SAND2021-15708C

Los Alamos NATIONAL LABORATORY

An Automatic Tool for Data QC in Seismic Arrays using Signal Singular Values. (IN42B-02)

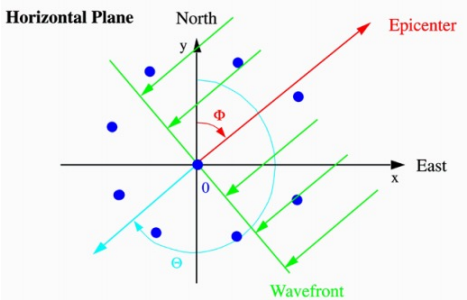
Charlotte Rowe, Los Alamos National Laboratory
Stephen Heck, Sandia National Laboratories

16 December, 2021

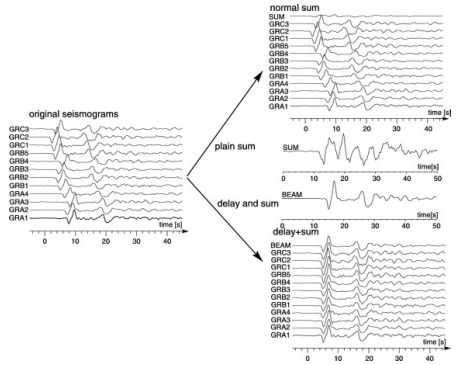
LA-UR-21-32104

Sandia National Laboratories is a multi-mission laboratory managed and operated by National Technology & Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under Contract Number DE-NA0003525

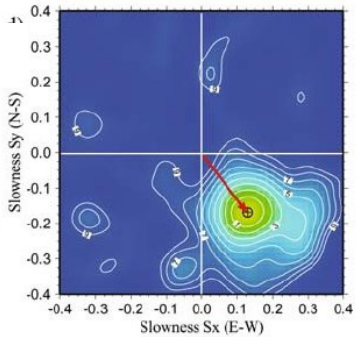
From Wikipedia: A **seismic array** is a system of linked seismometers arranged in a regular geometric pattern (cross, circle, rectangular etc.) to increase sensitivity to earthquake and explosion detection. A seismic array differs from a local network of seismic stations mainly by the techniques used for data analysis. The data from a seismic array is obtained using special digital signal processing techniques such as beamforming, which suppress noises and thus enhance the signal to noise ratio (SNR).



Seismic plane wave arriving from distant event to the northeast



Beamforming, a lag and sum technique based either on a priori knowledge of source direction, cross-correlation lags of array channels, or grid search for maximum coherence.

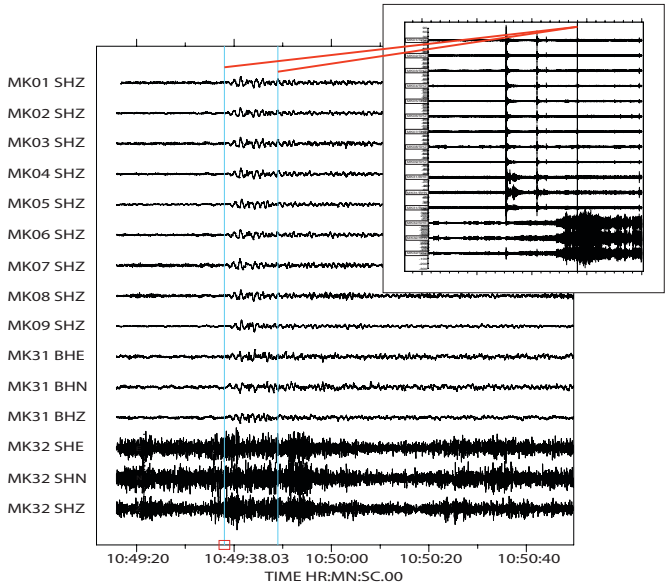


FK analysis, producing azimuth to source and slowness (1/velocity) for P-wave arrival at an array.

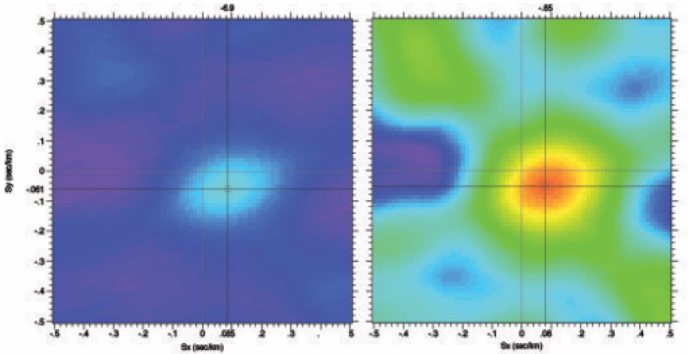
Impact of Noise on Array Analysis

The power of arrays has been well established, but as with any long time series, issues with signal quality can affect their performance.

Often a failure or contamination of one or a few components can compromise the results of array analysis methods.



Left: small Wenshuan aftershock observed at Makanchi Array. Below: FK plot for this event with (left) and without (right) noisy channels.



- The goal of this work is to build an automated algorithm that can perform quality checking on seismic array components within our database to flag those channels that may hinder the usefulness of the array for a particular research objective.

- In cases where one component is chronically bad, it can be eliminated a priori before a researcher undertakes an analysis on a long time series of data. But numerous instances of intermittent channel problems have been observed.

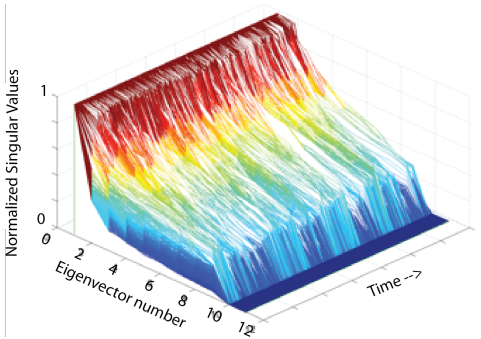
- We need to identify these without the need for laborious manual examination by our researchers to optimize their results. Flagging and removing bad channels can save time and resources, and expedite our analyses.

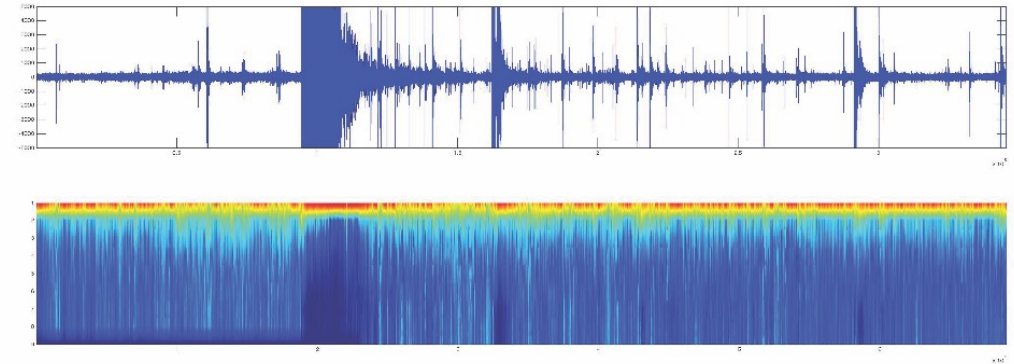
Principal components

- In our Array QC tool, we leverage a technique that has recently gained prominence among large computer system analyses, particularly in the continuing battle against malware.
- This approach focuses on anomaly detection among many hundreds or thousands of nodes on large systems. In the problem of monitoring IP flows, the signal (traffic) resides in a space that can approach 2^{100} dimensions.
- Characterizing the normal traffic and anomalous activity in a direct representation is intractable. Thus some sort of principal components or subspace method is applied.
- The sometimes independent, sometimes correlated, behavior of the computer nodes can be viewed as analogous to the sometimes independent and sometimes correlated behavior of time series across a seismic network or array.

Identifying Bad Channels

- In initial approaches, we examined the SVD function to find time periods of bad, noisy or missing array elements so that an automated analysis system might flag and/or discard them.
- Intermittent noise of sufficiently high amplitude will not be possible to characterize within the “normal” data dimension and will thus alter the the shape of the SVD curve.
- This bad channel may be undetected during ordinary operations if it is intermittent.





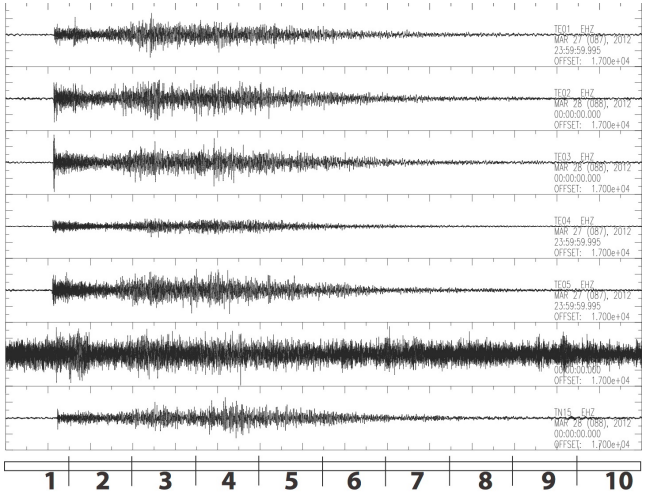
We decided to employ a jackknifing method to isolate the bad channel.

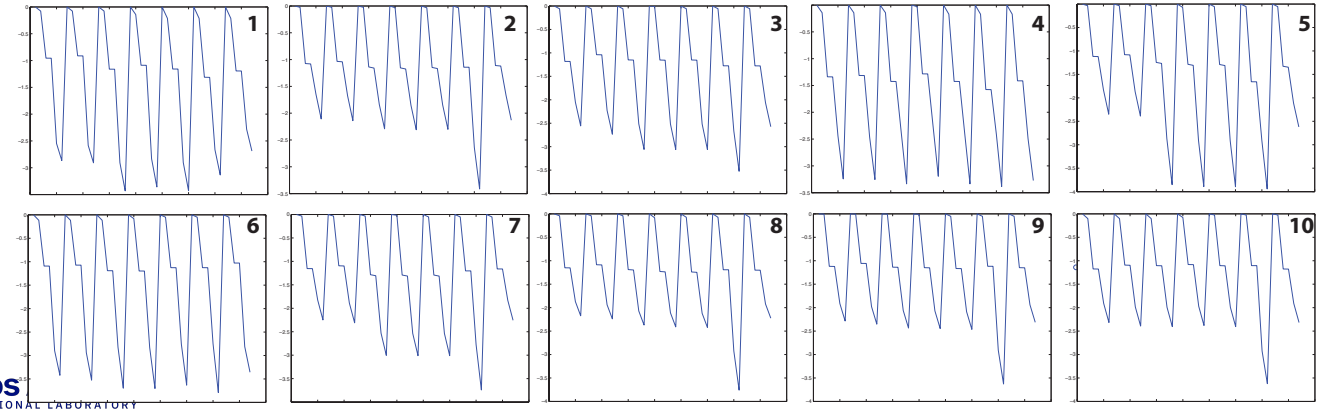
The logic behind this:

- The system of array channels, viewed as a matrix of independent vectors, should be represented by a characteristic set of singular values (or eigenvalues).
- If there is a bad channel in the system, removing a good channel shouldn't perturb the system very much (it is still skewed by the bad channel).
- If we jackknife through all the channels, we should see a significant change in the system only when the offending (bad, noisy) channel is the channel that is removed.

Jacknifing to Find the Culprit

- The impact of jackknifing on seven channels is illustrated. In each of the numbered time steps, the singular values are calculated for each of 7 jackknife realizations (below).
- When the noisy trace was removed, a change in the singular value function was generally observed for most time steps.
- Sawtooth functions in the numbered panels below represent each jackknife step for the time window.

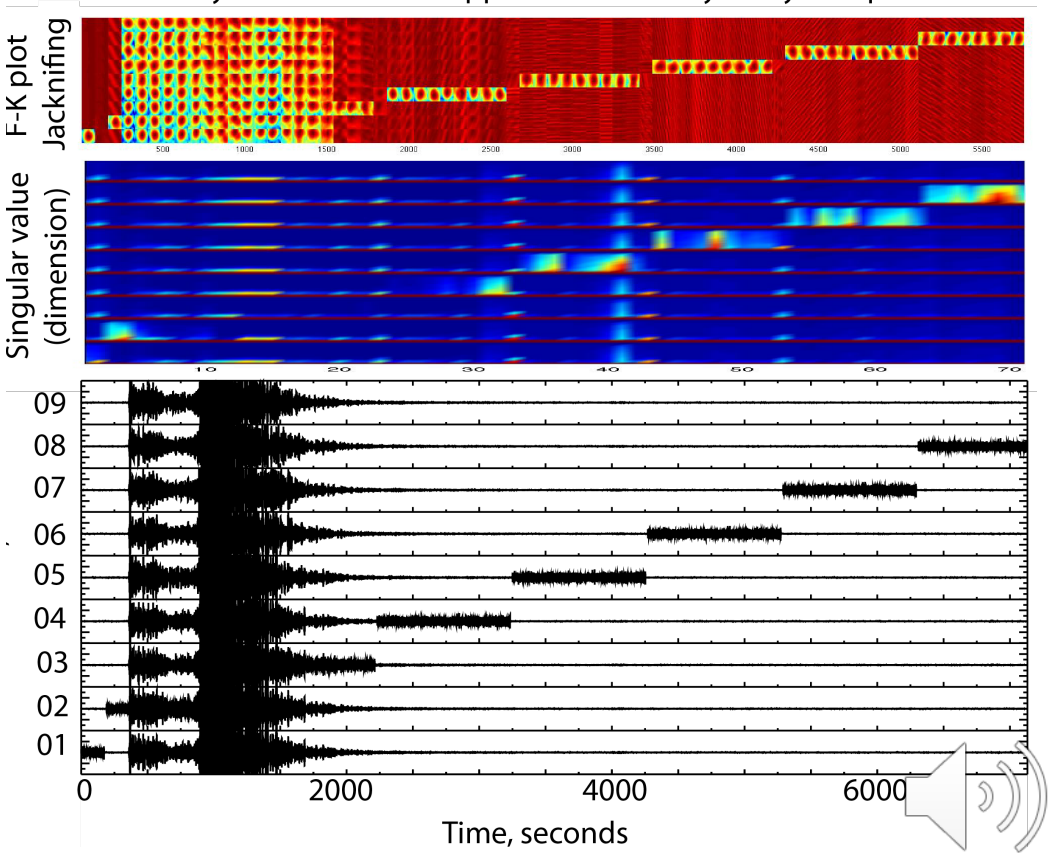


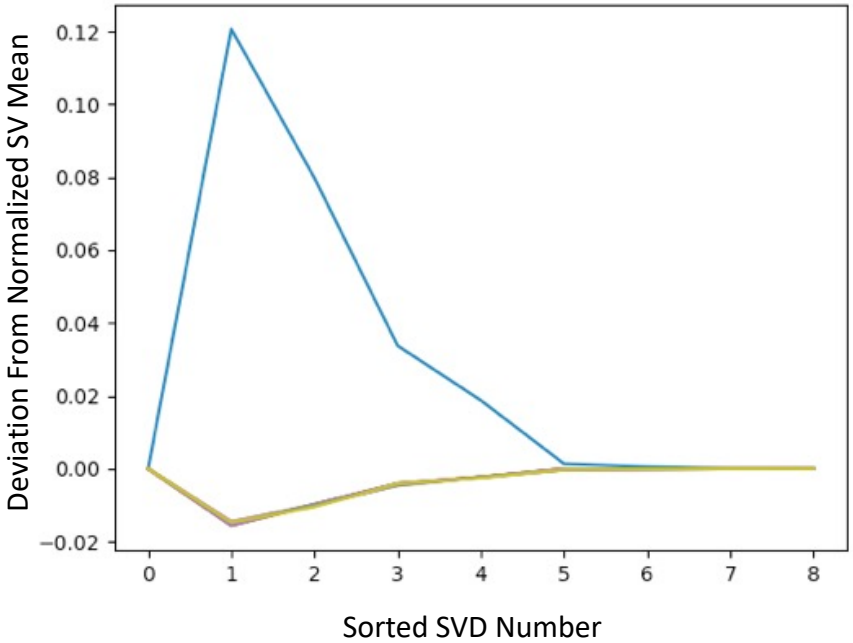


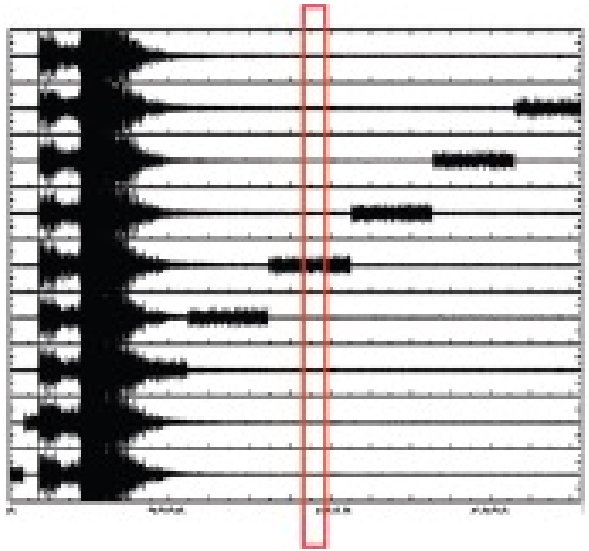
Testing on Another Array

- Array component waveforms are shown at the bottom. A calibration pulse is cycling through the channels.
- The center (blue) panels show the surfaces for sequential singular value functions, each of which has one component removed via jackknifing. The change when the offending channel is removed is obvious except during the much larger amplitude earthquake.
- The uppermost panels show sequential FK plots for the array, illustrating improvement achieved through removal of the calibrating trace.

Example: Automated Scanning Dimension test and
F-K analysis with Jacknife Approach to Identify Noisy Component

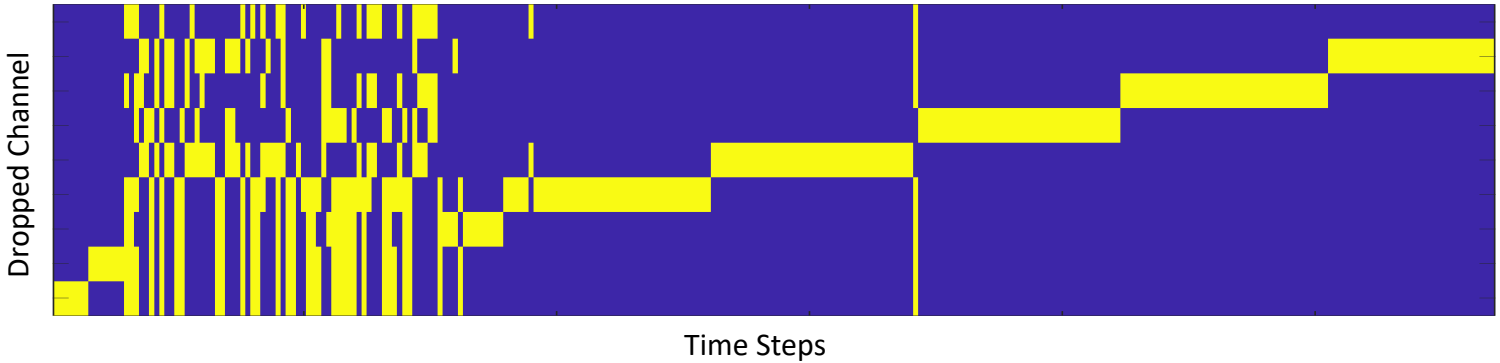






Time window (red) whose jackknifing results are shown

Example of difference between singular value functions when noisy channel (blue) is removed vs. any other channel (all the other colors).

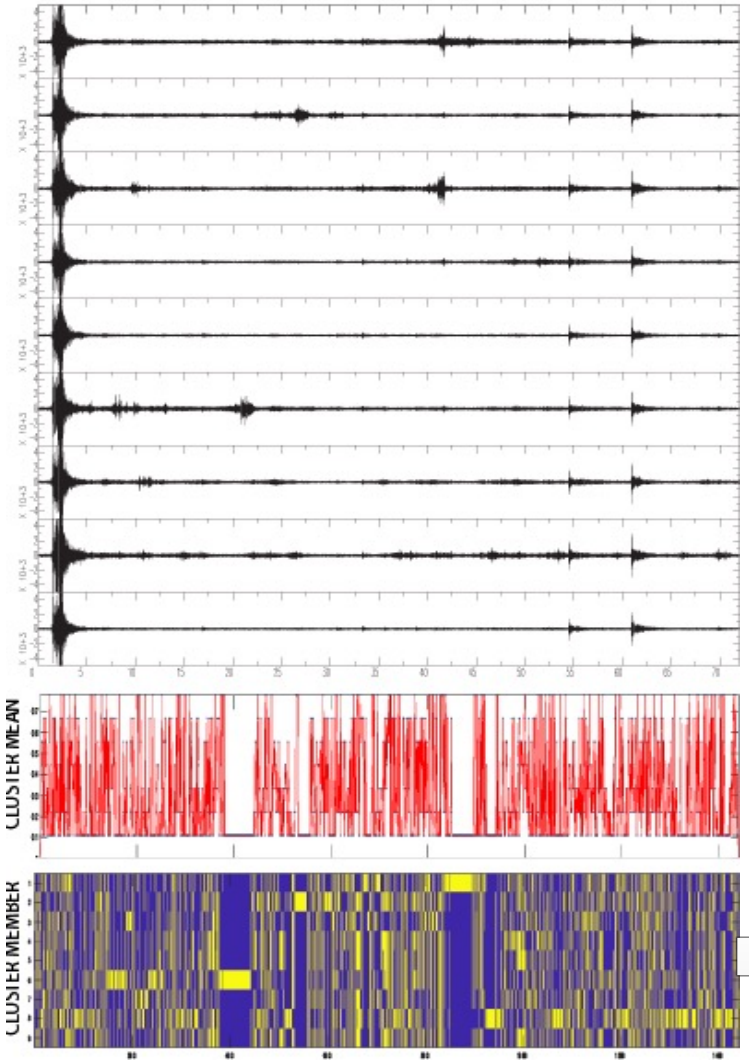


Using a K-means clustering method to assign the SVD functions into two families at each time step, we obtain a cluster membership through time as shown here. Except where the larger earthquake on these traces dominates over the cycling calibration pulse, a single function is flagged for the duration of its calibration as a member of one group, while all others cluster into the alternate group.

Here we show another example, of array traces and cluster membership.

The center (red and white) panel provides a running function of the cluster membership fraction. If only one channel belongs to Cluster 1 this value is 0.111, for instance, since we have nine channels.

We see a correspondence between time periods of a single offending channel and the associated noise appearing on that channel only. This channel could be dropped from the array analysis being undertaken.



Challenges going forward:

- Building the GUI
- Examining different clustering algorithms. We want to be able to determine if there is more than one bad channel.
- Applying a thresholding (Euclidean distance?) criterion for cluster membership. Currently the tool is too fussy and may flag a bad channel that isn't entirely useless.
- Size of system – in extremely large systems of traces, one offending channel may not sufficiently perturb the system. Randomized sampling and jackknifing will be tested (for instance in SEG-Y supergathers in active source situations, or Large-N deployments)

Script

Slide 1: Hello, I'm Charlotte Rowe from Los Alamos National Laboratory. Today I am presenting some work in collaboration with my co-author Stephen Heck at Sandia National Laboratories that aims to develop an automated tool for quality control on seismic array data.

Slide 2: I realize that perhaps not everyone in this session is a seismologist, so I thought I'd briefly introduce at a high level what seismic arrays are and how we use them.
- A seismic array differs from a network in that the sensors in an array are arranged in a carefully planned geometry intended to improve our ability to detect and characterize signals from a particular source region or regions. It is essentially a seismic antenna
- The Wikipedia definition says it well: "it is a system of linked seismometers arranged to increase sensitivity for detection of seismic sources of interest. Leveraging the special geometry of an array, the data are analyzed using specialized signal processing method to enhance the signals observed."
- We see here three hopefully helpful images:
on the left is a hypothetical circular array with one central component, and I show a hypothetical incoming seismic plane wave from a distant source. By observing the time of arrivals at the various sensors we can obtain the approximate azimuth to the source.

in the center I show the hypothetical arrivals - those nearest the source would come in earlier, while those farther away would come in later. By lagging and summing the traces we can obtain optimal alignment and then perform an addition or stacking of these lagged traces, creating a beam which has much greater signal to noise than the individual components. The lagging amounts can be determined in a number of ways: if we have other knowledge of the source direction from sensors elsewhere we can assume an azimuth. If we have muilticomponent sensors, we can use signal polarization information. We can do a grid search to maximize the stack or use cross-correlation of the components.

on the right I show what is known as an F-K (or frequency-wavenumber) plot. This is a complex, frequency-domain calculation that provides signal energy as a function of both azimuth and slowness (the inverse of velocity) across the array. Slowness provides us with an estimate of the source distance, as it is a function of the incidence angle of the seismic wave coming up from below. Very distant earthquakes will have near vertical incidence, for instance, whereas something nearby will be nearly horizontal. By having both slowness and azimuth, the array can provide an estimate of source location.

Slide 3: Our concern here is seismic noise that appears on array channels. Here I show as an example a small aftershock of the 2005 Wenshuan earthquake in China, as recorded the Makanchi seismic array in Kazakhstan. We can see from the inset that the lower three channels, which were relatively quiet, have become quite noisy, while the remaining components are still quiet. In the lower right, I show two F-K plots for you: The one to the left shows the analysis using all channels; the one on the right shows the analysis if we only use the quiet channels. Although the maximum energy in both plots seems to be at roughly the same position, the

amplitude of the anomaly is much greater on the right hand image This difference in amplitude is a concern, as many of the automated or semi-automated analyses researchers undertake may rely upon thresholds.

Slide 4: Our goal is therefore to build a tool that can identify bad channels in seismic array data so that users who wish to do significant analysis of large data sets will know what data to potentially exclude from their analysis. without painstaking and time-consuming manual review. For an array with a chronically bad channel, it's easy enough to just neglect it for the entire analysis, but for ephemeral signal problems, we'd like to be able to flag them for removal automatically.

Slide 5: We're building on the idea of array data dimension, or principal components to achieve this tool. This technique is employed for monitoring traffic on large computer systems, where subspace dimension can identify anomalies without the need to directly analyze the traffic. There are many parallels between node behavior and the behavior of seismic channels in an array or a small network, so we seek to apply a similar method to our problem.

Slide 6: We're relying on singular value decomposition for computational ease to test the method. If we examine a small time window on all the array channels, we can arrange the seismic traces as vectors in a matrix and perform SVD on this matrix. As we step through time windows, changes to the signal matrix will change the shape of the SVD function. On the left you see an elevation of SVD functions for the array, arranged side-by-side as we move our time window through the trace. The figure on the right shows one of the array traces above, and a map view of the SVD time variation below. One thing that stands out is that each of the earthquakes on this trace correspondes to a significant change in the shape of the SVD (essentially the dimension goes down, as all the array components are dominated by a correlated signal). But we are most interested in the lower left corner of this surface, where we clearly have lost a singular value. What happened? We must be missing a channel.

Slide 7: I went back manually and pawed through the data to discover that we had indeed lost one channel for a time before the mainshock, but I'd rather not have to do that. We opted to employ a jackknifing method, cycling through the traces, to identify the offending channel. At each time step, if we cycle through the data leaving one channel out at a time, when a good channel is removed, the singular values are not much affected, but when the bad channel is removed, we can see a significant change to the system if the bad channel has a large influence (i.e. the noise amplitude is high).

Slide 8: Here is a simple test where we have seven channels from an array in Chile. One of these channels is very noisy. We see that an earthquake is observed over most of this segment of the array. I divide the wavefroms into ten segments (shown at the bottom of the waveforms). I calculated seven singular value decompositions for each time step. Each singular value function corresponds to dropping one channel during the analysis. In each panel for the time steps, I show the 7 normalized SVD functions. We can see that, except during the

earthquake coda, there is a clear change for the singular values when the sixth channel is removed.

Slide 9: Here's another case. We show another array that has nine components. On the bottom you see the traces; there's an earthquake early in the traces but across this entire time period we see a calibration signal that is cycling through all the components. We can see singular value surfaces in the blue section, where except during the earthquake signal, the cycling calibration pulse is clearly represented in a change of singular values when the calibrating trace is the one removed during jackknifing. The upper, red, section just shows time varying F-K array analysis windows to demonstrate the significant impact that the calibration has in degrading that method.

Slide 10: So a qualitative, visual examination suggest the jackknifing SVD approach will work. but we need to quantify these differences if we want an automated tool to offer guidance on a bad channel. We have opted to compare each singular value function to a median of all nine, and use the resulting vectors in a cluster analysis. Here we see the traces again, with a red box showing the time segment that we are using for an example. The functions on the left are the nine median-differenced singular value functions for this time step, in which the calibration occupies the fifth trace. The blue curve is the one that corresponds to the system when the fifth channel is dropped. The other eight curves are all bunched together below it.

Slide 11: We use a k-means clustering tool in which we requested to have each time step clustered into two groups. Here you can see the results of each clustering step, plotted as a surface of blues and yellows. The yellows track the calibration pulse quite nicely, where a single channel seems to be separated from all the others in the clustering results, except when the earthquake signal dominates the system - at these places, cluster membership seems fairly random, as we would expect during quiet and non-calibrating times as well.

Slide 12: Okay, the blue and yellow matrix is still visual. We convert this clustering result into a numerical value in which we take the cluster zero membership fraction of the total number of traces, yielding values that we see in the red panel here. This function can be used to quickly flag time periods when there is a problem, but we then need to tie it to the cluster membership matrix to report the bad channel. Here you can see some glaringly obvious issues in both the red function/vector and the offending channel in the blue and yellow matrix. Above, we observe what signal is causing the tool to complain. Our plan is to assign a time tolerance, such that the problem needs to continue for so many seconds or minutes in order to be reported, since the random behavior of independent array channels will occasionally result in a one-to-eight clustering although nothing anomalous is going on.

Slide 13: We're still working on this tool and have additional improvements to make. For one, we are building a GUI. The tool is written in Python and currently wants the user to provide choices at command line, which for a large array is a lot of typing and opportunity for typos. We want to explore other clustering approaches, including possibly flexible cluster numbers. Our current clustering assumes that the median-difference singular value vectors are characterizing

Euclidean distance. Perhaps another metric could be used. We also wil explore assigning a noise threshold, as the tool seems unnecessarily sensitive due to the need to assign two families. Finally, one bad channel might not provide sufficient perturbation in a very large system with thousands of channels, such as multi-fold SEGY supergathers of industry or large-N deployments, as opposed to our monitoring arrays with at most a few tens of channels. We will be looking at a randomized jackknifing process of data subsets for these cases to explore the tractability of larger trace collections and optimal subdivision parameters.