geoStudio: An improved zero-lag cross-correlation code and GUI for the estimate of apparent slowness vectors using small-aperture seismic arrays

Cortés, G.[1], Almendros, J.[1,2]

[1]Andalusian Institute of Geophysics, University of Granada, Spain

[2]Department of Theoretical and Cosmos Physics, University of Granada, Spain

Version July 17, 2017, to be submitted to *Journal Name*

Abstract.

1. Introduction

Array methods constitute a powerful tool that is used to extract information about the seismic wavefield. They are able to analyze signals that lack identifiable seismic phases, as long as they keep a minimum of coherency among stations distributed in an area (or volume). The lack of identifiable phases can be due to a low signal to noise ratio. For example this happens in the analysis of distant, relatively small seismic events; the study of weak seismic phases produced at the discontinuities of the Earth structure; etc. This lack of identifiable phases can be also due to the intrinsic absence of seismic phases, for example in the analysis of volcanic tremor, background noise, etc. Moreover, array data can be used to investigate the fine structure of the Earth interior [REFS].

Array techniques were developed during the 60s, in relation with the advances in seismic monitoring of nuclear explosions in the framework of the Comprehensive Test Ban Treaty [e.g. *Ringdal and Husebye*, 1982; *Rost and Thomas*, 2002]. Several methods have been proposed to extract the information about the propagation characteristics of the seismic wavefields, including time-domain and frequency-domain beam-forming [*Lacoss et al.*, 1969], the high-resolution method [*Capon*, 1969], the multiple signal classification method [*Schmidt*, 1986; *Goldstein and Archuleta*, 1987], and the zero-lag cross-correlation method [*Frankel et al.*, 1991; *Del Pezzo et al.*, 1997].

In this paper, we focus on the implementation of the zero-lag cross-correlation method for the analysis of small-aperture seismic array data. We describe a code written around twenty years ago, that has been extensively used during these years. Finally, we describe several optimizations and improvements, including the development of a graphical user interface in Python, which simplifies the setting of input parameters and the representation and visualization of data and results.

2. The array-averaged cross-correlation method

The array-averaged cross-correlation method, or zero-lag cross-correlation (ZLCC) method, is based in calculations of correlation coefficients among the seismograms recorded by the stations of a seismic array [*Frankel et al.*, 1991; *Del Pezzo et al.*, 1997; *Almendros et al.*, 1999; *Almendros*, 1999]. The method computes the average cross-correlations among the seismograms, corrected for the delays related to wave propagation across the array. It is a simple and robust method, with a relatively mild dependence of the parameters of the analysis.

We start with the arrival of a plane wavefront to a seismic array composed of N stations, which are located in a given configuration on a homogeneous structure. Under these conditions, seismic wave propagation can be represented by an apparent slowness vector $\vec{s}$. The delay between wavefront arrivals at the array station i located at $\vec{r_i}$ and a reference point located at $\vec{r_0}$ (i.e. the array center) is given by [e.g. *Almendros et al.*, 1999]:

$$\tau_i(\vec{s}) = \vec{s} \cdot (\vec{r_i} - \vec{r_0}) \tag{1}$$

We correct all seismograms for these delays. With this operation, we expect to align the wavefront arrivals at the array center, so that all of them are in phase (which in practice is only true when the assumed apparent slowness vector does represent the real wavefront propagation). The delayed seismograms are:

$$u_i'(t, \vec{s}) = u_i(t + \tau_i(\vec{s})) = u_i(t + \vec{s} \cdot (\vec{r_i} - \vec{r_0})) \tag{2}$$

When the assumed apparent slowness vector corresponds to the real apparent slowness vector of the wavefield, all delayed seismograms are aligned in time, and their average cross-correlation will be maximum. The zero-lag cross-correlation of the aligned waveforms is expressed as:

$$C_{ij}(t, \vec{s}) =< u_i'(t, \vec{s}), u_j'(t, \vec{s}) > \tag{3}$$

In order to quantify the alignment of the seismograms, the method uses the array-averaged cross-correlation coefficient, computed as follows [*Frankel et al.*, 1991; *Del Pezzo et al.*, 1997]:

$$C(t,\vec{s}) = \frac{1}{N^2} \sum_{i,j=1}^{N} \frac{C_{ij}(t,\vec{s})}{\sqrt{C_{ii}(t,\vec{s})C_{jj}(t,\vec{s})}} \tag{4}$$

This quantity is interpreted as a measure of the probability that the apparent slowness vector $\vec{s}$ represents the actual slowness vector of the seismic wavefield at time t.

3. Implementation of the method

3.1. The Fortran program CC8MRE

The `CC8MRE` code was developed by *Almendros* [1999] based on a previous version by *Del Pezzo et al.* [1997]. It is a simple application of the zero-lag cross-correlation algorithm and was written in Fortran. Is has been extensively used for the analysis of small-aperture array data in volcanic settings [*Almendros et al.*, 1997, 1999, 2000; *Ibanez et al.*, 2000; *Saccorotti et al.*, 2001; *Ibanez et al.*, 2003; *Almendros et al.*, 2007; *Garcia-Yeguas et al.*, 2011; *Almendros et al.*, 2014; *Blanco et al.*, 2015].

We select an apparent slowness vector in a grid characterized by a spacing of Δs and extending from $-s_{max}$ to s_{max} in the East and North directions. Selecting a grid size that is an integer multiple of the grid spacing, the number of nodes is $N_s \times N_s$, where $N_s = 2s_{max}/\Delta s + 1$. When the apparent slowness vector is $\vec{s} = (n_x, n_y)\Delta s$, the number m of sampling intervals required by the seismic wavefronts to propagate from the array center to station i is:

$$m = (n_x \cdot (x_i - x_0) + n_y \cdot (y_i - y_0)) \frac{\Delta s}{\Delta t} \tag{5}$$

The delayed seismograms can be written as:

$$u'_i(k,n_x,n_y) = u_i\,(k+m) = u_i\left(k + (n_x \cdot (x_i - x_0) + n_y \cdot (y_i - y_0)) \frac{\Delta s}{\Delta t}\right) \tag{6}$$

(Since the argument of u_i must be an integer number, rounding to the nearest integer is implitic in this equation). The cross-correlation between these delayed seismograms, using a window of W samples starting at sample k, is given by:

$$C_{ij}(k, n_x, n_y) = \sum_{l=k}^{k+W-1} u'_i(l, n_x, n_y) \cdot u'_j(l, n_x, n_y) \tag{7}$$

The array-averaged zero-lag cross-correlation coefficient is then:

$$C(k, n_x, n_y) = \frac{1}{N^2} \sum_{i,j=1}^{N} \frac{C_{ij}(k, n_x, n_y)}{\sqrt{C_{ii}(k, n_x, n_y) C_{jj}(k, n_x, n_y)}} \tag{8}$$

In order to reduce the calculation times, we take advantage of some properties of the zero-lag cross-correlations. First, we observe that the terms with $i = j$ contribute with 1 to the sum, that is:

$$i = j \Rightarrow \frac{C_{ij}(k, n_x, n_y)}{\sqrt{C_{ii}(k, n_x, n_y) C_{jj}(k, n_x, n_y)}} = 1 \tag{9}$$

Following *Del Pezzo et al.* [1997], we also identify a symmetry in the zero-lag cross-correlations:

$$C_{ij}(t, n_x, n_y) = C_{ji}(t, n_x, n_y) \tag{10}$$

With this two considerations, we deduce that Equation 8 can be expressed as:

$$C(k, n_x, n_y) = \frac{1}{N} + \frac{2}{N^2} \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} \frac{C_{ij}(k, n_x, n_y)}{\sqrt{C_{ii}(k, n_x, n_y) C_{jj}(k, n_x, n_y)}} \tag{11}$$

The flow diagram of `CC8MRE` is provided in Figure 1. The steps of the analysis are: (1) read input parameters; (2) read seismic data files; (3) prepare data and perform preliminary calculations; (4) perform main calculations; and (5) generate output files.

Required parameters are:

- The name of the data files where seismic traces are stored. There are three versions of the program, that are able to read three different seismic formats. These include: SAD/DTS files (a format designed by the Andalusian Institute of Geophysics and used in early prototypes of the seismic arrays); SAC format (http://ds.iris.edu/ds/nodes/dmc/software/downloads/sac), and SEISAN files [*Havskov and Ottemoller*, 1999].

- Coordinates of the seismic array: they are given in a text file containing N rows and two columns with East and North coordinates of the array stations (in km). They can be relative coordinates with respect to the center of the array or any other reference point.

- Parameters for the filter: number of samples to skip, number of samples to read, and low and high limits for the filter. The filter is defined as a 2-pole, zero-phase, band-pass Butterworth filter.

- Parameters for the temporal windows: initial sample of the analysis, number of windows, window length (in samples), and advance factor of the time window (in units of window length). The window length is recommended to contain around 2-3 cycles of the signal at the frequency of interest [*Almendros et al.*, 1999].

- Parameters for the apparent slowness grid: extent of the domain and grid interval (both in s/km). The grid interval is recommended to be around the value $\frac{\Delta t}{A}$, where A is the array aperture. This is the minimum variation in slowness that produces a change of at least 1 sample in the delay between the most distant stations.

- Parameter for the uncertainty calculations: percentage of the maximum correlation used to obtain the error estimates. A reference value of 0.05 (equivalent to 95% of the maximum correlation) is generally used.

The program can be run with two or three command line arguments:

```
$ cc8mre test.inp test.out [test.res]
```

The first and second are mandatory, and the third is optional. The first argument is the name of the input file, containing all the parameters required for the analysis. The second and third arguments are output files.

The main output file (second argument) has one row for each of the `nwin` time windows analyzed. They contain the apparent slowness vectors corresponding to the maximum array-averaged correlation coefficients found in the given apparent slowness domain. There are eight columns: column 1 is the time of the center of the window (in seconds from the beginning of the file); columns 2-4 contain the apparent slowness (in s/km); columns 5-7 show the back-azimuth (in degrees); and the maximum cross-correlation is found in column 8. Apparent slownesses and back-azimuths are given in three columns as $l - c - h$, where c is the central value corresponding to the maximum cross-correlation, and l, h represent the lower and upper uncertainty limits.

The optional output file (third argument) contains the complete distributions of zero-lag cross-correlations on the apparent slowness vector domain for each of the time windows. In this case the file has three columns with the East and North components of the apparent slowness vector (in s/km) and the corresponding zero-lag cross-correlation. The file has `nslo*nslo*nwin` rows, so eventually it can be a large file. We recommend its application only to limited time spans.

3.2. Visualization software

In this section we show the routines used to display the results of the CC8MRE program. They are written in Matlab, and allow for quick representations that help in the interpretation of the results.

For example, we can visualize the beamforming array response for a given configuration of stations using the code `plotresp.m`. The array response is defined as:

$$R(\vec{s}) = \left| \frac{1}{N} \sum_{j=1}^{N} \exp\left(i2\pi f\left(\vec{s} \cdot \vec{r}_j\right)\right) \right|^2 \tag{12}$$

This function provides a glimpse of the array resolution and is very useful to take decisions about the input parameters. The program can be executed as:

```
>> plotresp(coordfile,[freq(Hz)],[stats],[smax(s/km])
>> plotresp('test.xy',2,[1:4 6:10],2)
```

where `coordfile` is the path and name of the file with the East and North coordinates in km; `freq` is the frequency to display the array response; `stats` is the index of the stations to be used in the calculation; and `smax` is the size of the apparent slowness domain. Parameters in brackets are optional, and default values are defined in the code.

The program `plotout.m` displays the results of `CC8MRE` as time series of average cross-correlation, apparent slowness, and back-azimuth, including the uncertainties. It is used as:

```
>> plotout(outfile,[tlim(s)],[ccmin],[slim(s/km)],[alim(deg)])
>> plotout('test.out',[120 600],0.6,[0.2 0.8],[90 180])
```

where `outfile` is the path and name of the file containing the results of the zero-lag cross-correlation method. Several filters of the data can be applied using the remaining (optional) parameters. For example `tlim` allows to select a particular time window; `ccmin` is a threshold cross-correlation value, only values above it are displayed; `slim` and `alim` are ranges of apparent slowness and back-azimuth, that allow to focus on apparent slowness vectors within particular limits.

The program `plotslow.m` shows the distributions of zero-lag cross-correlation in the apparent slowness domain, which is an optional output of the `CC8MRE` code. The program is used as:

```
>> plotslow(resfile)
>> plotout('test.res')
```

where `resfile` is the file containing the apparent slowness vectors and the corresponding cross-correlations (the third argument of the `cc8mre` command line).

Finally, the program `plothist2d.m` divides the apparent slowness vector plane in cells and shows a 2D histogram of the total number of solutions within each cell. The program is used as:

```
>> plothist2d(outfile,[tlim(s)],[ccmin],[shist(s/km)])
>> plothist2d('test.out',[120 600],0.6,[2 0.2])
```

where `tlim` and `ccmin` are the time limits and minimum correlation to be considered (as in `plotout` above), and `shist` contains the apparent slowness vector domain and bin sizes (in s/km) for the histogram counts.

3.3. Overlapping

During the applications of the `CC8MRE` code, we realized that very often we use overlapping windows. In those cases, part of the products and sums in l (Equation 7) were already calculated in previous steps. The analysis of long-duration signals such as volcanic tremor is usually addressed using overlapping windows.

3.4. The C program

In the `CC8MRE` code, the products of the aligned traces defined in Equation 7 are calculated for each window, and then forgotten for the next window. However, when the successive temporal windows overlap P samples, we find that P of the required products were already calculated in the previous window. This implies a repetition of operations already done, that can be avoided.

Let us assume that we want to compute array-average cross-correlations of array data, using M windows of W samples, overlapping by P samples, and an apparent slowness grid of $N_s \times N_s$ nodes. The length of data D involved in the analysis is:

$$D = W + (M-1)(W-P) = M(W-P) + P \Rightarrow M \approx \frac{D}{AW} \tag{13}$$

where we define the percentage of advance of the moving window $A = 1 - P/W$ and assume that the total length of data is large compared to the window overlapping ($D >> P$).

For each cross-correlation in Equation 7 we must perform $2W - 1$ operations (W multiplications and $W - 1$ sums). This is repeated three times to compute the cross-correlation in the numerator and the two auto-correlations in the denominator of Equation 11. With the additional operations required (multiplication of auto-correlations, square root, quotient) we need a total of $6W$ operations to compute the term within the sums. This is repeated: (1) for each of the $N(N-1)/2$ station pairs; (2) for each of the $N_s \times N_s$ apparent slowness vectors; and (3) for each of the M temporal windows. Therefore, the total number of operations is:

$$Q_1 = 3WMN(N-1)N_s^2 \tag{14}$$

$$Q_1 \approx \frac{3D}{A}N(N-1)N_s^2 \tag{15}$$

The C program takes a different approach. It is optimized in the sense that operations are performed as few times as possible. First of all, for each given apparent slowness vector $(n_x, n_y)\Delta s$, a matrix of delays (in samples) is calculated and stored, so they are not recalculated again for each window. To perform the correlations, samples are multiplied for the whole length of data D and stored in a matrix, so the products need not be calculated again. The number of multiplications is DN for the autocorrelations and $DN(N-1)/2$ for the cross-correlations, giving a total of $DN(N+1)/2$ multiplications. As for the sums, we divide the data length in segments with length $W-P$, which is the number of samples that the analysis window advances in each step. To obtain the correlation for a window of length W, we need to add the sum of the products for the segments contained in the window, plus the sum of the R remaining samples. The number of complete segments in a window is:

$$N_P = floor\left(\frac{W}{W-P}\right) \Rightarrow W = N_P(W-P) + R \Rightarrow N_P = \frac{W-R}{W-P} \tag{16}$$

The number of sums within each segment is $W-P-1$. We perform the sums of the $W-P$ products contained in the segments for the whole data length D, obtaining $D/(W-P)$ sums. These sums need not be calculated again. For each of the M windows, we just need to add the N_P sums and then the sum of the remaining R samples. Thus the total number of sums to compute the correlations is:

$$Q_{sums} = (W-P-1)\frac{D}{W-P} + (N_P - 1 + R)M$$

$$
\begin{aligned}
&= \frac{W-P-1}{W-P}D + \left(\frac{W-R}{W-P} - 1 + R\right)M \\
&= \frac{W-P-1}{W-P}(D+RM) + \frac{P}{W-P}M
\end{aligned} \tag{17}
$$

As before, we have to repeat this operation N times to calculate the auto-correlations, and $N(N-1)/2$ times to calculate the cross-correlations. The total number of operations is:

$$
Q_2 = \left(D\frac{N(N+1)}{2} + \left(\frac{W-P-1}{W-P}(D+RM) + \frac{P}{W-P}M\right)\frac{N(N+1)}{2} + 3M\frac{N(N-1)}{2}\right)N_s^2 \tag{18}
$$

$$
Q_2 \approx \frac{1}{2W}\left(\left(2AW + \frac{1-2A}{A} + R\left(1-\frac{1}{AW}\right)\right)\frac{N+1}{N-1} + 3\right)\frac{D}{A}N(N-1)N_s^2 \tag{19}
$$

The ratio between the number of operations in the two cases is:

$$
R = \frac{Q_2}{Q_1} = \frac{1}{6W}\left(\left(AW + (AW-1)\left(1+\frac{R}{AW}\right) + \frac{1-A}{A}\right)\frac{N+1}{N-1} + 3\right) \tag{20}
$$

$$
= \frac{1}{2W} + \frac{A}{6}\left(2 + \frac{1-2A}{A^2W} + \frac{R}{AW}\left(1-\frac{1}{AW}\right)\right)\frac{N+1}{N-1} \tag{21}
$$

This ratio is represented in Figure 4 for different numbers of stations N. It represents the reduction in the number of operations and, as a first approximation, in computing time. The performance of the C program improves with the amount of overlapping, reaching a minimum for large overlapping near 90%. In this case, the computing time is just 10% of the time required by `CC8MRE`. Performance also improves with the number of stations, the time reduction is more noticeable for large-N arrays. The worst situation is a tripartite array with no overlapping, when the computation time reduces just to 2/3 of the time required by `CC8MRE`.

3.5. The integrated GUI

We have developed a Graphical User Interface named `geoStudio`. It is based in Python and uses the libraries WxPython, Numpy, Scipy, Pyplot, Matplotlib, and Obspy. Although it is intended for more general purposes, here we will focus on the implementation of the array analysis.

The main control window has different options

4. Aplication to real data

In this section we illustrate the capabilities of geoStudio by performing an analysis of real data from Deception Island volcano, Antarctica. During the last two decades, we have deployed a seismic array at Fumarole Bay in three-month-long summer surveys. Since 2005 we use a 12-channel, 24-bit seismic array developed at the Andalusian Institute of Geophysics [*Abril*, 2007], which improved the capabilities of previous designs [*Almendros*, 1999; *Havskov and Alguacil*, 2004, chapter 9]. Sampling is synchronized by GPS and is set at 100 samples per second in each channel. Short-period geophones with response electronically extended to 1 Hz are connected by cable to the recording center. These stations are spread over a smooth slope, with an aperture of about 400 m. Seismic data are locally stored on an external device. In the most recent versions, data are also telemetered via wifi to a central recording site running seiscomp3 [*Carmona et al.*, 2014].

The data contains long-period seismicity

FIGURES OF ARRAY LOCATION AND CONFIGURATION, ARRAY RESPONSE, SEISMOGRAM AND SPECTROGRAM, SLOWNESS & BACKAZIMUTH VS TIME, CORRELATION VS SLOWNESS X,Y, ETC

5. Conclusions

IMPROVEMENTS OF THE SOFTWARE COMPARED TO THE PREVIOUS VERSION

SUGGESTIONS FOR FUTURE IMPROVEMENTS

Acknowledgments. We thank XXX and XXX.

References

Abril, M. (2007), Evolución, diseño y desarrollo de antenas sísmicas. Las antenas del Gran Sasso, del Vesubio y las nuevas antenas sísmicas portátiles del Instituto Andaluz de Geofísica. Aplicación a zonas tectónicas y volcánicas, Ph.D. thesis, University of Granada, Spain.

Almendros, J. (1999), Análisis de señales sismo-volcánicas mediante técnicas de array, Ph.D. thesis, University of Granada, Spain.

Almendros, J., J. M. Ibanez, G. Alguacil, E. Del Pezzo, and R. Ortiz (1997), Array tracking of the volcanic tremor source at Deception Island, Antarctica, *Geophys. Res. Lett.*, *24*(23), 3069–3072, doi:10.1029/97GL03096.

Almendros, J., J. M. Ibanez, G. Alguacil, and E. Del Pezzo (1999), Array analysis using circular-wave-front geometry: An application to locate the nearby seismo-volcanic source, *Geophys. J. Int.*, *136*(1), 159–170, doi:10.1046/j.1365-246X.1999.00699.x.

Almendros, J., J. M. Ibanez, G. Alguacil, J. Morales, E. Del Pezzo, M. La Rocca, R. Ortiz, V. Arana, and M. J. Blanco (2000), A double seismic antenna experiment at Teide volcano: Existence of local seismicity and lack of evidences of volcanic tremor, *J. Volcan. Geotherm. Res.*, *103*(1-4), 439–462, doi:10.1016/S0377-0273(00)00236-5.

Almendros, J., J. M. Ibanez, E. Carmona, and D. Zandomeneghi (2007), Array analyses of volcanic earthquakes and tremor recorded at Las Cañadas caldera (Tenerife Island, Spain) during the 2004 seismic activation of Teide volcano, *J. Volcan. Geotherm. Res.*, *160*, 285–299, doi:10.1016/j.jvolgeores.2006.10.002.

Almendros, J., R. Abella, M. Mora, and P. Lesage (2014), Array analysis of long-period events and volcanic tremor wavefields at Arenal volcano, Costa Rica, *J. Geophys. Res.*, *119*, 5536–5559, doi:10.1002/2013JB010628.

Blanco, M. J., E. Fraile-Nuez, A. Felpeto, J. M. Santana-Casiano, R. Abella, L. M. Fernandez-Salas, J. Almendros, V. Diaz-del Rio, I. Dominguez Cerdena, L. Garcia-Canada, M. Gonzalez-Davila, C. Lopez, N. Lopez-Gonzalez, S. Meletlidis, and J. T. Vazquez (2015), Comment on ”Evidence from acoustic imaging for submarine volcanic activity in 2012 off the west coast of El Hierro (Canary Islands, Spain)” by Perez NM, Somoza L, Hernandez PA, Gonzalez de Vallejo L, Leon R, Sagiya T, Biain A, Gonzalez FJ, Medialdea T, Barrancos J, Ibanez J, Sumino H, Nogami K and Romero C [Bull Volcanol (2014) 76:882-896], *Bull. Volcan.*, *77*(7), 1–8, doi:10.1007/s00445-015-0947-6.

Capon, J. (1969), High-resolution frequency-wavenumber spectrum analysis, *Proceedings of the IEEE*, *57*(8), 1408–1418.

Carmona, E., J. Almendros, R. Martin, G. Cortes, G. Alguacil, J. Moreno, J. B. Martin, A. Martos, I. Serrano, D. Stich, and J. M. Ibanez (2014), Advances in seismic monitoring at Deception Island volcano (Antarctica) since the International Polar Year, *Annals of Geophysics*, *57*(3), SS0321, doi:10.4401/ag-6378.

Del Pezzo, E., M. La Rocca, and J. M. Ibanez (1997), Observations of high-frequency scattered waves using dense arrays at Teide Volcano, *Bull. Seism. Soc. Am.*, *87*(6), 1637–1647.

Frankel, A., S. E. Hough, P. Friberg, and R. Busby (1991), Observations of Loma Prieta aftershocks from a dense array in Sunnyvale, California, *Bull. Seism. Soc. Am.*, *81*(5), 1900–1922.

Garcia-Yeguas, A., J. Almendros, R. Abella, and J. M. Ibanez (2011), Quantitative analysis of seismic wave propagation anomalies in azimuth and apparent slowness at Deception Island volcano (Antarctica) using seismic arrays, *Geophys. J. Int.*, *184*, 801–815, doi: 10.1111/j.1365-246X.2010.04864.x.

Goldstein, P., and R. J. Archuleta (1987), Array analysis of seismic signals, *Geophys. Res. Lett.*, *14*(1), 13–16, doi:10.1029/GL014i001p00013.

Havskov, J., and G. Alguacil (2004), *Instrumentation in Earthquake Seismology*, Springer, doi:10.1007/978-1-4020-2969-1.

Havskov, J., and L. Ottemoller (1999), SEISAN earthquake analysis software, *Seism. Res. Lett.*, *70*, 532–534.

Ibanez, J. M., E. Del Pezzo, J. Almendros, M. La Rocca, G. Alguacil, R. Ortiz, and A. Garcia (2000), Seismovolcanic signals at Deception Island volcano, Antarctica: wave field analysis and source modeling, *J. Geophys. Res.*, *105*(B6), 13,905–13,931, doi: 10.1029/2000JB900013.

Ibanez, J. M., E. Carmona, J. Almendros, G. Saccorotti, E. Del Pezzo, M. Abril, and R. Ortiz (2003), The 1998-1999 seismic series at Deception Island volcano, Antarctica, *J. Volcan. Geotherm. Res.*, *128*(1-3), 65–88, doi:10.1016/S0377-0273(03)00247-6.

Lacoss, R. T., K. E. J, and M. N. Toksoz (1969), Estimation of seismic noise structure using arrays, *Geophysics*, *34*(1), 21–38, doi:10.1190/1.1439995.

Ringdal, F., and E. S. Husebye (1982), Application of arrays in the detection, location, and identification of seismic events, *Bull. Seism. Soc. Am.*, *72*(6), S201–S224.

Rost, S., and C. Thomas (2002), Array seismology: Methods and applications, *Rev. Geophys.*, *40*(3), 1008, doi:10.1029/2000RG000100.

Saccorotti, G., J. Almendros, E. Carmona, J. M. Ibanez, and E. Del Pezzo (2001), Slowness anomalies from two dense seismic arrays at Deception Island volcano, Antarctica, *Bull. Seism. Soc. Am.*, *91*(3), 561–571, doi:10.1785/0120000073.

Schmidt, R. (1986), Multiple emitter location and signal parameter estimation, *IEEE Trans. Ant. Prop.*, *34*(3), 276–280, doi:10.1109/TAP.1986.1143830.

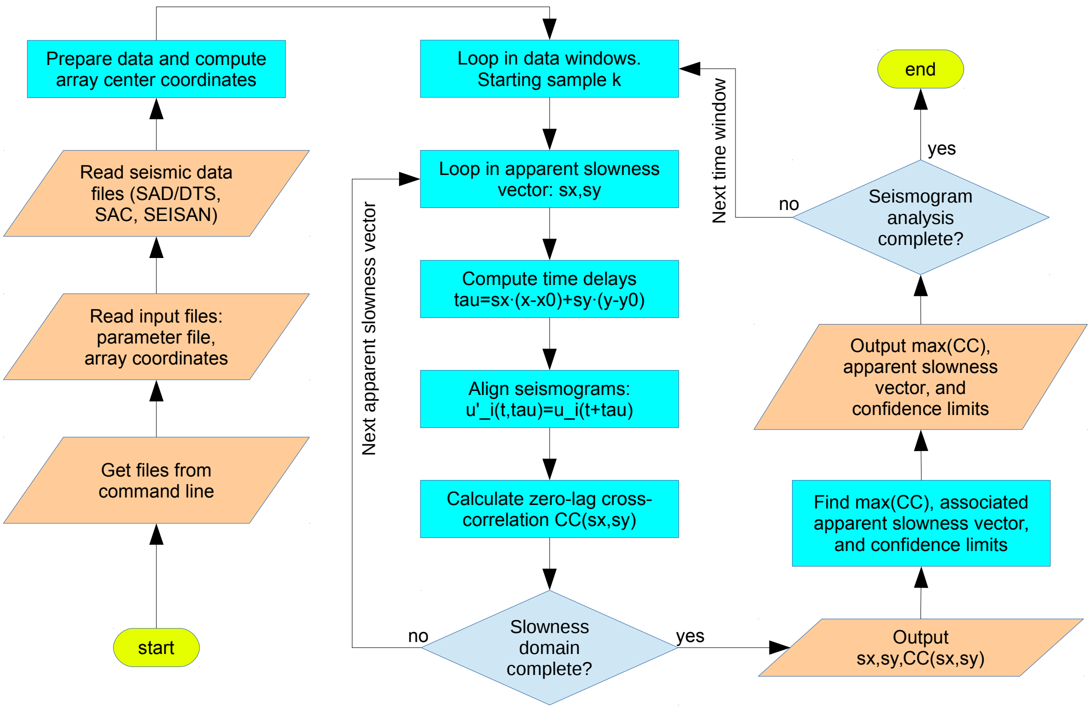

Figure 1. Flow diagram of CC8MRE.

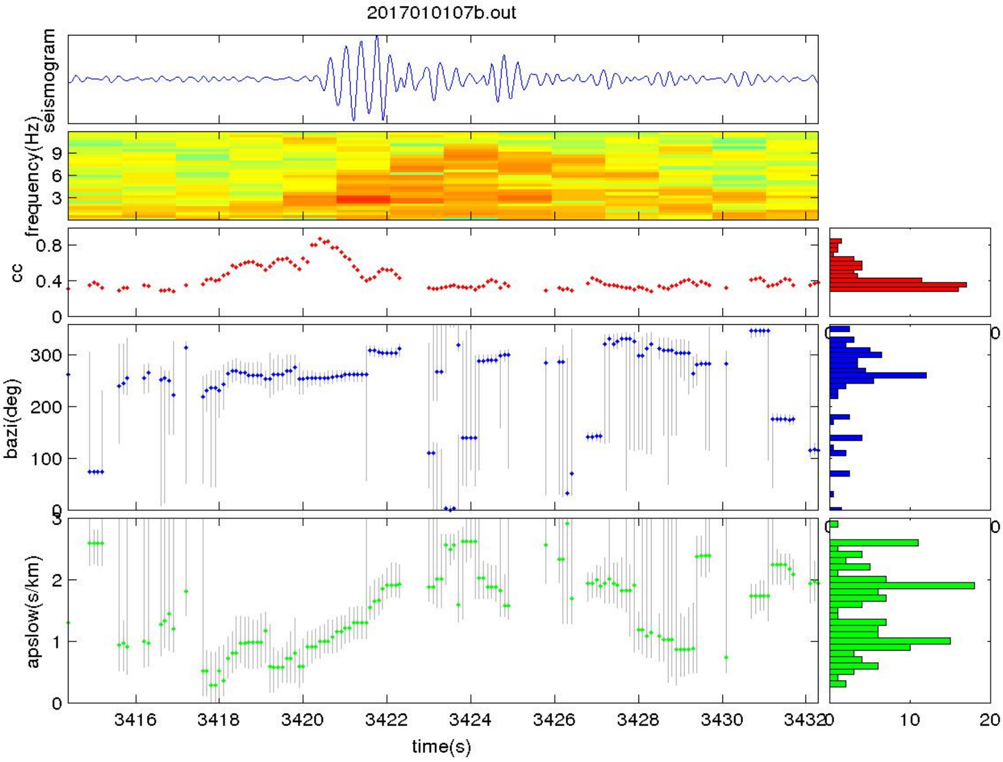

Figure 2. Figure caption.

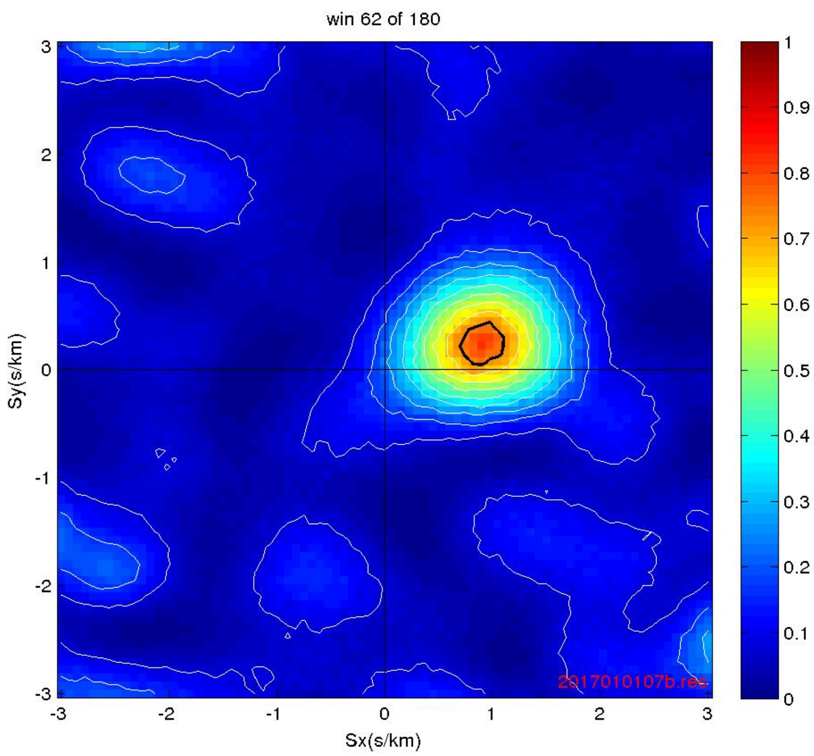

Figure 3. Figure caption.

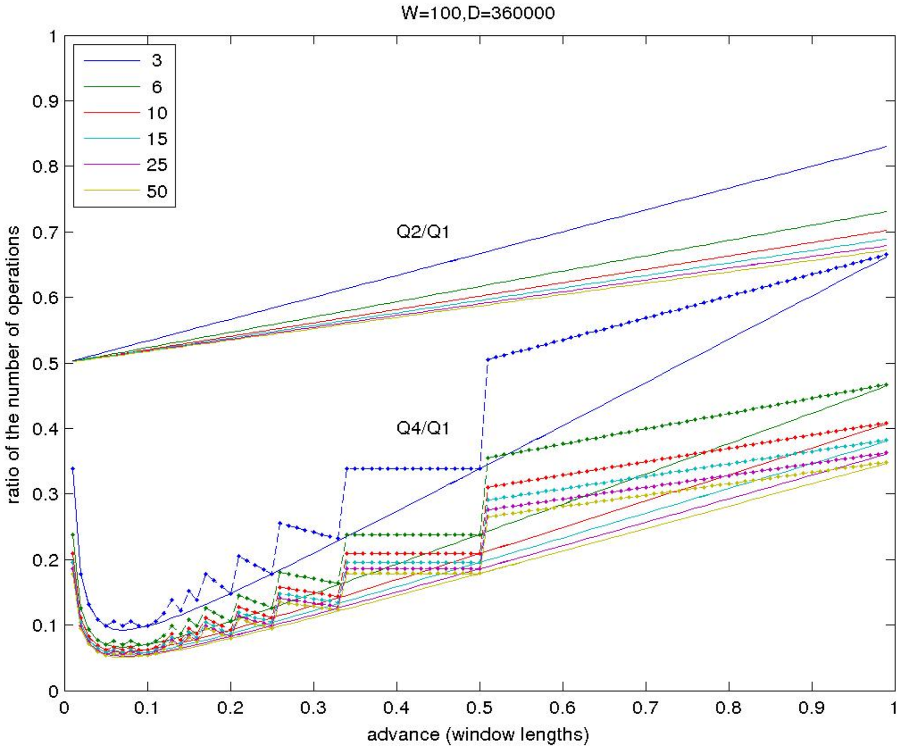

Figure 4. Figure caption.

Figure 5. Figure caption.