

AUTOMATED, EFFICIENT DETECTION AND LOCATION OF LOCAL AND REGIONAL SEISMIC EVENTS USING IRREGULAR NETWORKS AND ARRAYS

Michael A.H. Hedlin and Catherine D de Groot-Hedlin

**Institute of Geophysics and Planetary Physics
University of California, San Diego
9500 Gilman Drive
La Jolla, CA 92093-0225**

28 February 2021

Final Report

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION IS UNLIMITED.



**AIR FORCE RESEARCH LABORATORY
Space Vehicles Directorate
3550 Aberdeen Ave SE
AIR FORCE MATERIEL COMMAND
KIRTLAND AIR FORCE BASE, NM 87117-5776**

DTIC COPY

NOTICE AND SIGNATURE PAGE

Using Government drawings, specifications, or other data included in this document for any purpose other than Government procurement does not in any way obligate the U.S. Government. The fact that the Government formulated or supplied the drawings, specifications, or other data does not license the holder or any other person or corporation; or convey any rights or permission to manufacture, use, or sell any patented invention that may relate to them.

This report was cleared for public release by AFMC/PA and is available to the general public, including foreign nationals. Copies may be obtained from the Defense Technical Information Center (DTIC) (<http://www.dtic.mil>).

AFRL-RV-PS-TR-2021-0070 HAS BEEN REVIEWED AND IS APPROVED FOR PUBLICATION IN ACCORDANCE WITH ASSIGNED DISTRIBUTION STATEMENT.

//SIGNED//

Dr. Raymond J. Willemann
Program Manager/AFRL/RVB

//SIGNED//

For: Erin N. Pettyjohn, Chief
AFRL Geospace Technologies Division

This report is published in the interest of scientific and technical information exchange, and its publication does not constitute the Government's approval or disapproval of its ideas or findings.

REPORT DOCUMENTATION PAGE				Form Approved OMB No. 0704-0188	
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing this collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number. PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.					
1. REPORT DATE (DD-MM-YYYY) 28-02-2021		2. REPORT TYPE Final Report		3. DATES COVERED (From - To) 29 Aug 2019 – 28 Feb 2021	
4. TITLE AND SUBTITLE Automated, Efficient Detection and Location of Local and Regional Seismic Events Using Irregular Networks and Arrays				5a. CONTRACT NUMBER FA9453-19-C-0074	
				5b. GRANT NUMBER	
				5c. PROGRAM ELEMENT NUMBER C6601F	
6. AUTHOR(S) Michael A.H. Hedlin and Catherine D de Groot-Hedlin				5d. PROJECT NUMBER 1010	
				5e. TASK NUMBER	
				5f. WORK UNIT NUMBER VIGY	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Institute of Geophysics and Planetary Physics University of California, San Diego 9500 Gilman Drive La Jolla, CA 92093-0225				8. PERFORMING ORGANIZATION REPORT NUMBER	
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES) Air Force Research Laboratory Space Vehicles Directorate 3550 Aberdeen Avenue SE Kirtland AFB, NM 87117-5776				10. SPONSOR/MONITOR'S ACRONYM(S) AFRL/RVBN	
				11. SPONSOR/MONITOR'S REPORT NUMBER(S) AFRL-RV-PS-TR-2021-0070	
12. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release; distribution is unlimited (AFMC-2021-2178 dtd 12 Jul 21)					
13. SUPPLEMENTARY NOTES					
14. ABSTRACT We present results from the adaptation and application of the AELUMA method to data from single stations and arrays in the auxiliary and primary International Monitoring System (IMS) seismic networks. The report describes the code development phase which used data collected by stations in North America – largely from the Transportable Array. Many subsets of the network were used to test code performance on networks that approach the sparsity of the IMS. The code was then applied to a month of data collected by IMS seismic stations in Europe, Northern Africa and Asia. Throughout this project we gauged performance of the code using ground-truthed events, such as a large mine blast in Utah, and, mainly, seismic catalogs produced by United States Geological Survey's National Earthquake Information Center (USGS-NEIC). Principal issues in the application of the method to the IMS were 1) the particularly large distances between single stations and 2) the existence of dense arrays co-mingled with the single stations. While the code is fully operational, our tests indicate it detects fewer than anticipated events. Possible key remedies include 1) focusing on either P or S detections in the event location stage, and 2) finding a better event association method.					
15. SUBJECT TERMS array processing, signal detection, event location					
16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT Unlimited	18. NUMBER OF PAGES 98	19a. NAME OF RESPONSIBLE PERSON Dr. Raymond J. Willemann
a. REPORT Unclassified	b. ABSTRACT Unclassified	c. THIS PAGE Unclassified			19b. TELEPHONE NUMBER (include area code)

This page is intentionally left blank.

TABLE OF CONTENTS

	Page
1. Summary.....	1
2. Introduction.....	1
3. Methods, Assumptions, and Procedures of the AELUMA method.....	2
4. The datasets.....	3
4.1. Waveform data.....	3
4.2. Reference events.....	5
5. Single frequency analysis of DMC data.....	5
6. Multi-frequency analysis of DMC data.....	12
7. Decimation – how does the method perform with fewer stations?.....	17
8. Surface waves – how do they compare to the high frequency analyses above?.....	20
9. Sparse network approach.....	23
10. Application to IMS data.....	24
11. Results and Discussion.....	40
References.....	44
Appendices: Introduction, data processing at sparse networks and arrays.....	46
Appendix A Delaunay triangulation on a sphere	47
Appendix B Triad detection code	48
Appendix C Array detection code.....	55
Appendix D Event association and event location.....	65
Appendix E User manual - input.....	77
Appendix F User manual - output.....	85
Appendix G AELUMA code.....	87

	Page
Appendix H NEIC catalogs.....	87
Appendix I AELUMA catalogs.....	87
List of Symbols, Abbreviations, and Acronyms.....	88

LIST OF FIGURES

Figure	Page
1. Map of NEIC events used for comparison with AELUMA.....	7
2. On January 1, 2018 there were 473 stations in our database.....	8
3. Detections vs time for January 9, 2018 using a single filter that spans the entire band of interest (0.5-7.8 Hz.....	9
4. Triad size vs. mean number of detections at each triad within a bin.....	10
5. Comparison of AELUMA event detections vs NEIC events for 1 January 2018.....	11
6. Here we show detections vs time for January 9, 2018 using four overlapping filters that span the entire band of interest (0.5-7.8 Hz.....	13
7. The top panel shows the average number of triads per day vs mean side length of triad for January, 2018. The bottom panel shows the mean number of detections at each triad within the bins above, vs mean triad arm length, at each frequency band.....	14
8. Histogram of how well AELUMA does as a function of event magnitude - considering events from the entire month (January, 2018).....	16
9. The gray histogram is the number of events/day in the NEIC catalog. The red histogram shows the number of these events that were also detected by AELUMA. The gray histogram is identical to middle panel of figure 1).....	16
10. Figure showing triad configurations for 3 levels of decimation.....	18
11. Histogram of how well AELUMA does as a function of event magnitude - considering events from the entire month (January, 2018) – when just larger triads are used (side lengths from 160-640 km.....	20
12. The gray histogram is the number of events/day in the NEIC catalog. The red histogram shows the number of these events that were also detected by AELUMA when just using larger triads with side lengths from 160-640 km. The gray histogram is identical to middle panel of figure 1). Compare this with Figure 9.....	20
13. Map of events that generated surface waves that were detected by AELUMA.....	22
14. Left column shows the polarization analysis for a station about 3090 km from its Caribbean source. The right column shows the corresponding analysis for a station about 2340 km from the event.....	24
15. Events in the NEIC catalog during the month from mid-July to mid-August, 2017.....	26
16. Triangulation of the stations in the IMS network that were used in our analysis.....	27
17. Detections made by stations in the IMS.....	28
18. Compares AELUMA event detections vs NEIC event detections for day 219 2017.....	29
19. An event in both catalogs.....	30
20. All detections by arrays and triads in the IMS are represented by black and red dots respectively.....	32
21. This figure compares AELUMA event detections vs NEIC event detections for day 226 2017.....	33
22. The largest magnitude NEIC event that was NOT detected by AELUMA.....	34
23. Example of mb 4.4 event found by both NEIC and AELUMA.....	35

Figure	Page
24. See caption for figure above (except no red stars since it was not detected by NEIC).....	36
25. Top) Events in both the NEIC and AELUMA catalogs for the month of this study (black and red respectively), middle) events just in the AELUMA catalog, bottom) events just in the NEIC catalog.....	38
26. Histogram of how well AELUMA does as a function of event magnitude.....	39
27. Gray histogram gives the number of events each day in NEIC catalog within the area of interest. Red histogram shows the ones that were also detected by AELUMA.....	39
28. Signals recorded from one event by stations in the IMS.....	42

Table	LIST OF TABLES	Page
1. Names, latitudes and longitudes of the twenty one single stations that were used in the analysis. All data were obtained via vDEC.....		4
2. Names, latitudes and longitudes of the eighteen arrays in the study area. All data were obtained via vDEC.....		4

1. SUMMARY

We present results from the adaptation and application of the AELUMA method to data from single stations and arrays in the auxiliary and primary International Monitoring System (IMS) seismic networks. The report describes the code development phase which used data collected by stations in North America – largely from the Transportable Array. Many subsets of the network were used to test code performance on networks that approach the sparsity of the IMS. The code was then applied to a month of data collected by IMS seismic stations in Europe, Northern Africa and Asia. Throughout this project we gauged performance of the code using ground-truthed events, such as a large mine blast in Utah, and, mainly, seismic catalogs produced by United States Geological Survey's National Earthquake Information Center (USGS-NEIC). Principal issues in the application of the method to the IMS were 1) the particularly large distances between single stations and 2) the existence of dense arrays co-mingled with the single stations. While the code is fully operational our tests indicate it detects fewer than anticipated events. Possible key remedies include 1) focusing on either P or S detections in the event location stage, and 2) finding a better event association method.

2. INTRODUCTION

Here we summarize our research on extending software to sift through large geophysical datasets in search of signals and their sources. Although the method, AELUMA, has proved to be effective when applied to distinctly different types of geophysical data (seismic, barometric, infrasound) it had to date only been applied to networks of single stations. It had not yet been applied to networks that include both single stations and arrays. Further development of the AELUMA method was necessary for application to seismic data in the primary and auxiliary seismic networks in the IMS for the purpose of nuclear monitoring; this is the subject of this report.

Our research has been broken down into two tasks. In the first task we made the necessary changes to the software while testing the evolving code on data collected in North America and made available to us via the IRIS Data Management Center (DMC). While offering a much greater density of stations than in the IMS, and easy access to the data, this dataset allowed us to assess how the performance of the method would change as the density of stations decreased. This dataset included both stations in research networks (including the USArray Transportable Array) and in the IMS.

In the second task we used only IMS data in Europe for tests of the new algorithm under real (nuclear monitoring) circumstances. This task is particularly challenging not only due to the existence of numerous, often large, dissimilar, seismic arrays but due to the very low density of the network.

Under both tasks we compared catalogs of events produced by AELUMA with those from independent sources - chiefly from the USGS.

Here we summarize this work and conclude with our assessment of the utility of the method for use in detecting and locating seismic events of interest to the nuclear monitoring community. This report is organized as follows. The main body provides a general review of the method and the data used in our study. This section focuses on results from analyses of recorded data in North America and Europe. The report includes nine appendices in which we present various aspects of the method and tests on synthetic data. We point to catalogs and the AELUMA software package. Here in the main body of the report we make frequent references to the appendices which are used to describe the code in much greater detail.

3. METHODS, ASSUMPTIONS, AND PROCEDURES OF THE AELUMA METHOD

Here we provide a brief, conceptual, overview of the AELUMA method. We discuss the code that has been developed in this project in the main body of the report below and in appendices. We point to various appendices for more detailed explanations of the various processing steps taken by AELUMA.

AELUMA was developed in anticipation of ever expanding datasets and the increasing need to reduce analyst workload. In essence AELUMA is an automated routine designed to comb through arbitrarily large datasets in search of multiple recordings of signals that can be associated with a common source and then ultimately used to locate the source. AELUMA is simple in concept but has shown itself to be highly flexible in application to widely varying datasets. Very briefly, any network of stations can be recast as a web of small arrays with each array comprising a few neighboring stations. At each array location signal properties are estimated. In this initial step all arrays are processed independently - without any consideration of results from other arrays. Combining results from all arrays can shed light on where signals from a common source have been detected, how these signal properties vary across the region sampled by the stations and thus can provide insight into the nature of the source – its location, time, and size. The source’s size is inferred from the amplitude of the recorded signals and the number of arrays that recorded it. AELUMA does not provide an estimate of the event’s magnitude.

In practice AELUMA divides a network of sensors into 3-element arrays using Delaunay triangulation (Lee and Schacter, 1980, Appendix A). We refer to each array as a “triad”. A cross-correlation procedure is applied to data from each station pair within a triad to determine time delays (Appendix B). If the total time delay is consistent with a plane wave signal crossing all three stations, a detection is declared and *tau-p* processing is used to estimate the signal’s propagation speed and direction. The same processing is applied to each time window and each triad within the network. The software processes data from regularly spaced time intervals to yield a database of all detections made by all triads across the entire time interval of interest. Commonly, the network is too sparse to allow coherent processing of data. In such cases the data are converted to envelopes. Signal shapes, represented by envelopes, are often coherent between widely separated stations even when the actual recordings are not (e.g. Gibbons et al., 2008).

The previous step in which a network of stations is triangulated into a web of three-element arrays is used when the dataset is simply a network of individual stations. Datasets that also include arrays of stations require somewhat different processing to detect signals at each array. As summarized in Appendix C, when processing array data *tau-p* processing is used as with 3-element triads. However, when processing data from an N-element array where N would be expected to be larger than 3, data are cross-correlated between all station pairs within the array. All time offsets from that array are used to estimate the signal properties, speed and direction, using *tau-p* processing.

In the next stage AELUMA selects subsets of the detected signals that are all consistent with one source (Appendix D). These detected signals come from both triads and arrays. The selection of signals that are to be associated with a common event is based on their arrival time and direction of propagation. The two sets of detections are given equal weight in later processing to determine the source parameters (location and time). Once a subset of signals is associated with a common event they are removed, only the remaining detections are used to find other sources. A fitness function that considers signal arrival location, time, and direction of propagation is used to select the most likely source location and estimate the uncertainty in its location.

As mentioned above this is just a general overview of the AELUMA method. Below we discuss changes we have made to the software to accommodate IMS data in greater detail.

4. THE DATASETS

4.1 Waveform data: For us, the logical testbed for the AELUMA algorithm as we developed it for application to IMS seismic data was in North America. In this region we not only have access to a significant number of seismic stations but data from many of them are easily accessed through the IRIS Data Management Center (DMC). This allowed us to modify the code while taking advantage of existing code for data input. This region was also selected due to the unusually high density of stations that were deployed across the contiguous United States and Alaska/Yukon over many years. This allowed us to assess how the code would perform under widely varying conditions at a range of distances from seismic events. We were able to down-sample the network to closely examine change of code performance as the network became progressively sparser. In this part of the project we selected data from the continental US in 2007 as well as both the continental US and Alaska in 2017, 2018 and 2019. These data were used just in task 1.

The second database, used for task 2, was from IMS stations in Europe, Asia and Africa. This dataset was necessary as in this task we were to apply the code under conditions faced by the nuclear monitoring community. In this task our operating assumption was that the data would be exclusively from the IMS and not any additional single or network stations that might be available in some regions – although the method can easily accommodate such data. Throughout this part of the project we used data from primary

and auxiliary stations together. For this part of the contract we applied for and obtained a license to access IMS data via the Virtual Data Exploitation Centre (vDEC). We downloaded data from 4 weeks (July 15-August 15, 2017 - JD 196-227) in a region centered on the eastern Mediterranean. The study area extended from 6°N to 71°N, 9°W to 106°E. The dataset included data from 18 arrays and 21 single stations (Tables 1 and 2). Three stations in the study area, AS030 (FURI), AS062 (KOWA) and PS21 (THR), did not provide data.

Table 1. Names, latitudes and longitudes of the twenty one single stations that were used in the analysis. All data were obtained via vDEC.

AAK (AS060)	42.639100	74.494200
AKTO (AS059)	50.4348	58.0164
ARU (AS085)	56.4293	58.5615
ASF (AS056)	32.1723	36.8972
ATD (AS028)	11.53	42.847
BELG (AS094)	52.393374	47.624402
DAVOX (AS102)	46.7806	9.8797
DBIC (PS15)	6.6701	-4.8563
EIL (AS048)	29.6725	34.9519
GNI (AS003)	40.1495	44.7414
IDI (AS036)	35.288	24.89
JMIC (AS073)	70.986600	-8.505150
KBZ (PS032)	43.726898	42.8996
KEST (PS42)	35.7317	9.346
KIRV (AS082)	58.58504	49.36622
MDT (AS066)	32.814	-4.607
MLR (AS081)	45.4917	25.9437
OBN (AS084)	55.1138	36.5687
VAE (AS050)	37.469	14.3533
VRAC (AS026)	49.30828	16.59351
WSAR (AS074)	23.239	58.6307

Table 2. Names, latitudes and longitudes of the eighteen arrays in the study area. All data were obtained via vDEC.

AKASG (PS45)	50.7012	29.2242
ARCES (PS28)	69.534800	25.505700
BRTR (PS43)	39.725	33.639
BVAR (AS57)	53.024900	70.388500
EKA (AS104)	55.3332	-3.1588
ESDC (PS040)	39.6744	-3.963
FINES (PS17)	61.44363	26.07713

Table 2 (continued). Names, latitudes and longitudes of the eighteen arrays in the study area. All data were obtained via vDEC.

GERES (PS19)	48.845106	13.701559
GEYT (PS44)	37.92955	58.11706
HFS (AS101)	60.133474	13.69449
KURK (AS58)	50.721610	78.563360
KVAR (AS083)	43.95567	42.69517
MKAR (PS23)	46.770000	82.300300
MMAI (AS049)	33.01518	35.40311
NOA (PS27)	61.0397	11.2148
SONM (PS25)	47.834690	106.394990
TORD (PS26)	13.14771	1.6947087
ZALV (PS33)	53.948060	84.818810

4.2 Reference events: an essential component of our research was assessing the accuracy of locations obtained by AELUMA as well as the completeness of its catalogs. For this report, in our research in North America and the European region we used the catalogs produced by the USGS National Earthquake Information Center (NEIC), obtained online via <https://earthquake.usgs.gov/earthquakes/search/>.

The catalog used for the study in North America included events from the entire month of January, 2018. The search region extended from 25°-70° N, 175° - 65°W. The minimum magnitude of the 1679 events in the catalog was 2.5. For our analysis of IMS data in Europe, Asia and Africa, from mid-July to mid-August, 2017, we included events from 30°-60° N and 0° to 85°E. The minimum magnitude of the 109 events in the catalog was 3.8.

At times we used other catalogs, including GGCEL ([https:// seismo.com/ggcel/](https://seismo.com/ggcel/)). We processed recordings of a number of highly accurately constrained events (such as the Crandall Canyon mine collapse in 2007).

5. SINGLE FREQUENCY ANALYSIS OF DMC DATA

We have run various tests using the data in the continental United States. Some preliminary results were reported at the 2020 TIM in Melbourne, Florida. Here we illustrate the first phase of the work by presenting results from a different dataset - one collected during the time period from January 1-31, 2018. This time period provides a good opportunity to test the AELUMA method. As shown in Figure 1 this time period includes ~ 3 weeks of relative quiescence followed by a M7.9 earthquake just south of the Aleutians and then a significant aftershock sequence. Data from this time period gives us an opportunity to demonstrate how well AELUMA does as a function of both event magnitude and event frequency. The sharply different rate of seismicity before and after the main shock tests the method in the case that events should be relatively easy to separate in time, and also for the case that events are closely packed in space and time, and thus are more difficult to differentiate. It is this latter time, after the main shock, that

presents AELUMA with the greatest challenge. In part, the difficulty arises from the fact that most of the events were located outside the footprint of the network.

The geographic distribution of stations was highly uneven; although an irregular station spacing is more typical of the type of global station coverage to which AELUMA might be applied in future, this complicates the analysis. The distribution of stations in the TA and other networks during that time period also gives us an opportunity to examine how well AELUMA adapts to triads of varying dimensions. The density of stations in Alaska is fairly uniform, however across the entire network the density is highly variable overall. Some regions (e.g. Canada) have few stations in the database. The density of stations in the continental US, particularly the west coast, is relatively low.

Here we use broadband vertical channels, BHZ, to detect high frequency signals. The distance between stations was constrained to range from a minimum of 10 km to a maximum of 640 km. The minimum interior angle of each triad was set to 12° . This wide range of allowable interstation distances leads to a wide range in triad size. This, in turn, allows us to examine the role of triad dimension in detecting signals - one of our key goals, as it allows us to examine how the performance of the method changes as the density of the stations decreases and whether the method continues to be useful at the density of stations that are in the IMS.

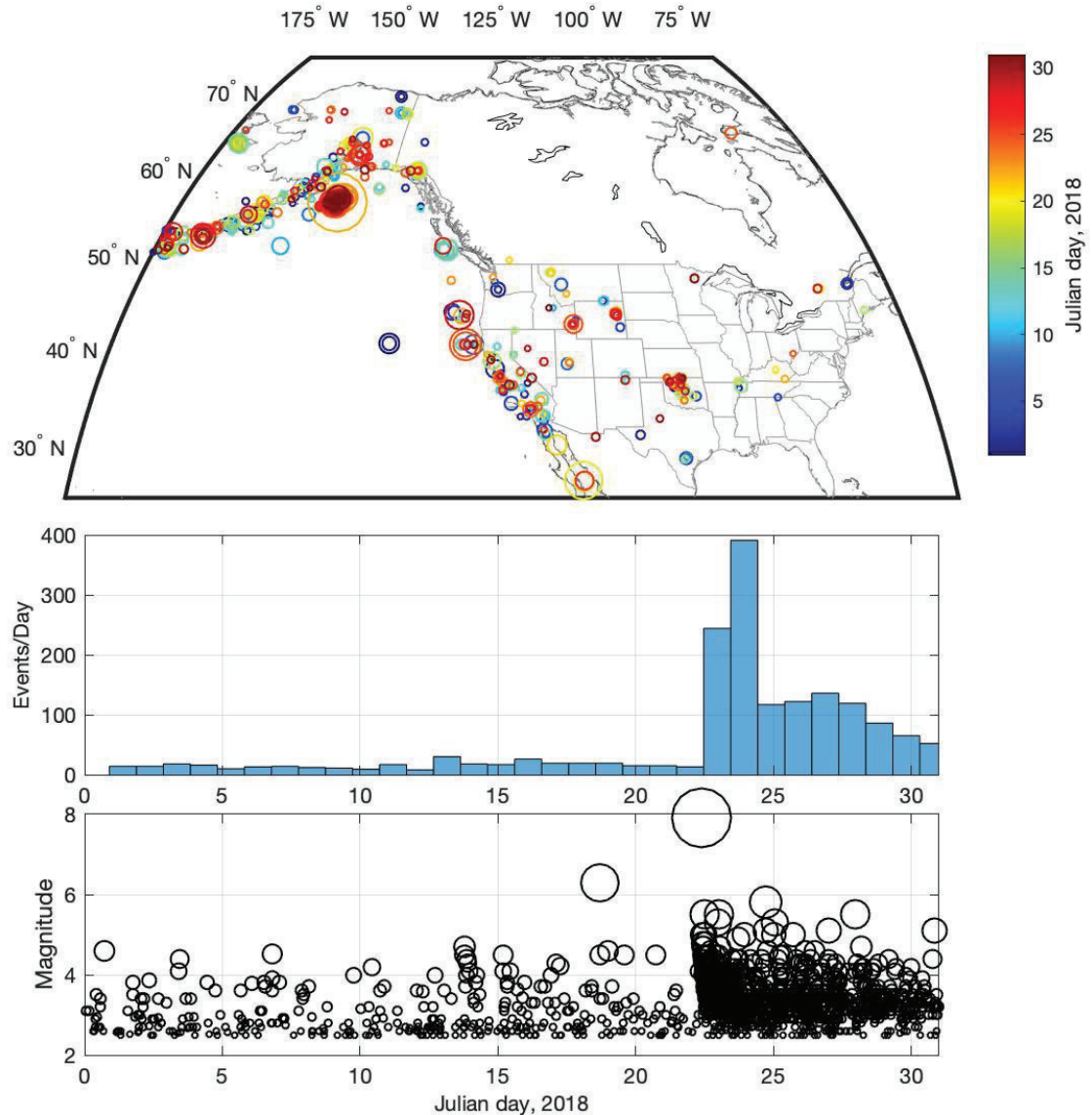


Figure 1. Map of NEIC events used for comparison with AELUMA. *In the top and bottom panels symbol sizes scale with event magnitude. The catalog is dominated by natural seismicity across much of the region and a few anthropogenic hotspots - in particular in Wyoming and Oklahoma. The middle panel shows normal, low, level of activity until the main shock and its aftershock sequence. Event sizes are plotted in the lowest panel.*

Figure 2 illustrates the widely variable triad dimensions over North America. This map uses the stations that were available from the DMC on January 1, 2018. On this day, 824 triads satisfied the interstation distance and interior angle criteria given above. The arrangement of triads does not change much over the month as station distribution is nearly unchanged over the entire study period. A comparison of Figures 1 and 2 shows how the events relate geographically with the available stations for this time period. Many stations were in Alaska during this time period, so events there should be well-

located. However, there are far fewer stations along the west coast of the contiguous United States to detect and locate the events off the west coast of the US. Several stations in the database were located in the Yukon Territory of Canada, close to Alaska.

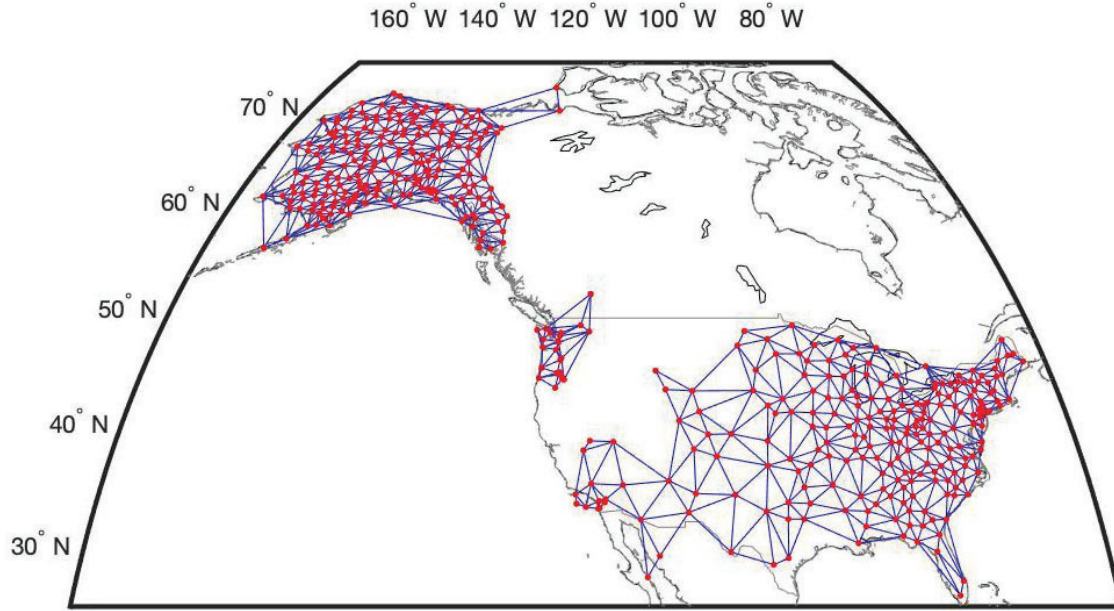


Figure 2. On January 1, 2018 there were 473 stations in our database. *AELUMA* found 824 triads that satisfy the conditions given above. Coverage in Alaska is clearly extremely good.

First, to get a baseline assessment of AELUMA, the code was applied to an entire month of data over a single band that spanned the entire band of interest (0.5-7.8 Hz). The wide variety of triad configurations is shown in Figure 2. Each day was divided into 573 overlapping time windows, each of length 600 seconds, with a time step of 150 seconds. The time step provides an overlap of 75%. The duration of the time window is governed by the distance for a signal to travel from one station to another at the maximum allowable interstation distance (see appendix E), so allowing large interstation distances leads to longer time windows. Since a maximum of one signal can be detected in each time window, events in a tight spatial cluster will not be readily detected if they are less than 150 seconds apart, at best.

The detections for a single day, January 9, 2018, are shown in Figure 3. Each point represents the time and amplitude of a signal that was detected at one triad. This day is relatively quiet as indicated in Figure 1. With this single band (0.5-7.8 Hz) AELUMA detected only 762 signals across all of the available triads. The detections come in bursts - with each burst likely associated with a single event. Large events can be recognized by a large number of detections at a range of amplitudes.

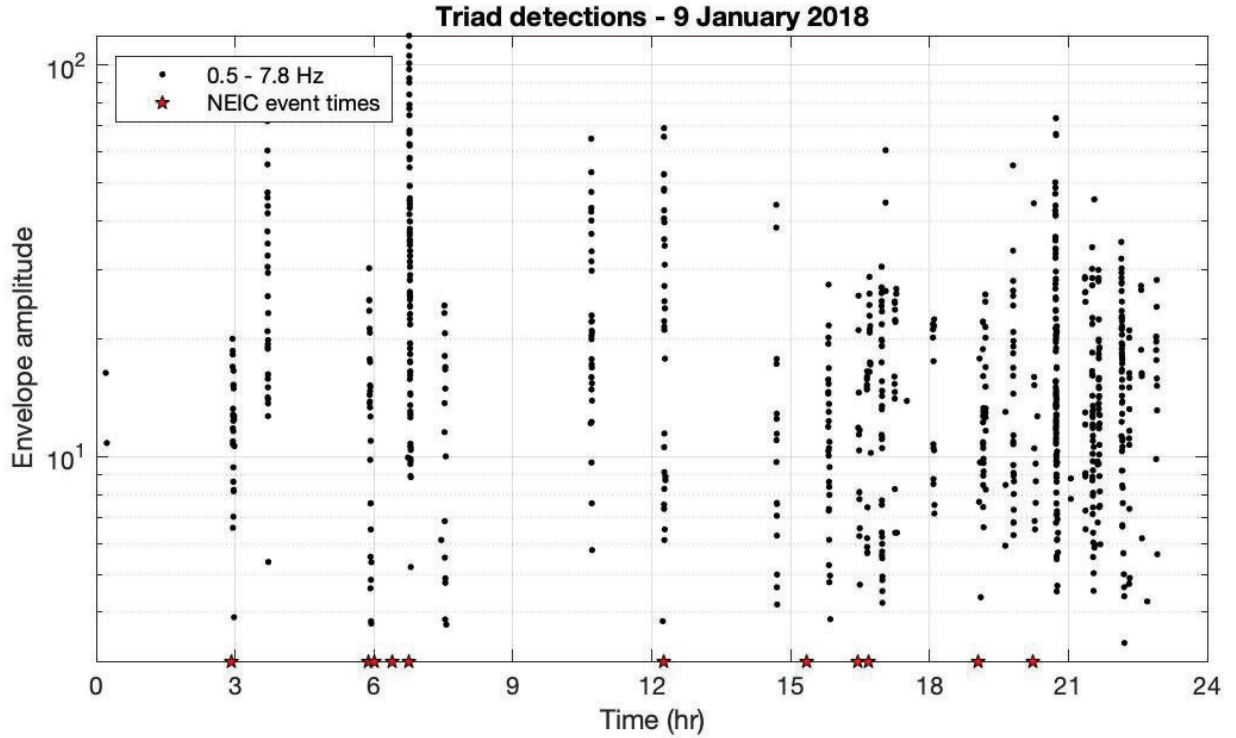


Figure 3. Detections vs time for January 9, 2018 using a single filter that spans the entire band of interest (0.5-7.8 Hz). *The time of NEIC events is shown at the bottom. Many detections appear to be associated with events not in the NEIC catalog, while several NEIC events (e.g. just after 15:00 UT) appear to have been missed by AELUMA in this initial run.*

The detections shown in Figure 3 come from a wide range of triad dimensions. Figure 4 shows how the detection rate varies with the size of the triads, averaged over the entire month of January, 2018. In this figure the upper panel shows a histogram of the number of triads as a function of mean side length; most triads have a mean side length of 50 to 200 km. The lower panel shows the average number of detections per triad, as a function of triad size. That is, each point represents the total number of detections for all triads within a given bin in the top panel, divided by the number of triads in that bin. Surprisingly, the number of detections remains robust with increasing triad size - i.e. as they approach the dimensions that are common in the IMS network. However, as the upper panel shows there are a vanishingly small number of triads in this dataset at these largest sizes. The result is, of course, dependent on where events are located relative to these triads. As we see below many of the events in this month were located in the Aleutians. The larger triads are located mainly in the continental US further from the events.

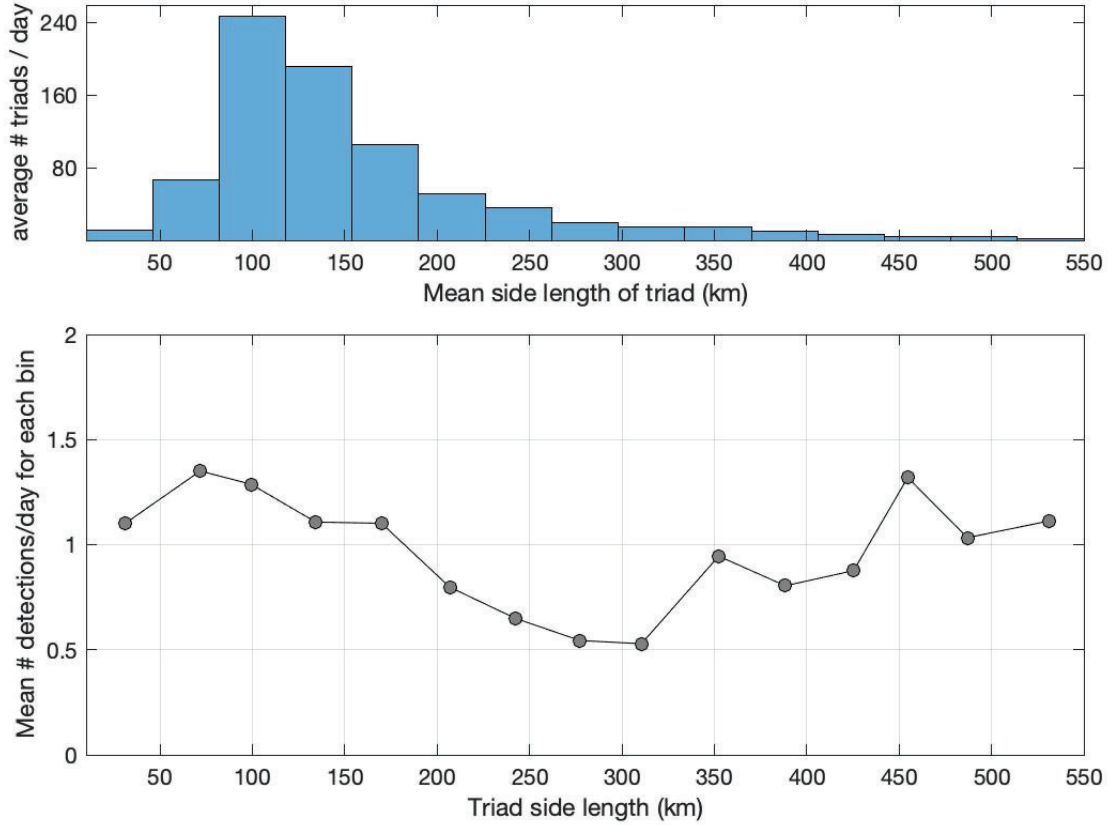


Figure 4. Triad size vs. mean number of detections at each triad within a bin. *The top panel shows the average number of triads per day vs mean side length of triad for January, 2018. Bottom. For each bin of the upper panel, shows mean number of detections at each triad within a bin vs. the mean triad arm length.*

The signal detections were sorted into clusters, each consistent with a single event, as described in Appendix D. At least 10 detections were required to define an event; events with a lower number of associated detections were discarded. An optimal source location and time was then estimated for each event cluster, with an associated error ellipse for each event, as also described in Appendix D. The resulting event parameters were compared to those in the NEIC catalog over the latitude range from 20 to 75° North, and -175 to -65° East. The catalogs were compared as follows. An event is selected from the AELUMA catalog. The NEIC catalog is scanned for events that fit within that event's uncertainty ellipse. If an event fits spatially, its origin time is then compared with that of the event in the AELUMA catalog. The allowed discrepancy in time between events to scale with the size of the uncertainty ellipse. The allowed time before or after the AELUMA origin time is the ellipse's major axis length divided by 2.2 km/s (the minimum allowable phase velocity) + 60 seconds. Later we quote the results of our comparison of the different catalogs. Not uncommonly the code would find more than one NEIC event that matched with any given event in the AELUMA catalog, due both to

the size of the error ellipse and the rapidity of nearly collocated events in the aftershock sequences.

In this case there were 1679 events in the NEIC catalog for the month of January 2018, with locations and magnitudes as shown in Figure 1. For the single frequency version of this code, AELUMA found only 618 events over this region for the same time period. Of these, 277 could be matched to NEIC events. Given the size of the error ellipses, some of the AELUMA events could be matched to more than one NEIC event; in all 422 NEIC events had matches to these 277 AELUMA events.

Figure 5 compares output from the single-frequency AELUMA code with our reference catalog produced by NEIC for a relatively inactive day - January 9, 2018; the detections were shown above in Figure 3. The NEIC catalog for that day includes 11 events, compared to 21 events in the AELUMA catalog. Of these events, only 5 were in both catalogs.

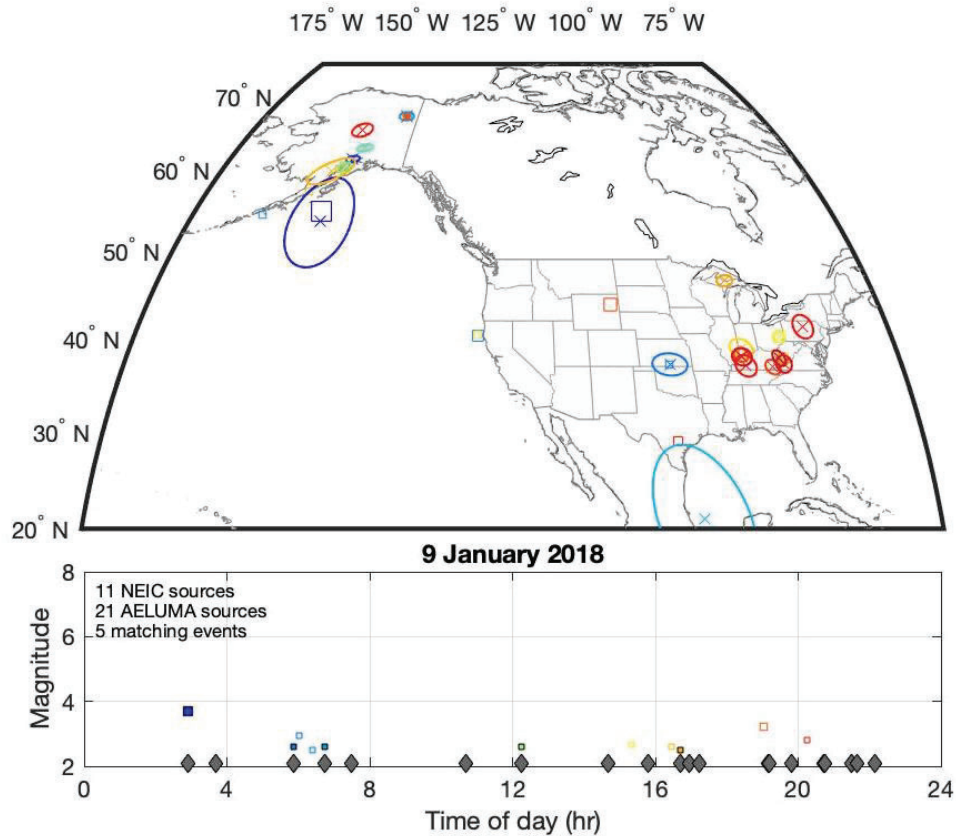


Figure 5. Comparison of AELUMA event detections vs NEIC events for 1 January 2018. The squares in the top panel are NEIC events on that day. The symbols are color-coded by time of day with the size of the symbol proportional to the event magnitude. The x's are events detected by AELUMA with the same color-code scheme. The 95% confidence location ellipses are shown for each event located by AELUMA. The bottom panel shows event magnitude versus time. The squares again represent events in the NEIC catalog. If

filled, the event was also detected by AELUMA. The x's at the bottom of the figure show the times of all events detected by AELUMA on that day.

Poor results are obtained using a single frequency to cover the single, rather wide, frequency band of interest; we detected only just over 25% of the NEIC events (422 of the 1679 events in the NEIC catalog). The problem with using a wide frequency band for detections is that, if the signal is rather narrow band, as is the case for most events, the SNR will be poor over a wider band and the signal correlations will likewise be poor. The resulting event association and source parameter estimation based on these detections is then also poor. This suggests that results would have been better if we had chosen a narrower band in which most of the signals lie. This is what was done in previous applications of AELUMA - that is, we selected an applicable frequency band for a signal of interest, then used that. More generally, however, the correct frequency band would not be known in advance. Also, signals of interest may lie in several frequency bands, depending on source type and the range from the source. To remedy this, the AELUMA code was updated to allow for multiple, overlapping frequencies. Detections were made over a series of narrower frequency bands, and then grouped into clusters, as before.

6. MULTI-FREQUENCY ANALYSIS OF DMC DATA

In previous applications (e.g. de Groot-Hedlin and Hedlin, 2015) and in the section above, the AELUMA method has made detections over a single frequency band. Such an approach is relatively simple to apply and sufficient when attempting to detect signals from sources with a known frequency band. We expect, however, that a single filter band spanning the entire frequency range of interest would be less effective overall, that is, when used to process recordings of dissimilar events that produce narrow-band signals that occur at a range of frequencies. The code was altered to allow multiple, narrow frequency bands. Here it is used to divide the band of interest into 4 narrow, overlapping, frequency bands (0.5-1.5 Hz; 0.87-2.6 Hz; 1.5-4.5 Hz; 2.6-7.8 Hz). To provide the fairest comparison of single- versus multi-band analysis, the combined, overlapping, bands span the same overall frequency band used in the single frequency analysis. Other than using multiple bands the number of stations used, the number of triads, the number of time windows, etc. are all the same. The only difference is that rather than using a single wide band we now combine results from the four overlapping bands. As shown in Figure 6 the altered code makes far more detections.

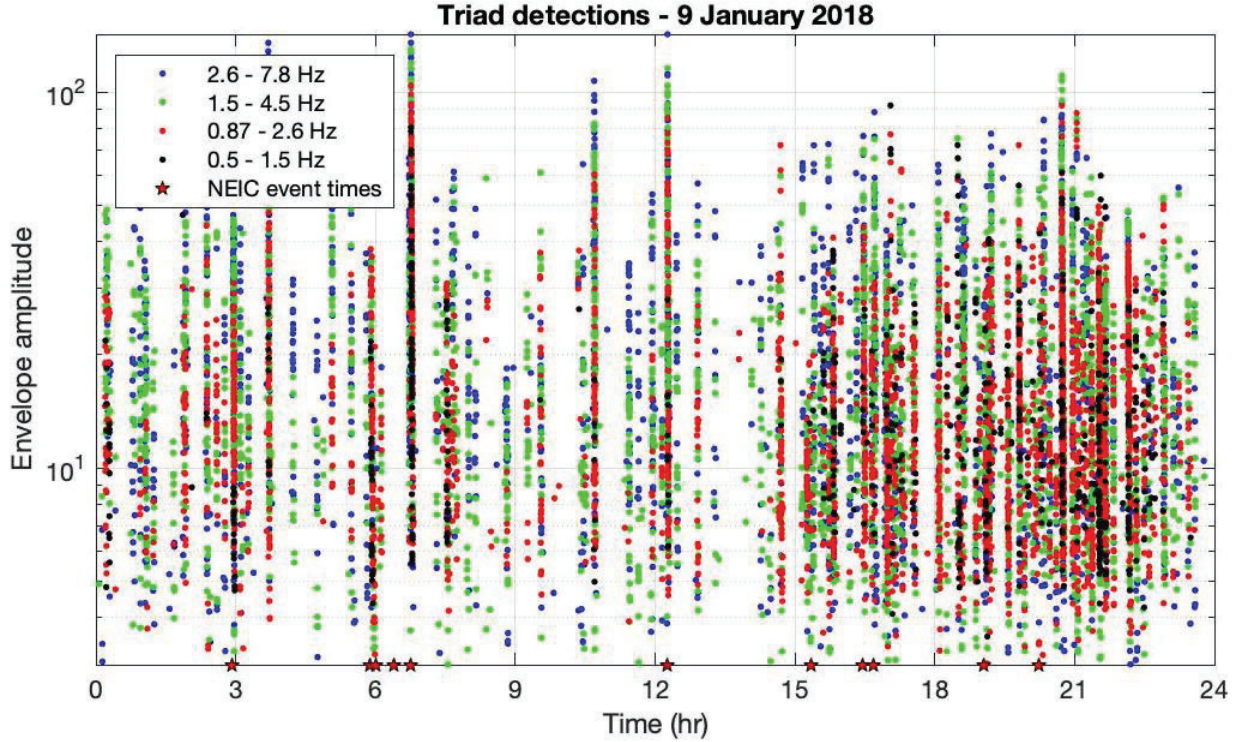


Figure 6. Here we show detections vs time for January 9, 2018 using four overlapping filters that span the entire band of interest (0.5-7.8 Hz). *The frequencies covered by each dot are given in the index. As in Figure 3 we superimpose the time of NEIC events at the bottom.*

Detections

On this day, January 9, 2018, AELUMA formed triads from the same stations as for the single frequency run. The code divided the day into the same 573 overlapping time windows of 600 seconds each, with time steps of 150 seconds. The four bands registered the following number of detections: 666 detections for the 0.5-1.5 Hz band, 2809 detections for the 0.87-2.6 Hz band, 3776 detections for the 1.5-4.5 Hz band, and 2934 detections for the 2.6-7.8 Hz band. There are a total of 10,185 for this day for these 4 frequency bands. That is ~ 13 times the number of detections that were made when AELUMA simply used a single filter to cover the entire band as shown in Figure 3.

Figure 6 shows that there is the highest number of detections at higher frequency bands, implying better cross-correlation - i.e. higher SNR - at the corresponding wavelengths. Recall that we are cross-correlating envelopes of the filtered recordings. At these station offsets we would not expect any detections if we processed filtered waveforms, rather than their envelopes. The number of detections is sharply lower in the lowest frequency band. This figure also shows that there are a significant number of detections at times corresponding to NEIC events suggesting that difficulties with locating events has to do with the event associations and not the detection part. The figure also reveals many of the detected events are not in NEIC catalog.

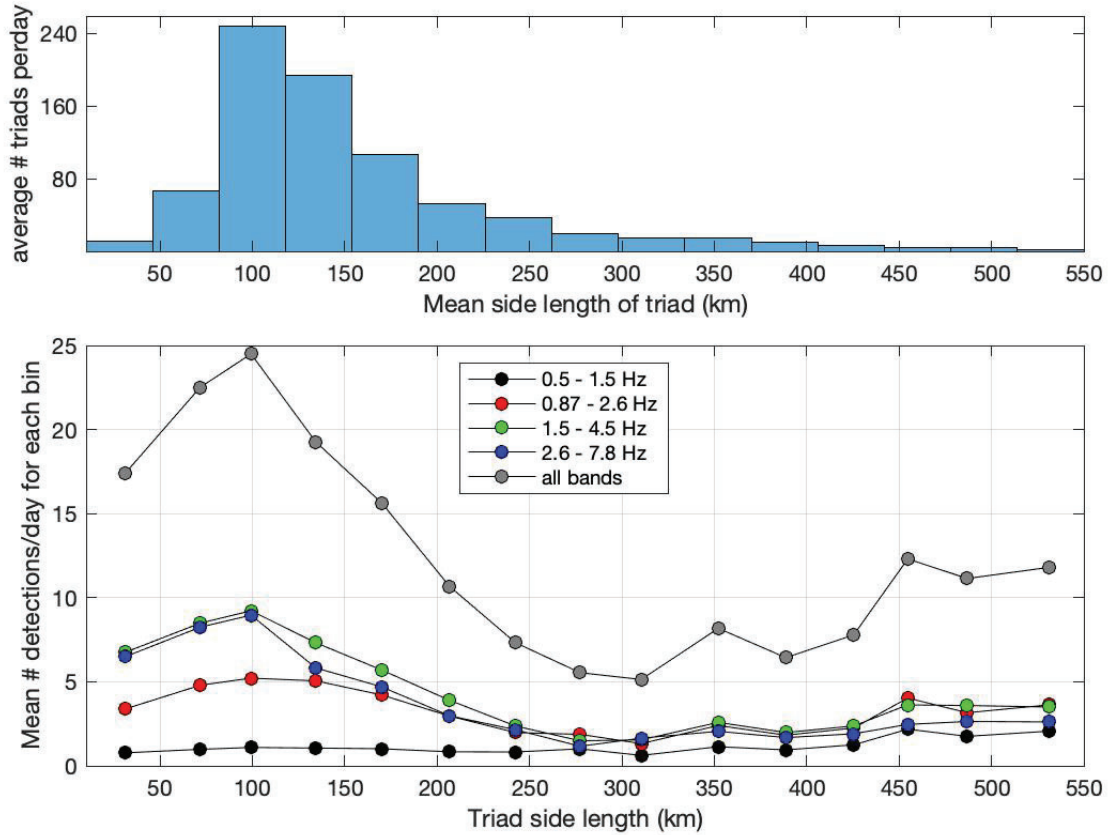


Figure 7. The top panel shows the average number of triads per day vs mean side length of triad for January, 2018. The bottom panel shows the mean number of detections at each triad within the bins above, vs mean triad arm length, at each frequency band. *The sum of all detections is shown in grey.*

Figure 7 repeats the analysis shown in Figure 4; here the different frequency bands can be more clearly discerned than in Figure 6. The average number of detections per triad within each band (and in all bands combined) are shown as a function of the size of the triad (as expressed as the mean triad side length) for January. The mean triad side length ranged from under 50 km to close to 550 km. A station separation of less than 200 km appears optimal although the detection rate continues to be substantial across all triad dimensions. However, the falloff in detection numbers is not steep, which is encouraging for data processing of sparse networks, including the IMS. The number of detections also depends on where the triads are located relative to the events. We return to this topic later in the discussion of decimation.

We note that the least effective frequency band at all triad sizes is the lowest one - from 0.5 to 1.5 Hz. Not surprisingly the detection rate in this band increases somewhat at the larger triads. Overall the higher frequency bands detect fewer events as the triad size

increases. This is expected. The minor increase in the detection rate in most bands at the larger triad sizes is largely due to a chance concentration of events near larger triads.

Event association

The processing that takes the large number of detections and associates subsets of them to individual events is discussed in appendix D and will not be repeated here. For this test a mincluster value of 15 was used; as explained in appendix E, this is the minimum number of triads needed to form an event association. Here the number of triads associated with the same event ranged from below the minimum accepted, 15, to 328. While the NEIC catalog had 1679 events during January, AELUMA found 3279 events. The number of AELUMA events that successfully matched with NEIC events was 908. The number of NEIC events that matched with AELUMA events was 1235, thus some AELUMA events matched with more than one NEIC event.

Figure 8 summarizes the performance of AELUMA as a function of event magnitude. The reference dataset, from NEIC, provided event locations and magnitudes. In general the percentage of events in the NEIC catalog that were also found by AELUMA decreases with decreasing magnitude. Below M 3.4 a third to half of the NEIC events are not matched. Overall 75% of the AELUMA events matched with NEIC events.

While a notable number of NEIC events were not matched by AELUMA events, many events in the AELUMA catalog were not matched with NEIC events. One possible explanation of this is that the AELUMA location uncertainties are underestimated. It is possible that some AELUMA events were actually false and due to chance correlation of unrelated noise bursts. The code applies a large number of checks on the data to weed out such false events. However, when mincluster is low, due to a shortage of stations, the probability of this occurring increases, particularly when there are a very large number of detections, as there were in this case. Here, the mincluster value of 15 should have been high enough to make forming events from noise bursts quite unlikely. The issue is discussed further in the section on IMS data. The hurdles applied by the code to avoid this problem are discussed in detail in the appendices.

Figure 9 presents the same information except plots detections as a function of day - before and after the main shock in Alaska. We expected that the rate of AELUMA events matching with NEIC events would decrease somewhat during the aftershock sequence, but this was not observed. The main shock occurred at 09:31 UT on Jan 23, 2018. The USGS reported 60 aftershocks greater or equal to magnitude 4.0 in the 48 hours following the event. The NEIC table we used for this study contained ~700 events near the main shock between the time of the main event and the end of Jan 25. That's an average rate of just over 10 events/hour. The rate of seismicity in the hours following the event was even higher (Figure 1). The AELUMA method used time windows of 10 minutes. This is greater than the average time between aftershocks (~ 6 minutes) between the main shock and the end of January 25 and much greater than the average time between aftershocks in the hours following the main shock. In this situation the code will cross correlate time series that often have more than one signal. Given that these are closely-spaced aftershocks the complicated multi-event waveforms should still

correlate well. That is, the code will likely estimate an average location when faced with time series that include more than one closely spaced event. Given the density of events in this aftershock sequence it is rather surprising that AELUMA matched as many events in the NEIC catalog as it did (Figure 9).

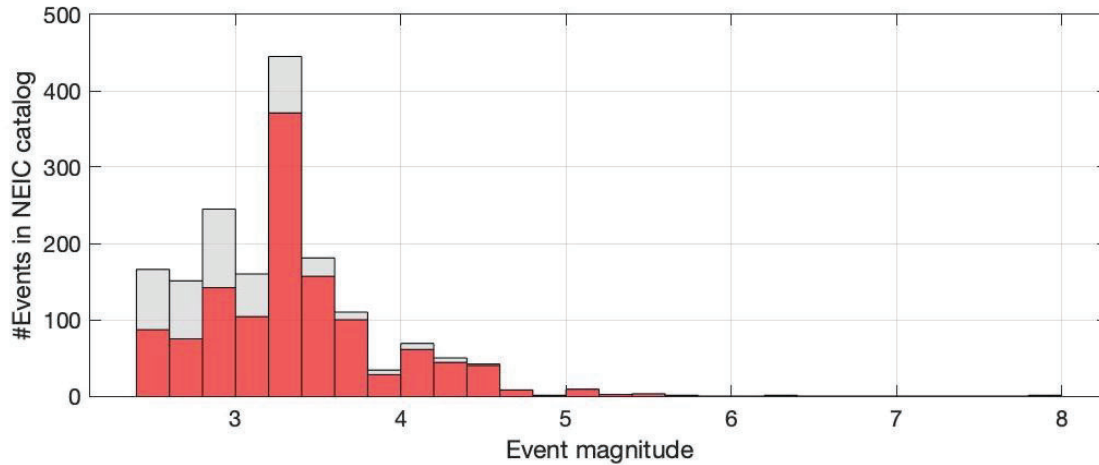


Figure 8. Histogram of how well AELUMA does as a function of event magnitude - considering events from the entire month (January, 2018). *The grey and red histograms are NEIC events and those events that matched with events in the AELUMA catalog respectively. This result is from the multi-band AELUMA code. We used four frequency bands, given earlier.*

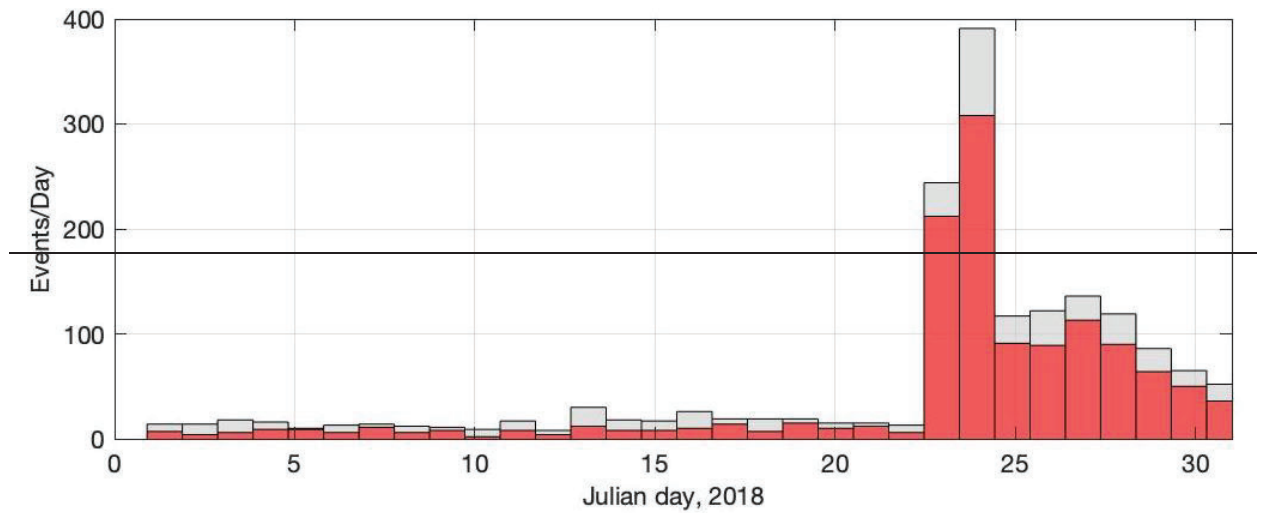


Figure 9. The gray histogram is the number of events/day in the NEIC catalog. The red histogram shows the number of these events that were also detected by AELUMA. The gray histogram is identical to middle panel of figure 1.

7. DECIMATION – HOW DOES THE METHOD PERFORM WITH FEWER STATIONS?

Figure 10 summarizes an experiment we conducted with progressively larger triads. We accomplished this by taking advantage of the unusually high density of stations available. By gradually increasing the acceptable triad side lengths we forced the code to form a progressively sparser mesh of triads. The parameters used to limit triad size are given in the caption for Figure 10. The different decimation levels forced trade-offs between different parameters. For example, time windows can be shorter for smaller triads. This is useful when processing recordings of rapid aftershock sequences. Shorter triad sizes however come with the drawback that many stations were discarded because they were too far from the rest of the network. That is, for smaller triads, we have shorter time windows but large areas have no triads, as shown in Figure 10a.

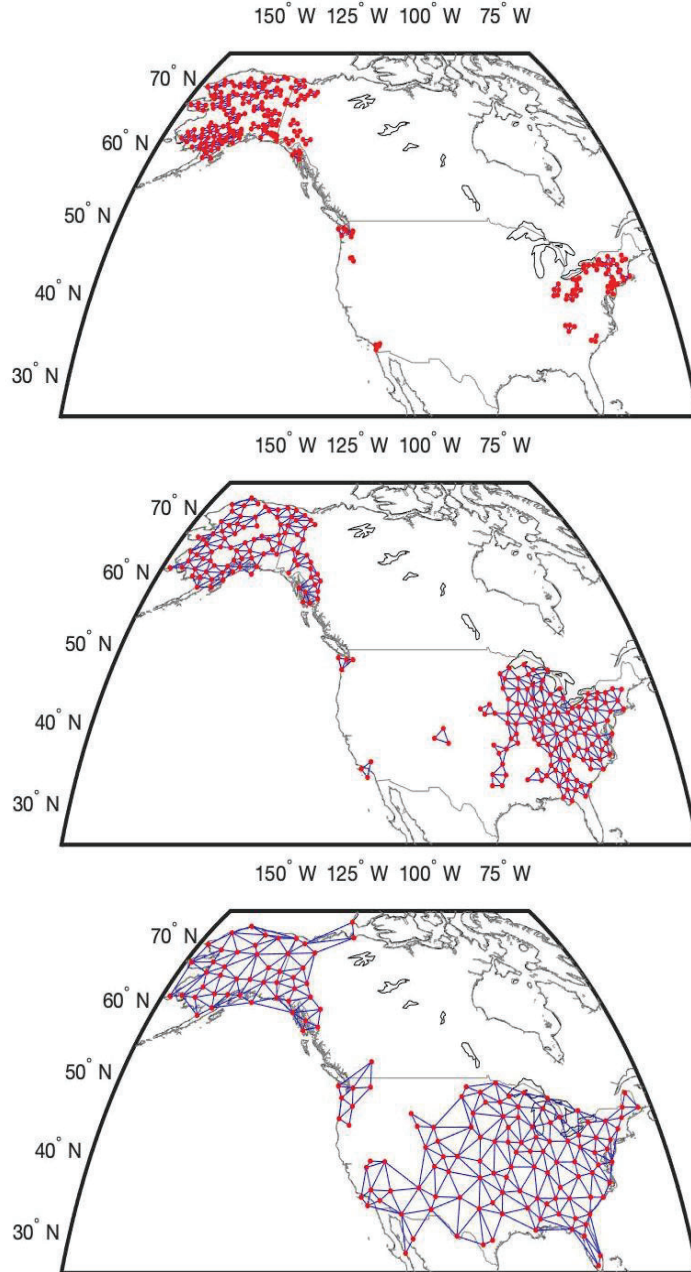


Figure 10. Figure showing triad configurations for 3 levels of decimation. *Top)* maximum interstation spacing 120 km; minimum interstation spacing = 30 km. *Middle)* maximum interstation spacing 230 km; minimum interstation spacing = 100 km. *Bottom)* maximum interstation spacing 640 km; minimum interstation spacing = 160 km.

The results of the decimation test are summarized below. In each case we used the same four frequency bands given above. We required a minimum of 10 triads to register a detection (i.e. mincluster = 10 for each run).

First run - interstation spacing from 30 to 120 km:

In this run we used 250 stations yielding 204 triads. As shown in Figure 10 above, many stations were eliminated because they were further than 120 km from their nearest neighbors. The time range was divided into 2301 windows of duration 150 seconds each. AELUMA found 2304 events of which 790 matched with NEIC events. The number of NEIC events that matched with AELUMA events was 1148. In some cases, the error ellipse for an AELUMA event is large enough to encompass more than 1 NEIC event. On average about half of AELUMA events match with 2 NEIC events.. The number of triads used for locating events ranged from 10 to 109.

Second run - interstation spacing from 100 to 230 km:

This run formed 270 triads from 222 stations. Here, we used 1597 windows of duration 216 s each. Here the performance is not much worse in the case of smaller-sized triads. More events are detected overall (2598 events), but fewer that match NEIC (ie more false alarms). We matched 676 AELUMA events with NEIC events. The number of NEIC events that matched AELUMA events was 908. In other words, again due to the size of some location error ellipses for AELUMA events was consistent with more than one NEIC event. The number of triads used for common events ranged from 10 to 143.

Third run - interstation spacing from 160 to 640 km:

Here 163 stations were used to form 254 triads. Here, we used 573 windows of duration 600 s each. Here the performance is not much worse than in the case of smaller triads. AELUMA found 1072 events for this case, i.e. fewer false alarms than for the smaller interstation distances tried, above. We matched 635 AELUMA events with NEIC events. The number of NEIC events that matched AELUMA events was 974. In other words, again due to the size of some location error ellipses, some AELUMA events were consistent with more than one NEIC event. The number of triads used for common events ranged from 10 to 128.

Results from this run show there is not a huge degradation in the number of events found with increasing triad dimension, but there are fewer ‘false alarms’. This might simply be because smaller triads see smaller events, which are missed by the big triads, i.e. ‘false alarms’ are not false, they are just small. Still only about 75% of events are being found by automated means. The issue and possible remedies is discussed later.

In figures 11 and 12 we plot the results from the sparsest version of the North American network as a function of event magnitude and date. Comparing these figures with figures 8 and 9 we can see poorer results, but not a significant degradation.

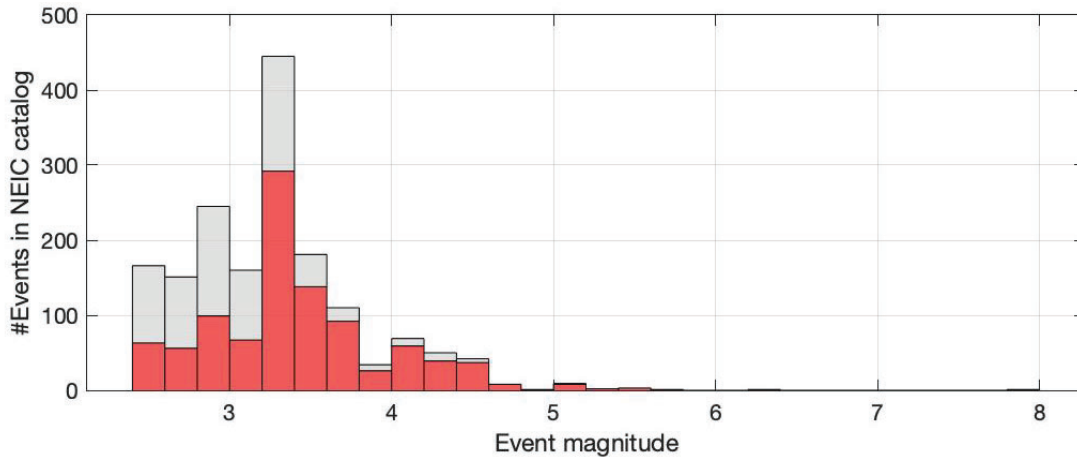


Figure 11. Histogram of how well AELUMA does as a function of event magnitude - considering events from the entire month (January, 2018) – when just larger triads are used (side lengths from 160-640 km). *The grey and red histograms are NEIC events and those events that matched with events in the AELUMA catalog respectively. This result is from the multi-band AELUMA code. We used four frequency bands, given earlier. We used the standard clustering method described in the appendix. Compare this figure with Figure 8.*

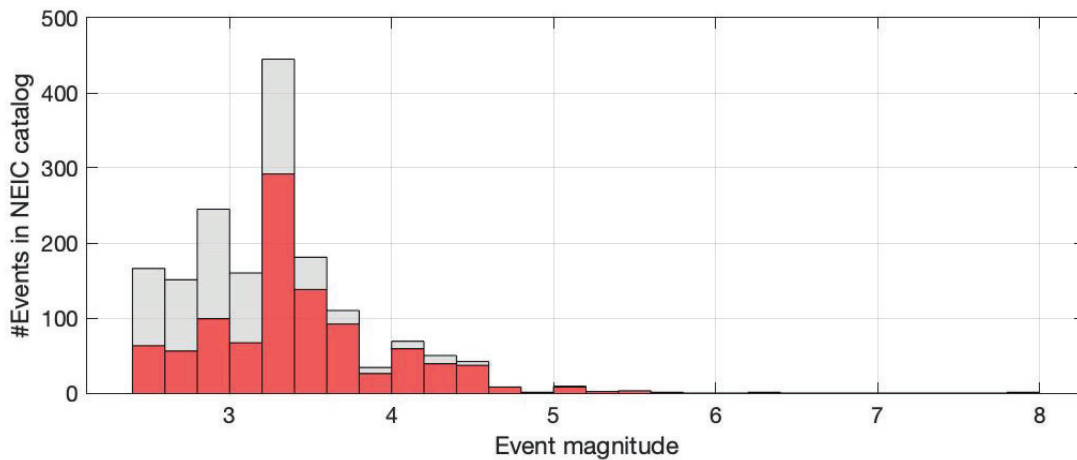


Figure 12. The gray histogram is the number of events/day in the NEIC catalog. *The red histogram shows the number of these events that were also detected by AELUMA when just using larger triads with side lengths from 160-640 km. The gray histogram is identical to middle panel of figure 1. Compare this with Figure 9.*

8. SURFACE WAVES – HOW DO THEY COMPARE TO THE HIGH FREQUENCY ANALYSES ABOVE?

With just a few changes in parameters, we can apply the code to look for events that generate surface waves (Fan et al., 2018). Surface waves can be detected at great distance. The problem with this is that surface waves from distant events can interfere

with locally-sourced signals, making it more difficult to detect small, local, events. Also, in this application AELUMA detects other signals not of interest to the nuclear monitoring community such as stormquakes (Fan et al., 2019). This is especially a problem for this time period (January).

We looked at surface waves in the period range from 13 seconds to 80 seconds (3 individual bands to cover this entire range). Many events were found, but most were outside of the region of interest. The map below shows the 380 events that were found in the region of interest. Most of these are probably stormquakes, not earthquakes. For instance, there are a cluster of events in a non-seismogenic region off the east coast on 5 January. There is another, larger cluster off the Pacific northwest of the US, extending well northward off the west coast of Canada on 18 January. However, as shown in Figure 1, NEIC also identified several earthquakes in that area around that time. So it is not clear whether the 18 January sources are related to storm activity or seismic activity. The events on 23 January and afterwards correlate with the seismic events in the Gulf of Alaska also identified in the NEIC catalog.

Overall 110 AELUMA events matched with NEIC events (Figure 13). The number of NEIC events that matched with AELUMA events was 167. The number of triads used for the AELUMA solutions ranged from 21 to 727.

Ironically, the very efficient propagation of surface waves makes them poor at identifying seismic waves detected by the NEIC catalog because (1) seismicity from events at great distance can swamp signals from local, small events (2) non-seismic sources (i.e. storm activity) also generate significant surface waves. Overall, the false alarm rate combined with the poor seismic detection rate makes surface waves a poor choice for nuclear monitoring efforts.

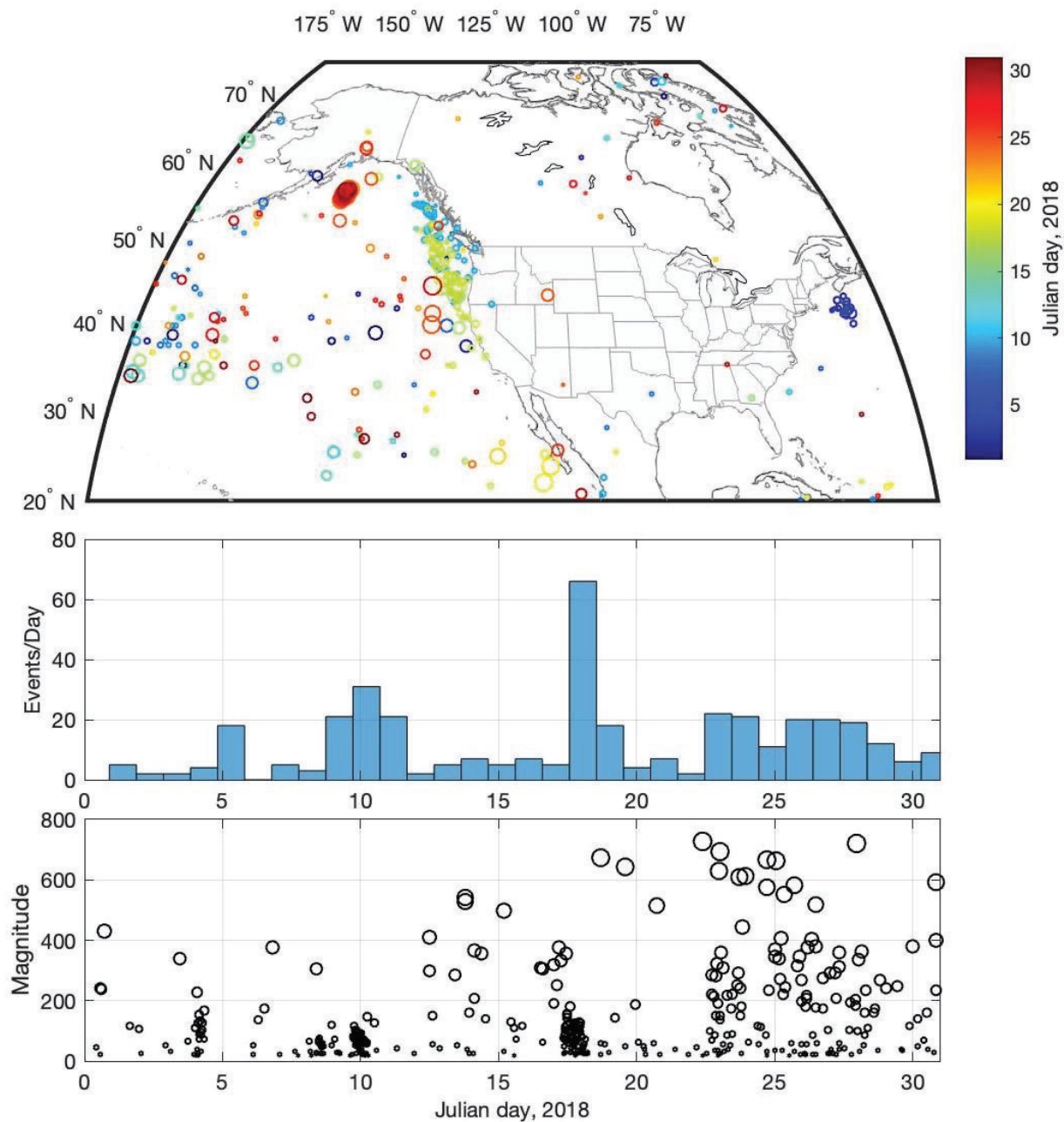


Figure 13. Map of events that generated surface waves that were detected by AELUMA. *In the top and bottom panels symbol sizes scale with the number of triads that detected the event. The events are generated both by storm activity and seismicity. The middle panel shows the number of surface wave generating events detected by day. The lowest panel shows the number of triads that detected each event.*

9. SPARSE NETWORK APPROACH

We had proposed using a standard STA/LTA approach of identifying signals at very sparse networks if very large triads proved ineffective at detecting signals. However, this proved unnecessary, at least for large signals. An initial test of the STA/LTA approach, which provides only arrival times and not signal azimuths, proved ineffective as far too many false detections were identified.

Instead, we investigated the use of polarization analysis at three components (3C) stations to identify the propagation azimuth and arrival time of signals; we considered that this might be useful at lower signal strengths, and thus effective for finding smaller events. A polarization analysis determines the particle motion for short time segments at each 3C station. As explained in Jurkevics (1988) the waveform covariance matrix is computed to determine the polarization ellipse, which describes the particle motion. From this the rectilinearity is estimated, which indicates the degree to which pure body wave motion dominates; the planarity indicates the degree to which the particle motion is consistent with a surface wave. The strike direction is in the direction of signal propagation for a P-wave and perpendicular to it for an S wave. That is, only the P-wave strike direction is a reliable measure of the signal propagation azimuth.

An example of a polarization analysis is shown in Figure 14, taken from de Groot-Hedlin (2020) for an Mb 6.4 event in the Caribbean recorded at USArray stations near the east coast of the United States. The polarization analysis was computed for 4 s windows for 1-3 Hz bandpassed data. The left column shows the results for a station approximately 3090 km from the event; the right column shows the results for a station about 2340 km from the event. Both columns show waveforms that include a P-wave, followed by an S-wave, followed by a surface wave. The direct source-receiver path, i.e. the signal azimuth, is shown by the dashed line in the lowest panels; the dash-dotted lines show the transverse direction. Even for this strong event, the polarization analysis does not divide arrivals into either purely rectilinear or planar motion, given the presence of noise. The azimuths for the body waves show a very high degree of scatter; the strike direction is consistent with the direct source-receiver path only for the P-wave arrival onset, as shown in the bottom panel. We conclude that a polarization analysis would be ineffective at determining signal azimuth except immediately at the signal onset (for the P-wave) and only then if the SNR was high enough; exactly the scenario that does not exist for smaller events.

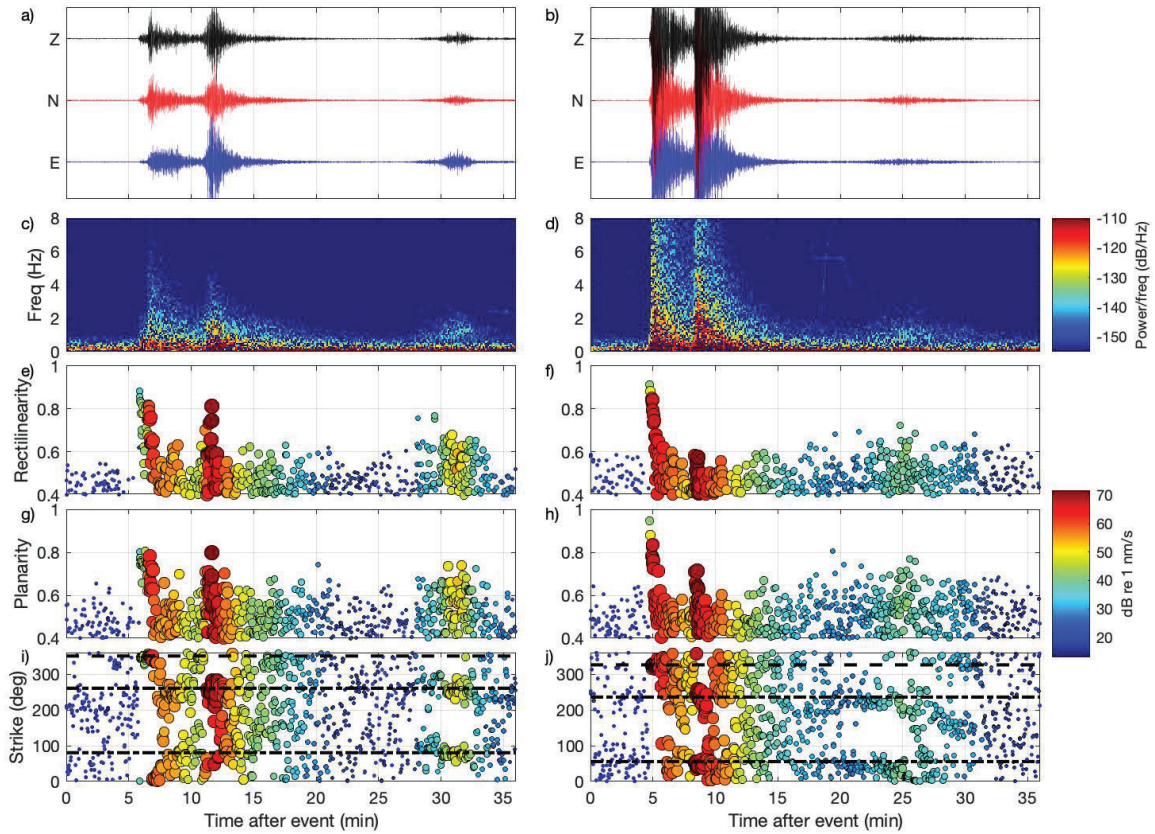


Figure 14. Left column shows the polarization analysis for a station about 3090 km from its Caribbean source. The right column shows the corresponding analysis for a station about 2340 km from the event. (a,b) 1–3 Hz filtered data for 36 min after the event; (c,d) spectrograms of the unfiltered Z component; (e,f) rectilinearity versus time. The marker sizes and color scale with the amplitude of the waveforms above. (g,h) Planarity versus time; (i,j) strike direction versus time. The dashed lines show the azimuth from source to receiver. The dashed-dotted lines show the transverse directions, that is 90° from the direct path.

10. APPLICATION TO IMS DATA

As mentioned above we assembled a dataset from one month (from mid-July to mid-August, 2017). We chose this time period because seismicity ranged from infrequent during much of the time period to very frequent during a protracted aftershock sequence near Turkey. In this time period the NEIC catalog contained 109 events ranging in magnitude from 3.8 to 6.6, with most smaller than magnitude 4.5. The sources occurred from 30°N to 75°N , 0°E to 85°E (see appendix H). The events are shown in Figure 15. In that figure we see that many of the events were located near Turkey and occurred early in the month. Most events were distributed across a broad range of longitudes largely between latitudes of 30 – 50°N . As with the test in Alaska, discerning individual

events in the aftershock sequence near Turkey using a sparse network is relatively difficult given the long time windows that are required. Although long windows are indeed needed when processing data from the large triads in the IMS, this is not the case when using the array data from the IMS as coherent processing of data in short time windows is possible.

Given the sparsity of the IMS, very long time windows (1,440 seconds) were used to analyze data from the triads. Much shorter time windows were used to analyze the array data (30 seconds). These parameters were chosen after an extensive series of tests. The user manual (Appendices E and F) discusses the rationale for choosing these and all other parameters. For this reason we expect that forming event locations during the aftershock sequence will rely mainly on array data.

The maximum triad side length (r_{tmx}) for this analysis was 1580 km. The minimum side length (r_{min}) was 30 km. The minimum interior angle was 10°. The IMS stations were triangulated with these parameters. For this analysis we chose one randomly chosen station from each array in addition to all single stations. Figure 16 shows the map of stations used for our test.

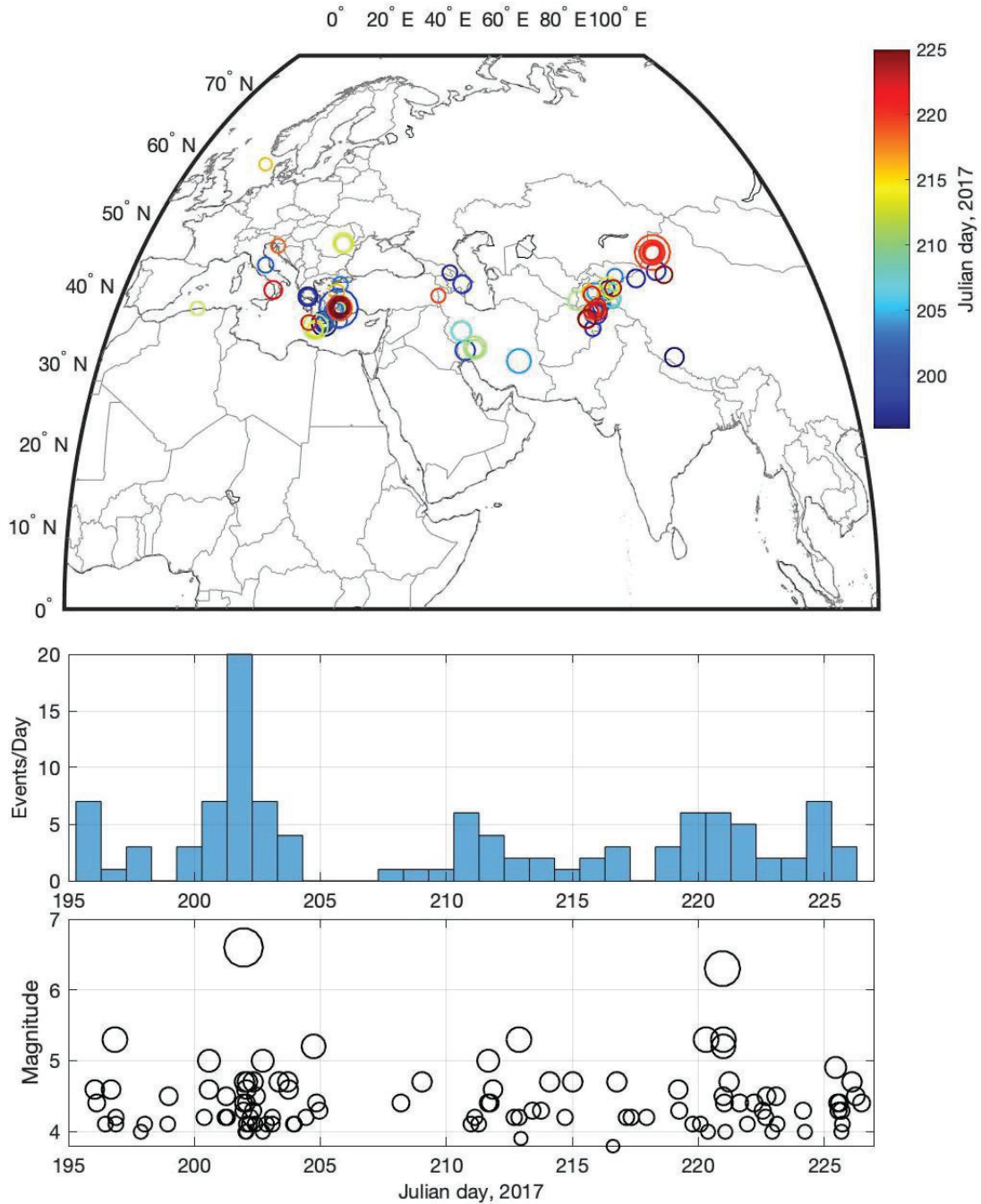


Figure 15. Events in the NEIC catalog during the month from mid-July to mid-August, 2017. *This time period included a large earthquake and aftershock sequence off the shore of Turkey. The symbols on the map represent these events and are color-coded by date. The middle panel gives the distribution of events in time. The sequence of events in Turkey happened just after JD 200. The lower panel gives the event magnitude as a function of time.*

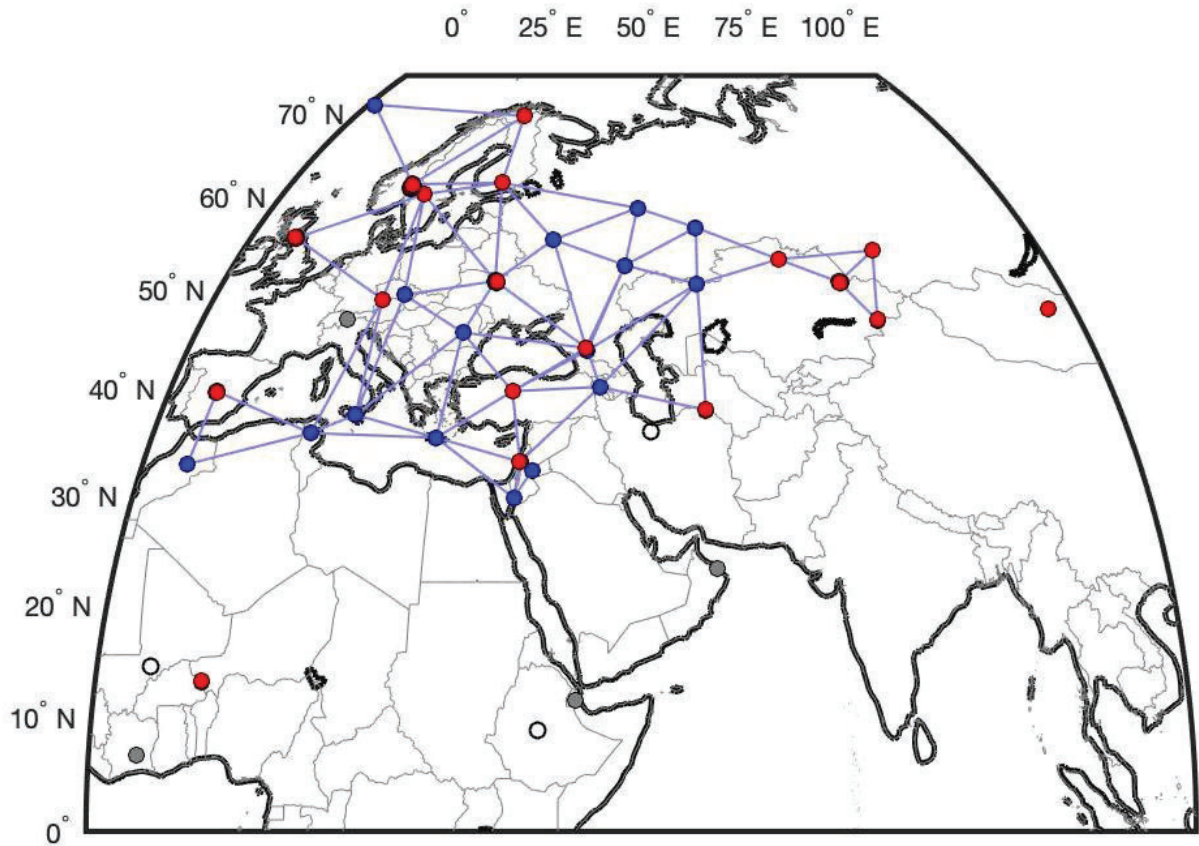


Figure 16. Triangulation of the stations in the IMS network that were used in our analysis. *Red circles represent arrays, blue circles single stations, empty circles represent stations from which there was no data. Grey circles represent stations that were unusable due to the station being located too far from other stations to be incorporated in a triad. The light-blue lines are triads.*

Results are illustrated by highlighting detections and formed events in two average days. We present detections made by AELUMA on these days, the event locations that were formed using the detections, record sections of sample events that were cataloged by AELUMA, NEIC or both. Although the focus is on these two days to illustrate the performance of the algorithm, results from all days from this month-long period are included with this report. Refer to Appendix I for the entire AELUMA catalog.

Results for day 219.

Detections:

The first day, JD 219 (August 7), occurred shortly before the sequence of events in Turkey. On this day three earthquakes were listed in the NEIC catalog ranging in magnitude from 4.1 to 4.6. Figure 17 shows the distribution of detected signals on this day as a function of envelope amplitude and time of day. Most signals were detected by arrays - a small minority by triads. As mentioned earlier the time window used for

processing of data from triads was very long (1,440 seconds). We expected this would reduce the utility of triads for detecting/locating events during the aftershock sequence. The utility of triads at other times (relative to the arrays) was unknown). As seen in Figure 17 the vast majority of signal detections came from the IMS arrays and not the triads. In general the shorter time windows used on array data makes it easier to correlate data to find signals for two reasons: (1) for shorter windows, each offset by 25% of the window length, you're combing through the data in greater detail than with long windows and thus it should not be surprising that more signals are detected and (2) cross-correlations of long time series of noisy data are less likely to be coherent, thus only larger signals will be detected. Figure 17 buttresses this latter argument.

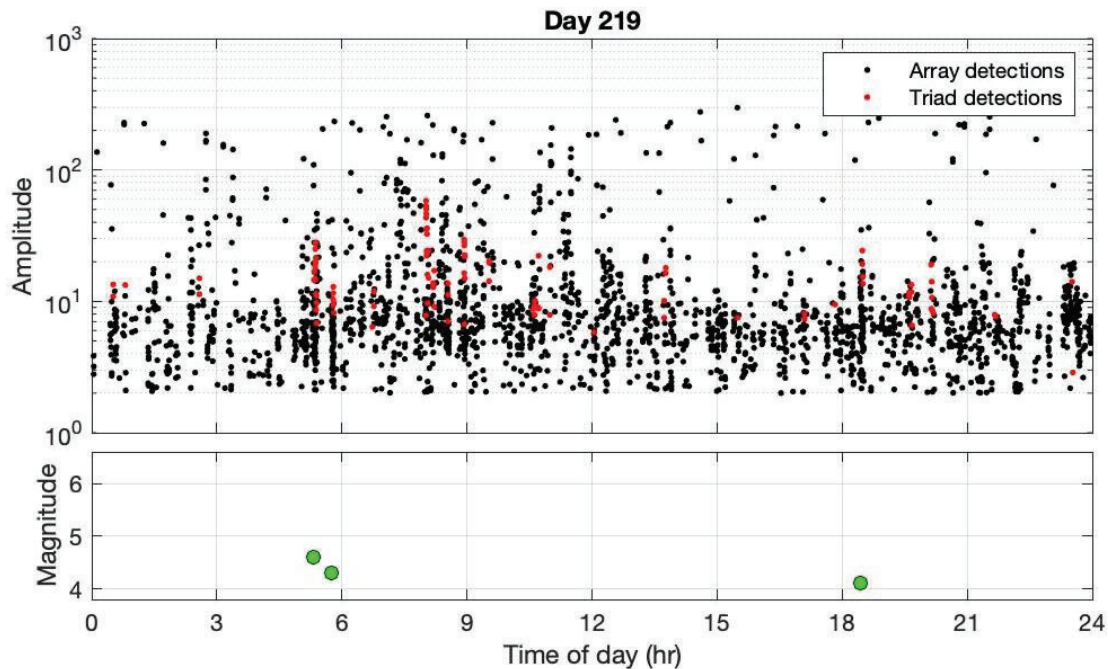


Figure 17. Detections made by stations in the IMS. *The upper panel shows the amplitude of each detection versus the signal's time of arrival for triad detections (red) and arrays (black). The bulk of detections were made by arrays. The lower panel shows the magnitude/time of events in the NEIC catalog that day. Compare this result to figures 3 and 6.*

Event associations:

Figure 18 shows the events in the NEIC and AELUMA catalogs on this day. All three NEIC events were found by AELUMA. AELUMA found an additional 4 events. Figure 19 shows one of the events found by both NEIC and AELUMA; it is notable that it detects a signal even in the presence of very noisy data.

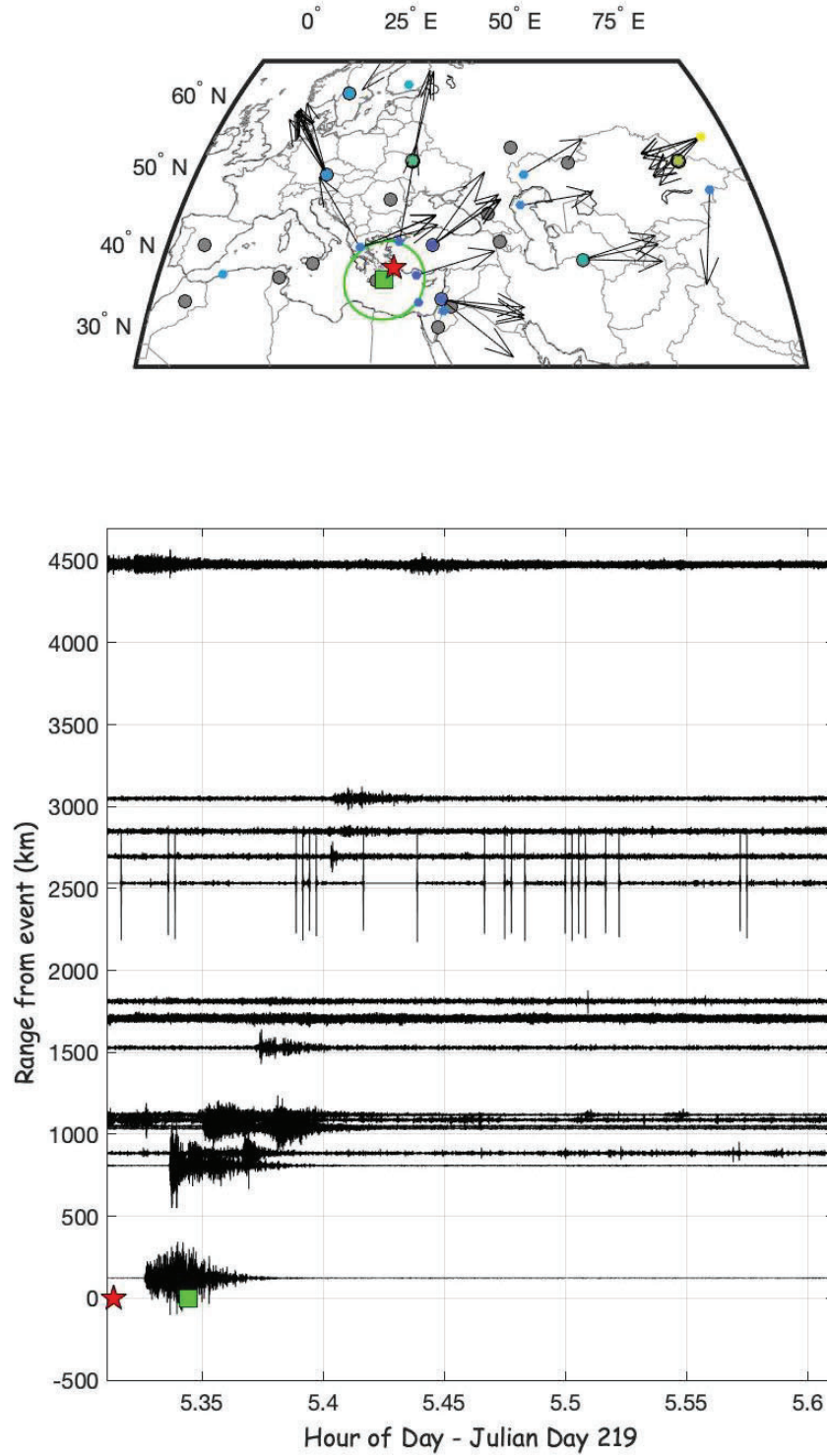


Figure 19. An event in both catalogs. *The upper panel shows detected signals at both triads and arrays. The arrows represent the direction of signal propagation. In the record section the data were bandpassed from 0.5-7.8 Hz. The red star represents the origin time from the NEIC catalog. The green square is the OT from AELUMA.*

Typically AELUMA OTs are late due to the fact that the code uses the time of peak energy and not signal onset time.

Figure 19 gives an example of an event found accurately despite noisy data. It was found by both NEIC and AELUMA. In the top panel of Figure 19 the red star represents the NEIC event location. The green square represents the location produced by AELUMA. The green ellipse is the estimate of location error (95% confidence ellipse). The colored circles show where all (unsorted) signals were detected within the 18 minute time period shown above, with arrows showing the apparent direction of signal propagation. In this case there were enough consistent signal directions and arrival times to define an event location for AELUMA. Gray circles mark IMS stations and arrays at which AELUMA detected the event. In the bottom panel are the waveforms for IMS stations or arrays at which AELUMA detected this event, plotted as a function of distance from the AELUMA source location. The NEIC event origin time is shown by the red star; the AELUMA estimate of the origin time is shown by the green square. AELUMA successfully detected and located the event despite elevated noise at some stations.

Results for day 226.

Detections:

As before detections are displayed as a function of time of day in the upper panel of Figure 20. Again, on this day, there were three events in the NEIC catalog. Those event times correlate with an increase in the number of low-amplitude signal detections. There are several other times when the number of signals detected by IMS stations increased - such as at the very start of the day. Overall the multifrequency AELUMA code yielded a significant number of detections that day. Again, most detections came from the IMS arrays and not the triads.

Event associations:

Figure 21 shows a map view of events in the NEIC and AELUMA catalogs. The lower panel shows the time and magnitude of all three events in the NEIC catalog that day along with all detections in the AELUMA catalog. The three events in the NEIC catalog ranged in magnitude from 4.4 to 4.7. Two of the NEIC events were detected. The mid-sized, magnitude 4.5, event was not. AELUMA located a total of 6 events – 4 of which were not in the NEIC catalog.

The missed event is shown in Figure 22; this is the highest magnitude event that was missed by AELUMA. The NEIC origin time is represented by the red star. Despite the large magnitude of the event the signals from this 96 km deep event are not particularly large in the band chosen for this figure (0.5 to 7.8 Hz). Likewise, not enough signals were detected in the four narrower frequency bands to form an event location.

The circles mark IMS stations and arrays within 3500 km of the event. The colored circles show where all (unsorted) signals were detected within the 18 minute time period shown above, with arrows showing the apparent direction of signal propagation. There were insufficiently consistent signal directions and arrival times to define an event

location for AELUMA. In the bottom panel of Figure 22 we show the waveforms for IMS stations within 3500 km of the NEIC event, plotted as a function of distance from the event. The event origin time is shown by the red star.

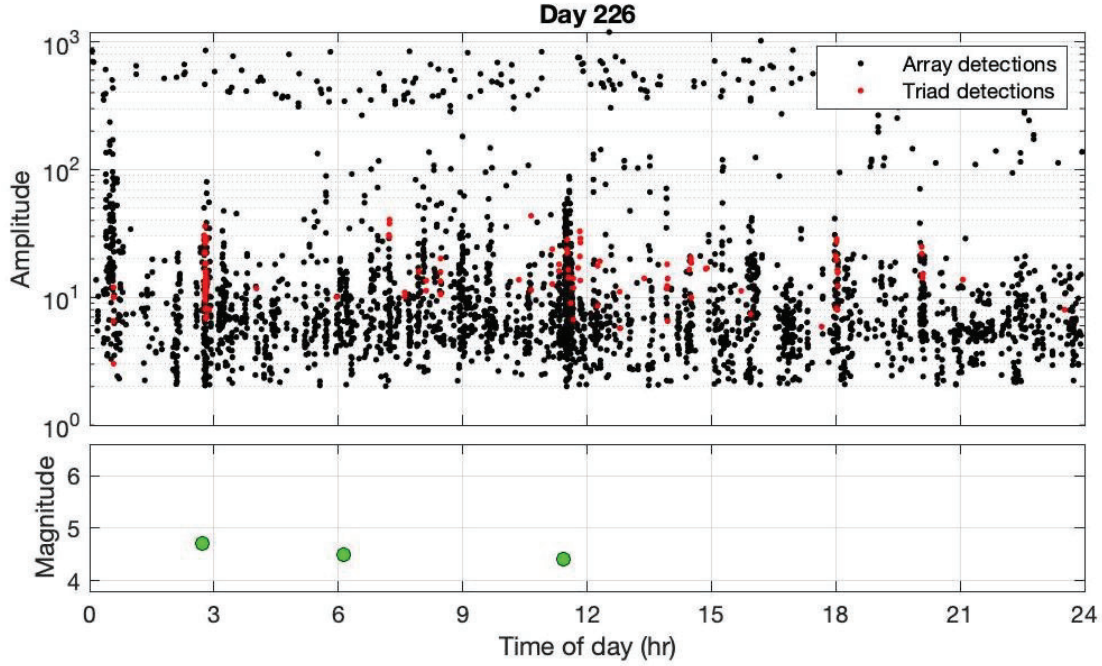


Figure 20. All detections by arrays and triads in the IMS are represented by black and red dots respectively. *Most detected signals are relatively low in amplitude. The lower panel shows NEIC events and by comparison with the upper panel shows marked increases in the number of detections at a number of times including at the time of the three events in the NEIC catalog. Compare to Figures 3, 6, and 17.*

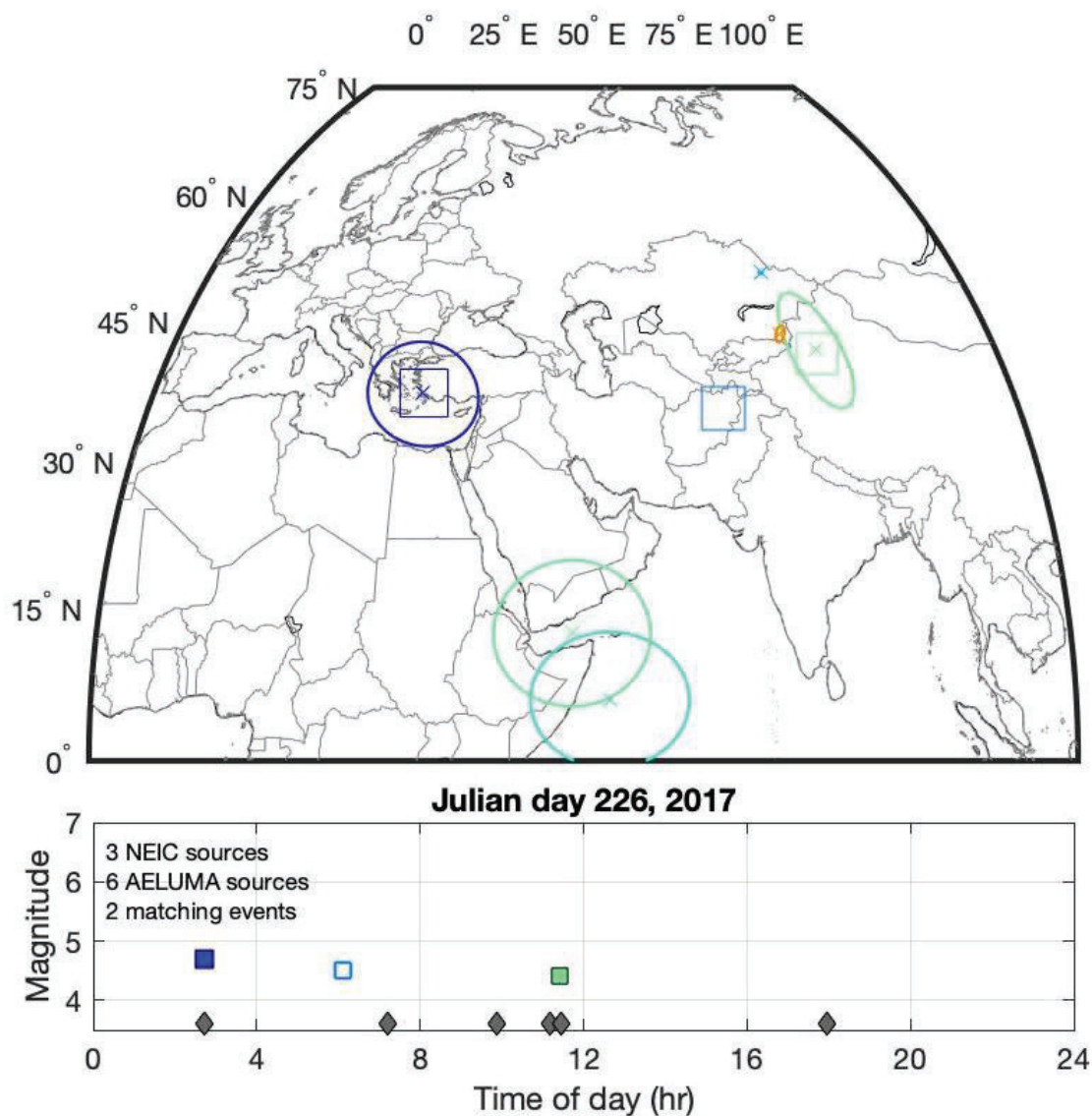


Figure 21. This figure compares AELUMA event detections vs NEIC event detections for day 226 2017. See figure 5 for captions. This day two of three NEIC events were matched by events in the AELUMA catalog. Four AELUMA events were not found in the NEIC catalog.

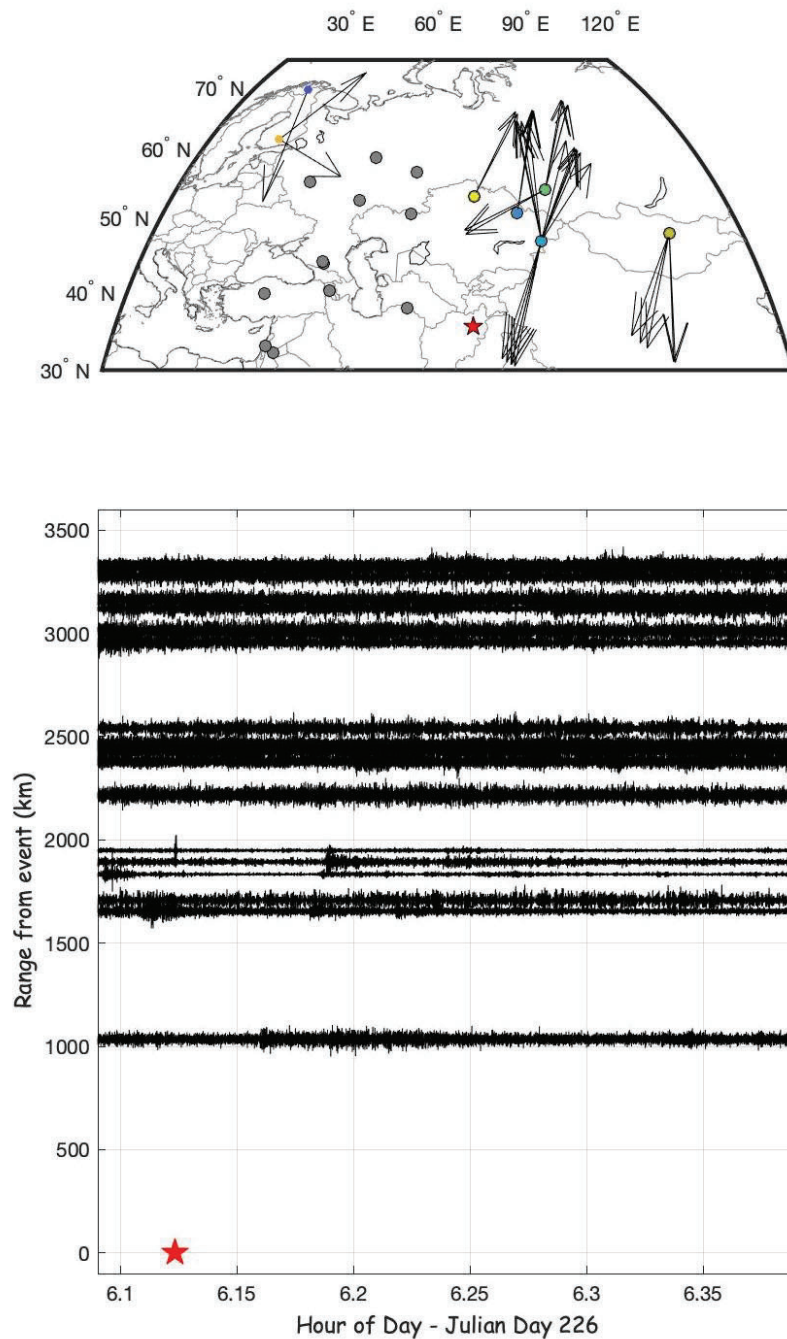


Figure 22. The largest magnitude NEIC event that was NOT detected by AELUMA. *This is a 4.5 mb event at 35.5N, 69.3 E. The upper panel shows the NEIC source location (red star) and the detections registered by AELUMA. The low SNR data yielded inconsistent detections and an event was not formed. The timeseries shown here were bandpassed across the entire band of interest (from 0.5 to 7.8 Hz).*

AELUMA did not find the 4.5 mb event in the figure above. But it did find the mb 4.4 event below.

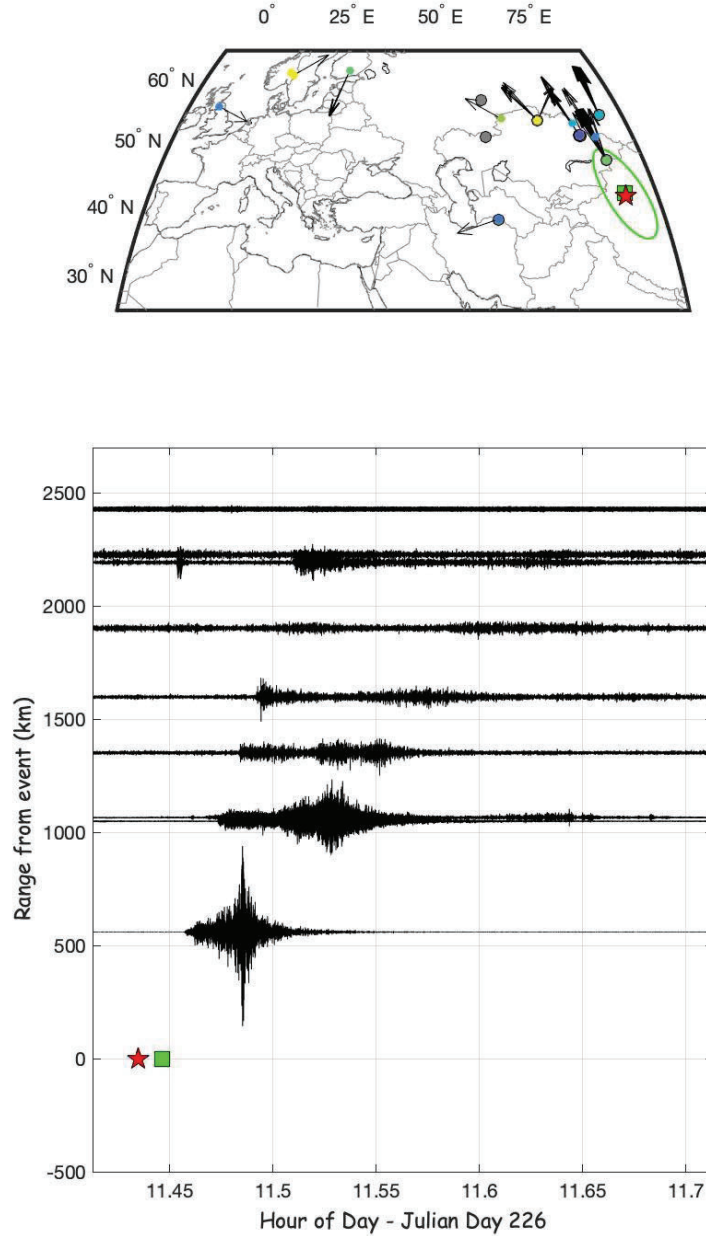


Figure 23. Example of mb 4.4 event found by both NEIC and AELUMA. In the top panel the red star represents the NEIC event location. The green square is the location from AELUMA location. The green ellipse is the estimate of location error (95% confidence ellipse). The colored circles show where all (unsorted) signals were detected within the 18 minute time period shown above, with arrows showing the apparent direction of signal propagation. In this case there were enough consistent signal directions and arrival times to define an event location for AELUMA. Circles mark IMS stations and arrays at which AELUMA detected the event. In the bottom record section the waveforms for IMS stations or arrays at which AELUMA detected this event, plotted as a function of distance from the AELUMA source location. The NEIC event origin time is shown by the red star; the AELUMA estimate of the origin time is shown by the green square.

Finally, in Figure 24 we show an example of an event detected ONLY by AELUMA, not NEIC.

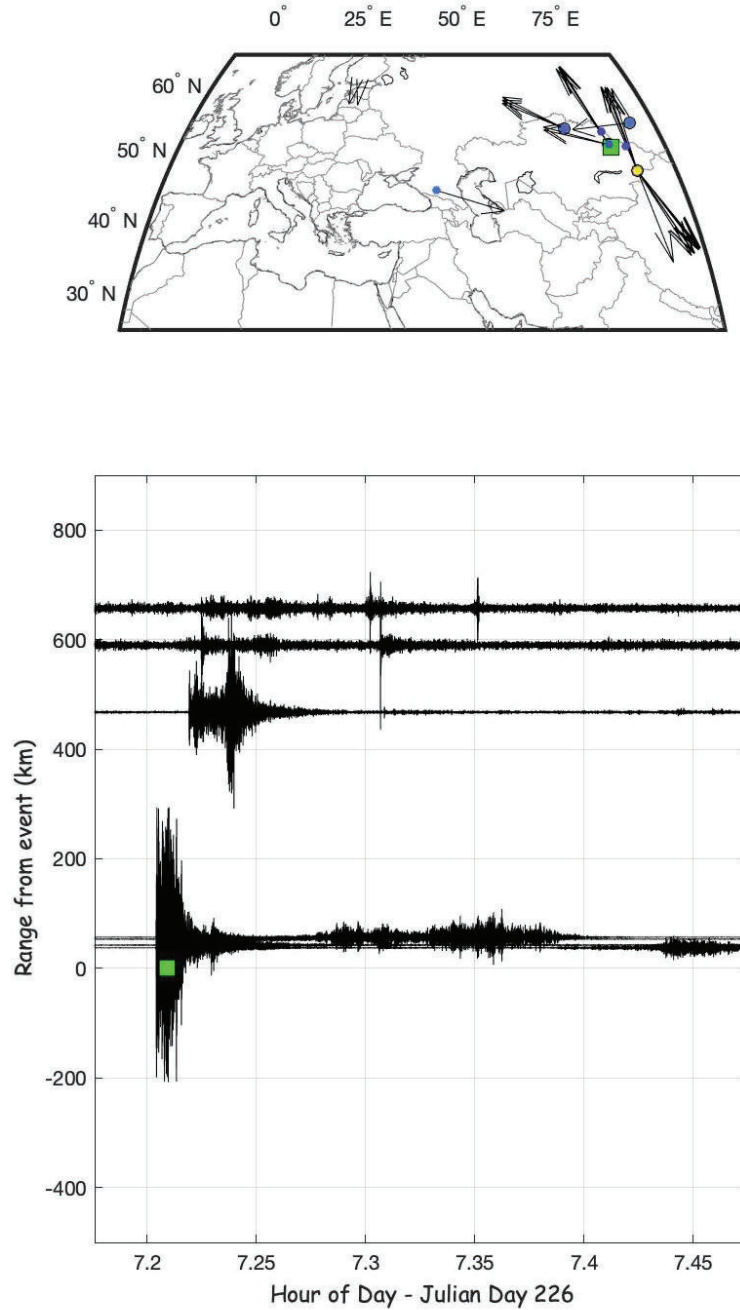


Figure 24. See caption for figure above (except no red stars since it was not detected by NEIC).

Overall results:

The catalogs (NEIC and AELUMA) are included (see Appendices H and I). We present some summary information here. Maps of events in the NEIC and AELUMA catalogs are shown in Figure 25. In all panels of this figure AELUMA located events are shown in red, NEIC events are shown in black. The upper, middle and lower panels show matched events (that is events in both catalogs), events just in the AELUMA catalog and events in the NEIC catalog respectively. Although we are encouraged by clustering of the AELUMA-only events shown in the middle panel the dispersed cloud does not accurately represent real seismicity. This highlights an issue that still needs to be resolved with the AELUMA code. Many of the unmatched events in the AELUMA catalog might have been poorly located due to low SNR and/or the low number of triads used in the solution. Due to the small number of stations available we allowed an unusually low number of triads and arrays when forming events (a combined minimum of 5). Although the IMS arrays significantly improved our solutions some error was introduced into the source location estimates by using so few triads – particularly when signals were immersed in noise in all frequency bands.

Some of the events plotted in the middle panel might have been the result of the chance correlation of unrelated noise bursts. This issue becomes more likely when a smaller number of triads is used when forming event locations. Still, even when using few triads, the detected signals are subjected to a number of stringent criteria (Appendices B-D) and we believe this is relatively uncommon. Nonetheless to date this remains an issue that will require further work. We discuss possible remedies in chapter 10. An approach using polarization analysis has not improved processing at the low event magnitude at which the problem is most serious (Chapter 8).

Figures 26 and 27 show histograms of events detected by AELUMA vs in the NEIC catalog as a function of event magnitude and date (respectively). The AELUMA catalog has all of the events magnitude 4.5 and above. The percentage of events matched gradually decreases with decreasing magnitude.

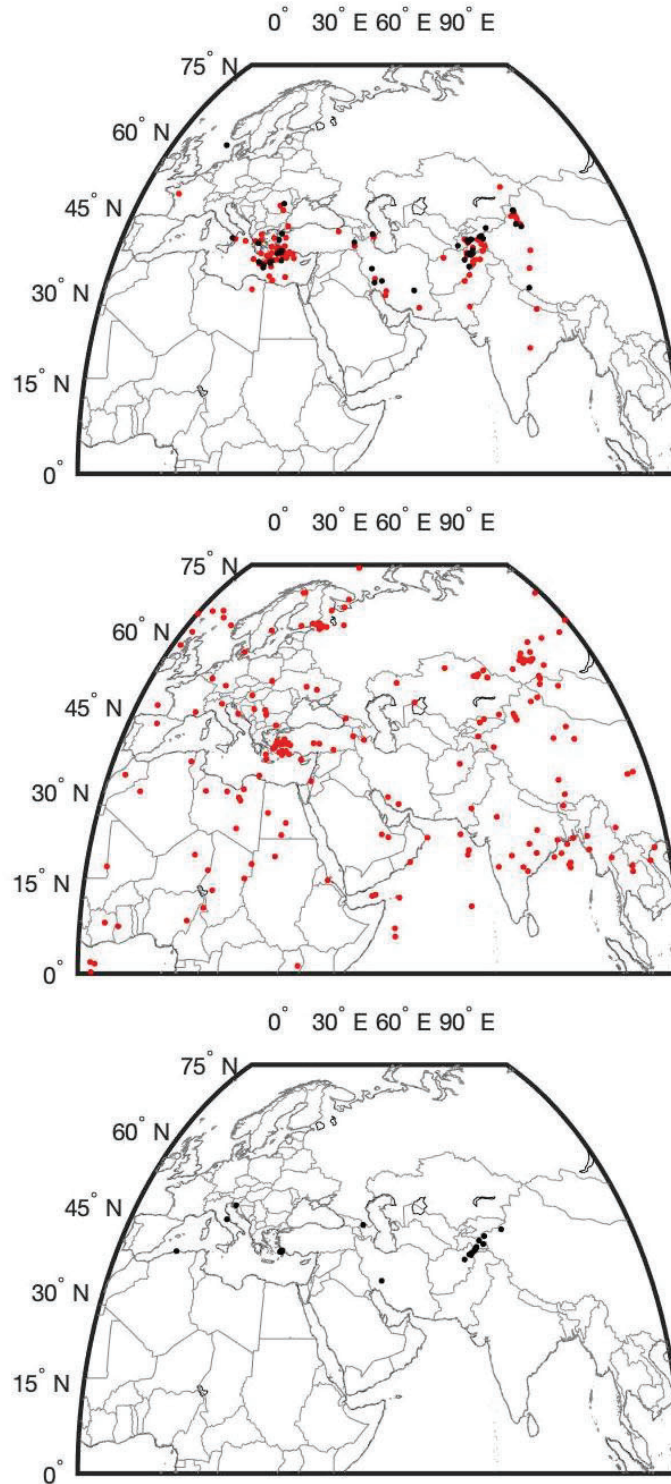


Figure 25. Top) Events in both the NEIC and AELUMA catalogs for the month of this study (black and red respectively), middle) events just in the AELUMA catalog, bottom) events just in the NEIC catalog. *The clustering of events in the middle panel is encouraging. The dispersed cloud of events is clearly not representative of the real distribution of seismic events. The issue is discussed below.*

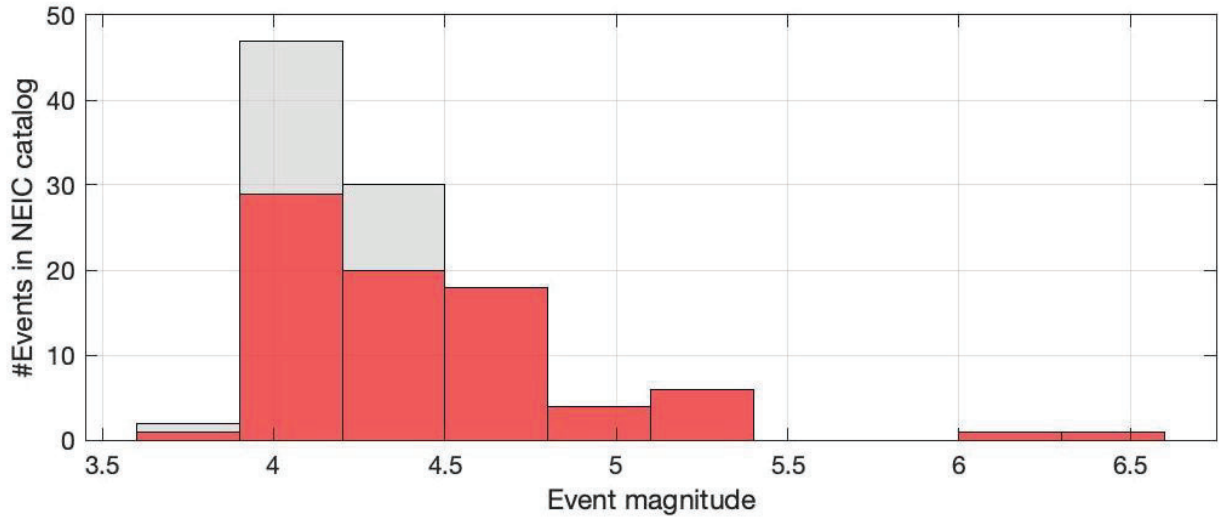


Figure 26. Histogram of how well AELUMA does as a function of event magnitude. Events found as a function of magnitude over the whole period (mid-July to mid-August, 2017). Here we present results from the multi-band AELUMA code, using both triad detections and array detections. We used five frequency bands. We used the standard clustering method described in the appendix. The capture rate above $M4.5$ is perfect.

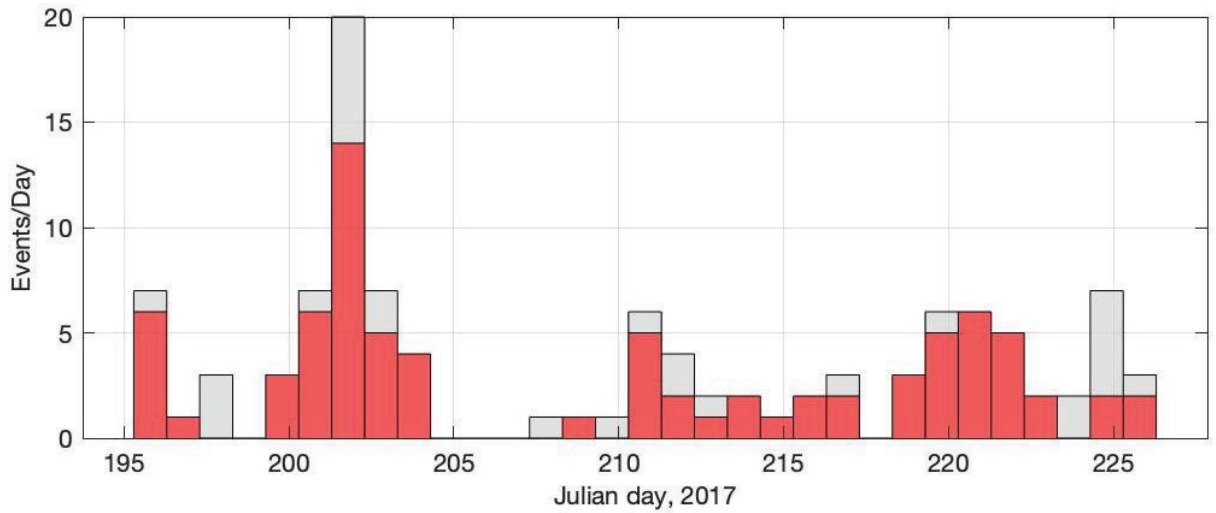


Figure 27. Gray histogram gives the number of events each day in NEIC catalog within the area of interest. Red histogram shows the ones that were also detected by AELUMA. The gray histogram is identical to the middle panel of figure 14). There were 109 events in the NEIC database. AELUMA cataloged 280 events. The number of AELUMA events with matching NEIC events was 85.

11. RESULTS AND DISCUSSION

We've presented the results of modifying the AELUMA code for application to sparse, arbitrarily large networks that include single sensors and arrays. Here we discuss various issues identified during the work. We discuss both areas in which the method appeared to work as well as expected or better, as well as where the code underperformed, and suggest various remedies. Finally we make a few general comments about the method and its application to IMS data.

What worked well?

Signal detection using the multifrequency code: The multifrequency version of AELUMA detected far more signals than the pre-existing single-band code. In fact there were so many signals detected that this actually created some problems downstream in the event association part of the code (see below). This problem has not been fully resolved (see below).

Inclusion of arrays. As discussed in the appendices a modified detection approach was used to incorporate dense array data into the signal detection module of AELUMA. These arrays have significantly improved the detection of signals. In fact the arrays in the IMS far outperformed the single stations. This was in part due to the sheer number of data from the arrays. It was also largely due to the use of shorter windows that can be used when processing array data (due to their relatively small aperture). The very large triads required windows of significant length (> 1400 seconds) that made finding and separating individual events more difficult.

Where did AELUMA underperform?

Event association at small magnitudes. An important issue that to date has not been fully resolved is event association at small magnitudes. Ultimately we believe AELUMA should perform far better than it has at magnitudes below 4.5 – even when the network of stations is very sparse. As noted earlier we did not detect all of the events in the NEIC catalog. There are various possible explanations of this.

- 1) Plethora of signal detections. As mentioned above the signal detection module of AELUMA has been modified significantly during this project and has made it difficult for the event association mode to function properly. Our approach has found it difficult to separate out subsets of these signals from such a large population.
- 2) Accuracy of error estimates. If the location error is underestimated, an event in the NEIC catalog could be missed. Also, restricting ourselves to automatic processing of just IMS data, or just data available to us in North America via the DMC portal may put us at a disadvantage. As stated at the USGS website “The NEIC collects data through the operation of national and global networks, and through cooperative agreements”. We do not know which stations were used to detect and locate individual events in the NEIC catalog however we expect that it is common for substantially more stations to be involved in building the NEIC

catalog. We also expect that analysts work to ensure events do not go undetected. We have various ideas for improving the code to reduce the number of missed events. These are discussed below.

- 3) Accuracy of event locations outside the network. This is discussed further in Appendix D. When the code is applied to the full global network this issue would presumably disappear.

The issue of extra events, i.e. ‘false alarms’ in the AELUMA catalog is potentially important. That is, an event in the AELUMA catalog may not be real - for example, due to the chance correlation of unrelated noise bursts. This is more likely when the minimum number of triads required to define an event is small. This occurs when the network is very sparse; for example, we allowed a minimum of 5 triads when analyzing the IMS data.

We believe real progress on this issue will require training of the algorithm (see below).

Possible remedies

Use of machine learning. Having processed a large amount of seismic data we see potential in a trainable algorithm. The current algorithm first finds events with many associated detections with celerities between upper and lower celerity limits - we used 2.8-7 km/sec to encompass both S and P wave arrivals - then progressively finds events with fewer and fewer detections. Currently only the arrival times and azimuths associated with each detection are used, although each detection carries more information, including amplitude, phase velocities, mean cross-correlation values, etc. Figure 28 shows some attributes of the detections for a large event found by AELUMA. The top panel shows the initial OT (blue star) and celerity limits (blue lines) for the coarse solution. The red line shows how the OT is found for the finer solution (the NEIC OT lies between these 2 values). The lower panel shows how the amplitude (the SNR values) changes with distance from the source.

Since event source parameters (time and location) are fairly robust for events with a large number of detections, machine learning techniques could be used to find other events that closely match these attributes. That is, it could be seen from Figure 28 that there are two branches for the arrival times corresponding to P and S branches. The algorithm could be used to ‘learn’ a 1D velocity model which would then be used to find further events with detections along these 2 branches (Alternatively, a known velocity model could be input to a future version of AELUMA). Giving AELUMA the correct 1D velocity model would help it hone in on signal detections along these arrival time branches. Other characteristics of the detections could be used to improve event associations. As shown in Figure 28, there is more scatter in the amplitudes, yet amplitudes could be used to eliminate outliers in some cases.

Although these ideas are worth pursuing, they are beyond the scope of this project.

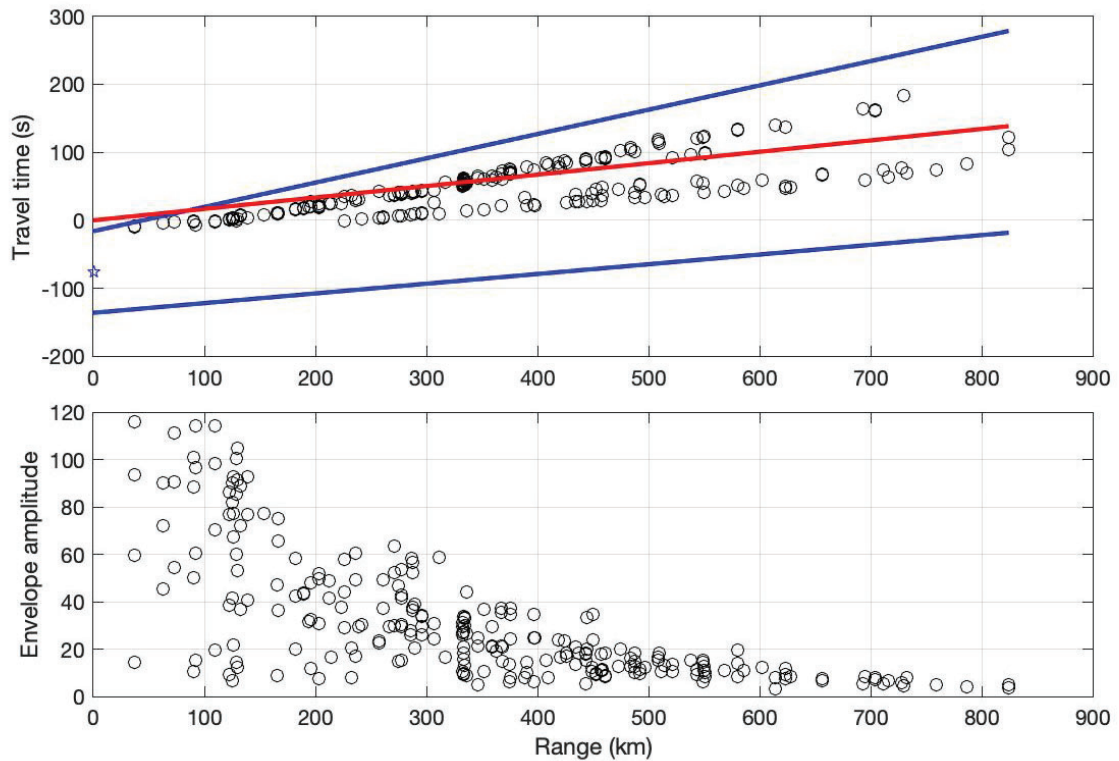


Figure 28. Signals recorded from one event by stations in the IMS. *In the upper and lower panels we show signal travel times and amplitudes as a function of range from the event respectively. The upper panel shows the broad corridor within which AELUMA searches for signals. The single corridor approach has proven to work well in media, such as the atmosphere, in which there is one propagation velocity. In the earth, of course, the separation in time between compressional and shear waves increases with increasing distance from the source. The red curve shows the average travel time curve that best fits all detected signals. A future version of the method could perform better when applied to seismic data by accommodating compressional and shear waves separately. An attenuation model that estimates drop-off of signal amplitude with range could make the code less likely to form event locations from unrelated noise bursts.*

General comments

Comparison with NEIC Despite this we observed that AELUMA did find a significant number of events that were not in the NEIC catalog – although this isn’t necessarily due to the AELUMA method performing well. There are various possible explanations of this. First, some events in the AELUMA catalog are likely due to the chance combination of unrelated noise bursts (see below). Some events in the AELUMA catalog should have matched an event in the NEIC catalog but didn’t due to an error in the AELUMA location or us underestimating the location error. Also, some events were not included in the NEIC catalog. We understand that some events (e.g. mining blasts) are excluded from the catalog.

Why the method works for events with complex signals. Given that the method relies entirely on cross-correlations of signal envelopes between adjacent stations, the signal can be somewhat complex. Even for a complex source, most stations would still record signals that are similar, but time-lagged with respect to each other (and would have high SNR). So the cross-correlations to determine relative time lags and the subsequent *tau-p* solution to determine source azimuth would be accurate. We regard that as a strength of this method - that neither the nature of the source nor the precise velocity model are needed for cross-correlations - since stations within a triad (or array in the case of the IMS) are close enough to record time-lagged versions of the signal. This could break down a bit once the inter-station spacing within a triad increases although our analysis suggests that networks with station separations up to ~ 600 km can still be processed using AELUMA.

Once the azimuths and arrival times are determined at each triad the detections are sorted into groups, each associated with a single event - in this, it is assumed each event is a point source. Since the stations cover a very broad range from the source, a point source is an adequate approximation even for large events. This likely would break down for a mega-event such as the 2004 Sumatra-Andaman event that ruptured over a 1000+ km range but works well enough for our purposes.

REFERENCES

- Arrowsmith, S.J., Whitaker, R., Katz, C., and Hayward, C., 2009, The F-Detector Revisited: An Improved Strategy for Signal Detection at Seismic and Infrasound Arrays, *Bulletin of the Seismological Society of America*; 99, DOI: <https://doi.org/10.1785/0120080180>
- Blandford, R. R., 1974, An automatic event detector at the Tonto Forest Seismic Observatory, *Geophysics*, 39, pp. 633–643.
- de Groot-Hedlin, C. D., 2020, Seismic T-Wave Observations at Dense Seismic Networks, *Seismol. Res. Lett.*, 91, pp. 3444–3453, DOI: 10.1785/0220200208.
- de Groot-Hedlin, C.D., Hedlin, M.A.H., and Walker, K.T., 2014, Detection of gravity waves across the USArray: A case study, *Earth and Planetary Sciences Letters*, DOI: 10.1016/j.epsl.2013.06.042.
- de Groot-Hedlin, C.D. and Hedlin, M.A.H., 2015, A method for detecting and locating geophysical events using groups of arrays, *Geophys. J. Int.*, 203, pp. 960-971, DOI: 10.1093/gji/ggv345.
- de Groot-Hedlin, C. D. and Hedlin, M. A., 2018, A new automated approach to detecting and locating seismic events using data from a large network, *Bull. Seism. Soc. Am.*, DOI: 10.1785/0120180072.
- de Groot-Hedlin C and Hedlin M., 2019, Detection of infrasound signals and sources using a dense seismic network. In: Le Pichon A, Blanc E, Hauchecorne A (eds) *Infrasound monitoring for atmospheric studies*, 2nd edn. Springer, Dordrecht, pp. 669–699.
- de Groot-Hedlin, C. D., and Vernon, F. L., 1998, An evolutionary programming method for estimating layered velocity structure, *Bull. Seismol. Soc. Am.*, 88, pp. 1023–1035.
- Fan, W., de Groot-Hedlin, C.D., Hedlin, M.A.H., and Zhitu, M., 2018, Using surface waves recorded by a large mesh of three-element arrays to detect and locate disparate seismic sources, *Geophys. J. Int.*, 215, pp. 942-958, <https://doi.org/10.1093/gji/ggy316>.
- Fan, W., McGuire, J.J., de Groot-Hedlin, C.D., Hedlin, M.A.H., Coats, S., and Fiedler, J.W., 2019, Stormquakes, *Geophysical Research Letters*, DOI: 10.1029/2019GL084217
- Gibbons, S. J., Ringdal, F., and Kvaerna, T., 2008, Detection and characterization of seismic phases using continuous spectral estimation on incoherent and partially coherent arrays, *Geophys. J. Int.*, 172, pp. 405–421, DOI: 10.1111/j.1365-246X.2007.03650.x.
- Havskov J., and Ottemöller L., 2010, Array Processing. In: *Routine Data Processing in Earthquake Seismology*, Springer, Dordrecht, https://doi.org/10.1007/978-90-481-8697-6_9
- Saalfeld, A., 1999, Delaunay Triangulations and Stereographic Projections, *Cartography and Geographic Information Science*, 26:4, pp. 289-296, DOI: 10.1559/152304099782294168
- Schweitzer, J., Fyen, J., Mykkeltveit, S., Gibbons, S. J., Pirli, M., Kühn, D., and Kværna, T., 2012, Seismic Arrays - In: Bormann, P. (Ed.), *New Manual of Seismological*

Observatory Practice 2 (NMSOP-2), Potsdam: Deutsches GeoForschungsZentrum GFZ, pp. 1-80, https://doi.org/10.2312/GFZ.NMSOP-2_ch9.
Vasco, D. W., Johnson, L. R., and Majer, E. L., 1993, Ensemble inference in geophysical inverse problems, *Geophys. J. Int.*, 115, pp. 711–728, DOI: 10.1111/j.1365-246X.1993.tb01489.x.

APPENDICES: INTRODUCTION, DATA PROCESSING AT SPARSE NETWORKS AND ARRAYS

In these appendices, we describe methods to detect seismic signals crossing either sparse networks or dense arrays, then how to group these detections into clusters associated with a single event, and finally, to find accurate source parameters for each event, including the event time and epicentral location. The method of detecting signals at a sparse network, such that the signals of interest are uncorrelated from station to station, then using these detections to find source locations is called AELUMA, for Automated Event Location at a Mesh of Arrays. The application of AELUMA to infrasound signal detection and source location is described in some detail in de Groot-Hedlin and Hedlin (2015). Its extension to seismic signal detection and source location is described in de Groot-Hedlin and Hedlin (2018). Modifications for cases where the wavelengths of signals of interest are greater than the interstation spacing are described in de Groot-Hedlin et.al (2014) and Fan et. al. (2018). In this project, we have broadened the AELUMA method to include signal detections made at dense seismic arrays. The following appendices provide detailed explanations of each step of the methods.

The signal detection algorithms are outlined as follows. For a very sparse network, signals are uncorrelated from station to station so the network is broken up into many small sub-arrays of three stations, called triads. The triangulation of points on a sphere is described in Appendix A. Since the signals are uncorrelated, array processing is performed on envelopes of the data for each sensor within a triad to obtain signal azimuth, phase velocity, and signal arrival time. Array processing at each triad is described in Appendix B. Standard array processing may be done to find the signal azimuths and phase velocities at dense seismic arrays, because the seismic signals are correlated. The taup-p processing algorithm is described in appendix C, including a method to improve results for noisy data.

Signals that are detected either at sparse networks or dense arrays include information on the signal azimuths and arrival times, as well as supplementary information including phase velocity, mean cross-correlation values and amplitudes. The supplementary information is used primarily for quality control, ie. to decide whether the detections meet given cutoffs. The remaining signal azimuths and arrival times are used for event association, that is, to assemble the detections into groups, each consistent with a single source. The optimal source epicenter and onset time is then found for each group. The algorithms for event association method and optimal source location are described in Appendix D.

Some parameters that depend on the signals of interest and the network configurations must be set for both the signal detection and source location methods. The code documentation in Appendix E explains these parameters. Appendix F explains the output. Together, appendices E and F serve as the user manual for the code. The final appendices point to the source code (Appendix G), and the NEIC and AELUMA catalogs (Appendices H and I respectively).

Appendix A

DELAUNAY TRIANGULATION ON A SPHERE

There are many ways to triangulate any series of randomly scattered points. Signal azimuths derived through array analysis are most accurate at triads with nearly equal sides. Therefore, we use Delaunay triangulation since this algorithm maximizes the interior angle within the triangles. But standard Delaunay libraries, including the matlab routines used in this code, apply to points in a 2D plane rather than on a sphere. Given the size of the networks that the AELUMA code is applied to, the area cannot accurately be approximated as a flat surface. However, it has been shown that the Delaunay triangulation of points mapped onto the 2D plane using stereographic projection corresponds to the Delaunay triangulation of vertices on a sphere (Saalfeld, 1999).

To compute the Delaunay triangulation over a large region, we map the points representing the station locations onto a 2D plane using a stereographic projection. This mapping is conformal in that the angles between stations are preserved, although the distances and areas are not. An example is shown in Fig A1, for stations in the US and Canada with seismic data available online through the IRIS DMC for day 195 of 2017. We require the triads to be reasonably compact, so we culled the available stations to include only those within 200 km of at least two stations, but no less than 40 km. This left 387 stations for this day. Most of them are part of the USArray Transportable Array (TA) in Alaska or the Central and Eastern United States Network (CEUSN). Several dozen more along the west coast were available. The seismic station locations are mapped onto a 2D plane using a stereographic projection as shown in Fig. A1a and the Delaunay triangulation is applied to these points. The resulting Delaunay triangulation is shown on a Mollweide projection in Fig A1b. In this case, 510 triads are formed from the 387 stations, that is, each station can be included in several triads.

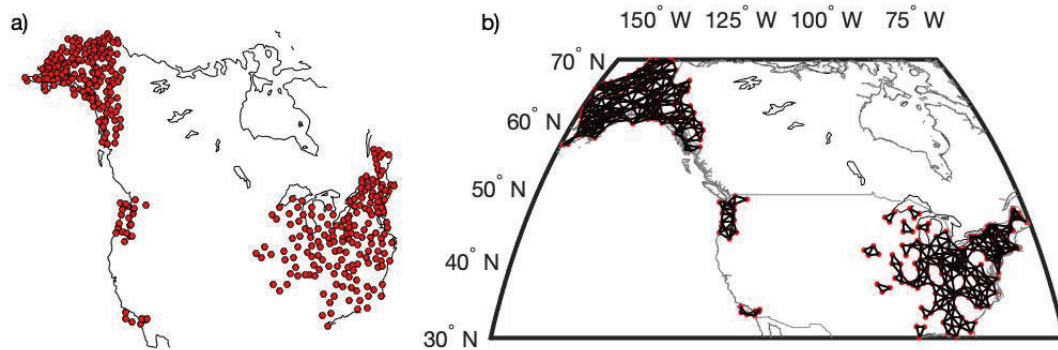


Figure A1. a) Stereographic projection of seismic stations for day 195 of 2017. The seismic station locations are shown in red. 387 stations were located within at least 200 km of two other stations, but no more than 40 km. b) The triads are shown by the black lines. 510 triads were formed from these stations.

APPENDIX B

TRIAD DETECTION CODE

The network is divided into a mesh of 3-sensor arrays, which we call triads. Figure A1 of the previous section showed a large network and how it is divided into a mesh of triads. Whether or not signals are coherent from station to station depends on how the inter-station distance compares to the wavelength of the signal of interest. If the inter-station spacing is far greater than several wavelengths, then the signals are not coherent between stations and incoherent processing is used to process the signals. If the spacing is comparable to the wavelength, then coherent processing may be performed. AELUMA has been applied to several types of incoherent signals; high frequency seismic body waves (de Groot-Hedlin and Hedlin, 2018) and infrasound signals (de Groot-Hedlin and Hedlin, 2015), with wavelengths of hundreds of meters to several kilometers, were detected at the USArray, where the average station spacing was 70 km. Atmospheric gravity waves (de Groot-Hedlin et.al, 2014) and seismic surface waves (Fan et. al, 2018) are examples of coherent signal detection by the AELUMA code. For the purposes of this project, incoherent processing is applied to the signal processing at stations that are not part of a dense array.

(i) Incoherent processing for triads

A synthetic example is used to demonstrate how incoherent signals traversing a triad are processed. Figure B1 shows a sample triad configuration, with an average station spacing of approximately 70 km, similar to the inter-station spacing at the US Array.

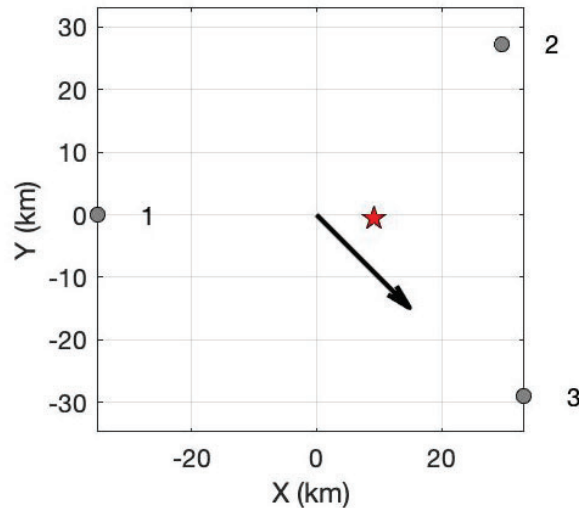


Figure B1. Triad configuration. *The red star shows the centroid of the triad. The signal propagation azimuth is shown by the arrow.*

Synthetic data were generated with a center frequency of 2 Hz, with random noise added in the frequency range from 1-4 Hz. The propagation azimuth was set 135° clockwise from north, that is, the signal crosses the triad from northwest to southeast. The average

station spacing is 70 km. The signal velocity was set at 3.5 km/s, so the wavelength at the center frequency is 1.75 km, far less than the inter-station spacing, so that signals must be processed incoherently. The data at each station are converted to envelopes using the standard method of Hilbert transforming the data, followed by application of a short-term average to long-term average (STA/LTA) filter. The STA and LTA time spans were chosen to be 3 s and 60 s, respectively. The amplitudes of the envelopes are a measure of the signal-to-noise ratio (SNR). The noisy data are shown in light grey in Fig B2a, with envelopes shown by the heavy black lines.

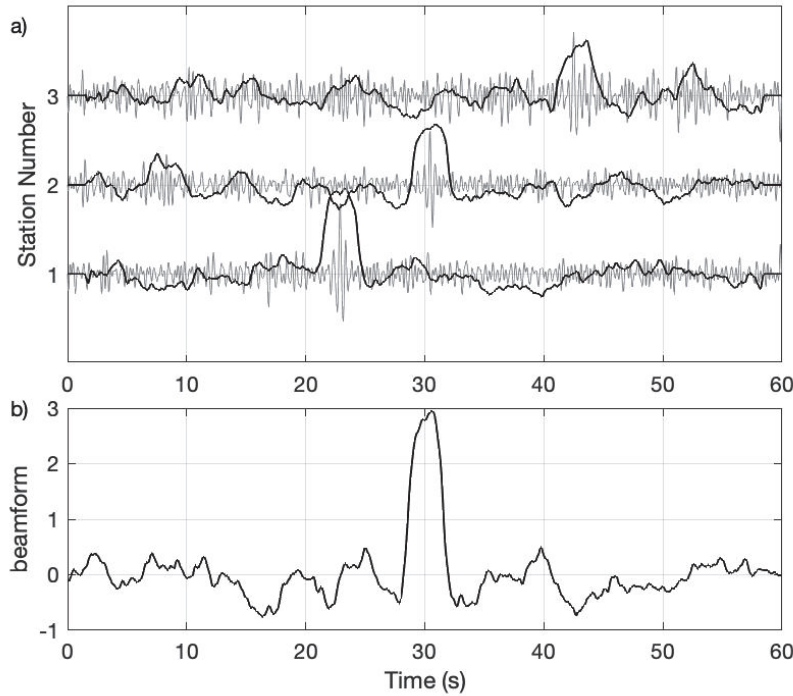


Figure B2. a). Synthetic data and envelopes, shown in gray and black, respectively. *The numbers identify the station locations, from Fig B1.* b) The beamformed envelope. *The peak of the beam-formed envelope is designated the arrival time at the triad centroid, shown by the red star in Fig. B1.*

The relative arrival times, or delay times, between all station pairs are derived by cross-correlating the de-meaned envelopes. Ideally, the delay times sum to zero for a closed path around the triad, however due to waveform distortions and the presence of noise, the time lags are said to be consistent with a coherent signal if they satisfy the criterion

$$T_{ij} + T_{jk} + T_{ki} < T_{res}, \quad (B1)$$

(Cansi, 1995), where T_{ij} is the time delay of the signal at station j relative to station i . The value of the residual misfit in time T_{res} is usually chosen empirically based on the average dimension of the triad in the study as well as the expected phase velocity (eg. de Groot-

Hedlin and Hedlin, 2015, 2018; Fan et.al., 2018). To simplify the choice of the cutoff value, we define the consistency criterion T_{cons} here as

$$T_{\text{cons}} = \frac{|T_{ij} + T_{jk} + T_{ki}|}{|T_{ij}| + |T_{jk}| + |T_{ki}|} \quad (B2)$$

The denominator increases with increasing triad dimension and decreasing phase velocities, so defining the consistency criterion as a ratio should make it applicable to a wide range of triad dimensions and signal velocities. This ratio ranges from 0 for a perfect fit to 1.0 for a signal inconsistent with a plane wave.

The propagation azimuth and phase velocity may be derived using tau-P array processing (e.g., Havskov and Ottemöller, 2010; Schweitzer et al., 2012). The signal delay time between stations is given by

$$T_{ii} = \mathbf{p} \cdot (\mathbf{r}_i - \mathbf{r}_i) = p_x(x_i - x_i) + p_v(y_i - y_i) \quad (B3)$$

where $\mathbf{p} = [p_x, p_v]$ is the signal slowness vector, and $\mathbf{r}_i = [x_i, y_i]$ denotes the coordinates of station i . We solve for $\mathbf{p}$ using least squares, then solve for the azimuth and phase velocity:

$$\theta = \tan^{-1}(p_x/p_v) \quad (B4)$$

$$|v| = \frac{1}{\sqrt{p_x^2 + p_v^2}} \quad (B5)$$

A minimum of three stations are needed to unambiguously derive the propagation direction and phase speed but Eq. (B3) can be generalized to more than three stations as shown in appendix C, on array detections. The derived azimuth and velocity are then used to compute a beam-formed envelope from the de-meaned envelopes at each station, and the time associated with its maximum amplitude is identified as the signal arrival time at the triad centroid location.

For the envelopes shown in Fig B2a, the mean cross-correlation between pairs of envelopes is 0.66, and the consistency criterion T_{cons} has a value of 0.0025. The azimuth and phase velocity solutions are derived as 135.2° and 3.43 km/s, very close to the input values used to generate the solution. These values were used to derive a beamformed envelope corresponding to a location at the center of the triad. The beamformed envelope is shown in Fig Bb. The time corresponding to peak amplitude of the beam-formed envelope is designated the arrival time at the centroid location.

(ii) Thresholds for processing Triads

In any detection method, some threshold or cutoff criterion must be satisfied to declare a detection, that is, to determine whether or not a solution is reliable. Rather than choosing empirical cutoff criteria for the triad method described above, effective cutoff values

were determined by repeating the above computations for an ensemble of 4000 realizations. The phase velocity for each computation was varied from 2.0 km/s to 7.0 km/s and the source azimuth was varied from -180° to 180° . Noise levels at each envelope varied from 10% of the amplitudes of the synthetic data to 130%; the upper level was chosen such that about half the members of the ensemble yielded solutions that fit the true source azimuth to within 10° . The azimuth and phase velocity errors were computed for each computation, as well as the following statistical measures

- a) the mean cross-correlation between envelopes within the triad,
- b) the consistency criteria given by eqn. B2,
- c) the SNR of the final beam-formed envelope.

Because the beam-formed envelope is derived from de-measured envelopes at each station, the SNR is given by one plus the peak amplitude of this envelope.

The azimuth misfit vs. each of these statistical measures is shown in Fig B3 for each member of the ensemble. The fit to the azimuth is shown here rather than the phase velocity, since the derived azimuth is more important in the next step of deriving source locations (Appendix D). For this ensemble, 1966 members of the population fit the azimuth to within 10° , shown in green, and 2034 members had a poor fit to the azimuth, shown in grey. Figure B3a shows that members of the ensemble with large azimuth misfits have low mean cross-correlation values but there is considerable overlap between the two populations of points. Figure B3b shows that solutions with small azimuth misfits are strongly concentrated at very low T_{cons} values, as expected. Figure B3c shows that points with low SNR values have large misfits in the azimuth. Note that given an azimuth cutoff of 10 degrees (i.e. ± 10 degrees), a portion of the solutions (1/18) would be expected to lie by chance within the azimuth cutoff even for pure noise. This may account for the fact that there are some points with low azimuth misfits for high T_{cons} values (Fig B3b), or low SNR values (Fig B4d).

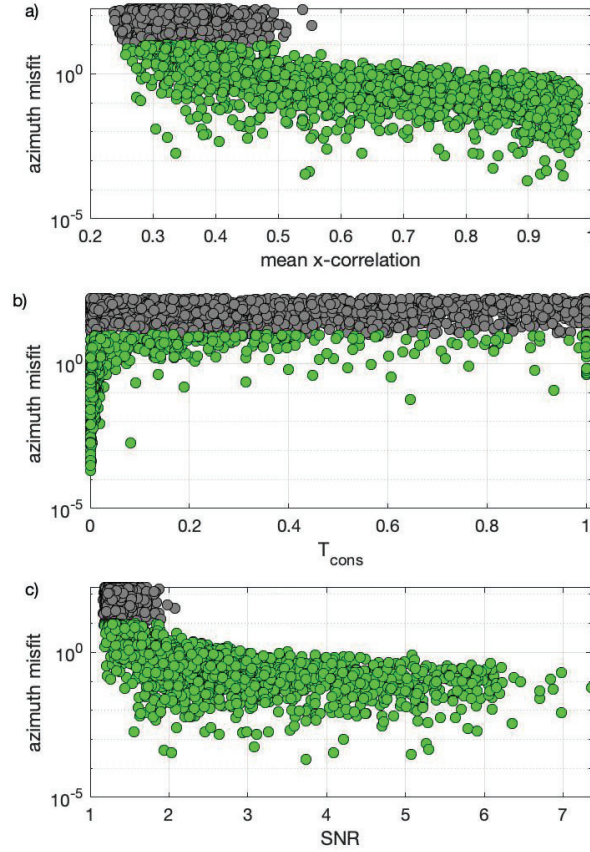


Figure B3: The results from 4000 realizations of the computations indicated in Figure B2, for varying phase velocities, azimuths, and noise levels. *The misfit between the original azimuth and the derived value is shown as a function of a) the mean interstation cross-correlation between the envelopes, b) the time consistency value (eq. B2), c) the SNR value of the beam-formed envelope. Realizations with azimuth misfits less than 10^0 are shown in green; higher misfits are shown in grey.*

The results of Fig. B3 are displayed in histogram format in Fig B4, where the poorly fitting population is again shown in grey and the population with adequate fits to the azimuth is shown in green for each of the statistic measures. The histograms confirm that there is considerable overlap between the populations when comparing their cross-correlation values, suggesting that this may be a poor proxy for determining whether a solution is accurate. Almost all solutions with an azimuth misfit of less than 10^0 have T_{cons} values of less than 0.1, suggesting that this is a good cutoff value for T_{cons} . A second criterion, that the mean SNR of the beam-formed envelopes must be greater than 1.5, is also applied based on these results.

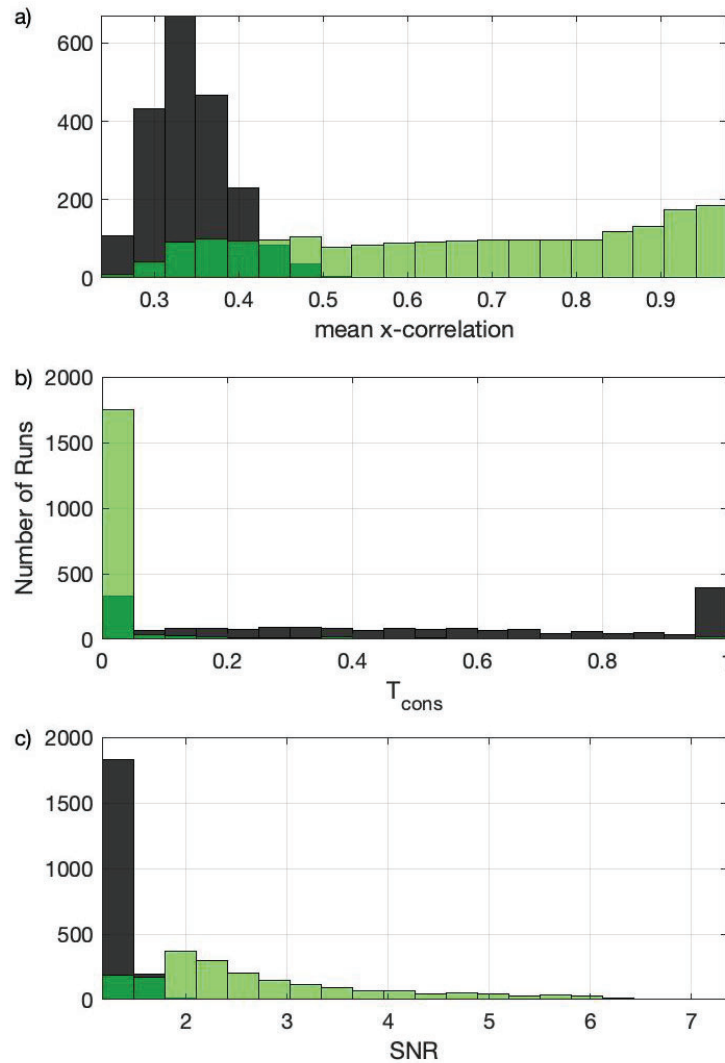


Figure B4. The results in figure B3 are shown in histogram format here to more easily view the separation in populations between good fits to the azimuth (green) and poor fits (grey).

Although not shown here, the SNR of the individual envelopes at each station strongly correlate with the mean SNR of the beam-formed envelopes. This can be used to speed up the computations as indicated in the pseudo-code, shown below.

(iii) Pseudocode

The pseudocode for signal detection using triads is described here. A pseudocode is an outline of the operating principles of the algorithm.

Before running the detection code, parameters governing the region of interest, station configuration, and allowable sample rates must be set in `Triad_param1.m`. Parameters for the frequency bands of interest, the short- and long-term time averages and time

window lengths are also set within this code. Cutoff values for the SNR and T_{cons} values are also set here. The parameters that must be set in Triad_param1.m are discussed in Appendix E.

Pseudocode for Triad_detections(requestedYear, requestedDay)

```

select stations of interest for julian day 'requestedDay' of year 'requestedYear'
read in the data
form triads
set null signal defaults for (1) Xcorrelation_values (2) time_consistency, (3)
signal_azimuth, (4) signal_velocity, (5) signal_SNR, (6) arrival_time
FOR each frequency band
    bandpass filter and form envelopes
    FOR each triad
        FOR each time window
            find SNR for each envelope in the triad
            IF mean(SNR) < SNRcutoff
                this is a non-detection, continue to next time window
            END
            compute Xcorrij and Tij (delay time) for each station pair
            compute (1) Xcorrelation_values = mean cross-correlation
            compute (2) time_consistency
            IF time_consistency > Tconscutoff
                this is a non-detection, continue to next time window
            END
            solve for (3) signal_azimuth, (4) signal_velocity
            do beamform for this signal_azimuth and signal_velocity,
            find (5) signal_SNR, (6) arrival_time for beamformed waveform
        END          % go to next time window
    END              % go to next triad
output detections for this frequency
END                  % go to next frequency

```

APPENDIX C ARRAY DETECTION CODE

LIST OF FIGURES

Figure C1. The gray circles show the sensor positions for a 12-element array. The numbers correspond to the numbered waveforms shown in figure B2. The mean sensor position is indicated by the red star. The angle of signal propagation across the array is shown by the heavy arrow.	56
Figure C2 a) Noisy synthetic data, with numbers corresponding to locations shown in figure C1. A faint signal is apparent within the noise. b) A beam-formed envelope based on an initial, approximate estimate of the signal azimuth and velocity across the array.....	57
Figure C3. Left) F values computed for each beam derived by delay and stacking. The largest F value, marked by the white star, is optimal. Right) L1 time misfits corresponding to time delay misfits computed for the noisy data of Fig. C2.	59
Figure C4. a) interstation time delays. b) t_{misfit} values.	60
Figure C5. The black line shows the beamform based on original solution for the azimuth and phase velocity. The beamform associated with the corrected solution is shown in red..	61
Figure C6. The results from 4000 realizations of the computations, with varying inputs for phase velocity, azimuth, and noise level.	62
Figure C7. The black line shows the signal power for each 15-second time window for the Nevada array on julian day 218 of year 2007.	63

APPENDIX C

ARRAY DETECTION CODE

The tau-p method is used to estimate the phase velocity and direction of plane waves crossing each seismic array. There are many excellent references on seismic array analysis that describe tau-p analysis (e.g., Havskov and Ottemöller, 2010; Schweitzer et al., 2012). In this appendix, we describe the tau-p method using a synthetic example and discuss its application to recorded seismic array data.

(i) *Coherent processing for arrays*

Figure C1 shows the array configuration, consisting of 12 stations with an average grid spacing of 800 m in the x and y positions, with some variability in the locations. The source is located at 115 degrees clockwise from North. The signal propagates across the array at an azimuth of -65 degrees with respect to North at a phase velocity of 3.45 km/s.

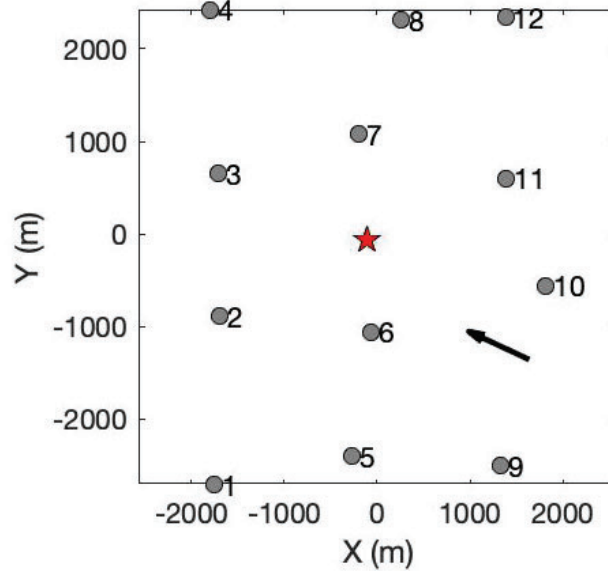


Figure C1. The gray circles show the sensor positions for a 12-element array. The numbers correspond to the numbered waveforms shown in figure B2. The mean sensor position is indicated by the red star. The angle of signal propagation across the array is shown by the heavy arrow.

A synthetic signal was generated with a center frequency of 2 Hz. Array data are processed coherently as the signal wavelengths are typically greater than the interstation spacing. For this example the signal velocity is 3.45 km/s, so the wavelength at the center frequency (about 1.7 km) was greater than the interstation spacing for many stations in this array. Noise in the frequency band from 1 to 3 Hz was added to the signal. The noisy synthetic data are shown in figure C2a. Because the spatial dimension of an array is smaller than for a triad, shorter time segments can be used for the analysis. The time segments analyzed are the maximum of twice the signal transit time across an array or 10-15 times the period at the lowest frequency. For the triads, the former time duration places a lower limit on the time windows; for the arrays, the latter constraint controls the

length of the time window. Here, time windows of 15 seconds were used, as shown in Fig C2. A faint signal is buried within the noise.

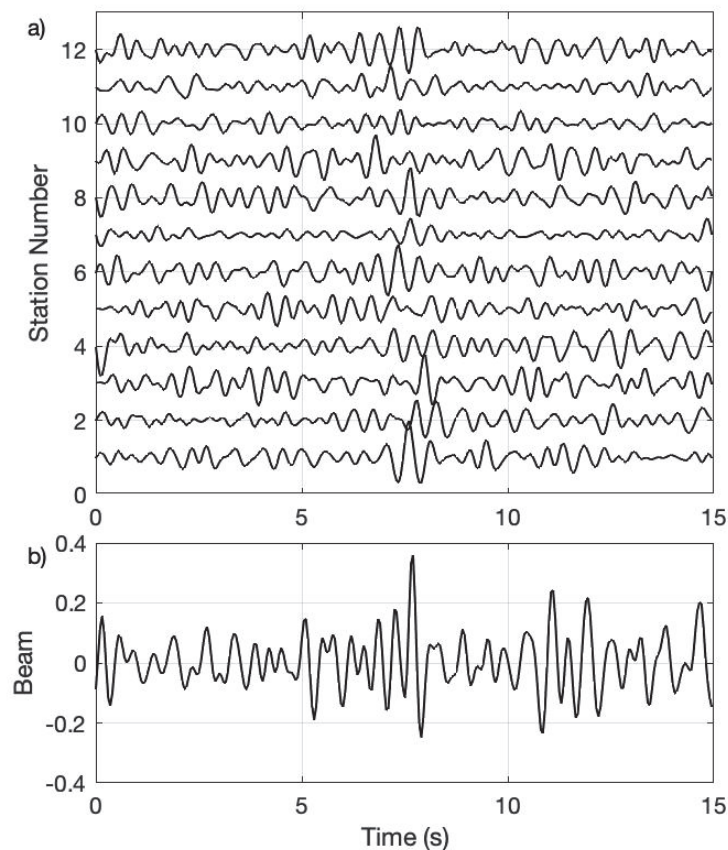


Figure C2 a) Noisy synthetic data, with numbers corresponding to locations shown in figure C1. A faint signal is apparent within the noise. b) A beam-formed envelope based on an initial, approximate estimate of the signal azimuth and velocity across the array. *The peak of the beam-formed envelope is designated the arrival time corresponding to a point at the center of the array, shown by the red star in figure C1.*

The tau-p equation, Eq. B3, can be generalized for an arbitrary number of sensors. For the 12-station array in the example here, there are 66 station pairs in total. Cross-correlating the corresponding waveforms yields the time delays between each station pair. For noisy data especially, there can be inaccuracies in the interstation time delays, due to cycle skipping or loss of signal coherency between the furthest sensors within an array. We minimize the effect of outliers in the time delays using the L1 norm to derive the slowness p , instead of using the L2 norm. (We note that the L1 and L2 solutions for slowness are identical in the case of 3 sensors, i.e. for triads). The slowness is then used to derive the signal azimuth and direction (Eqs B4 and B5), which are then used to compute the beam-formed data.

For the noisy data shown in Fig C2a, the inversion of the time delays led to a solution with a signal direction of 131° and a phase velocity of 6.2 km/sec, well off the input values of 115° and 3.45 km/s. The beam-form solution based on these values is shown in

Figure C2b; This is a non-optimal solution, resulting from noisy data. The problem is that the tau-p method yields the best fit to the time delays; for noisy data, which is not identical to the solution that would result in the best waveform stack.

The problem can be better illustrated using two metrics; one for the time delay misfits, and the other a measure of how well the waveforms stack up. The delay time misfit can be calculated by computing the slowness p corresponding to a given azimuth and phase velocity and applying Eq. B3 to derive the expected time delays, given the known station positions within the array. The expected time delays - denoted T_{ij}^e for the expected time delay between stations i and j - may then be compared to the time delays T_{ij} derived from waveform cross-correlations. The L1 norm misfit in time is then given by

$$T_{misfit} = \frac{\sum_{ij} |T_{ij}^e - T_{ij}|}{\sum_{ii} |T_{ii}|}, \quad (C1)$$

where the sum is over all station pairs. The measure of how well the waveforms stack up is given by an F detector quantity discussed in Blandford (1974) and Arrowsmith et.al, (2009). They define an F detector as proportional to the power of the beam divided by the power of the mean difference between the individual shifted waveform at each sensor and the beam. It is given in Blandford (1974) as

$$F = \frac{N - 1}{N} \frac{\langle (\sum \mathbf{B})^2 \rangle}{\langle \sum (w_i - \mathbf{B})^2 \rangle}, \quad (C2)$$

where N is the number of sensors within the array, $\mathbf{B}$ is the beam, and w_i is the waveform at the i^{th} sensor. For a well correlated signal, the denominator reduces to residual noise and the signal beam power in the numerator increases.

The difference between these two quantities is illustrated in Fig. C3, which shows the F values and the time misfits for a grid of points over a phase velocity range from 2 - 12 km/s, with an increment of 0.1 km/s, and an azimuth range from 1 to 359°, with an increment of 2°. The waveforms are shifted and summed to compute the beam and associated F values at each grid point. The L1 time misfit is also computed. These values are compared in Fig C3. The optimal solution corresponds to the largest F value, which is given by 115° and 3.5 km/s, marked by the white star, which agrees with the input solution to the resolution of the test points. The optimal time misfit corresponds to the minimum, at 131° and 6.2 km/s, which was the solution given by the tau-p method. Note that for less noisy data, the optimal solutions would be identical. The red square in both these plots corresponds to the initial tau-p solution, that was based on noisy data.

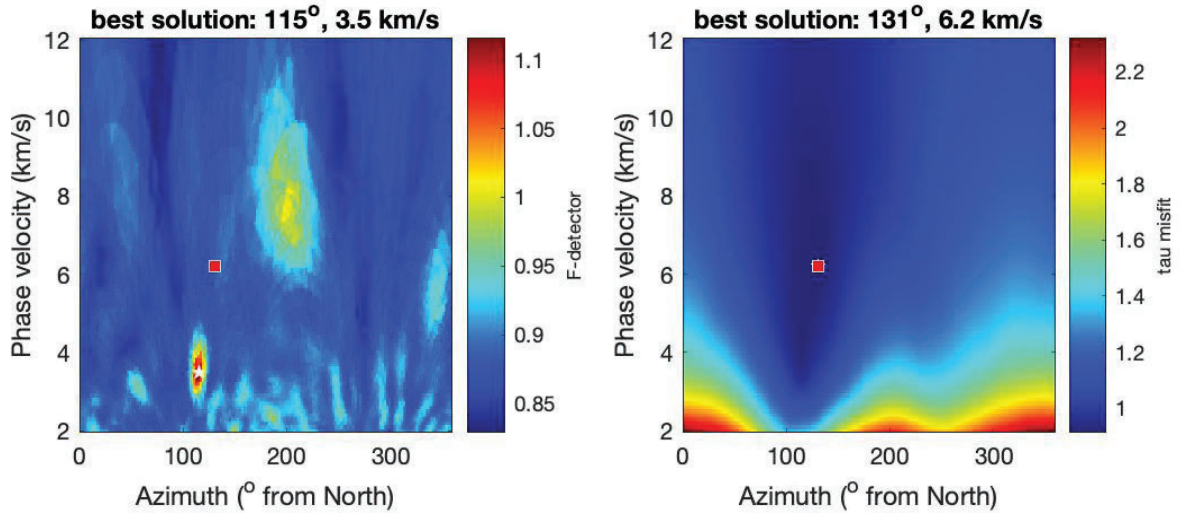


Figure C3. Left) F values computed for each beam derived by delay and stacking. The largest F value, marked by the white star, is optimal. Right) L1 time misfits corresponding to time delay misfits computed for the noisy data of Fig. C2. The smallest value is optimal. *The red square in each plot shows the initial tau-p solution computed applied to noisy data.*

As shown, the tau-p solution only approximates the values that yield the optimal waveform stack. Rather than computing the tau-p solution, we could have instead computed the solution in the frequency wavenumber domain - the FK solution - using the waveforms at each station. However, this method is about ten times slower than the tau-p solution and not necessarily more accurate. Another method would be to do delay and sum beamforming, as in the left panel of Fig. C3. Although this yields the most accurate solution, it is even slower, approximately 100 times as slow as tau-p.

There are several possible other ways to reach the optimal solution. For instance, one could do a directed search in azimuth, phase velocity to find the optimal F-value solution using the tau-p solution as the initial point. We took another approach. We assumed that only some of the time delays were poor; the others correctly measured the signal time delay between stations. The problem reduces to finding which time delays to discard. Our approach was to discard the time delay values associated with the largest misfits to the time delays.

The method is demonstrated Fig. C4, which shows the interstation time delays for the station pairs in the top panel and the corresponding time misfits in the bottom panel. The circles in the top panel show the time delays for each of the 66 station pairs, derived by cross-correlating each waveform pair. The optimal slowness was computed for these points as described above to give an azimuth of 131° and phase velocity of 6.2 km/s. Equation B3 is applied to find the expected time delays given this solution. The empty circles in the lower panel show the absolute value of the misfits between the expected time delays and those derived from the cross-correlation. Time delays associated with the largest misfits - those more than twice the median value - were discarded. The gray circles in the top panel show the remaining points; in this case 55 of the original 66 time

delays were retained. These 55 remaining points were used to re-compute the slowness, which yielded an azimuth of 113.8° and phase velocity of 3.95 km/s, much closer to the optimal solution of 115° and 3.45 km/s. The expected time delays given by this new solution are shown by red squares in Fig C4a and the red circles in Fig C4b correspond to the time misfits.

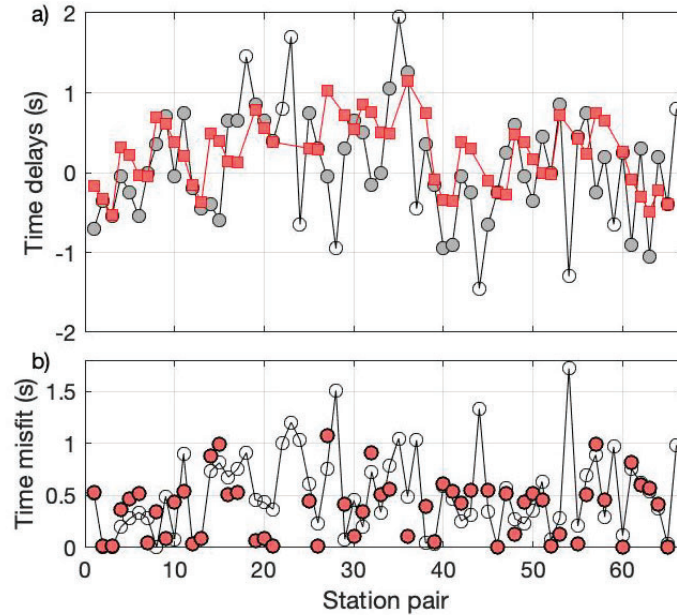


Figure C4. a) interstation time delays. Circles show the time delays derived by cross correlation of waveforms at each station pair. Empty circles are associated with large t_{misfit} values; those marked by grey are used to invert for a corrected slowness solution, p_{corr} . Red squares are the expected time delays corresponding to p_{corr} . b) t_{misfit} values. The empty circles are t_{misfit} values corresponding to the initial solution for slowness. The red circles are those corresponding to p_{corr} ; these are derived from the difference between the gray circles and red squares in a).

Figure C5 compares the beamform for the corrected solution in red with the original beamform, in black. The black beamform is the one shown in Fig. C2b. The T_{misfit} value for the original solution was 0.91; it decreased to 0.79 for the corrected solution. The F-value was 0.8844 for the original solution, which increased to 1.02 for the corrected solution. Clearly, the solution is improved by dropping the worst time delays in this case. To demonstrate that this is generally true and to find adequate thresholds for determining whether a solution is adequate, we repeated the above computation 4000 times, as shown in the next sub-section.

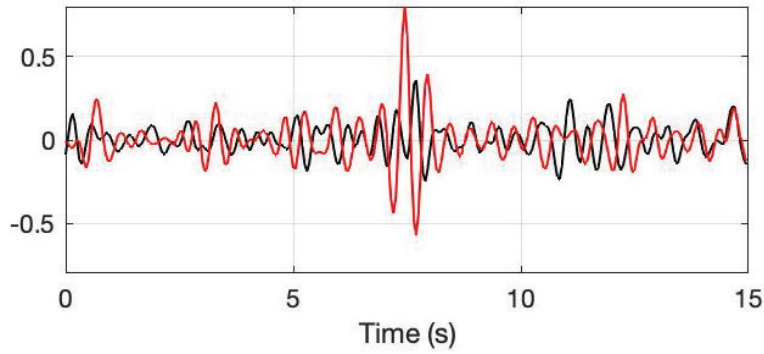


Figure C5. The black line shows the beamform based on original solution for the azimuth and phase velocity. The beamform associated with the corrected solution is shown in red

(ii) *Thresholds for array processing*

We repeated the above computations 4000 times to determine, first, whether the method outlined above significantly improves the accuracy of the solution for slowness (equivalently, azimuth and phase velocity) and second, to find thresholds for determining the accuracy of a solution. We note that a low T_{misfit} value serves only as a proxy to the solution that yields the slowness value corresponding to the optimal stack. That is, the method outlined above will yield an improved T_{misfit} value, but is not guaranteed to improve the azimuth solution. As in Appendix B, the phase velocity for each computation was varied from 2.0 km/s to 7.0 km/s and the input source azimuth was varied from -180° to 180° . The signal to noise levels were allowed to vary from 30% to 140% of the amplitude of the synthetic waveforms. As before, the objective was to ensure that about half the members of the ensemble had azimuth misfits within 10° for the initial solutions. For each realization, the solution was corrected if the initial T_{misfit} value was greater than 0.1. As before, several measures of the accuracy were computed, both for the initial and corrected solutions: the mean cross-correlation between waveforms, the T_{misfit} values (eq. C1), and the F-values (eq. C2).

Figure C6 shows the results for the initial solutions in the left column, and the corrected solutions in the right column. As in Appendix B, the histograms show the population with an adequate misfit to the azimuth (within 10° of the true value) in green; grey indicates population members with a poorer fit to the azimuth. The most significant difference between the populations is the improvement in the T_{misfit} values, as expected. It also results in some improvement to the azimuth solution, as hoped for. Of the 4000 initial solutions, 1999 of them had adequate fits to the azimuth. After dropping the outliers in time delays, 2157 realizations had azimuth misfits of less than 10° , approximately an 8% improvement.

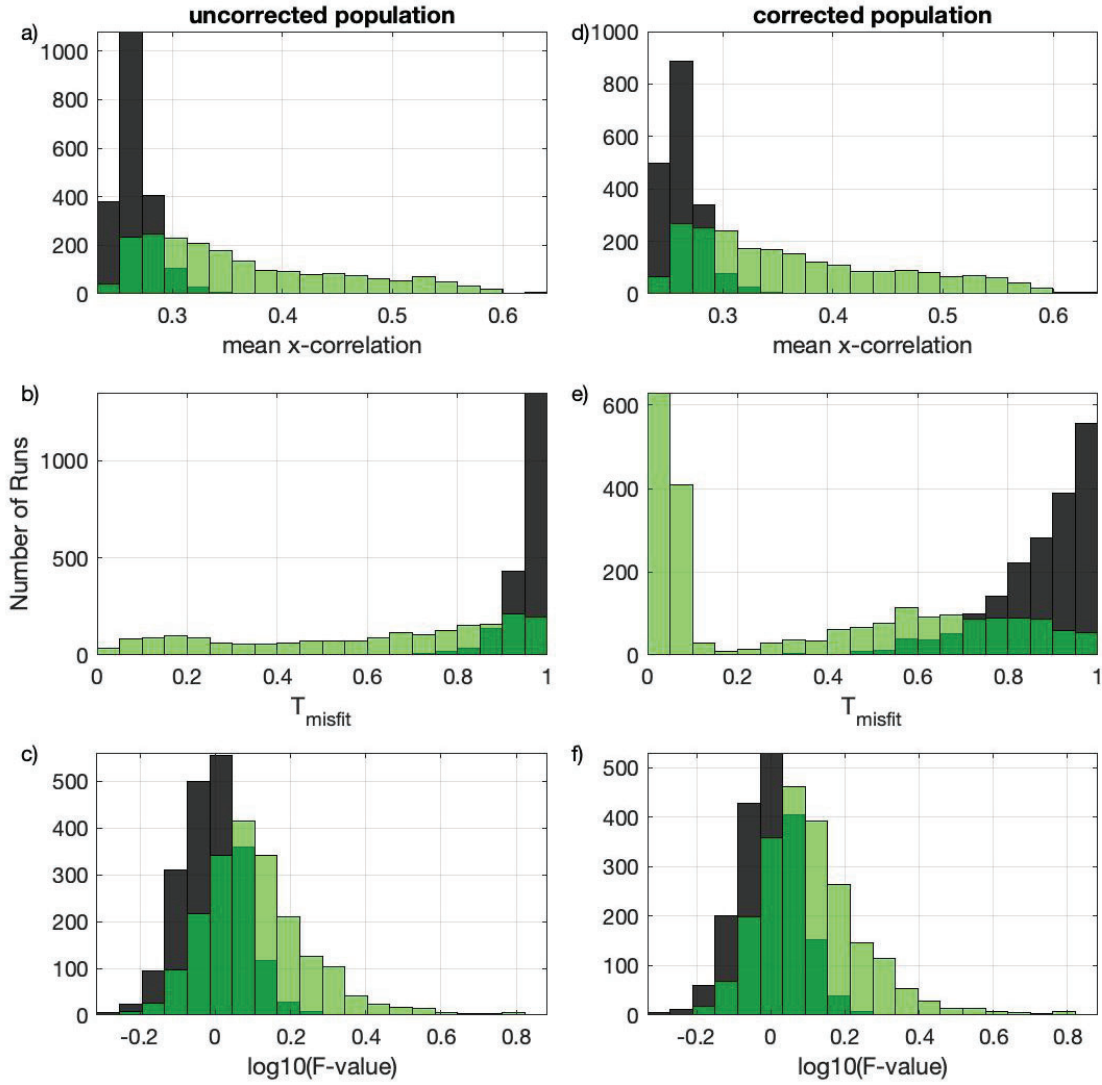


Figure C6. The results from 4000 realizations of the computations, with varying inputs for phase velocity, azimuth, and noise level. *Results for initial solutions are shown at right; results for the population with corrected population are shown at left (i.e. time delays with the highest misfits to the solution are dropped).* In each plot, realizations with azimuth misfits less than 10° are shown in green; higher misfits are shown in grey. From top to bottom, the number of runs with either a good (green) or poor (gray) fit to the input azimuth are shown for a) the mean interstation cross-correlation between waveform pairs, b) the average time misfit (eq. C1), c) the log of the F-value (eq. C2).

A possible drawback of this method is that it reinforces the initial azimuth and slowness estimate. We compared this method with one where instead of dropping the time delays with the largest misfits, we dropped the time delays associated with the poorest cross-correlations between waveforms. This method does not perform as well, with only a 1-2% improvement in the azimuth solution. We note that the code is very modular, that is,

if there is a better method of improving the fit to the azimuth, that method could be used instead.

The results shown in Fig C4 were derived for a 12-sensor array. More generally, it was found that acceptable mean cross-correlation values and T_{misfit} values depended on the array size. However, the statistics for F-value, the measure of how well the waveforms align, are fairly independent of array size. As shown, most realizations with an F-value greater than 1.3 have an adequate fit to the input solution (note that F-values are shown in the $\log_{10}$ domain in Fig. C4). This includes some false detections, which would be eliminated later in the source location stage (Appendix D). Fewer false detections would result from a higher F-value cutoff.

Array processing is more computationally intensive than the processing for triads. We reduce the processing time by computing results only for those time windows in which the seismic signals have higher signal power than the background level. That is, prior to the array processing steps, we compute the signal power in each window and compare it to the median signal power level over a large number of windows. For example, Fig. C7 shows the signal power computed for each 15-second window over an entire day. The red line indicates the running median signal power level computed over a large number of time windows. Array processing was done for time windows in which the signal power was approximately 1.6 times the median signal power. Note that this is similar to an STA/LTA criterion for the processing step, except that a median value is used instead of the LTA, as this results in smoother values for the denominator. We justify doing the array processing only for windows with significant power levels by noting that we are interested mainly in impulsive events. If instead we were interested in background noise, we would skip this step and process every window.

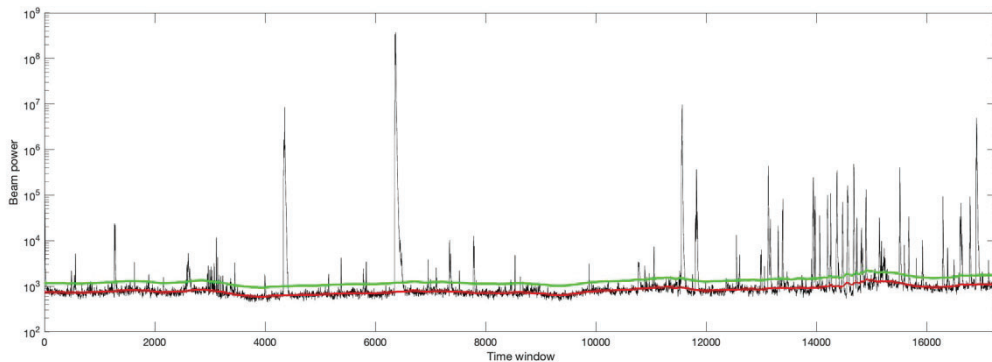


Figure C7. The black line shows the signal power for each 15-second time window for the Nevada array on julian day 218 of year 2007. *The median signal power, computed over one hour of time windows is shown in red. Array processing was done for windows in which the signal power was over 1.6 times the median signal power; this cutoff is shown by the green line.*

Note that an option exists within the code to form envelopes of the data prior to array processing. This can be useful for relatively sparse arrays, or detection of high frequency

signals (Gibbons et.al, 2008). In practice, we found that it led to too many false detections at the frequencies of interest in this study.

(iii) *Pseudocode*

The pseudocode for signal detection using arrays is described here.

Before running the detection code, parameters governing the region of interest, array configuration, and allowable sample rates must be set in Array_param1.m. Parameters for the frequency bands of interest and time window lengths are also set within this code. These parameters are discussed in Appendix E.

Pseudocode for Array_detections(requestedYear, requestedDay)

```

select stations of interest
read in the data for that day
group stations into arrays
set null signal defaults for    (1) mean Xcorrelation_values (2) time_misfit (3)
signal_azimuth (4) signal_velocity, (5) beam_Fval, (6) arrival_time
FOR each array
    FOR each frequency band
        bandpass filter the array data
        FOR each time window
            compute the ratio of the signal power to the median power
            IF the ratio is too small
                this is a non-detection, continue to the next time window
            END
            compute Xcorrij and Tij for each station pair
            compute (1) mean cross correlation
            compute (2) Tmisfit, the average time misfit
            IF Tmisfit < 0.1
                solve for the (3) signal azimuth and (4) signal velocity
            ELSE
                remove Tij values corresponding to the worst misfit
                for the remaining time delays, re-compute (1) the mean
                correlations and (2) the average time misfit
            cross
            Tmisfit
                solve for the (3) signal azimuth and (4) signal velocity
            END
            do beamform for this signal_azimuth and signal_velocity,
            compute (5) beam_Fval, (6) arrival_time for beamformed
        waveform
    END          % go to next time window
END          % go to next frequency
output detections for this array
END          % go to next array

```

APPENDIX D **EVENT ASSOCIATION AND EVENT LOCATION**

LIST OF FIGURES

Figure D1. Detections for day 218 of 2007 that satisfy detection thresholds described in the text, shown as a function of SNR vs time of day.	67
Figure D2. Map of spatial misfit, showing the number of unique triads with detection azimuths that fit each spatial gridpoint.	68
Figure D3. Plots of arrival time versus range for a gridpoint at 40°N, 111°W.	69
Figure D4. Fitness function map, showing the maximum number of unique triads with detection azimuths and seismic velocities consistent with a given time point and spatial gridpoint.	70
Figure D5. A map of detections consistent with an approximate source location of [40°, -111°], at 0849UT.	71
Figure D6. Map of L1 misfit to the group of detections associated with the Crandall mine disaster (the map of azimuths for the signal detections in this group was shown in Figure D5.	72
Figure D7. Plots of arrival time versus range for the optimal source location shown in Figure D6.	73
Figure D8. Map of event detections for August 6, 2007 (julian day 218).	74

APPENDIX D

EVENTASSOCIATION AND EVENT LOCATION

Both the triad and array detection codes return files containing information on a large number of signal detections. For the triad detection code, the output is contained in a file of the form `triadsRaw_Nday_Nyear_flagM.out`, where `Nday` is the julian day, `Nyear` is the year, and `flagM` is the flag number set in the parameter file (explained in Appendix E). The output file contains, for each window and each triad within the network, whether or not a signal detection has been declared. Each declared signal detection is associated with an arrival time and azimuth information, as well as the signal phase velocity across the triad, the cross-correlation value averaged over each station pair, the SNR of the beam envelope, and the consistency T_{cons} (see eqn. B2). The array detection code, contained in files called `arraysRaw_Nday_Nyear_flagM.out`, produces similar quantities, except that the time misfit T_{misfit} (eqn C1) is returned instead of the consistency and the F-value (eqn C2) is returned instead of SNR amplitude. Only the arrival times and signal azimuths are used to estimate source times and epicentral locations so the information from arrays and triads may be combined almost seamlessly.

(i) *Read in detections*

Low quality signal detections may be eliminated by setting thresholds for the phase velocity, mean cross-correlation values, amplitudes (either SNR or F-value) and time misfits (either T_{cons} or T_{misfit}). For instance, signals detected by either triads or arrays could be infrasound signals instead of seismic signals. Since only seismic signals are of interest here, setting phase velocity limits consistent with seismic propagation eliminates other signal types. Also, declared detections that result from spurious correlations within a noisy dataset can mainly be eliminated by setting appropriate lower limits on the mean cross-correlation values, amplitude and time misfits. Finally, since highly localized events that are seen only at a very limited number of triads or a single array are generally not of interest, we require each event to be defined by at least N detections. Lower values of N lead to more false alarms but larger values will tend to miss smaller events.

An example of the event association and source location is shown for triad detections made on August 6, 2007, which is julian day 218. Most stations used are part of the TA which was located near the west coast of the US on this date. Some stations in Alaska are also included. Several anthropogenic events including the Crandall mine disaster and an explosion at UTTR occurred on this date, as well as several earthquakes. Figure D1 is an example showing the detections prior to forming groups of detections, each associated with a single event. There were 362 stations associated with 620 triads on this date. The detection method is demonstrated for a single frequency band from 0.5 to 5 Hz. Figure D1 shows the 4093 detections that remain after applying the following thresholds: phase velocities are from 2.2 - 14 km/s, the mean cross-correlation values are at least 0.45, the minimum SNR is 2, and the T_{cons} value is 0.05 or less. Because there is significant overlap between time windows (based on parameters set in `Triad_param2.m`) we require that at least three azimuth estimates at a given arrival time agree within a cutoff value set in `Triad_param2.m`. No limits are placed on the consistency of phase velocities between consecutive detections. We required each event to be associated with 20 detections; any

detection that was isolated in time such that it was inconsistent with a grouping of less than 20 detections was eliminated. Event associations are made with the remaining detections. Detections made using arrays are culled in an analogous manner, however, since there are far fewer arrays than triads, the threshold of detections required to define an event is set to a lower value.

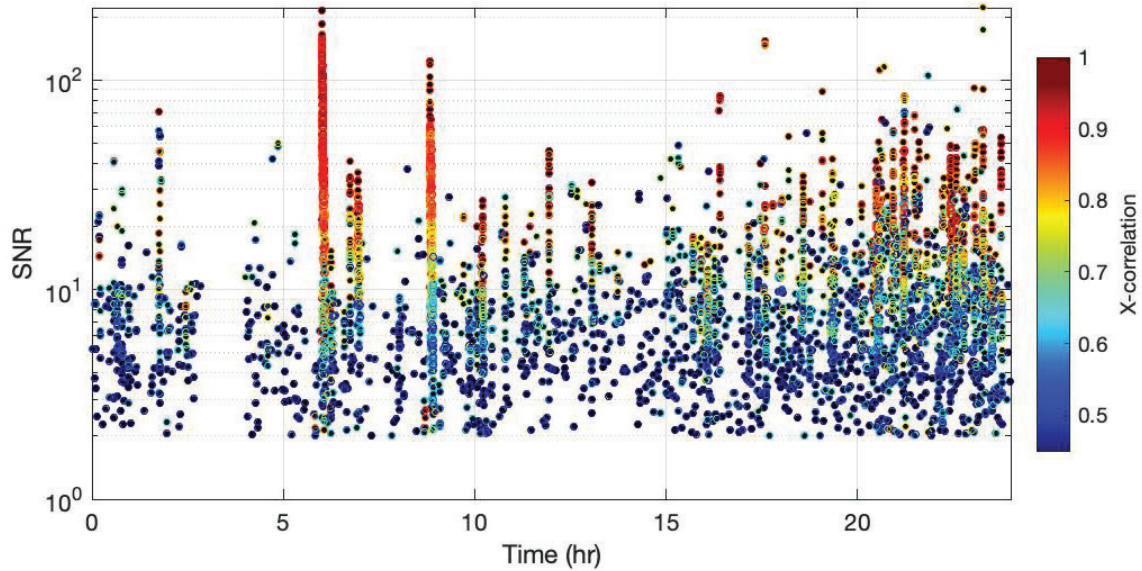


Figure D1. Detections for day 218 of 2007 that satisfy detection thresholds described in the text, shown as a function of SNR vs time of day. *The points are color-scaled as a function of cross-correlation.*

As shown in Fig D1, there are many detections associated with the Crandall mine collapse $[39.46^\circ, -111.167639^\circ]$ at 0848 UT and also with an earthquake that occurred at about 0600 UT. After about 1500 UT, there is an increase in the number of detections associated with anthropogenic noise that occurs during the working day (de Groot-Hedlin and Hedlin, 2018). The detections associated with the UTTR explosion $[41.131^\circ, -112.8965^\circ]$ at 2033 UT occur during this interval. The next step is to form event associations, that is, sort the detections into groups, each associated with a unique seismic source.

(ii) *Event association and coarse estimation of source parameters*

We form event associations by performing a search over a fairly coarse search grid and counting the number of detections with azimuths consistent with each node. Figure D2 shows a map of the initial spatial fit to the detections for a coarse grid that includes 1 gridpoint per degree in both latitude and longitude. For each gridpoint, the distance and azimuth to the triad centroid is computed for all coherent signal detections, and a subset of detections with propagation azimuths consistent with travel from that gridpoint is found. We derive the spatial misfit map by counting the number of unique triads with detections that fit a given gridpoint within a given azimuth error; a threshold of 15° is

used here. For this example, a large number of azimuths point to a source location in the Utah area.

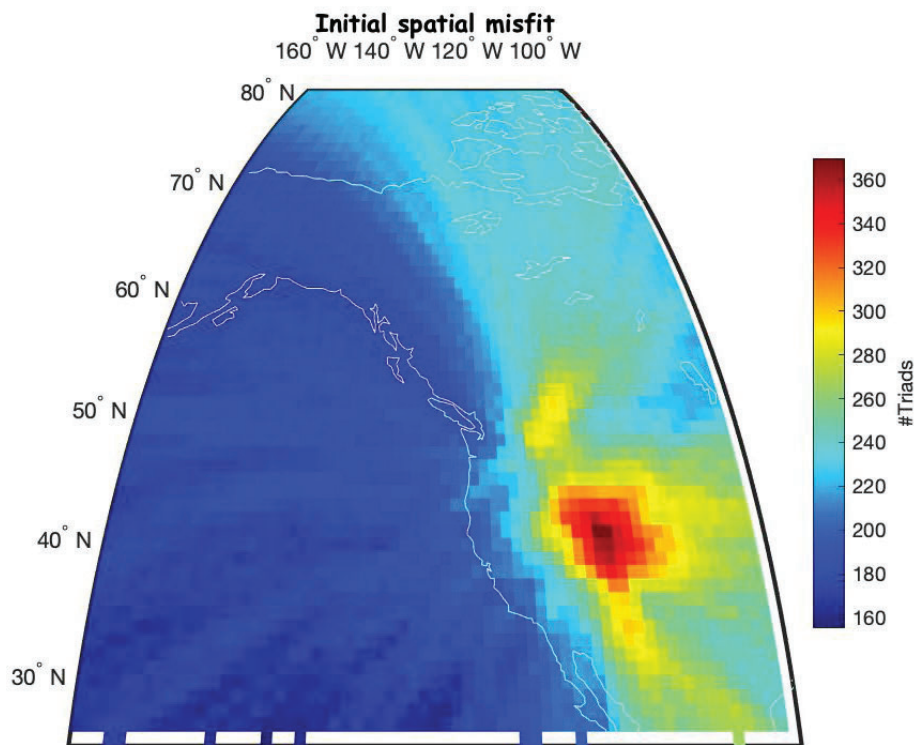


Figure D2. Map of spatial misfit, showing the number of unique triads with detection azimuths that fit each spatial gridpoint.

The spatial misfit map does not take into account the arrival times of the detections. In the next step, we compute the fitness function for a subset of gridpoints with a high spatial fit to the data, indicated by the yellow to red regions in Figure D2. For these points, we sweep through a vector of potential source times (which are sampled at 60 second intervals here) and for each time, we count the number that are consistent with group velocities within a given range. Here we use an allowable velocity range from 2.8-9 km/s. The number of unique triads that agree with each sampled time are counted for each gridpoint within the subset. This step is demonstrated in Figure D3 for a gridpoint at $[40^\circ, -111^\circ]$, near the site of the Crandall mine collapse. All detections from 0830 to 0930 UT are shown in black. Detections with azimuths consistent with that gridpoint are shown in green; those consistent with celerities from 2.8-9 km/s are shown in red. In this case, there are 237 unique triads that are consistent with this time and gridpoint shown. The fitness function for each gridpoint is given by the maximum number of unique triads with detections that are consistent with a sampled source time. In some cases, there may be ambiguity in the optimal location as the computed fitness functions are identical. A tedious discussion of the application of a penalty function to break ties between potential source locations may be found in de Groot-Hedlin and Hedlin (2019). The resulting fitness function is shown in Figure D4. Dark blue areas show regions where the fitness

function was not computed because the spatial fit was low, as shown by the map of Figure D2.

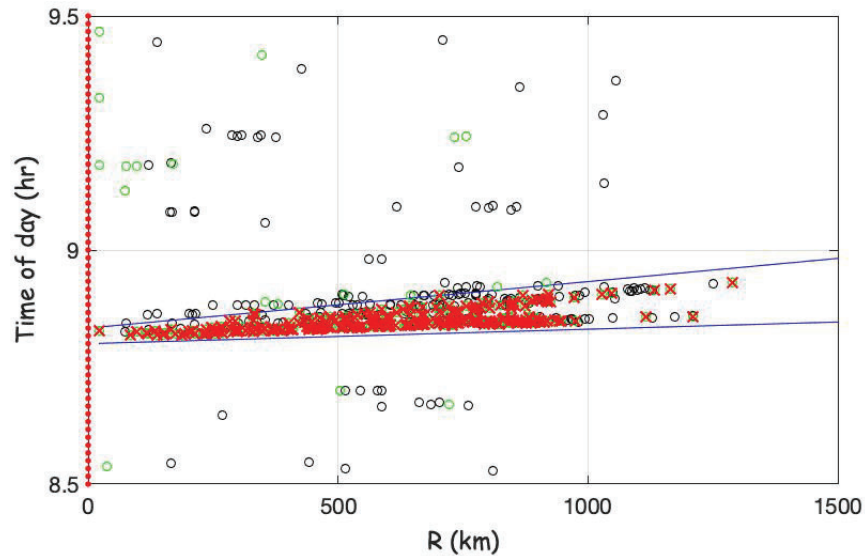


Figure D3. Plots of arrival time versus range for a gridpoint at 40°N, 111°W. *The red dots at zero range indicate the vector of potential origin times. Blue lines bracket the minimum and maximum allowable arrival times for a given point, for a celerity range from 2.8-9 km/s. Black circles show any detection time from 0830-0930 UT, green circles indicate detections with azimuths consistent with this gridpoint, and red x'es indicate those with both azimuths and arrival times consistent with this particular gridpoint and test origin time.*

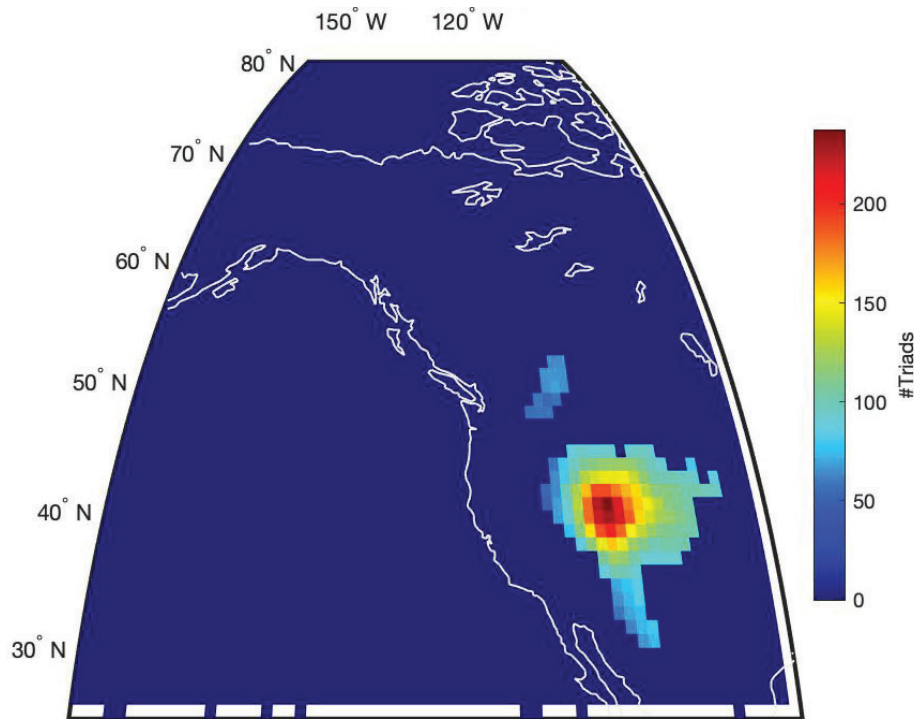


Figure D4. Fitness function map, showing the maximum number of unique triads with detection azimuths and seismic velocities consistent with a given time point and spatial gridpoint.

The gridpoint with the highest fitness level is taken as an approximate source location and the corresponding optimal time at that point is taken as the initial estimate of the source time. Figures D2 through D4 illustrate a single iteration of the event association method. Each iteration identifies a single source time and location, along with its associated detections. Following each iteration, the detections associated with the largest event group are removed from the ensemble and the remaining detections are used to locate another source. When no additional groups of at least N detections can be identified, the iterations are terminated.

The procedure above produces a coarse catalog of source locations and time. In the example above the source locations are found to a 1° resolution in latitude and longitude and 60 s accuracy in time. Aside from providing coarse origin times and locations, the steps above define clusters of detections, each associated with a single event. Figure D5 is a map view of the signal detections associated with the Crandall mine collapse, showing the direction of signal propagation, color-coded as a function of time after the coarse origin time. The initial coarse location of this event is shown by the black star. The origin times indicate that the times do not increase uniformly with distance from the source. This is because some detections are associated with the faster P waves; other with the slower S waves. This is also seen in Fig D3, which shows arrival time vs range.

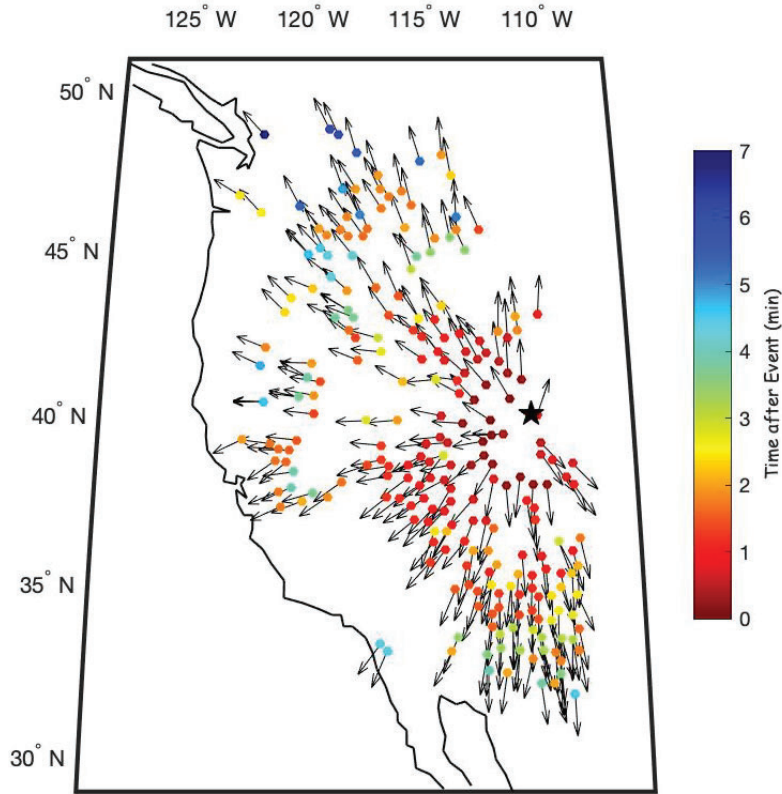


Figure D5. A map of detections consistent with an approximate source location of $[40^\circ, -111^\circ]$, at 0849UT. The arrival times and azimuths of these detections are used to refine the source time and location.

(iii) Fine-scale estimation of source parameters

Once the signal detections are grouped into clusters, each associated with a particular event, the initial coarse locations and origin times are refined. For each cluster of detections, a fine scale spatial grid is defined around the coarse source location. The optimum location is based on the best L1 misfit in azimuth, that is, the optimal source location is defined by the azimuths only. The source location error is estimated from a set of all points with L1 azimuth misfits within 50% of the minimum value. The covariance matrix of this set of locations is computed, and the eigenvalues and eigenvectors of this matrix define the axes of the error ellipse (Vasco et al., 1993; de Groot-Hedlin and Vernon, 1998). This step is done twice to obtain an accurate spatial location, refining the spatial grid each time based on the size of the error ellipses. Figure D6 shows the result for the detections associated with the Crandall mine collapse. The initial coarse location estimate was $40^\circ\text{N}, 111^\circ\text{W}$; the refined source location is $39.54^\circ\text{N}, 111.11^\circ\text{W}$. The error ellipse for this source estimate encompasses the known source location $[39.46^\circ, -111.167639^\circ]$.

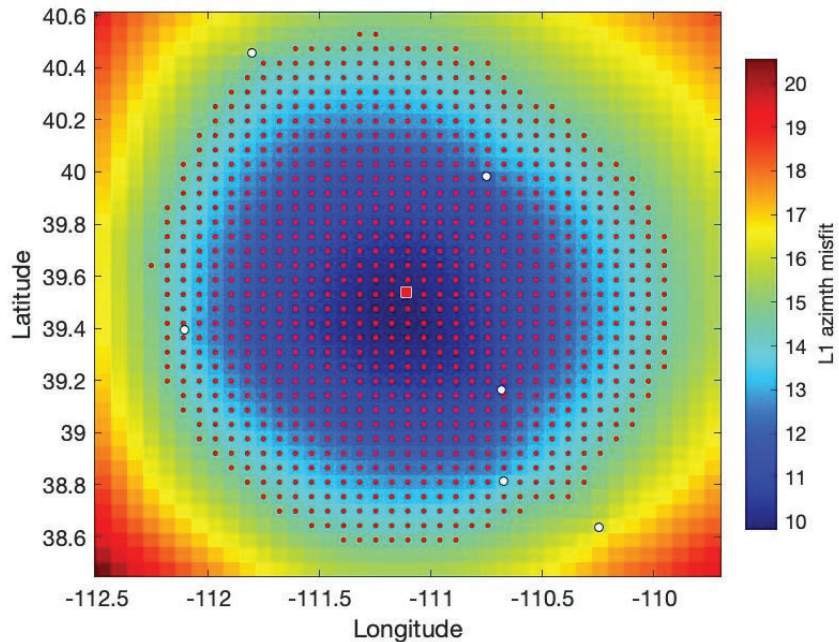


Figure D6. Map of L1 misfit to the group of detections associated with the Crandall mine disaster (the map of azimuths for the signal detections in this group was shown in Figure D5). *The optimal source location is shown by the red square near the center. All gridpoints with L1 azimuth misfits within 50% of the minimum are marked by red dots. White dots show the locations of the nearest triad centers in this cluster.*

The source location estimate is used to estimate the source time using as shown in Figure D7. The distances to each detection are derived based on the optimal source location and a linear regression is used to find the time intercept at $R=0$. In this case, the time estimate is shortly after 0849 UT, about a minute after the known source time. The inaccuracy in time is attributable to the fact that a very imprecise seismic velocity model is used; we assume a constant velocity model in the linear regression. The slope of the line is the estimated slowness across the network. Source locations are flagged if either: 1) the optimal source location is found along an edge of the fine scale grid, or 2) the derived slowness is negative.

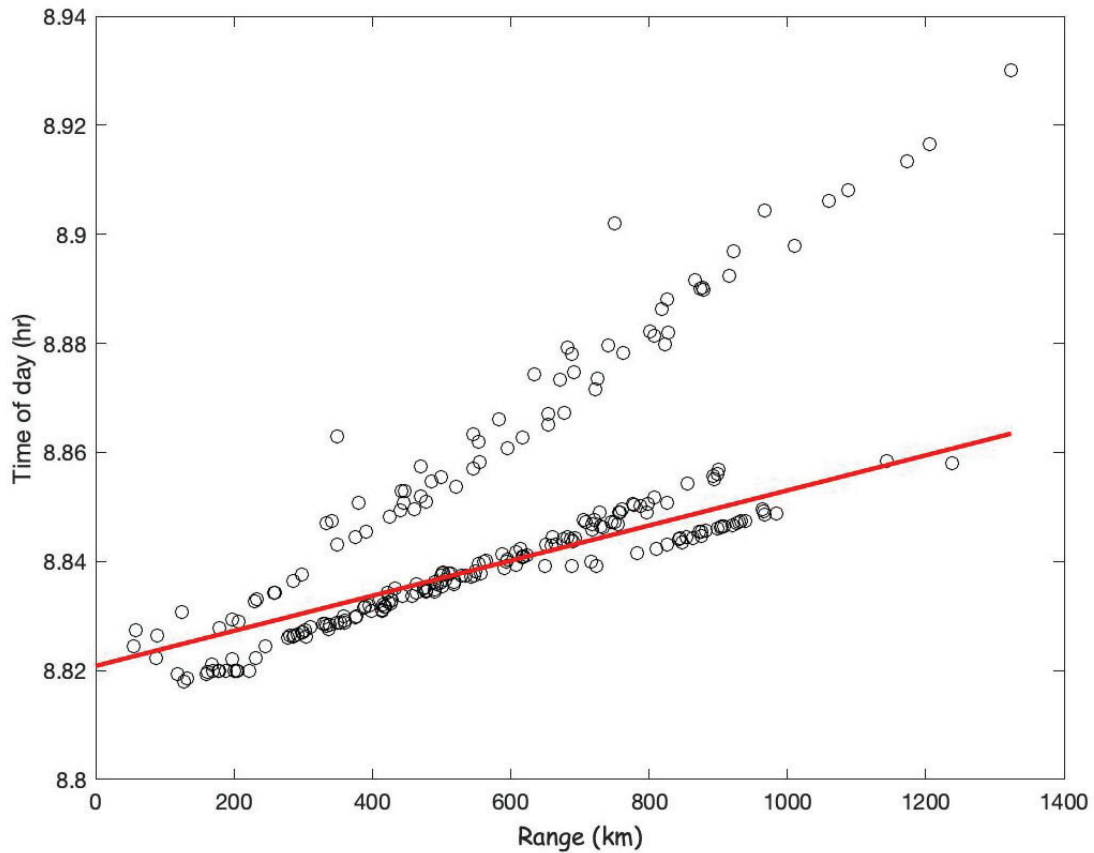


Figure D7. Plots of arrival time versus range for the optimal source location shown in Figure D6. *The red line indicates the best linear fit to the data, which yields the origin time.*

A map of the derived source locations for this day, along with their ellipse errors is shown in Figure D8. All source locations were derived from at least 20 triad detections. For this day, one of the source locations is associated with a negative slowness value and was therefore flagged. This location and its error ellipse are shown in red. Green points and error ellipses correspond to locations with acceptable source location estimates. Error ellipses are much smaller for source locations within the bounds of the network; they are much larger outside the network.

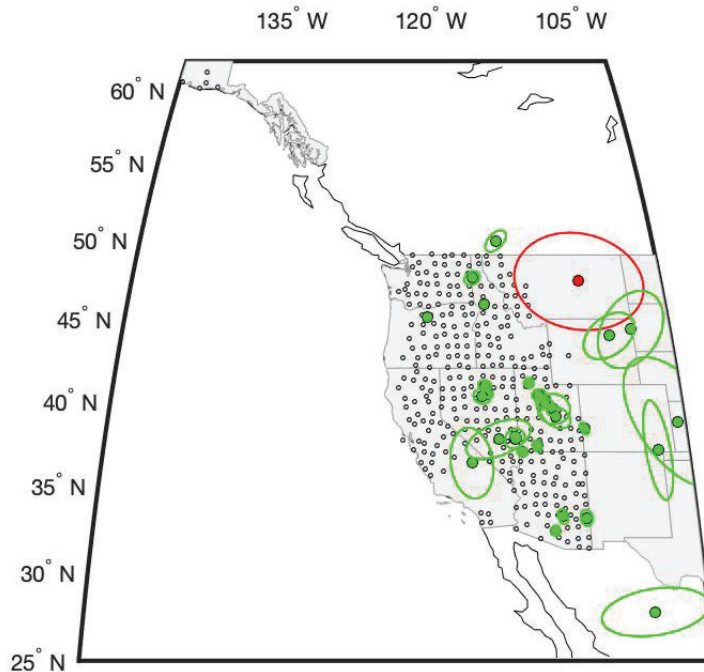


Figure D8. Map of event detections for August 6, 2007 (julian day 218). *Gray dots indicate stations at which any detections were made for this day. The green dots and ellipses around them indicate source location estimates and associated location errors. The single red dot and surrounding ellipse are flagged as a bad point, as it is associated with a negative celerity.*

Note that we have only imposed loose constraints on the seismic velocities; a precise velocity model is not required either for the detections, the event association step, or for the source location. Although it is possible that source parameters could be more precisely estimated using an accurate velocity model, the advantage of the algorithms described here is that they are more broadly applicable for other types of signals. If the search parameters are defined appropriately, the algorithm may be applied to the detection of other geophysical phenomena, including infrasound (de Groot-Hedlin and Hedlin, 2015) or gravity waves (de Groot-Hedlin et.al., 2014). Note that, although the discussion in this section was confined to deriving source locations based on detections made at triads, the methods apply as well to detections made at arrays.

(iv) *Pseudocode*

The pseudocode for source location using only triad detections is described here

Before running the source location code (Triad_locateSources) parameters governing the region of interest, grid sampling rates for the coarse grid location, and cutoff thresholds must be set in Triad_param2.m. These parameters are discussed in Appendix E.

Pseudocode for Triad_locateSources(requestedYear, requestedDay, optional:hour1, hour2)

A.

Read in the triad detections associated with julian day ‘requestedDay’, year ‘requestedYear’, (use optional parameters hour1 and hour2 (from 0-24) if only a portion of the day is of interest)

Discard low quality detections based on threshold values

Remove isolated detections (because there is a significant overlap between time windows)

B.

WHILE number of detections > N (*minimum to define an event*)

 Compute the spatial misfit map for a coarse spatial grid
 in areas of high spatial fit (~ 5% of grid), compute fitness by sweeping thru arrival times

 estimate rough spatial error limits

 save properties of K detections with highest fit

 output coarse solution to file

 continue with remaining detections

END

(*at this point M clusters of detections have been defined; each defines an event*)

C.

FOR each cluster

 Loop 1

1. Define a finer scale based on the coarse source location and error estimates
2. Find new optimal source location for this grid based on L1 azimuth minimum
3. Compute new spatial error limits (gridpoints < 50% of L1 minimum)
4. Use linear regression to find the origin time & slowness for this source location

 Loop 2

 Define a finer scale based on the spatial location and error estimates of Loop 1.

Repeat steps 2 through 4 of Loop 1
Flag the solution if the location is
 a) found along an edge of the fine scale grid, or
 b) the derived slowness is negative.
Output the finer source parameter solution to file
END

Code alterations for the case where only array detections are used

The code Array_locateSources estimates source locations based solely on array detections. Prior to running it, parameters governing the region of interest, grid sampling rates for the coarse grid location, and cutoff thresholds are set in Array_param2.m. These parameters are discussed in Appendix E. The source parameter estimation code for this case follows that described for the triad detections.

Code alterations for the case where both triad and array detections are used

The code LocateSources estimates source locations based on both triad and array detections. The file TRIADArray_param2.m, described in Appendix E must be set up prior to using it. Again, the only difference between this code and those described previously is in part A of the pseudocode. Both the triad and array detections are read in, then merged before proceeding to the event association and source location steps.

APPENDIX E

USER MANUAL - INPUT

Appendices A through D explain how the code works. This appendix and the next serve as the user manual for the code; this appendix explains how to set up the parameter files for signal detection and source location, the next appendix describes the output files that are written.

*(1) Signal detection using triads – set up **Triad_param1.m***

The code **Triad_detections(Year, JDay)**, located in /bin is run to detect signals at triads.

Year – year of interest

JDay – julian day of interest

The workings of the code are described in Appendices A and B. Before running it, information about the signal that is of interest must be supplied by the file

Triad_param1.m, also located in /bin. The parameters to be set in **Triad_param1.m** are explained here, along with suggested values. The parameters deal with the type of station to be used, the network configuration, and properties of the signal under investigation.

ifDMC – if data available through the DMC data are of interest, this parameter is set to 1. Suggested value = 1 for a first pass at running the code – the user must have a good internet connection since the data are accessed online. **If this parameter is set to another value, for instance to read in data on disk, the user must supply their own code to read the data and put it in a format usable for the Triad_detections code.**

triadflag – this allows the code to be run using different parameters for a particular year and day of interest without over-writing the previous file for that date. This might allow the user, for instance, to run the code to search for seismic body waves for one run, and infrasound data for another run. Suggested value = 1 for the first time it is used for a given day and year.

The next two parameters are specific to the case where DMC data are read in. They do not need to be set if ifDMC=0.

net – this is the network code needed to read DMC data. Check the DMC pages for information on the networks available. Setting the value net = '*' searches through all stations available on the day of interest.

loc – a location code needed to read in DMC data. Suggest setting loc = '' (blank)

The next few parameters deal with data type/sampling rate for the signal of interest

chan – channel code for seismic (or infrasound) data. This may be listed as a string of codes if more than one channel type is wanted, i.e chan = 'SHZ, BHZ, HHZ' could be used for high frequency seismic data

fsmin - minimum allowable sample rate. Suggested value fsmin = 20 for high frequency seismic data

fsmax - maximum allowable sample rate. fsmax must be > fsmin

The next parameters deal with the geographic region of interest

ifregion – set to 0 if stations are not limited by geographic area (i.e. read data from anywhere in the world) or 1 to specify a particular geographic region= 1; Suggest setting ifregion=1

lon1, lon2 – west/east longitudes of region of interest. $\text{lon1} < \text{lon2}$. For instance, to specific longitudes within contiguous United States, $\text{lat1} = -130$; $\text{lon2} = -60$;

lat1, lat2 – south/north latitudes of region of interest. $\text{lat1} < \text{lat2}$. For instance, to specific latitudes within contiguous United States, $\text{lat1} = 25$; $\text{lat2} = 50$;

Note if ifregion=0, the lat and lon values are treated as dummy variables.

The next few parameters are dependent on the network geometry

rtmx – the maximum triad arm length in kilometers. This depends on the network density. The sparser the network, the higher this number will have to be. For the USArray, suggested value = about 150.

rmin - the minimum triad arm length in km. Suggested value approximately $\text{rtmax}/4$.

minang – minimum interior angle to define a triad (equilateral triangles are ideal). Suggest $\text{minang}=20$.

maxang – maximum interior angle of triad. Suggest $\text{maxang}= 180-2*\text{minang}$;

The next set of parameters depend on the signal of interest.

fmin – minimum signal frequency of interest

fmax – maximum frequency of interest. fmax must be $> \text{fmin}$. Also, fmax must be $< \text{fsmin}/2$ to satisfy the Nyquist criterion.

numf –number of frequency sampling bands. numf must be an integer, the simplest choice is 1. The code will compute the frequency range for each overlapping frequency band.

freqs – the code will compute these based on values of fmin , fmax , numf

flo – the code will compute these based on values of fmin , fmax , numf

fhi – the code will compute these based on values of fmin , fmax , numf

vphmin – a minimum estimate of the phase velocity in km/s. suggest vphmin approx. 2.2km/s to capture seismic waves. (vphmin approx. .23 to capture infrasound)

ifenv: =0 use waveforms for signal cross-correlation and detection

=1 use envelopes for signal cross-correlation and detection

The next few parameters are specific to the case where envelopes are being used. They do not need to be set if $\text{ifenv}=0$.

fsenv – sampling rate used for the envelopes. Must be set to at least $2*\text{fmax}$, also must be less than fsmin . For low values, the code runs faster but the precision of the azimuth and phase velocity may be slightly reduced.

tsta, tlta – short and long time averages (in seconds) for forming envelopes. Suggest setting tsta to be $\geq 1/\text{fmin}$. Suggest setting tlta to be 30 to 100 times tsta .

snrcut - each envelope (or waveform amplitude) must have this SNR to do further computation

twin – duration of time window (in seconds) to analyze for each detection. Choose the maximum of a) 2xtransit time across array (which depends on both rtmx and vphmin), b) 10 cycles at the lowest frequency (depends on fmin, c) 300 points (depends on fsmin for waveforms, on fsenv for envelopes)

The following values could be changed by the user but should not be. Change at your own risk.

tcons - consistency cutoff = ratio of $|t_{12}+t_{23}-t_{31}|/(|t_{12}|+|t_{23}|+|t_{31}|)$. Strongly suggest Tcons=0.1

In the next few lines, twin (which is set above) is altered to the nearest number such that an integer number of time windows fit in a 24-hour period.

tstep – time step between successive time windows. Suggest setting to twin/4 so that there is significant overlap between windows.

makeplots – =0 to suppress plotting maps, suggest setting = 1 for a first run-through of the program.

A few more variables are set here. Leave them alone; they have to do with plotting in case makeplots=1.

Running the Triad_detections code creates a binary file named triadsRaw_YYY_YYYY_flagN.out where YYY and YYYY are the julian date and year, respectively, that are set when calling Triad_detections(YYYY,YYY) and N is the value of triadflag set in the Triad_param1.m file above. This file is output to the directory /data/detections/YYYY and is read in by the program that sorts detections into source locations, see Appendix E.3.

(2) Signal detection using arrays – set up Array_param1.m

The code **Array_detections(Year, JDay)**, located in /bin must be run to compute signal detections at arrays. The values Year and JDay are as described in Appendix E.1.

Appendix C describes how the code works. Before running it, information about the signal of interest must be supplied by the file **Array_param1.m**, also located in /bin. Most of the parameters are as described in Appendix E.1. For completeness, they are listed here. **Only new parameters are shown in red.**

ifDMC - as described in Appendix E.1

arrayflag - similar to the description for triadflag in Appendix E.1, but for array output

As above, the next 2 parameters are only necessary for ifDMC = 1

net – as described in Appendix E.1

loc as described in Appendix E.1

chan - as described in Appendix E.1

fsmin - as described in Appendix E.1
fsmax - as described in Appendix E.1

ifregion - as described in Appendix E.1
lon1, lon2 - as described in Appendix E.1
lat1, lat2 - as described in Appendix E.1

Occasionally, there may be arrays within the area of interest that perform poorly at the date of interest, for instance due to a persistent noise source nearby (for instance, an array located near an airport). In that case, it may be desirable to exclude that array from the study by specifying an exclusion zone surrounding the array. Several zones may be specified; the latitude and longitude limits for each zone must be given. An example for 1 exclusion zone is shown here.

ifexclusion = 1; - number of exclusion zones. Suggest ifexclusion = 0.

The following values are only specified if ifexclusion is not 0. For ifexclusion=N, N latitude and longitude limits must be specified; each must be an N x 2 matrix.

ifexlon(1,:) = [-123.46 -123.44]; - longitude limits of exclusion zone

ifexlat(1,:) = [48.64 48.66]; - latitude limits of exclusion zone

The next few parameters are dependent on the network and are used to group the sensors into arrays.

rmax - maximum radius across array in km. Consider the signal wavelength to set this value.

namin - minimum number of stations in an array. The minimum allowable value is 3.

Parameters that depend on the signal of interest

numf - as described in Appendix E.1

fmin - as described in Appendix E.1

fmax - as described in Appendix E.1

freqs - as described in Appendix E.1

flo - as described in Appendix E.1

fhi - as described in Appendix E.1

ifenv - as described in Appendix E.1. **I suggest using ifenv = 0 for arrays**

vphmin - as described in Appendix E.1

twin - as described in Appendix E.1

Although I recommend not using the option to form envelopes of the data prior to array processing, there may be applications in which it is useful (see Gibbons et al, 2008). In that case the following variables are needed.

tlta, tsta - as described in Appendix E.1

The following values could be changed by the user but should not be. As described in Appendix E.1 above, twin is altered in the next few lines such that an integer number of time windows fit in a 24-hour period.

tstep - as described in Appendix E.1, however more overlap is required for arrays than for triads. **Suggest setting tstep to twin/4 so that there is 75% overlap between windows.**
makeplots - as described in Appendix E.1

As described in Appendix E.1, a few more variables are set here but should be left alone.

Running the Array_detections code creates a binary file named arraysRaw_fff_yyyy_flagN.out where fff and yyyy are the julian date and year, respectively, that are set when calling Array_detections(yyyy,fff) and N is the value of triadflag set in the Array_param1.m file above. This file is output to the directory /data/detections/yyyy and is read in by the program that sorts detections into source locations, see Appendix E.4.

(3) *Event association and source locations using triad detections –set up triad_param2.m*

The code **Triad_locateSources(Year, JDay,hr1,hr2)**, located in /bin is run to associate the signal detections made at triads with an event and then to estimate the source parameters.

Year – year of interest

JDay – julian day of interest

hr1, hr2 – these parameters are optional. hr1>=0; 24>=hr2 >hr1. Only use these if a partial day is of interest. Generally, these parameters are left out to examine the entire day. Note that if these parameters are used, the detections will be displayed in several formats, as a function of amplitude vs. time of day, cross-correlation vs. time of day, etc. This can be helpful for debugging, or for choosing optimal parameters.

Appendix D describes how the code works. The Triad_detections code must be run before Triad_locateSources. Also, some more information must be supplied by the file **Triad_param2.m**, also located in /bin. The parameters to be set in Triad_param2.m are explained here, along with suggested values. In the first line, Triad_param1 is called as the source location code carries over some information set previously, for instance, triadflag.

The first parameters either set a region for the event search, or let the program decide on limits based on where the sensors were in the Triad_detections code.

ifregion2 – region in which to do a coarse source location search. I suggest setting ifregion2=1. If it is set to 0, the code sets limits based on the sensor locations.

wstlon, estlon – west/east longitudes of coarse source location search grid. lon1<lon2.

sthlat, nthlat – south/north latitudes of coarse source location search grid. lat1<lat2.

I recommend setting a broader search region than was set for the call to Triad_detections, as sources may fall outside the network specified in Triad_param1.m. *If ifregion2=0, then wstlon, estlon, sthlat, nthlat are set to null values.*

The next two parameters define the coarse time and spatial search grids.

tincoarse – coarse sample start time increments, in seconds. (see y-axis of Figure D3 in Appendix D).

coarsegrid – integer number of search points per degree for the coarse spatial grid, that is, the higher the value, the finer the search grid.

I recommend setting the time increment, `tincourse`, to be about the time it takes for a signal to transit from one coarse grid point to another. That is, the slower the signal, the greater `tincourse`.

The next few parameters define thresholds that each detection must satisfy. Low quality detections are discarded.

ampcut – minimum signal to noise ratio. I recommend setting it to a higher value than for SNR in the `param1.m` file. Set to 0 to ignore this threshold entirely.

xccut – minimum cross-correlation value. I recommend `xccut= 0.5`, possibly reducing this to 0.4 for very large, poorly correlated triad signals.

Tconscut - fractional consistency in seconds. I recommend $0.05 \leq Tconscut \leq 0.1$

vphmin – minimum phase velocity associated with detection (km/s).

vphmax - maximum phase velocity associated with detection (km/s).

I suggest fairly loose constraints for these values; `vphmax > vphmin`. Note that `vphmin` is already set in `Triad_param1.m`, see Appendix E.1 Comment the value out if you don't want to reset it.

The next few parameters govern how signals are grouped together to form an event association. See Figure D3 in Appendix D.

celermin – minimum group velocity (km/s)

celermax – maximum group velocity (km/s)

I suggest tighter constraints than for phase velocities, so

`vphmax > celermax > celermin > vphmin`

mincluster - minimum number of detections to define a source

Afit – the source receiver azimuths must fit to within `Afit` degrees. I suggest `Afit = 15`.

clock – set this value to 1 for careful de-bugging, i.e to look at cluster of detections for each event. I recommend `clock= 0`

A few more variables are set here. Leave them alone; they have to do with plotting in case `clock = 1`.

Running the `Triad_locateSources` code creates ascii format output files with the names `CoarseLocations_YYYY_JJJ_TriadflagN.txt`, `StationLocs_YYYY_JJJ_TriadflagN.txt` and `StationLocs_YYYY_JJJ_TriadflagN.txt` where `YYYY` and `JJJ` are the year and julian date, respectively, that are set when calling `Triad_locateSources (YYYY, JJJ)` and `N` is the `triadflag` value set in the `Triad_param1.m` file. These files are output to the directory `/data/events/YYYY`. The output files are explained further in Appendix F.

(4) Event association and source locations using array detections – set up array_param2.m

The description in Appendix D of how detections are sorted into events and then into

source parameter estimates applies as well to array detections.

The code **Array_locateSources(Year, JDay,hr1,hr2)**, located in /bin is run to associate the signal detections made at arrays with events and then to estimate the source parameters for each. The values Year, JDay, hr1, and hr2 are described in Appendix E.3 above. The Array_detections code must be used to detect sources before Array_locateSources is run. Also, some more information must be supplied by the file **Array_param2.m**, also located in /bin. Almost all parameters are as described in Appendix E.3 above. For completeness, all parameters are listed here; **new ones are shown in red.**

ifregion2 – as described in Appendix E.3

wstlon, estlon - as described in Appendix E.3

sthlat, nthlat - as described in Appendix E.3

tincoarse - as described in Appendix E.3

coarsegrid - as described in Appendix E.3

xccut - as described in Appendix E.3. I recommend xccut= 0.4 for arrays.

Tconscut - minimum fractional time misfit for arrays (see Eq. C1). **I recommend Tconscut = 0.6 for arrays, i.e. much higher than for triads.**

fdetmin – minimum threshold for F detector (see Eq. C2). I recommend fdetmin = 3.

vphmin - as described in Appendix E.3

vphmax - as described in Appendix E.3

celermin - as described in Appendix E.3

celermax - as described in Appendix E.3

mincluster - as described in Appendix E.3 (Obviously this must be <= the number of arrays)

Afit - as described in Appendix E.3

clock and remaining variables - as described in Appendix E.3

Running the Array_locateSources code creates ascii format output files with the names CoarseLocations_YYYY_jjj_ArrayflagM.txt, StationLocs_YYYY_jjj_ArrayflagM.txt and StationLocs_YYYY_jjj_ArrayflagM.txt where YYYY and jjj are the julian year and date, respectively, that are set when calling Array_locateSources (YYYY,jjj) and M is the triadflag value set in the Array_param1.m file. These files are output to the directory /data/events/YYYY. The output files are explained further in Appendix F.

(5) *Source locations using both triads and arrays* – set up TRIADArray_param2

Finally, the **LocateSources(Year, JDay,hr1,hr2)** code, located in /bin is run to associate the signal detections made at both triads and arrays with events and then to estimate the source parameters for each. The values Year, JDay, hr1, and hr2 are as described in Appendix E.3. Both the Triad_detections and Array_detections code must be run before LocateSources. Some information must be supplied by the file **TRIADArray_param2.m**, also located in /bin. Most parameters have been described in Appendices

E.1 through E.4. For completeness, all parameters are listed here; **new ones are shown in red.**

triadflag - as described in Appendix E.1

arrayflag - as described in Appendix E.2

ifregion2 – as described in Appendix E.3

wstlon, estlon - as described in Appendix E.3

sthlat, nthlat - as described in Appendix E.3

tincoarse - as described in Appendix E.3

coarsegrid - as described in Appendix E.3

The next few parameters are thresholds for the triad detections

triadxccut – minimum cross-correlation value for triads (see Appendix E.3 for xccut)

triadTconscut – minimum fractional consistency for triads (see Appendix E.3 for Tconscut)

ampcut - as described in Appendix E.3

The next few parameters are thresholds for the array detections

arrayxccut – minimum cross-correlation value for arrays (see Appendix E.4 for xccut)

arrayTconscut – minimum fractional time misfit for arrays (see Appendix E.4 for Tconscut)

fdetmin - as described in Appendix E.4

Afit - as described in Appendix E.3

vphmin - as described in Appendix E.3

vphmax - as described in Appendix E.3

celermin - as described in Appendix E.3

celermax - as described in Appendix E.3

mincluster - as described in Appendix E.3 (this is the number of triad + array detections)

clock and remaining variables - as described in Appendix E.3

Running the LocateSources code creates ascii format output files with the names CoarseLocations_YYYY_jjj_BothflagMN.txt, StationLocs_YYYY_jjj_BothflagMN.txt and StationLocs_YYYY_jjj_BothflagMN.txt where YYYY and jjj are the julian year and date, respectively, that are set when calling LocateSources (YYYY,jjj) . The value MN is given by 10*triadflag + arrayflag, i.e., M and N are the triadflag and arrayflag values, respectively, from the TRIADarray_param2.m file, if each is less than 10. These files are output to the directory /data/events/YYYY. The output files are explained further in Appendix F.

APPENDIX F

USER MANUAL - OUTPUT

Running the Triad_detections and Array_detections codes produce binary output files of the type triadsRaw_ddd_yyyy_flagN.out and arraysRaw_ddd_yyyy_flagN.out as described in Appendix E.1 and E.2. These files give information on the attributes of the detections (i.e. their arrival time, azimuth, amplitude, etc). These are only interim files and their formats won't be described here. An interested reader could examine the Triad_detections and Array_detections codes to see how each is formatted, or examine read_rawTriads and read_rawArrays to see how these files are read in.

The ascii files produced by the Triad_locateSources, Array_locateSources, and LocateSources sources are described here.

(1) Coarse location files

The coarse location file produced by Triad_locateSources has 11 columns for grouping of detections that are common to each event and lists rough source parameter estimates for each. The events are not listed chronologically within each day file, but instead from the events with the greatest number of detections at unique triads, to the least.

column 1 – year
column 2 – julian day
column 3 – hour
column 4 – minute
column 5 – second
column 6 – approximate source latitude
column 7 – approximate source longitude
column 8 – number of triad detections used to define the event
column 9 – number of **unique** triads to define the event (always <= than column 8)
column 10 – minimum latitude error
column 11 – minimum longitude error

The coarse location file produced by Array_locateSources is similar except that column 8 is the number of array detections used to define the event and column 9 is the number of unique arrays to define the event (sometimes a given triad or array detects a signal more than once given the time overlap between windows).

The coarse location file produced by LocateSources, which finds events based on both triad and array detections, is similar except that it has 12 columns.

Columns 1-7 are as before,
column 8 - total number of detections from both triads and arrays
column 9 – number of unique triads that define the event
column 10 – number of unique arrays that define the event
Columns 11 and 12 are the minimum latitude and longitude errors, as above.

(2) Source location files – finer scale location and origin times

The source location file produced by Triad_locateSources has 19 columns for each event. These are sorted into approximate chronological order for each day, based on initial time estimates for the coarse locations.

Column 1 – event ID (epoch time)

Column 2 – year

Column 3 – julian day

Column 4 – hour

Column 5 – minute

Column 6 – second

Column 7 –source latitude (degrees)

Column 8 –source longitude (degrees)

Column 9 – minor axes of error ellipse (in km)

Column 10 – major axes of error ellipse (in km)

Column 11 – angle of error ellipse (in degrees) with respect to North

Column 12 – mean azimuth misfit (in degrees)

Column 13 – group slowness estimate (sec/km)

Column 14 - total number of triad detections for the event

Column 15 – number of unique detecting triads that detected the event

Column 16 - number of stations included in detecting triads

Column 17 – range from source to nearest detecting station (in km)

Column 18 – angle subtended by all detecting stations as seen from the source (see notes below)

Column 19 – gridfit (see notes below for description)

Column 18: The angle, in degrees, subtended by all stations that detect the source, as seen from the source. This value lies between 0 to 360. For instance, if the detecting stations surround the source, the value will be close to 360 degrees. Conversely, this value will be small if all the detecting stations lie within a narrow wedge to one side of the source.

Column 19: Gridfit is a quality control value. The value is 1 if the final source location in this file is beyond the source location estimate given in the coarse location file, meaning that the final value is far from the initial estimate. The value is 2 if the vslow estimate (column 13) is less than or equal to zero. The default for this value is 0. However, there are also several other ways of assessing quality of the estimated source location. The larger the area of the error ellipse (given by $\pi \times \text{column 9} \times \text{column 10}$) the poorer the source location. The number of unique triads used to detect the event (column 15) is a measure of the both the source magnitude and its location accuracy. The range from the nearest detecting station (column 17) and the angle subtended by all stations that detected the event (column 18) may also be used to assess the quality of the source estimates.

The source location file produced by Array_locateSources also contains 19 columns. The only difference is that, for lines 14 through 16, the description would read 'arrays' instead of 'triads'.

The source location file produced by LocateSources has 20 columns. Columns 1 through 13 are as described above. The remainder of the columns are
Column 14 – total number of detections for the event, both triads and arrays
Column 15 – number of unique detecting triads that detected the event
Column 16 – number of unique detecting arrays that detected the event
Column 17 – - number of stations included in detecting triads and arrays
Columns 18 through 20 are as described for columns 17 through 19 above.

(3) Station Location files

All station location files have 3 columns, whether they are produced by Triad_locateSources, Array_locateSources, or LocateSources.

Column 1 – event ID (epoch time)
Column 2 – station latitude
Column 3 – station longitude

The event ID of column 1 can be matched against that of the source location files above. All station locations within each triad or each detecting array are listed.A

APPENDIX G AELUMA CODE

The complete software package was delivered to AFRL at the same time as the report.

APPENDIX H NEIC CATALOGS

Two catalogs are used in this report to gauge efficacy of the AELUMA method. They were delivered to AFRL at the same time as the report. One is in the file, "NEIC_Jan2018.txt," and was used as part of our code development in North America. The second catalog, "NEIC 2017.txt," was used to test the method when applied to IMS data in Europe, North Africa and Asia.

APPENDIX I AELUMA CATALOGS

We also delivered to the AFRL two of the AELUMA catalogs produced during this contract. The catalogs cover the same regions and time ranges that were covered by the NEIC catalogs that were discussed in Appendix G. The two files are: "AELUMACatalog_January2018," which lists events detected in the North American study region in January 2018, and "AELUMACatalog_midJulmidAug2017," which lists events that were detected by AELUMA in the IMS study.

Approved for public release; distribution is unlimited.

List of Symbols, Abbreviations, and Acronyms

AELUMA	Automatic Event Location Using a Mesh of Arrays
AFRL	Air Force Research Laboratory
AFSPC	Air Force Space Command
AFWA	Air Force Weather Agency
IMS	International Monitoring System
NEIC	National Earthquake Information Center
TA	Transportable Array
USGS	United States Geological Survey

DISTRIBUTION LIST

DTIC/OCF	
8725 John J. Kingman Rd, Suite 0944	
Ft. Belvoir, VA 22060-6218	1 cy
AFRL/RVIL	
Kirtland AFB, NM 87117-5776	1 cy
Official Record Copy	
AFRL/RVB/Dr. Raymond J. Willemann	1 cy

This page is intentionally left blank.