

Jan 13, 16 7:58	README_noise.rgcg3	Page 1/3
Noise Cross-Correlation Utilities Documentation -----		
Programs: split_data, noise_preprocess, noise_xcorr Author: Jamie Barron Date: 2nd February 2011		
Modifications: Robert Green Date: Summer 2014		
1) Modifications to noise_xcorr to add options to compute the autocorrelation of each station in addition to the cross-correlations of all station pairs.		
2) Modifications to noise_xcorr so that the Noise Correlation Functions (NCFs) have an odd number of data points which is required for functionality in AFTAN.		
3) Modifications to noise_xcorr so that it outputs shorter NCFs by default. There is an option to also output the longer ones if needed.		
Compilation: -----		
- requires the boost C++ libraries (Makefile looks for the libraries in the system library path and the include files in /usr/include/boost -- if not set up this way you will need to edit the Makefile to point to the correct places including possibly giving a -L<dir> argument to the compiler (add to the LIBBOOST variable in the makefile) to point it at the correct directory for the libraries).		
- requires the GNU scientific libraries (Makefile also looks for these libraries in the system library path, so if you have your libraries in a different place you will have add a -L<dir> argument to the LGSL variable in the makefile to point it at the correct library location).		
Once the libraries are installed and the Makefile set up correctly to know where they are, then the programs should compile by running 'make'.		
To remove the binaries and object files and recompile, run 'make clean' and then re-run 'make'.		
 split_data -----		
Program to break long SAC files up into hour or day blocks		
Run with no arguments or -h argument to get the usage message.		
You can choose to break into hour or day blocks (hour blocks is default) and to output the split files into a specific directory (default is names split_files).		
 noise_preprocess -----		
Program to prepare the data for cross-correlation		
Run with no arguments or -h argument to get usage message.		
It looks for a configuration file which contains the information on filtering and the running mean normalization. The default file it looks for is noise.cfg. Here is an example of the contents of the file		
----- [prefilter]		

Jan 13, 16 7:58	README_noise.rgcg3	Page 2/3
bp_low_corner = 0.01 bp_high_corner = 0.2 bp_npoles = 4 [runmean] window_length = 50 bp_low_corner = 0.01 bp_high_corner = 0.05 bp_npoles = 4 -----		
Note that the file is only needed if the data are bandpass filtered before normalisation and/or the running mean normalization is used. If the running mean normalization without filter is used then you don't need to specify the filter parameters (bp_*) in the [runmean] block. If the data are not bandpass filtered before normalization then you don't need to specify the [prefilter] block. But it's probably just best to copy the example noise.cfg file in this directory and then modify these arguments if you are using these options - it doesn't hurt for them to be there when you're not using them...		
You can specify an alternative configuration file with the -c option -- if you don't do this and it can't find noise.cfg when options that need it are specified, then the code will stop with an error.		
By default, the code carries out the following steps: <ul style="list-style-type: none"> - remove mean and trend - taper the signal using a Hanning window - bandpass filter (using [prefilter] parameters in config file) - remove mean and trend of bandpass filter - apply time domain normalization (see Bensen et al, GJI, 2007) - apply spectral whitening 		
You can turn off the removal of mean and trend (in both places simultaneously) by using the --no-remove-mean and --no-remove-trend options. You can turn off the bandpass filter by using the --no-band-pass option, and the pre-bandpass taper by using the --no-taper option. The taper window can be changed from its default of 100 samples by using the -t option.		
You probably only want to use the above options if you've already bandpass filtered the data in a way that you're happy with and just want to do the normalization.		
You can turn off the post-normalization spectral whitening by using the --no-whiten option.		
The normalization is set using the -n option to one of the following <ul style="list-style-type: none"> clipping - clip to RMS, and scale signal so that this RMS level=1 onebit - one bit normalization (+/- 1) runmean - running mean normalization runmean_filt - running mean normalization with filter (see Bensen et al, GJI, 2007 for details of the method).		
The input SAC files are not overwritten unless you specify the --overwrite option. By default they're written out to the preproc_files directory, but this can be changed with the -o option.		
 noise_xcorr -----		
Program to do the noise cross correlation.		
Run with no arguments or -h to get the usage message.		
If you use the --dry-run option, you will get the lists of paired up files. This is useful to find out which files are required for a single station pair		

Jan 13, 16 7:58

README_noise.rgcg3

Page 3/3

correlation -- so that these files can be put together and the correlation for individual station pairs run on the cluster.

The default is to correlate day blocks. I tried hour blocks and it mostly seemed to work although there were some problems with some station pairs, so I probably need to debug this some more.

You can change the sample rate but if you've decimated your data to 1 sample per second then this is the default anyway. All the data has to be consistent with the sample rate specified (i.e. you must have all the data at, say, 1 sample per second, and can't have some at 1 sps and some at 2 sps).

Running correlations on the cluster

To be added