Manual for Ambient Noise Data Processing Code (ANDP2.3)

General Descriptions:

This seismic ambient noise data processing code (latest version 2.3, updated on October 11, 2020) with parallel computing is dedicated to automatically process seismic ambient noise dataset at small or large scale based on user-defined settings. We keep the user interference as simple as possible. Only three steps are needed to launch this program. First, prepare your continuous waveform dataset and set the input parameters for the program. Second, compile the source code and install the executables. Third, run a script. This code is written using Message Passing Interface (MPI) with Fortran 2003 standards, which makes it well suited for modern PCs with multicores and also high performance machines, including clusters and supercomputers.

Main Features:

1. Fully automated
2. Minimum user interference
3. Support parallel computing
4. Continuous waveform with any length
5. Continuous waveform with any sampling rate
6. Produce uncertainty measurements for dispersion data
7. Output ASCII cross-correlation files for Prof. Huajian Yao’ code
8. Provide linear, phase-weighted and RMSR_SS three different stacking schemes

System Requirements:

This program has been successfully tested on CentOS 7 and should be working on other Unix-like systems, such as Ubuntu, Fedora and so on. The following compilers, libraries and software packages are required to run this program:

Compilers: **gfortran**, **openmpi**

Libraries: **fftw3**

Software packages: **sac**, **python3**

NOTES:

1. If you are going to process seismic data that have sampling rate larger than 1Hz, you are encouraged to modify the following sac source code and recompile and reinstall the whole package. The change mentioned here is to increase the maximum number of points set for the sac 'smooth' command. However, this operation is optional since using a number large than 128 in the running time average process is not likely to produce significant changes.

${SACHOME}/src/scm/xsmooth.c:

#define MHALF 128   replaced by   #define MHALF 1024

2. If you choose to perform the rms ratio selection stacking scheme, you'd better to modify the following sac source code and recompile and reinstall the whole package. The change mentioned here is to increase the maximum number of files that can be read each time.

${SACHOME}/inc/mach.h:

#define MDFL 1000   replaced by   # define MDFL 100000

Input Parameters:

All the input parameters needed for this program are placed in <**input.dat**>. Here we explain each of them in detail (Important parameters that you are likely to modify are indicated by bold blue color).

INPUT PARAMETERS FOR AMBIENT NOISE CROSS-CORRELATION

Line #4:	LHZ	Channel name (no need to change).
Line #5:	**180 150 3 2**	Period limits used in the removal of instrument response. Period band set here should be larger than the following period band of band-pass filtering.
Line #6:	**70 60 3 2**	period limits for the band-pass filtering. I'd like to pick the dispersion data at period band of 4-50 s, so here I set it to be 70 60 3 2 to include my interested period band. You should adjust these parameters according to your own case. **NOTES:** The minimum period set here should be at least twice larger compared to the sampling period (e.g., 1 in this case) to satisfy the well-known Nyquist-Shannon sampling theorem.
Line #7:	**200 21200**	tBegin and tLength. tBegin is relative to the event folder time and tLength is the time length in second for each event. Take the event 20111201_000000 for example, tBegin=200 and tLength=21200 means the program will cut the data from 2011-12-01 00:03:20 to 2011-12-01 05:53:20. You should modify these two parameters when you process continuous waveforms that are cut into different length.
Line #8:	1	Power of cosine tapering function (no need to change).
Line #9:	Y 40 N 50 15	If doing running time average, half-window length, if doing earthquake band-pass filtering, period band of the band-pass filtering. Running time average generally produce better results than one-bit normalization method. Doing band-pass filtering to suppress earthquake signals in the running time average operation is optional. In my experience, earthquake band-pass filtering is not good.
Line #10:	N	If doing one-bit normalization (usually not)
Line #11:	20	Half-window length in spectral whitening (no need to change)
Line #12:	N 0.5	If repressing the notch at 26s and the retaining factor (usually not)

Line #13:	**1800**	: lag time (in second) for the cross-correlation. You should set this parameter according to the aperture of your array. lag=1000 means your cross-correlation function will have length of 2000 s (-1000 ~ 1000 s). **CAUTION**: <u>longer cross-correlation function will significantly increase the computing time if you choose to output the phase-weighted stacking (pws) cross-correlation functions as indicated below</u>.
Line #14	**N 1**	: If output the pws results, weight number for pws. Larger number will increase the weight, leading to stronger suppression of the noise.
Line #15	**N**	: If performing the rms ratio selection stacking method. 'passbands.dat' is required if this option is turned on.

INPUT PARAMETERS FOR BOOTSTRAP ANALYSIS

Line #21	N 100	: If doing bootstrap analysis to measure group/phase velocity uncertainties, number of repeating times for the bootstrap analysis.
Line #22	**2_2**	: Type of dispersion data to write down as final results. For the first number, 1 represents basic frequency-time analysis results; 2 represents phase-matched filtering results. For the second number, 1 represents preliminary dispersion results without quality control like jump detecting; 2 represents clean dispersion results after imposing quality control like jump detecting. Clean phase-matched filtering results are recommended to use.

INPUT PARAMETERS FOR AUTOMATIC FREQUENCY-TIME ANALYSIS

Line #28	-1.0	: piover4 (phase_shift=pi/4*piover4)
Line #29	**1.5 5.0**	: vMin, vMax. Minimum and maximum group velocity used for tapering the signal window.
Line #30	**4.0 50.0**	: tMin, tMax. Minimum and maximum period for group and phase velocity dispersion measurements. The measuring band here should be included in the band-pass filtering band as shown in Line #6.
Line #31	**Y 1**	: If limiting the maximum period to satisfy the time harmonic expression of surface wave propagation in the far field as constrained

		: by the formula: nLambda*period*ref_velocity<dist. nLambda=2 or 3
		: are commonly used in various literature but nLambda=1 also works
		: according to Luo et al., 2015;
Line #32	N	: If output all the original dispersion results produced by AFTAN
		: program.
Line #33	20.0	: tresh-jump detection factor (used in trigger subroutine, if(abs)>tresh)
Line #34	1.0 **2.0**	: ffact1 and ffact2. alpha values used in the narrow-band Gaussian
		: filters for basic and phase-matched filtering, respectively.
		: (alpha=20*ffact*sqrt(dist/1000); y=x*exp(-alpha*((f-f0)/f0)**2))
		: If the alpha value for the phase-matched filtering function (aftanipg
		: function) is set to be 2.0, more phase velocity measurements will be
		: obtained but the group velocity dispersion curve will become less
		: smooth! Recommended values are 2.0 or 1.0
Line #35	0.5	: taperl. Factor for the left-end seismogram tapering
		: (ntapb = nint(taperl*tmax/dt))
Line #36	0.2	: fsnr. Not used (for future extension)
Line #37	2.0	: fmatch. Factor for the length of phase matching window

INPUT PARAMETERS FOR OUTPUT SETTINGS

Line #43	Y	: If output the sac file record and cross-correlation record
Line #44	Y	: If verbose all the processing details

Compile.sh:

Run ./**Compile.sh** to do the compiling and installing work. sudo permission is required since it installs the executables to the /usr/local/bin path by default. You can change the path of the compiled executables by modifying INST_DIR variable in <Compile.sh> and the two Makefiles under AND_Driver and AFTAN folders.

Run.py:

<**Run.py**> is the only script that you need to execute. Just type **./Run.py >& Run.log** and check the log file to see if there is any error within it after the program completes. This script will invoke the

<AND_Driver>, <TF_PWS> and <AFTAN_PROG> three executables to deal with the input seismic waveforms based on input parameters as defined in <**input.dat**>. Within this script, you need to modify the following four variables:

SACfolder	: Set the input data folder path. The data structure for input continuous
	: waveforms in sac format should be ***/year/month/event/net.sta.cha.SAC**
	: e.g. DATA/2018/01/20180101_120000/DB.EW01.LHZ.SAC
	: The input sac files can have any sampling rate but the sampling rate of all these
	: files must be consistent. The event folder name represents the starting time of
	: this event. For example, 200190101_000000, 200190101_060000,
	: 200190101_120000, 200190101_180000 means the input continuous
	: waveform data are cut into 6-hour long segments.
RESPfolder	: Set the input folder path of instrument response files. Individual or combined
	: full RESP or PZ file(s) are all recognizable.
tarfolder	: Set the output data folder path.
ncores	: Set the processor numbers. You should modify this number according to your
	: own computing resources.

ref1Dmod.dat:

<**ref1Dmod.dat**> contains 1-D reference phase velocity values which is used in AFTAN program for the phase velocity dispersion measurements. The two columns are period and corresponding reference phase velocity value.

myTools.py

<**myTools.py**> contains the GetPZ function which converts preliminary RESP or PZ files (single or combined) to standard single PZ files.

Output Files:

CC_AFTAN

This folder contains all the cross-correlation functions and corresponding raw dispersion results.

For example:

WD.TA20_WD.TA21_ls.SAC : Linear stacking cross-correlation function in sac format

WD.TA20_WD.TA21_ls.SAC.txt	: Linear stacking cross-correlation function in ASCII format which can be directly used in Prof. Huajian Yao's code.
WD.TA20_WD.TA21_pws.SAC	: Phase-weighted stacking cross-correlation function in sac format
WD.TA20_WD.TA21_pws.SAC.txt	: Phase-weighted stacking cross-correlation function in ASCII format which can be directly used in Prof. Huajian Yao's code.
WD.TA20_WD.TA21_rms.SAC	: RMS ratio selection stacking cross-correlation function in sac format
WD.TA20_WD.TA21_rms.SAC.txt	: RMS ratio selection stacking cross-correlation function in ASCII format which can be directly used in Prof. Huajian Yao's code.
WD.TA20_WD.TA21_ls.SAC_1_1	: Basic preliminary dispersion results
WD.TA20_WD.TA21_ls.SAC_1_2	: Basic clean dispersion results
WD.TA20_WD.TA21_ls.SAC_2_1	: Phase-matched filtering preliminary dispersion results
WD.TA20_WD.TA21_ls.SAC_2_2	: Phase-matched filtering clean dispersion results

FINAL

This folder contains all the final dispersion results.

For example:

LINEAR/WD.TA20/WD.TA20_WD.TA21.dat	: Final dispersion data with the result of Line #22 based on linear stacking
PWS/WD.TA20/WD.TA20_WD.TA21.dat	: Final dispersion data with the result of Line #22 based on phase-weighted stacking
RMS/WD.TA20/WD.TA20_WD.TA21.dat	: Final dispersion data with the result of Line #22 based on RMS ratio selection stacking

DataRecord.lst and **CCRecord.lst**

DataRecord.lst contains the data information for each input sac file.

CCRecord.lst shows the number of stacking for each station pair.

These two files will be generated if isOutput in Line #43 is set to be ‘Y’.

RMSR_SS.lst

RMSR_SS.lst contains the rms ratio selection information for each station pair at each passband. This file will be generated if isOutput in Line #15 is set to be ‘Y’.

stations.lst and **events.lst**

stations.lst contains all the staion information (network, station, lon, lat, dt)

events.lst contains all the event information (absolute path of each event folder)

Run a test

The folder ‘TEST-FanXL’ contains all the computing results on my computer based on the example data in ‘DATA_1Hz’ and corresponding response files from ‘RESPfiles’. Before processing your own seismic data, you should first run your own example under the folder ‘TEST-Your’ to make sure it produces the same results as mine.

Acknowledgement:

The seismic ambient noise cross-correlation computing algorithm and the automatic frequency-time analysis algorithm of this program is based on Barmin's code (http://ciei.colorado.edu/Products/). The phase-weighted stacking program is modified from Guoliang Li' code (guoleonlee@gmail.com). Any bug reports, suggestions or comments related to this program can be directed to Xingli Fan via <fanxldengcm@gmail.com>.