the real-time event detection. In [Parameter Studies](#) section, we rerun the offline code using the optimal parameter values to assess the detection times and accuracy of event times and locations. We summarize the findings in the [Summary](#) section.

SIMULATION ALGORITHM AND IMPROVEMENTS IN THE QCN SOFTWARE

The QCN software is made up of two parts: (1) the client-side program and (2) the server-side program. A client-side program runs on volunteer computers, detecting strong new motions and sending digital trigger data to the QCN server. The server-side program monitors the incoming trigger signals for possible events. In this study, we improve the server-side software to maximize the robustness, efficiency, and maintainability of the code. Specifically, the code has been rewritten in C++ (prior version utilized legacy C code), several bug fixes are applied, and we simplify several conditional statements. Throughout the article, the code reflecting these changes is identified as version 2.0, whereas version 1.x (or online) denotes the operational server code in the real-time system.

The updated software detects additional earthquakes that version 1.x missed. In this study, all simulations obtained by version 2.0 were executed in offline mode whereas version 1.x results have been taken from the real-time system (online mode). We use offline code and version 2.0 synonymously in this article.

The pseudocode of the simulation algorithm is presented in Figure 2, and we describe the algorithm below in general terms. For a given simulation start and end time, each loop simulates a time advance of one second. Inside the loop (lines 6–24 in Fig. 2), the code queries the trigger memory table for the previous 200 s (same as the real-time system) of simulated real time. Looping over every single pair of triggers, the algorithm finds correlated triggers (lines 9–12) by checking three conditions: (1) the sensor pair is separated by less than the maximum correlation distance (D_{\max}), (2) the time difference of the pair is within the maximum correlation time (T_{\max}), and (3) the time difference between the two triggers is smaller than the propagation time of an S wave (plus 3 s) for the straight-line distance between the two stations. The default values of D_{\max} and T_{\max} are 200 km and 90 s (see Table 1 for the other default simulation parameters). When the total count of correlated triggers is greater than or equal to the minimum correlation count (CNT_{\min}) (set at the default value, five), a probable earthquake is declared. To locate the event (line 16), the code undertakes a recursive 3D grid search that minimizes the misfit between the predicted and actual wave arrivals. After

```

input : The SQL database storing the triggers and the start and end times of simulation (YYYY-MM-DD HH:MM:SS)

output: Simulating real-time QCN system behavior

1 Read crustal velocity from Crust2.0 model
2 Set user specified parameters:  $D_{max}$  (maximum correlation distance),  $T_{max}$  (maximum correlation time),  $CNT_{min}$  (minimum
   correlation count),  $MISFIT_{max}$  (Maximum average travel time misfit)
3 Convert the input times to UTC times ( $t_{start}$ ,  $t_{end}$ )
4 Connect the SQL database storing the past triggers of the QCN sensors
5 Initialize the simulated real time ( $t_{now}$ ) as  $t_{start}$ 
6 do
7     Query database and load it into memory every an half hour
8     Select the triggers occurred in the last 200 seconds of the simulate real time into the memory
9     Loop over the set of the triggers to find correlated triggers:
10         /*  $d_{ij}, t_{ij}$ : a distance and time difference between  $i_{th}$  and  $j_{th}$  triggers. */
11         if  $d_{ij} < D_{max}$  and  $t_{ij} < T_{max}$  and  $ABS(t_{ij}) < d_{ij}/V_s + 3$  then
12             Correlate  $i^{th}$  and  $j^{th}$  triggers
13         end
14         /*  $v$ : a vector storage of triggers,  $N$ : total number of triggers */
15         for  $i=0$  to  $N$  do
16             Get the total count of correlated triggers (cnt) for  $v_i$ 
17             if  $cnt \geq CNT_{min}$  then
18                 Locate an event (e) minimizing the arrival times using Crust2 velocity model and observed times of the
19                 triggers that correlates to  $v_i$ 
20                 Compute the correlation coefficient ( $r^2$ ) using the differences  $dt_i^{obs} = t_i^{obs} - e_t$  and  $dt_i^{est} = t_i^{est} - e_t$ 
21                 if  $\bar{d} = \frac{1}{cnt} \sum_{n=0}^{cnt} ABS(dt_n^{est} - dt_n^{obs}) \leq MISFIT_{max}$  and  $(r^2) > 0.5$  then
22                     Issue an earthquake for this event and update the quake table
23                 end
24             end
25         end
26     March the simulated real time 1 second to advance ( $t_{now} + = 1$ )
27 while  $t_{now} < t_{end}$ ;

```

▲ **Figure 2.** The offline QCN algorithm using the past QCN data for simulating the online QCN system.

Parameter Values	Default Parameters	Optimal Parameters
Minimum Correlation Count (CNT _{min})	5	5
Maximum Correlation Distance (D _{max})	200 km	100 km
Maximum Correlation Time (T _{max})	90 s	30 s
Maximum average travel-time misfit (MISFIT _{max})	2 s	1 s

finding the event location and origin time, we measure the observed time difference (d_i^{obs} , the recorded time difference between sensor i and j), the estimated time difference (d_i^{est} , the computed time difference using the Crust_2.0 velocity model, [Laske et al., 2012](#), between sensor i and sensor j), and the correlation coefficient r^2 (see line 17 in Fig. 2). The correlation coefficient is computed for x and y data sets as follows:

$$r^2 = \frac{(\sum xy - n\bar{x}\bar{y})^2}{(\sum x^2 - n\bar{x}^2)(\sum y^2 - n\bar{y}^2)}, \quad (1)$$

in which n represents the total number of elements in x and y datasets, and \bar{x} and \bar{y} are the averaged values from the data sets. To check, we compute the average travel-time misfit (line 18), first summing the differences and then averaging them ($\bar{d} = \frac{1}{N} \sum_{i=1}^N d_i^{\text{obs}} - d_i^{\text{est}}$, in which d_i^{obs} is the observed arrival time at a station i , and d_i^{est} is the estimated arrival at station i based on the event location). The code flags an event as a likely earthquake if the maximum average travel-time misfit (MISFIT_{max}) is less than or equal to some threshold value (set in version 1.x to be 2 s) and the correlation, r^2 , is greater than 0.5. The code increments the simulated time by 1 s (line 23), and repeats all the above steps until the end of simulation (line 24). The offline algorithm reconstructs the timing of trigger arrivals from archived data (lines 3–4); otherwise, the code is the same for both online and offline versions.

VERIFICATION OF THE NEW CODE

We first verify the new code by comparing results with version 1.x. Although the sensor data have been flowing into the QCN server since 2008, we are limited to the period after 25 September 2010, when the real-time detection algorithm was first implemented. The last date included in the simulations described herein is 1 August 2013. The server-side code has undergone many changes since its inception. We label the past code as version 1.x. Over the years, version 1.x has evolved, and versioning of the code was not implemented because modifications were made. For this reason, it is hard to identify which parameters were implemented in the code for a specific event. The version 1.x code should be envisaged as the past code at a

time of a specific event. Our assumption is that performance of the system improved through time since the real-time system was initiated in September 2010, but there is no way to verify this assumption.

The two codes perform quite similarly, yielding similar but not identical results. We present (1) detection time, (2) event time, (3) horizontal location, and (4) depth differences for matched events between the online system and the simulation of the version 2.0 code. The simulation of 3 years' worth of data takes about 12 h on a Intel Xeon (1.66 GHz) machine. We define detection time as the total time between the event origin time and the server-side detection time, including the travel times of waves to the sensors, the delays (network latency, on average about 2–3 s) that occurred in the transmission, and the computation time (on average, less than a second) in the QCN server. We compare the difference in the detection times for the two codes in Figure 3a. The detection times show that version 2.0 performs better on average. Based on the simulation results by version 2.0, the average detection time for all the events is about 10 s; and 95% of all of the events are detected in less than 14 s. It is difficult to pinpoint the cause of the large detection time differences between the two codes because the exact parameters used by version 1.x of the code for particular events are unknown.

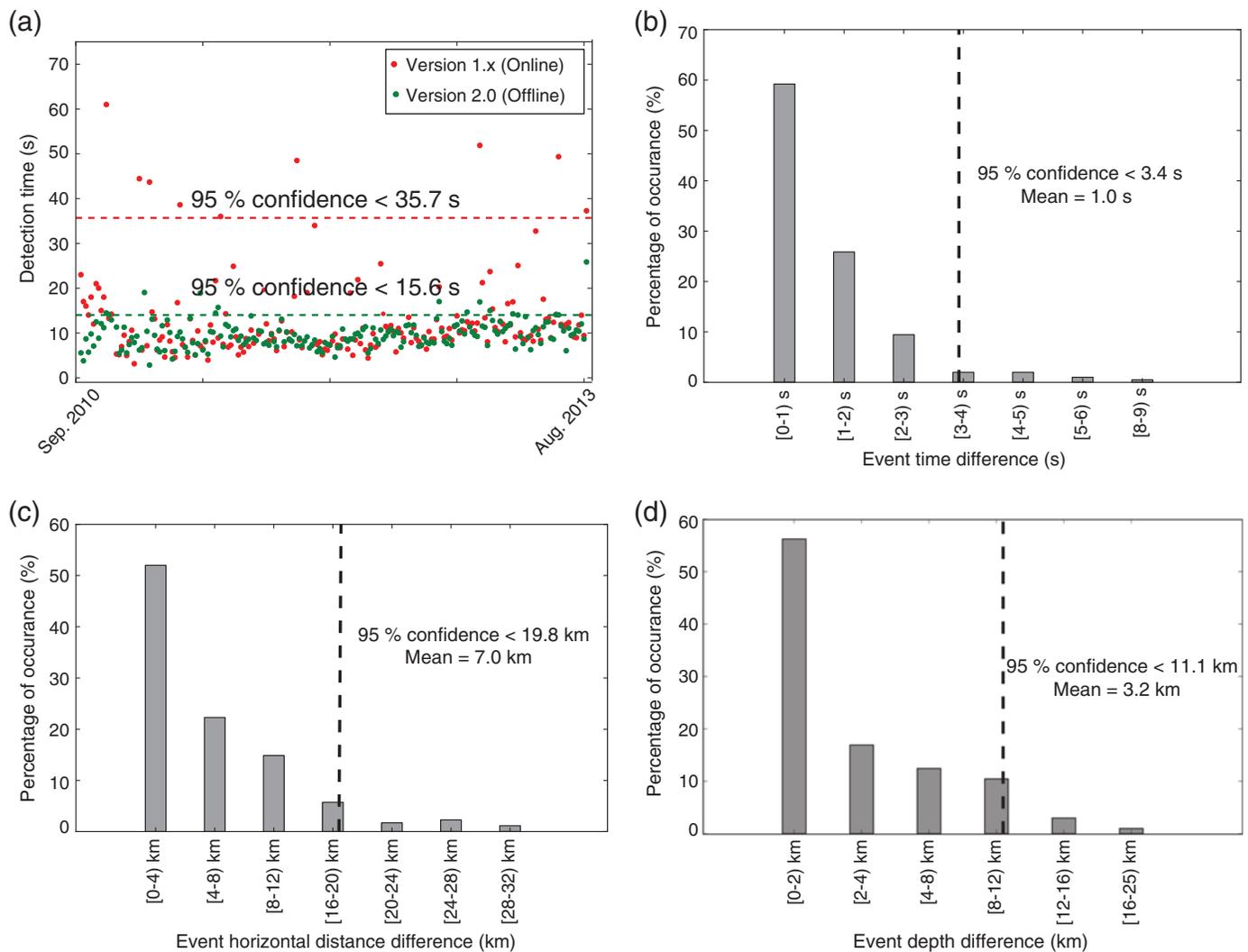
We show the absolute difference between event origin time from both codes in Figure 3b; the origin times of about 60% of events are within 1 s. Only about 7% of events show origin times that differ by more than 3 s. The difference in the horizontal distance (e.g., epicentral distance) and depth follows similar trends such that for the majority of events the two codes are in close agreement with one another (see Fig. 3c,d).

Code-to-code comparison of the above quantities verifies that the version 2.0 of the code detects earthquakes faster, and differences between event origin times, horizontal location, and depth are minimal. We attribute some of the difference in event information to the use of the Crust_2.0 velocity model ([Bassin et al., 2000](#); [Laske et al., 2012](#)) used in version 2.0, compared with the simple 1D velocity used in Version 1.x.

VALIDATION WITH NEIC AND GNS CATALOGS

Next, using NEIC ([NEIC, 2010–2013](#)) and GNS Science catalogs ([Geonet, 2010–2013](#)), we investigate whether version 2.0 of the code improves the rapidly determined origin time and location of the events. The majority of earthquakes recorded by the QCN system occurred in New Zealand following the Darfield earthquake. Most other recorded earthquakes are occurred in Chile and the United States. Because the majority of recorded events occurred in New Zealand, following the 2010 M_w 7.2 Darfield, New Zealand, earthquake, we include the GNS catalog in addition to the NEIC catalog. When we find an event in both catalogs, the reference event is taken from the GNS catalog because this catalog uses additional local data to locate events.

We consider four event characteristics in this section: the origin time, the epicenter, the depth, and the magnitude.



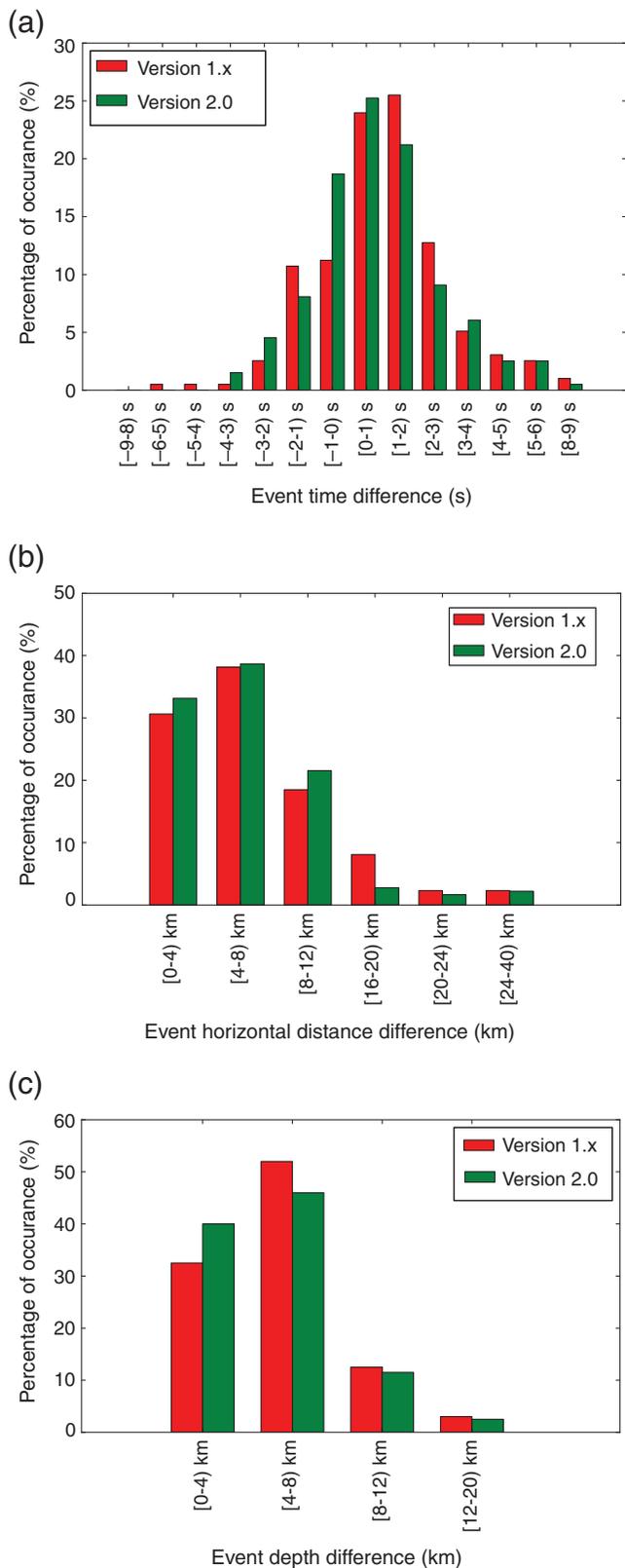
▲ **Figure 3.** Distribution of (a) difference in detection time, (b) difference in event origin time, (c) difference in horizontal (epicentral) distances, and (d) difference in depths are plotted for the matched events between the version 1.x and version 2.0. Comparison is for events that occurred between 25 September 2010, when the online QCN real-time system detected its first earthquake, to 1 August 2013. The dashed line represents 95% confidence level.

Figure 4a shows the event origin time difference between the QCN system and the available catalog (either GNS or NEIC) for both version 1.x and version 2.0 detections. The event origin time differences are clustered around a peak at about 1 s difference. Both codes perform well and estimate the origin times of a majority of events within 3 s of the catalog-reported origin time. We continue the catalog comparison and examine differences in horizontal (epicentral) distance between the matched events. According to Figure 4b, the distribution of differences between horizontal distance difference for version 1.x and 2.0 is small; however, version 2.0 performs slightly better, detecting locations closer to the catalog locations for a slightly higher percentage of events. Depth is the most difficult value to determine in an event. Both codes deviate similarly from the catalogs in estimating event depths. Most detections (about 90% of them) are within 8 km depth and 12 km epicentral distance to the corresponding catalog event (Fig. 4b,c).

To compute the magnitude of events, the QCN software uses peak ground acceleration (PGA) values at the trigger time and 1, 2, and 4 s after the trigger (see Fig. 5). Typically, at the trigger time, the corresponding PGA value is so small that without using subsequent values at 1, 2, and 4 s, the magnitude relation underestimates the magnitude of the event. Until March 2013, we only saved the PGA values at the trigger time but not the PGA values at 1, 2, and 4 s after the trigger. Because of the missing PGA values in the database, we do not compare magnitudes at this time.

RETROSPECTIVE RUNS

Version 2.0 of the code is also applied to archived data from 1 September 2009 to 1 August 2013. The number of total events, true-positive events (confirmed events from GNS or NEIC catalogs), and false-positive events are presented in Table 2.

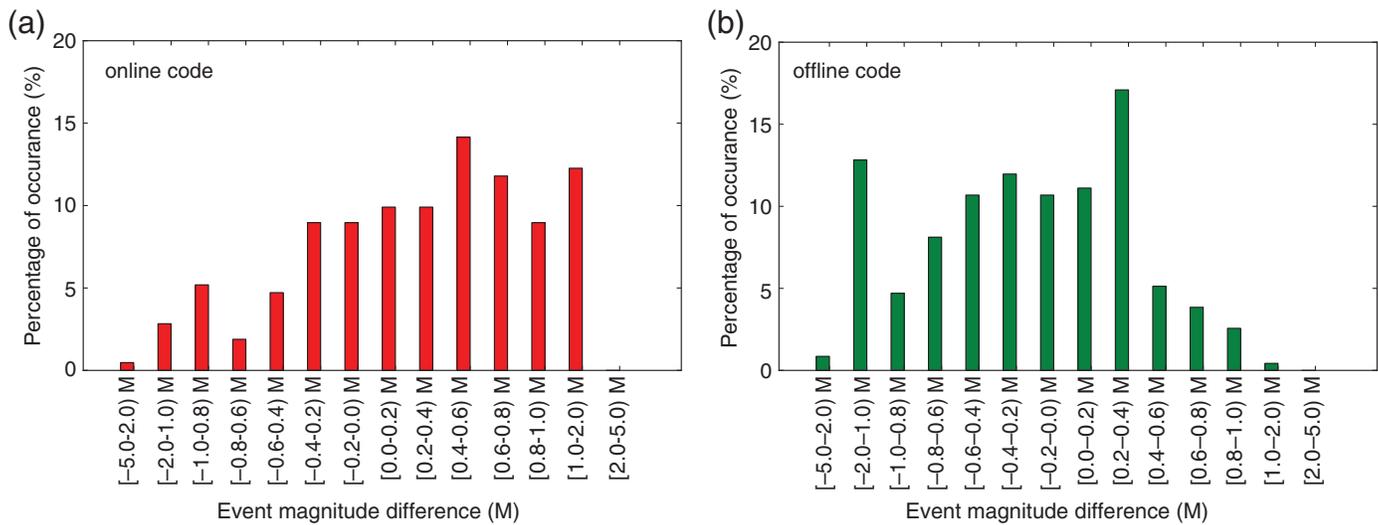


▲ **Figure 4.** Distribution of (a) difference in event origin times, (b) difference in horizontal (epicentral) distance, and (c) difference in depths between the NEIC or GNS catalog and version 1.x and version 2.0 are plotted for matching events that occurred between 25 September 2010 and 1 August 2013.

We show the results for three different date intervals: (1) 1 September 2009 to 8 September 2010, prior to the installation of sensors in New Zealand, (2) 9–24 September 2010, when online detection was not implemented but over 100 sensors recorded ground motions from the Darfield, New Zealand, aftershock sequence, and (3) 25 September 2010 to 1 August 2013, the time period that version 1.x was live. During the period from 1 September 2009 to 8 September 2010, version 2.0 recovers 94 events, 11 of which are confirmed in the NEIC catalog, whereas 83 of them were found to be false-positive events (we checked the waveform manually to confirm this). During this time period, QCN was using lower-resolution sensors (10–12 bit) (please refer to Table 3 for more information about the USB sensors used in this study). Version 2.0 of the code detects many false-positive events because the high noise levels of the old sensors causes a large number of false triggers. During the period from 9 to 24 September 2010, we recovered 42 events, confirmed with the GNS catalog. Forty of the recovered events are confirmed in the GNS catalog, and only two are false events. From 25 September 2010 to 1 August 2013, version 2.0 detects 228 true-positive events (17 false-positive events), with 16 additional events confirmed during this period. Between 1 September 2009 and 1 August 2013, the version 2.0 code recovers a total of 67 new confirmed events in the combined NEIC and GNS catalogs. Although version 2.0 of the code finds many additional earthquakes for the same time period, version 1.x detects fewer false-positive events (see Table 2).

The majority of the events detected by QCN were aftershocks of the Darfield, New Zealand, earthquake. Through comparison with the GNS catalog, we determine the percentages of events detected and missed. We limit the GNS catalog search to events within 75 km of the epicenter of the Darfield, New Zealand, earthquake; QCN sensor network was clustered at epicentral distances between 20 and 40 km northeast of the mainshock epicenter. In addition, we consider only events with magnitudes greater than 3; QCN sensors typically are not sensitive enough to detect lower-magnitude events (Evans *et al.*, 2014). The events are sorted into four magnitude bins: 3–4, 4–5, 5–6, and 6+. To match true-positive events, we take the reference magnitude from the GNS catalog, but for false-positive events (not matching any catalog), we use the QCN-determined magnitudes. Figure 6a presents the distribution of true-positive events from this simulation. The QCN offline code matches 96 of 2743 GNS events (3.5%) varying in magnitude range from 3 to 4. In the 4–5 magnitude bin, QCN detects 121 events compared with 303 events reported by GNS (a 40% success rate). QCN misses only three earthquakes out of 24 events reported in the GNS catalog with magnitudes between 5 and 6. Only three events with magnitude over 6 occurred, and all were detected (100%).

Figure 6b shows a histogram of false-positive events by magnitude from the offline code. Most of the false-positive events occur in M_w 4–5 range. The magnitude relation that Lawrence *et al.* (2014) used had an estimation bias centered around M_w 4 (Chung *et al.*, 2015). We attribute the fact that



▲ **Figure 5.** Distribution of difference in event magnitude between (a) version 1.x and (b) version 2.0 of the code and the NEIC or GNS catalog are plotted for matching events that occurred between 25 September 2010 and 1 August 2013.

most false-positive events are in the M_w 4–5 range to the bias in the magnitude formulation.

QCN detects a higher percentage of events within 35 km of the epicenter of the Darfield earthquake likely due to the large number of stations in the urban area of Christchurch which is located between 20 and 40 km from the mainshock. Looking at Figure 6c, we see 100% success for events with magnitudes greater than 5.0 that occur within 35 km of the mainshock. For events with magnitudes between 4 and 5, the success rate is around 50% for epicentral distances between 15 and 35 km, and then it decreases slowly as the epicentral distance increases. We see a relatively low percentage (<10%) of true-positive events with magnitudes between 3 and 4 across all epicentral distances.

PARAMETER STUDIES

There are several parameters that play an important role in controlling the accuracy of the algorithms. We identified four

important parameters: minimum correlation count (CNT_{min}), maximum correlation distance (D_{max}), maximum correlation time (T_{max}), and maximum average travel-time misfit ($MISFIT_{max}$). By operating over four years and storing thousands of waveforms, the QCN archive provides a comprehensive suite of data for testing. Being able to reprocess four years of sensor data in a relatively short amount of time (a single run takes only about 12 hours) allows us to investigate how these parameters influence the QCN system detections.

Minimum Correlation Count

To detect an event with the QCN algorithm (see Fig. 2), we first look for correlations among the triggers. Minimum correlation count (CNT_{min}) sets the threshold for the system to declare an event. During initial testing, we observe that setting the CNT_{min} to less than five yields many false-positive events, whereas increasing CNT_{min} reduces the number of true detections.

Number of Events	Ver1.x (Online)		Ver1.x (Online)		Ver2.0 (Offline)		Ver2.0 (Offline)	
	Ver1.x (Online)	True Positive	False Positive	Ver2.0 (Offline)	True Positive	False Positive		
1 September 2009 to 8 September 2010	N/A	N/A	N/A	94	11	83		
9 September 2010 to 24 September 2010	N/A	N/A	N/A	42	40	2		
25 September 2010 to 1 August 2013	217	212	5	245	228	17		

Events found in NEIC and GNS catalogs are defined as true positive. Events defined as false positive were not validated by NEIC or GNS catalogs.

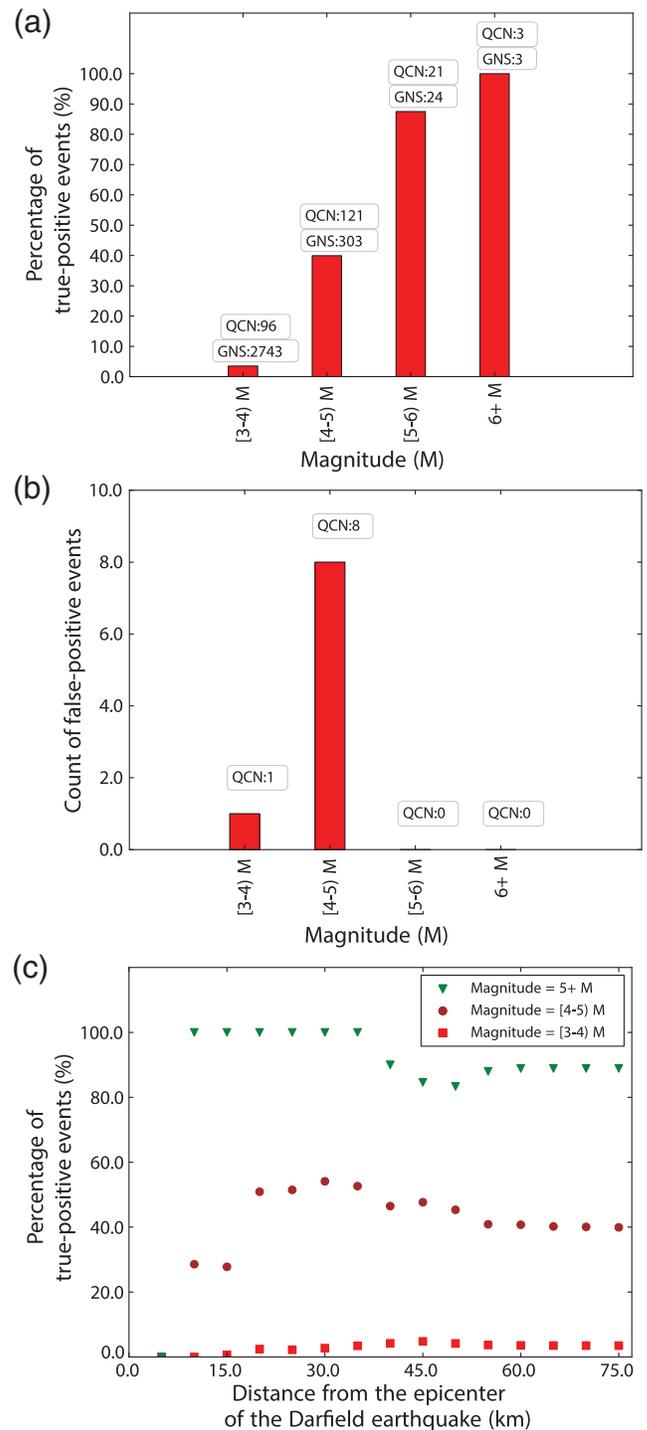
Model	Dynamic Range (g)	Resolution (g)	Frequency Range (Hz)
MotionNode Accel	± 2	$\pm 1.0 \times 10^{-3}$	0.05–25
JoyWarrior	± 2	$\pm 4.0 \times 10^{-3}$	0.05–25
O-NAVI 60mG	± 2	$\pm 6.0 \times 10^{-5}$	0.05–25

We simulate the offline system with four values of CNT_{\min} : 4, 5 (default), 6, and 7. The resulting count of true-positive and false-positive events are plotted in Figure 7a. The counts of both true-positive and false-positive events decrease as CNT_{\min} increases. Ideally, the system should retain as many true-positive events as possible while minimizing the number of false-positive events. We note that whereas a CNT_{\min} of four detects the highest number of true-positive events (a little over 300), it also generates the highest number of false-positive events (175), yielding about 64% detection efficiency with one-third of the events as false positives (see Fig. 7b). As a complementary figure, we present the percentage of the events classified as true positives or false positives in Figure 7b. Increasing CNT_{\min} to five, we see that the reliability rises to about 93%. As we expected, further increases in CNT_{\min} results in higher reliability, although the detection efficiency decreases. For example, when CNT_{\min} is increased to seven the reliability is 99.1%, but yields the fewest number of true-positive events (about 225).

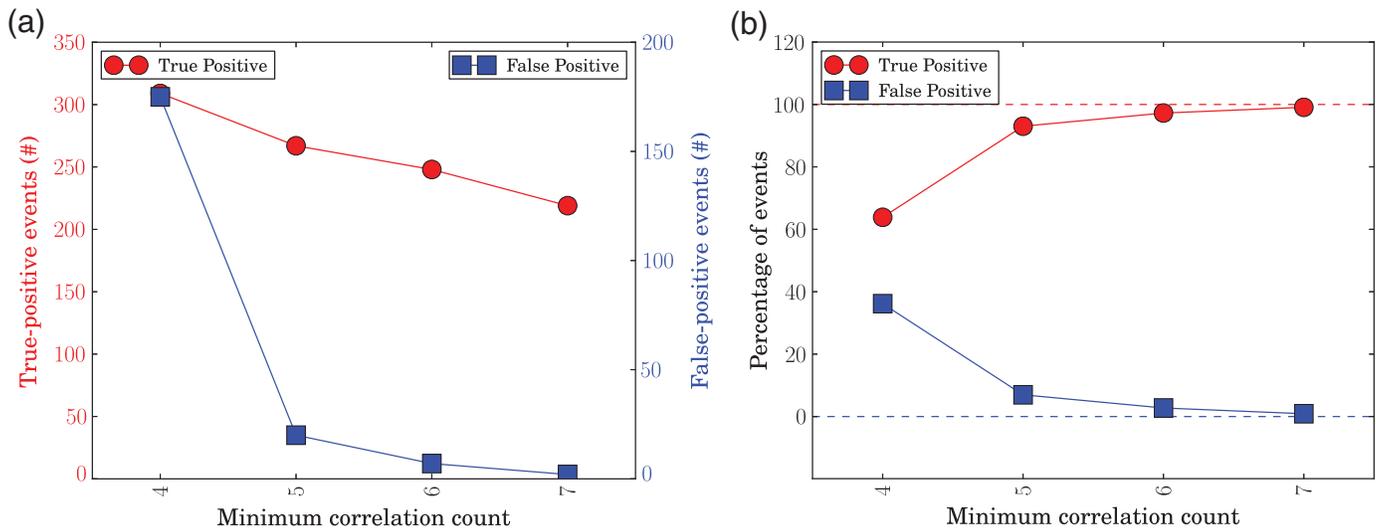
Applying a higher threshold value for CNT_{\min} increases the time detection, adding several seconds to detection times which is not ideal for an early warning system. This study suggests a CNT_{\min} of five is the optimal choice for the current state of the QCN sensor distribution. Here, we define the optimal choice as one that maximizes the number of true-positive detections while minimizing the number of false-positive detections. We note that as the QCN sensor density increases, a higher minimum correlation number might be preferred.

Maximum Correlation Time and Distance

Using a preliminary check for triggers occurring within a specific time window and distance range (such as 90 s time window and 200 km), the algorithm eliminates obviously uncorrelated triggers. The correlation time window and the region are controlled by the maximum correlation time (T_{\max}) and distance (D_{\max}) parameters. The default values of these parameters are set to 90 s and 200 km in the online code. The QCN sensors operate in 92 countries, and almost continuously receives the sensor triggers from various locations around the world. Increasing the T_{\max} and D_{\max} above optimal values will tend to slow down the detection algorithm and increase the chance of miscorrelation, leading to more false events.



▲ **Figure 6.** (a) Distribution of true-positive events that were detected by QCN and confirmed in the GNS catalog. (b) Distribution of false-positive events that were detected by QCN but not listed in the GNS catalog. Variation in the percentage of true-positive events detected by QCN versus the distance of the aftershock from the Darfield mainshock. Events with magnitudes greater than 3 in the GNS catalog are considered. The reference magnitudes are from GNS catalog. Spatial search in the GNS catalog is limited to a radius of 75 km around the epicenter of Darfield earthquake for events that occurred between 9 September 2010 and 1 August 2013.



▲ **Figure 7.** (a) Count and (b) percentage of true-positive and false-positive events versus minimum correlation counts.

A larger number of false-positive events are expected when a larger maximum correlation distance (D_{\max}) or time (T_{\max}) is used. We first test the system using T_{\max} values of 10, 15, 20, 30, 60, 90, 120, and 150 s. During all of the simulations, D_{\max} is kept at 200 km (see Fig. 8a,b). For low values of maximum correlation ($T_{\max} < 20$ s), we observe that the number of both false-positive and true-positive detections increases as T_{\max} increases, but for larger T_{\max} values the count of detections reaches a plateau (Fig. 8a,b). Accordingly, the optimal value of T_{\max} lies between 20 and 30 s for a D_{\max} of 200 km.

We also conduct runs with a range of D_{\max} values: 25, 50, 75, 100, 200, and 300 km. Although in the previous simulation, we vary only a single parameter (T_{\max}), here we vary T_{\max} along with D_{\max} for each simulation. Here, $T_{\max} = D_{\max}/V_S$, in which V_S is the average S -wave velocity, 3.4 m/s. The number of false-positive detections increases smoothly up to T_{\max} of 0.25 (75 km), flattens between 0.25 (75 km) and 0.33 (100 km), and rises again sharply after 0.33 (100 km) (Fig. 8c,d). Simulation time also increases from half a day when D_{\max} is 200 km to two and a half days when D_{\max} is 300 km, an indication that the real-time system would be subjected to larger central processing unit demands for larger D_{\max} . A D_{\max} of 100 km, with a corresponding T_{\max} of 30 s, is optimal for the data set studied here.

Average Travel-Time Misfit

The QCN algorithm finds the location of an earthquake with a 3D search that minimizes the difference between estimated and observed relative arrival times at the sensors. The algorithm defines the maximum average travel-time misfit (MISFIT_{\max}) allowed (Fig. 2). If MISFIT_{\max} is smaller than a user-specified value (the default value is 2 s), then the event is considered as a true earthquake. We investigate the sensitivity of MISFIT_{\max} by testing values of 0.1, 0.25, 0.5, 0.75, 1.0, 2.0, 3.0, 4.0, 8.0, and 16.0 s.

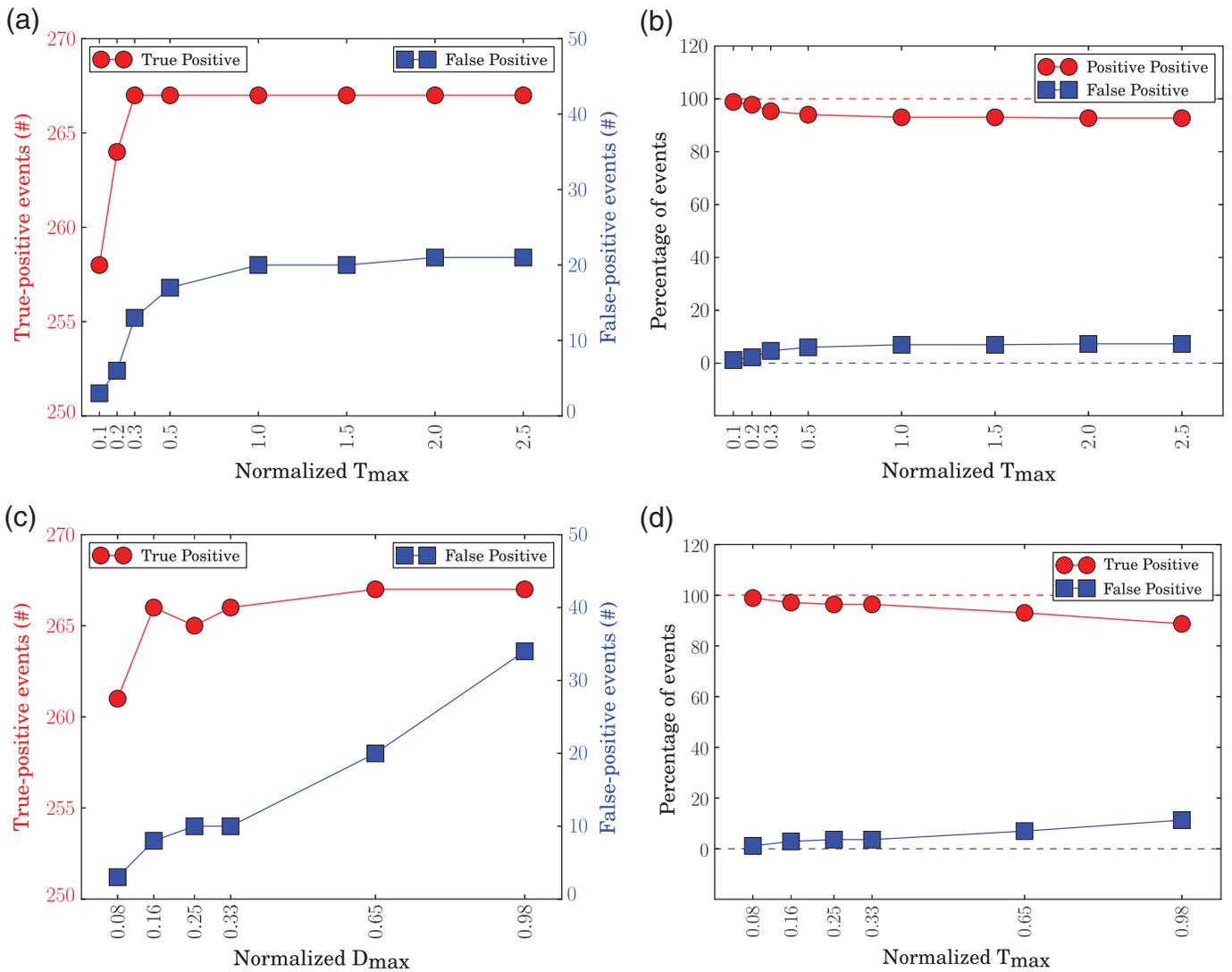
For the above MISFIT_{\max} values, we show the counts of true-positive and false-positive events in Figure 9a. For larger values, MISFIT_{\max} does not influence the number of true-positive events; however, it raises the count of false-positive events, lowering the reliability of the detection (see Fig. 9b). Limiting MISFIT_{\max} to smaller values, the algorithm effectively trims false events as seen in the figure. For the MISFIT_{\max} of 0.1, 0.25, 0.5 s, we see the number of false events declining, at a slower rate, but there is a sharp decrease in true-positive events as well. The optimal value for MISFIT_{\max} is ~ 1 s, which balances the number of false positives with the number of true positives.

We devise a probability fit that is a function of MISFIT_{\max} (Fig. 9c). The fit asymptotically converges to a value of 70 for large MISFIT_{\max} . Using this functional relation, we can estimate the probability that an event is a true positive and skip the algorithm for a final check of the average travel-time misfits. Because these results are based on limited QCN data, the functional relationship needs to be tested with additional data (such as the events recorded by different networks) for the operational code. The expected relationship may vary by region and by sensor or network type.

Two-Dimensional Parameter Space

By varying a single parameter while holding the other parameters fixed, we determine that CNT_{\min} and MISFIT_{\max} strongly affect the code performance (Table 4). We dedicate this section to investigating how these two variables control the code performance when they are varied together. The parameter space includes $\{\text{CNT}_{\min} = 4, 5, 6, 7\}$ and $\{\text{MISFIT}_{\max} = 0.1, 0.25, 0.5, 0.75, 1.0, 2.0, 3.0, 4.0 \text{ s}\}$, yielding results for 32 simulations (Fig. 10).

False detections increase as the average travel-time misfit increases and the minimum correlation count decreases; the percentage of false events reaches its maximum value (about 50% false events) for minimum correlation count of four



▲ **Figure 8.** (a) Count and (b) percentage of true-positive and false-positive events versus normalized maximum correlation times ($T_{\max}/(D_{\max}/V_S)$), in which D_{\max} is fixed at 200 km. (c) Count and (d) percentage of true-positive and false-positive events versus normalized maximum correlation distances ($D_{\max}/(T_{\max} \times V_S)$), in which T_{\max} is computed by dividing the corresponding D_{\max} value by an average S -wave velocity of 3.4 km/s.

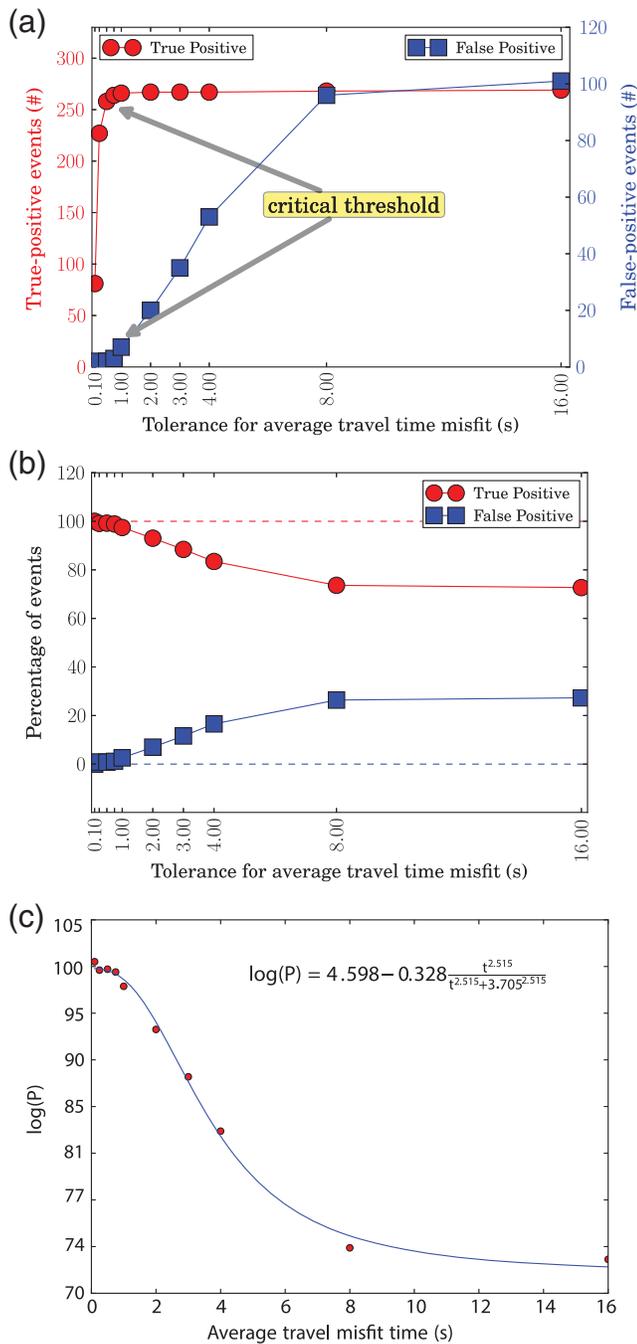
and average travel-time misfit around 4 s. Decreasing the average travel-time misfit (down to 0.5) and increasing minimum correlation count (up to 7), the system gives zero false events, but the code misses many true-positive events (see the values of true-positive event contour lines on the upper-left corner in Figure 10). We qualitatively define an optimal region, framed by the rectangle in Figure 10, in which the QCN detects a high number of true-positive events and few false-positive detections. This result suggests that we should set $MISFIT_{\max}$ to 0.75 or 1 s and CNT_{\min} to five or six.

Run with the Optimal Parameters

Using the optimal parameter values, we conduct a new simulation to compare the default run. The proposed parameters are five for the minimum correlation count, 100 km for the maximum correlation distance, 30 s for the maximum corre-

lation time, and 1.0 s for average travel-time misfit. The number of total events for both runs, including the counts of true-positive and false-positive events, is tabulated in Table 5. We compare the simulation on two different epochs: (1) 1 September 2009 to 9 September 2010, an era wherein the QCN had less reliable sensor and received many false triggers and (2) 9 September 2010 to August 2013, a period wherein the QCN detections stabilized.

For the default parameters, we find 94 events, 11 of which were the confirmed earthquakes from NEIC catalog (see the left super column in Table 5) in the first period, but 83 of them were false events. Using the optimal parameter values, only 11 false events are found rather than the 83 of the default run, a significant improvement. For this period, the simulation with the new parameters misses a single true-positive event among those detected by the default run. For the second



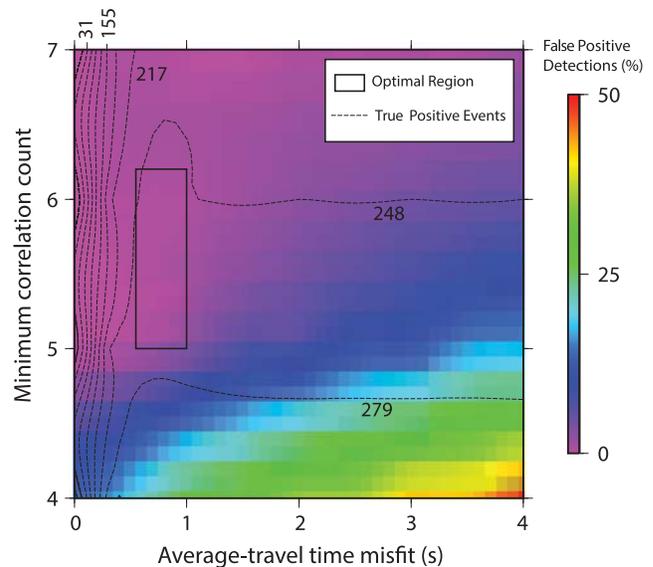
▲ **Figure 9.** (a) Count and (b) percentage of true-positive and false-positive events versus misfit time tolerances. (c) Probability of true-positive events as a function of maximum average travel misfit time (blue line). The red dots represent the probability of true-positive events for different misfit time tolerances (0.1, 0.25, 0.5, 1.0, 2.0, 4.0, 8.0, and 16.0 s) from the offline (version 2.0) code.

period, 9 September 2010 to 1 August 2013, although the new run misses two true-positive events, it filters an additional 14 fewer false events than the default run, increasing the reliability of the system from 93rd percentile to 98th percentile.

Table 4
Influence of the Parameters on the Count of True-positive and False-positive Counts

	Min. Corr. Count	D_{\max}	T_{\max}	Misfit Time
Influence on true-positive count	Mild	Weak	Weak	Strong
Influence on false-positive count	Strong	Mild	Weak	Strong

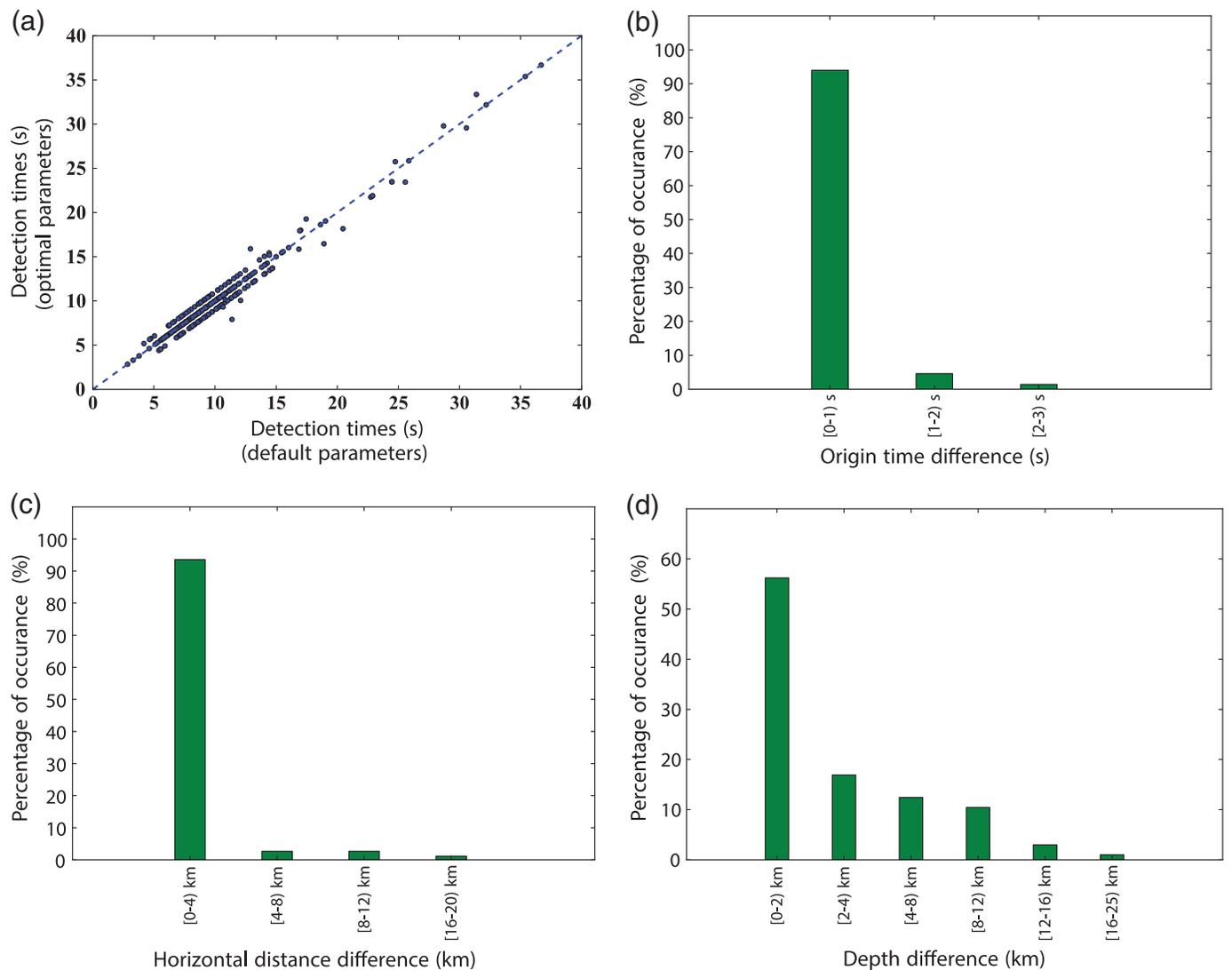
Modifying the parameters also effects the detection time (event origin to detection). We show the detection time, origin time, horizontal, and depth differences between the default and the optimal runs in Figure 11. Because the optimal parameters search over a smaller space-time window for correlating triggers, we might also expect the time to detection to improve. The new runs indeed have faster detection times on the average, as shown in Figure 11a, but the difference is negligibly small. The event origin times, in over 90% of the matching events, have differences within (0–1) s for both runs (see Fig. 11b). Plotting the distribution of horizontal distance differences in Figure 11c, little over 90% of the events are within 4 km distance. Despite the fact that we have very similar



▲ **Figure 10.** Plot contrasting the percentage of false-positive detections versus count of true-positive detections for different maximum average travel-time misfit and minimum correlation count. Background color represents the percentage of false-positive detections and the contour (dashed) lines represent the count of true-positive events. The rectangle outlines the approximate extent of the optimal region in the 2D parametric space. The optimal region is qualitatively defined as the region that minimizes the percentage of false-positive detections while maximizing the number of true-positive detections.

Date Range	Default Parameters			Optimal Parameters		
	Total Events	True Positive	False Positive	Total Events	True Positive	False Positive
1 September 2009 to 8 September 2010	94	11	83	21	10	11
9 September 2010 to 1 August 2013	287	268	19	271	266	5

Events found in NEIC and GNS catalogs are defined as true positive. Events defined as false positive were not validated by NEIC or GNS catalogs.



▲ **Figure 11.** (a) Detection times from the runs using default and optimal parameters (see Table 1) and distribution of (b) difference in event origin times, (c) difference in horizontal (epicentral) distances, and (d) difference in depths plotted for the matching events that occurred between the runs using the optimal and default parameter values. Comparison period is between 1 September 2009 and 1 August 2013.

results for the detection times, event times, and horizontal locations in both codes, the depth is more sensitive to the parameter changes (Fig. 11d).

The run using the optimal parameters reduces the number of false-positive events and keeps the true-positive events nearly as large as the default run. We also show that the new parametrization achieves almost the same event characterizations as the default.

SUMMARY

We test and optimize the new QCN software using a new feature that enables retrospective runs. Besides being a valuable software verification tool, the new feature allows us to investigate how some parameters in the code influence the performance of the system.

Comparing the offline (version 2.0) and online (version 1.x) between September 2010 and August 2013, we find that, on the average, the new code detects the events faster. Validated on the combined GNS and NEIC catalogs, both codes yield similar values of origin time, location, and depth. Overall, version 2.0 provides slightly improved locations, on average.

We simulate recorded data prior to activation of the real-time detection code. For the period from 1 September 2009 to 8 September 2010, the system exposes 11 new earthquakes and 40 new earthquakes between 9 September 2010 and 24 September 2010 that version 1.x missed. In the period that the online code was active (after 25 September 2010), the offline (version 2.0) code detects 228 true-positive events, exceeding the detections of the operational code by 16. We note that the majority of earthquakes examined in this study were recorded near Christchurch, New Zealand, due to a dense aftershock array deployed following the 2010 Darfield, New Zealand, earthquake.

The region near Christchurch, New Zealand, is the region where most of the earthquakes detected by the QCN system occurred. This provides us with a real experiment to assess a social cyber-seismic network. With about 100 volunteers, joining the network part time on an average day, this experiment answers two questions: (1) how many earthquakes the QCN recovers and (2) how many of them QCN misses. The results suggest that the QCN system is very effective for events with magnitudes larger than five, and the system is able to detect 40% of events M_w 4–5 (Fig. 6a). Only 3.5% of the events with M_w 3–4 are detected, but this is perhaps expected due to the low sensitivity of the sensors used (mostly 14-bit) and the distribution of sensors (most sensors were located between 20 and 35 km from the mainshock epicenter). The latest generation of sensors (16-bit) used by QCN would likely have a higher recovery of M_w 3–4 events (Evans *et al.*, 2014).

We investigated the influence of parameters in the code. The four parameters, minimum correlation count, maximum correlation distance, maximum correlation time, and average travel-time misfit are the influential parameters. This study determines the optimal values for these parameters based on the performance of retrospective runs. Running the system with

these optimal parameter values, we confirm that the new code reduces false events significantly while keeping the number of true-positive events as high as possible. The optimal values reduce the number of false-positive events from 83 to 11, for the period before 9 September 2010. For the period after 9 September 2010, the false events are also reduced from 19 to 5. The reliability of the system, for this period, increases from 93rd to 98th percentile.

Although the optimal parameters will improve the accuracy and performance of the QCN rapid detection system for most detections, the same parameters may need to be updated as volunteer participation increases in future. For example, it may be possible to further reduce the maximum average travel-time misfit allowed as sensors may be located closer to earthquake sources. Currently the density of QCN sensors around the world has large disparities with some regions of very dense sensor coverage and others with little to no coverage; further improvement on QCN rapid detection system may include the ability for the system to adapt the parameters dynamically based on the density of the local sensor distribution. In addition, prospective tests with synthetic data or waveform data collected by more traditional sensors will be valuable for further improving the performance of the rapid detection system. ✉

ACKNOWLEDGMENTS

We thank the hundreds of Quake-Catcher Network (QCN) volunteer hosts and the field crew; without them this study would have never happened. The research was supported by National Science Foundation (NSF) Award Number EAR 1027802.

REFERENCES

- Anderson, D., and J. Kubiawicz (2002). The world-wide computer, *Scientif. American* **286**, 40–47.
- Anderson, D. P. (2004). Boinc: A system for public-resource computing and storage, in *Proc. of the 5th IEEE/ACM International Workshop on Grid Computing, GRID '04*, IEEE Computer Society, 8 November 2004, 4–10.
- Anderson, D. P., C. Christensen, and B. Allen (2006). Grid resource management—Designing a runtime system for volunteer computing, ACM Press, New York, New York, 126 pp.
- Azzam, R., T. Fernandez-Steeger, C. Arnhardt, H. Klapperich, and K.-J. Shou (2011). Monitoring of landslides and infrastructures with wireless sensor networks in an earthquake environment, in *5th International Conference on Earthquake Geotechnical Engineering, IEEE Computer Society*, Santiago, Chile, 10–13 January 2011.
- Bassin, C., G. Laske, and G. Masters (2000). The current limits of resolution for surface wave tomography in North America, *Eos Trans. AGU* **81**, F897.
- Christensen, C., T. Aina, and D. Stainforth (2005). The challenge of volunteer computing with lengthy climate model simulations, in *e-Science*, IEEE Computer Society, Washington, D. C., 8–15.
- Chung, A., C. Neighbors, A. Bellmonte, M. Miller, H. Sepulveda, C. Christensen, R. Jakka, E. Cochran, and J. Lawrence (2011). The quake-catcher network rapid aftershock mobilization program following the 2010 M 8.8 Maule, Chile earthquake, *Seismol. Res. Lett.* **82**, no. 4, 526–532.

- Chung, A. I., E. S. Cochran, A. Kaiser, C. M. Christensen, B. Yildirim, and J. F. Lawrence (2015). Improved rapid magnitude estimation for a community-based, low-cost MEMS accelerometer network, *Bull. Seismol. Soc. Am.*, doi: [10.1785/0120140232](https://doi.org/10.1785/0120140232) (in press).
- Clayton, R., T. Heaton, M. Chandy, A. Krause, M. Kohler, J. Bunn, R. Guy, M. Olson, M. Faulkner, M. Cheng, *et al.* (2012). Community seismic network, *Ann. Geophys.* **54**, no. 6, 738–747.
- Cochran, E., J. Lawrence, C. Christensen, and A. Chung (2009). A novel strong-motion seismic network for community participation in earthquake monitoring, *IEEE Instrum. Meas. Mag.* **12**, no. 6, 8–15.
- Cochran, E., J. Lawrence, A. Kaiser, B. Fry, A. Chung, and C. Christensen (2012). Comparison between low-cost and traditional MEMS accelerometers: A case study from the M 7.1 Darfield, New Zealand, aftershock deployment, *Ann. Geophys.* **54**, no. 6, 728–737.
- Cochran, E. S., J. F. Lawrence, C. Christensen, and R. S. Jukka (2009). The quake-catcher network: Citizen science expanding seismic horizons, *Seismol. Res. Lett.* **80**, no. 1, 26–30.
- Cranswick, E., and R. Banfill (1990). Proposal for a high-density people's seismograph array, *Eos Trans. AGU* **71**, no. 43, 1469.
- Cranswick, E., B. Gardner, S. Hammond, and R. Banfill (1993). Recording ground motions where people live, *Eos Trans. AGU* **74**, no. 21, 243–244.
- D'Alessandro, A. (2014). Monitoring of earthquakes using MEMS sensors, *Curr. Sci.* **107**, 733–734.
- D'Alessandro, A., and G. D'Anna (2013). Suitability of low cost 3 axes MEMS accelerometer in strong motion seismology: tests on the lis331dlh (iphone) accelerometer, *Bull. Seismol. Soc. Am.* **103**, no. 5, 2906–2913.
- D'Alessandro, A., D. Luzio, and G. D'Anna (2014). Urban MEMS based seismic network for post-earthquakes rapid disaster assessment, *Adv. Geosci.* **40**, 1–9.
- Earle, P. S., and P. M. Shearer (1994). Characterization of global seismograms using an automatic-picking algorithm, *Bull. Seismol. Soc. Am.* **84**, no. 2, 366–376.
- Evans, J. R., R. Allen, A. Chung, E. S. Cochran, R. Guy, M. Hellweg, and J. F. Lawrence (2014). Performance of several low-cost accelerometers, *Seismol. Res. Lett.* **85**, 147–158.
- Geonet (2010–2013). <http://info.geonet.org.nz> (last accessed March 2015).
- Horiuchi, S., Y. Horiuchi, S. Yamamoto, H. Nakamura, C. Wu, P. A. Rydelek, and M. Kachi (2009). Home seismometer for earthquake early warning, *Geophys. Res. Lett.* **36**, no. 5, [10.1029/2008GL036572](https://doi.org/10.1029/2008GL036572).
- Kohler, M. D., T. H. Heaton, and M.-H. Cheng (2013). The Community Seismic Network and Quake-Catcher Network: Enabling structural health monitoring through instrumentation by community participants, *Proc. SPIE* **8692**, 86,923X–86,923X-8.
- Koide, E., N. Fukuwa, K. Masaki, T. Hara, K. Ohta, and K. Itoigawa (2006). Development of low-cost seismograph for building response through internet, *J. Architect. Build. Sci.* **23**, 453–458.
- Laske, G., A. Dziewonski, and G. Masters (2012). *The Reference Earth Model*, <http://igppweb.ucsd.edu/~gabi/rem.html> (last accessed March 2015).
- Lawrence, J. F., E. S. Cochran, A. Chung, A. Kaiser, C. M. Christensen, R. Allen, J. W. Baker, B. Fry, T. Heaton, D. Kilb, *et al.* (2014). Rapid earthquake characterization using MEMS accelerometers and volunteer hosts following the m 7.2 Darfield, New Zealand, earthquake, *Bull. Seismol. Soc. Am.* **104**, no. 1, 184–192.
- Luetgert, J. H., D. H. Oppenheimer, and J. Hamilton (2010). The Net-Quakes Project—Research-quality seismic data transmitted via the Internet from citizen-hosted instruments (invited), *AGU Fall Meeting Abstracts*, San Francisco, California, 13–17 December 2010.
- Naito, S., H. Azuma, S. Senna, M. Yoshizawa, H. Nakamura, K. X. Hao, H. Fujiwara, Y. Hirayama, N. Yuki, and M. Yoshida (2013). Development and testing of a mobile application for recording and analyzing seismic data, *J. Disast. Res.* **8**, no. 5, 990–1000.
- National Earthquake Information Center (NEIC) (2010–2013). <http://neic.usgs.gov> (last accessed March 2015).
- Taylor, S. R., P. E. Harben, S. Jarpe, and D. B. Harris (2011). Development of mine explosion ground truth smart sensors, *Tech. Rept.*, DTIC Document.

Battalgazi Yildirim
Angela Chung
Carl M. Christensen
Jesse F. Lawrence
Department of Geophysics
Stanford University
Stanford, California 94305 U.S.A.
yildirim@stanford.edu

Elizabeth S. Cochran
Earthquake Science Center
U.S. Geological Survey
525 & 535 S. Wilson Street
Pasadena, California 91106-3212 U.S.A.

Published Online 1 April 2015