November, 2010

BIDO Version 2.0

Original text in Japanese written by I. Cho and translated into English by T. Tada.

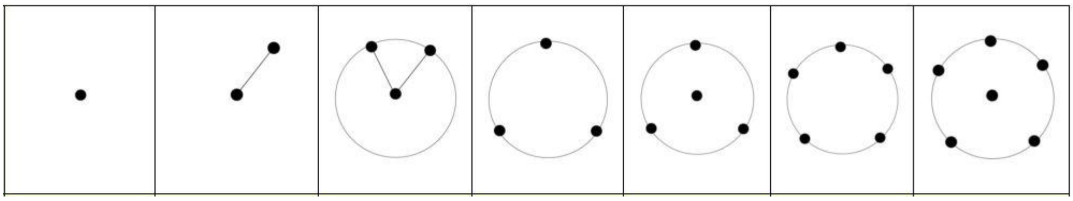
What is BIDO?

BIDO is a package of analysis codes to identify properties of surface waves using circular-array records of microtremors (ambient vibrations; bidô in Japanese).

Left: Simultaneous measurement of microtremors using a circular array of six seismic sensors with a radius of 30 cm. Right: A scene from a mobile measurement campaign. The entire recording system, including the set of seismic sensors shown in the left, was loaded on a vehicle. (See A6-1-8) (Related publication in Japanese: reference [6]).

What Array Geometries Is BIDO Good for?

The new version can be applied to single-station measurements (with one seismic sensor alone; left end panel in the figure below) , huddle tests (tests to check the homogeneity of equipment properties by huddling all seismic sensors at a single location), and preprocessing (including elimination of the trend, application of tapers, bandpass filtering, decimation and correction for differeces in instrumental response). It can also be applied to irregularly shaped arrays, such as linear arrays of two seismic sensors and L-shaped arrays of three seismic sensors (second and third panels on the left). However, the analyzable quantities differ for different numbers of sensors and for different record components (vertical/horizontal/three-component). It would not be easy to design the analysis procedures while taking account of the correspondence relations, which are fairly complicated (Appendix 1). We have therefore designed the program in such a way that simply replacing the file that describes the sensor locations and the record components allows the program to automatically identify the analyzable quantities and proceed with analysis.



User Manual

Contact:

Ikuo Cho (**ikuo-chou@aist.go.jp**)

Contents

1. Outline

BIDO is an analysis tool we offer free for the array exploration of microtremors. The software can be used to identify properties of surface waves that travel on the ground surface by analyzing circular-array records of microtremors.

Since 2000, our research group, centered on the (former) Shinozaki laboratory at the Tokyo University of Science (joint research by Professor Yuzo Shinozaki, Dr Taku Tada, myself [Ikuo Cho] and graduate students) have undertaken generalization of the SPAC method theory ("generic formulation for circular-array analysis", reference [2]), and have developed methods that allowed phase velocities of Rayleigh waves to be identified into much longer wavelength ranges than the traditional SPAC method (the CCA method and other derivative methods, reference [1, 3, 4, 5]). Our theories have also made it possible to identify phase velocities of Love waves (reference [5]), signal-to-noise ratios (reference [3, 8]), horizontal-to-vertical amplitude ratios (R/V spectra) of Rayleigh waves (reference [2]), and Rayleigh-to-Love power ratios (reference [5, 7]) with simple methods unknown in traditional approaches. BIDO is an analysis tool for microtremor data (circular-array records) that uses these methods to identify properties of surface waves.

Array Exploration of Microtremors

The ground surface is constantly trembling because of industrial activities, ocean waves and winds. They are, of course, too small to be felt by human bodies, and can be detected only by microtremor sensors (high-sensitivity seismic sensors). These small tremors are called microtremors (random noise). Simultaneous measurement using more than one microtremor sensors installed on the ground surface is called array measurement of microtremors. Array measurement of microtremors allows one to infer phase velocities of surface waves (propagation velocities of waves called Rayleigh waves and Love waves), on the basis of which one can then infer soil properties (velocity structures). By the term "array exploration of microtremors" we refer to the whole flow of procedures that start with array measurement of microtremors and end with evaluation of subsurface structures.

The SPAC Method

A technique of microtremor exploration to analyze phase velocities.

There are two major categories of analysis methods for phase velocities, namely the Capon method (also known as the FK method) and the spatial autocorrelation (SPAC) method. The spatial autocorrelation method was published by Keiiti Aki in 1957, whereas the Capon method was published by Jack Capon in 1969. The spatial autocorrelation method could be called more classic in that sense, but one had to await the activities of Hokkaido University's Hiroshi Okada and coworkers (publication years: 1983-200?) and the University of Tokyo's Kazuyoshi Kudo and coworkers before one could see it becoming a practical method of microtremor exploration. It is only after they began their activities that the spatial autocorrelation method came to be known by a diminutive(?) acronym, the SPAC method.

The SPAC method is strictly constrained by the requirement that the seismic array should be circular (a disadvantage), but it is also characterized by the ability to analyze wavelengths that are fairly long relative to the array size (an advantage). The number of seismic sensors in the circular array can be relatively small, and may even be reduced to just one central sensor plus one peripheral sensor (!?) in an ultimate case, according to an idea that emerged recently. We have also endeavored to help build a theoretical framework for this idea, which bore fruit in a recent publication (reference [9]).

An overview of international publications suggests that the SPAC method began to obtain worldwide recognition in the mid-1990s.

2. For Whom It Is Meant

We have meant this code package to be used by engineers with expertise in microtremor exploration and by researchers investigating microtremor exploration and array analysis methods (the program is easy to use, but you need expertise to interpret the output appropriately and to infer soil properties).

3. Usage

1) Create microtremor waveform data files, one for each sensor, and place them all within a single folder.

2) On activating the program, you will be asked for a number of parameters. With appropriate input given, the program automatically proceeds and plots the analysis results for you. The results are stored in an output folder that is automatically created under the data folder.

See Section 6 for details.

4. Technical Information / How to Install
Basic Information

A string of core programs written in Fortran (compilers: g77 and ifort; unchecked for others), linked together via B shell, are executed one after another. Operability confirmed on Windows (XP, Vista, and Windows7) and Linux (CentOS5, Fedora 10).

If you are using Windows,…

simply download the archive, decompress and execute. The registry is not rewritten, so simply dump the folder into the recycle bin to uninstall. The folder may be saved anywhere, like on the Desktop or in the root of the C drive.

If you are using Linux, …

it is recommendable to recompile the Fortran codes, although executable files are included in the package. To compile and install, decompress the archive, enter the src directory, and execute Install_linux.sh. **Gnuplot** (free) is used in drawing graphs, so you have to install it separately unless it is already installed on your PC.

Program Implementation on Windows

The development and operation environment is based on Linux. For use on Windows, the Fortran source codes were compiled using **Cygwin** (free), and a shell environment was implemented using **MSYS** (free). The graphic tool gnuplot is also included in the package (compilation finished, but bundled with source codes). If you are to rewrite source codes, you have to recompile them after installing Cygwin on your PC. To compile and install, decompress the archive, enter the src directory, and execute Install_win.sh.

PC Performance Requirement (Example)

The following is an example of the PC performance requirement, described for the case of 6. Demo Data processing under this program's development environment. On a Windows XP/Linux dual boot PC (CPU: Intel(R) Pentium(R) D CPU 3.00 GHz; memory: 2 GB), the CPU time requirement ("user" output of the "time" command) was about 2 min 30 sec on both operation systems (g77 compiler used in both), and the memory requirement was about 300 MB (adjustable by modifying array dimensions by editing PARAM.h when compiling source codes). However, the real time requirement ("real" output of the "time" command) was a little short of 3 min on Linux, while it was about 15 min on XP (shell processing by MSYS and file I/O may possibly account for the time on XP). If this difference can be generalized to all cases is difficult to say, but this outcome seems to recommend the use of Windows only for trial runs and Linux for massive calculations (with the Intel compiler=ifort).

The program itself only occupies about 14 MB of hard disk space (both on Windows and Linux), but this demo requires nearly 120 MB (including the program itself and demo data). If you set parameters so that all intermediary data are deleted except for the final analysis results and minimal log files (you will be asked about the choice on activating the program), only less than 50 MB worth of files will be left when the calculations are over (including the program itself).

5. Program Description and Algorithm

● Program Description

\BIDO-win.bat A batch file to activate MSYS on Windows. Not used on Linux.
\bin Executable files for use on Windows are stored in \bin\winbin, while executable files for use on Linux are stored in \bin\linbin. Executable files for use on MSYS are stored just beneath \bin.
\demo Contains demo data (used both Windows and Linux).
\etc Contains scripts to activate MSYS. Not used on Linux.

\run.sh A B shell script to activate the program (used both Windows and Linux).
\script Contains B shell scripts to link Fortran codes (used both Windows and Linux).
\src Contains Fortran codes (used both on Windows and Linux).

● Algorithm
Description of the general flow and individual procedures. In parentheses are the file names of relevant B shell scripts and Fortran codes.

General Flow (\script\circle.sh)
1) Select portions of the data that are good to use.
2) Azimuthally average data around the circle (output from step 1) is not used here)
3) Estimate spectral densities by using the output from steps 1) and 2).
4) Estimate spectral ratios, phase velocities, NS ratios etc.
5) Repeat steps 3) and 4) as many times as there are segment clusters.
6) Calculate means and standard deviations using the output from step 4).
7) Plot the output from step 6).

[1] Automatic Selection of Segments (\script\mksegment.sh, \src\evalrms.F, \src\segment.F)
Segments are selected as follows:
1) The following procedure is performed on all components of all sensors.
- Subtract a linear trend from the original waveform data and calculate the RMS (let this be called $\mathrm{RMS}_{\mathrm{all}}$).
- For every portion of the data with a prescribed segment duration into which the original waveform is divided (the portions are extracted so that they mutually overlap by half), subtract a linear trend and calculate the RMS. Normalize the RMS values by $\mathrm{RMS}_{\mathrm{all}}$ (let these be called $\mathrm{RMS}_{\mathrm{seg}}$).
2) Make a histogram of all $\mathrm{RMS}_{\mathrm{seg}}$, for all components of all sensors, at intervals of 0.1. Identify the interval of the largest frequency in the histogram.
3) Pick up data segments in which all $\mathrm{RMS}_{\mathrm{seg}}$ of all components and all sensors fall into the interval of the largest frequency simultaneously. Mark them as the segments good to use, and catalog them in the segment file.

[2] Azimuthal Averaging of Data around the Circle (\script\mkcrcle*.sh, \src\mkcrcl_*.F)
All methods adopted in this program start with taking weighted azimuthal averages of records around the circumference. Weighted azimuthal averaging corresponds to calculating Fourier coefficients in the Fourier series expansion around the circle. Our program calculates Fourier coefficients of the zeroth and first orders by default. Our theory is adaptable to unevenly spaced sensors around the circle, and adaptability to practical cases has been investigated for the CCA method (Reference [1]). We have not, however, closely investigated the adaptability for all methods, so we recommend the use of equidistant arrays to the extent that that is possible

(we are particularly uncertain about the adaptability to methods that use cross-spectral densities).

[3] Estimating Spectral Densities (\script\estspec*.sh, \src\estspec.F)

This program eliminates the trend from each segment, applies a cosine-type taper with a 50-percent overlap (Carter et al., 1973), and estimates spectral densities by using both techniques of segment averaging and spectral smoothing for the raw FFT spectral densities (Bendat & Piersol, 1971). The raw spectral densities are smoothed with a Parzen window before they are segment-averaged. The segment duration and the number of segments in the averaging over multiple segments are given in the dialogue on activating the program: "Duration of data segments for the evaluation of spectra" and "Number of data segments over which averages are taken. Enter 0 or a very large number if you wish to use all segments simultaneously." The bandwidth of smoothing with a Parzen window is given in the dialogue: "Band width of the Parzen spectral window."

[4] Calculating Spectral Ratios (\script\specratio*.sh)

Ratios are taken, with no frills, between spectral densities estimated in the above-described procedure, except when the denominator is zero. Different types of spectral ratios are linked to the phase velocities via formulae described in References [2, 5]. Note that the autocorrelation coefficient of the SPAC method is defined here by a spectral ratio according to the formulation of Reference [2] (in usual practice, the SPAC coefficient is calculated as an azimuthal average of complex coherences between the central point and a peripheral point).

Calculating the H/V spectrum

Starting with Version 1.2.2, I added a feature that estimates horizontal-to-vertical (H/V) spectral ratios, provided that the data have three components, at the one station that is indicated at the top of the seism file (A6-1-3) (H/V refers to the ratio of the power-spectral density of the horizontal components to that of the vertical component. The power of horizontal motion is defined as the sum of the NS and EW component powers). Accordingly, even when the seism file describes a single measurement station alone (even if this does not constitute an array), H/V spectral ratios are calculated as long as there are three-component records. Once the calculation is over, the logarithmic mean and standard deviation are plotted.

Calculating Phase Velocities (\script\spec2pv*.sh, \src\sctr2pv.F)

Spectral ratios are equated to Bessel functions according to the formulae, and a root-solving method that combines bisection and the secant method (Shampine & Watts, 1970) is used to search for *rk* (radius times wavenumber), the argument of the Bessel functions, in the range [0, $rk_{\max}$]. $rk_{\max}$ corresponds to the first maximum or minimum of the function value. The *rk* obtained is used to calculate the phase velocity, c=2p f/k (f stands for frequency).

[5] Repeat Steps [3] and [4] as Many Times as There Are Segment Clusters

[6] Calculating Means and Standard Deviations (\script\mkave.sh, \src\calave.F)

If the number of segments (number of segments over which averages are taken when estimating spectral densities with the segment averaging method. In other words, the integer value that you enter in the dialogue "Number of data segments over which averages are taken. Enter 0 or a very large number if you wish to use all segments simultaneously" on activating the program) satisfies

(number of all segments catalogued in the segment file (A6-1-4)) > (number of segments),

then more than one spectral density estimates are obtained from the given waveform data. If we define

(number of segment clusters) = (number of all segments catalogued in the segment file) / (number of segments),

there will be as many estimates of spectral densities, spectral ratios derived from them and phase velocities as the number of segment clusters (the remainder of division is discarded). This program calculates means and standard deviations on the basis of those estimates.

When the program is executed, folders with names \RESULT\(a number) are generated beneath the data folder. The number here represents that of a segment cluster, and these numerically named folders contain the corresponding analysis results. The \RESULT\ave folder contains output of statistical processing of the analysis results stored in those numerically named folders.
To eliminate "outliers," the maximum and minimum values are excluded from the statistical processing if and only if there are more segment clusters than the number NROBUST4AVERAGE_INC indicated in \src\PARAM.h. NROBUST4AVERAGE_INC is set at 8 by default. Some of the analysis results are averaged arithmetically and others logarithmically. Logarithmic averaging is used when averaging ratios like AmpRV_R.d, nsr.d, nsrlim_cca.d, nsrlim_cca.lwapx.d and powratio_R2L.d.

Bendat, J. S., and A. G. Piersol, *Random Data: Analysis and Measurement Procedures*, John Wiley & Sons, 1971.

Carter, G. C., C. H. Knapp, and A. H. Nuttall, 1973, Estimation of the magnitude-squared coherence function via overlapped Fast Fourier Transform processing: IEEE Transactions on Audio Electroacoustics, **AU-21**, 337–344.

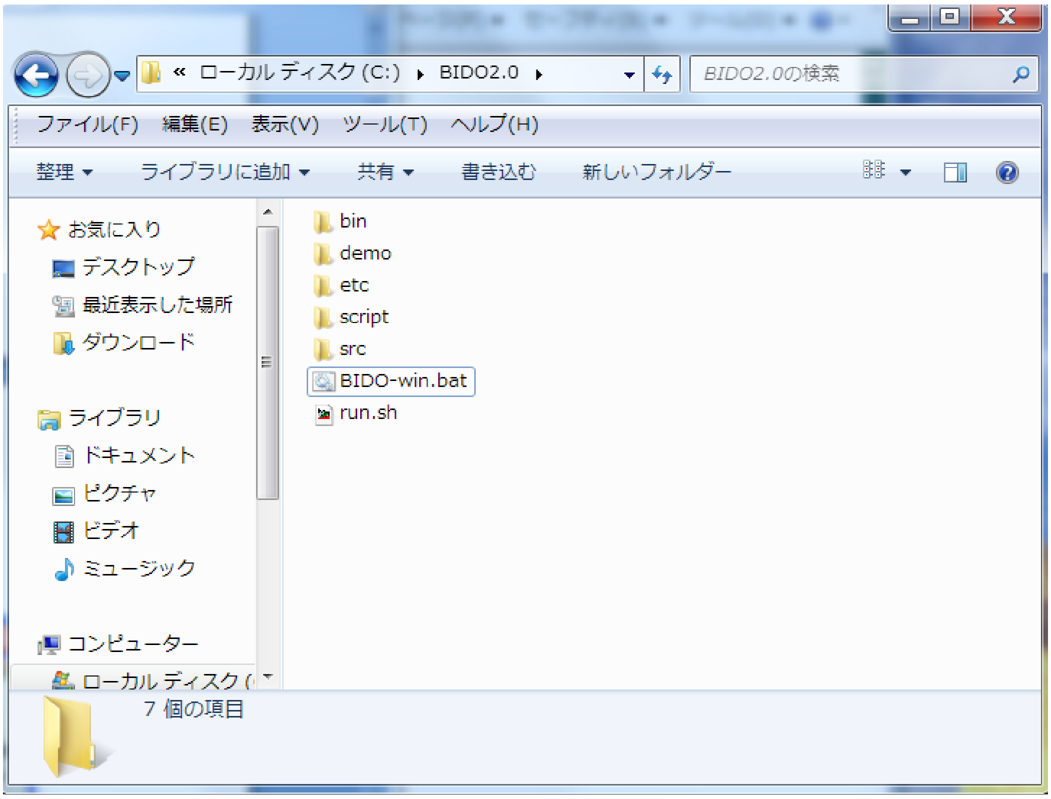
Shampine, L. F., and H. A. Watts, FZERO, a root-solving code, Report SC-TM-70-631, Sandia Laboratories, 1970.

6. Example of Program Execution Using Demo Data

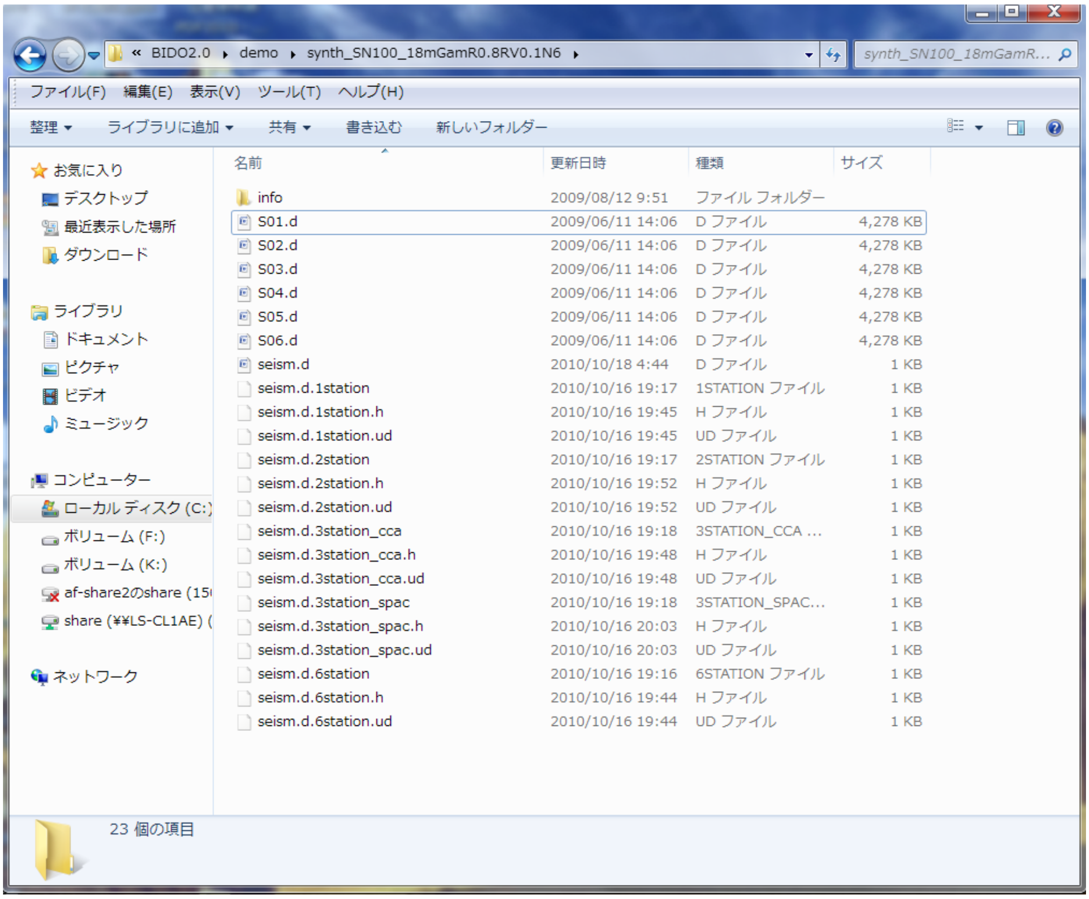
6-1 Demo of Array Data Analysis

We use synthetic demo data simulating array observation to illustrate how to execute the program. The following illustrates how it can be executed on Windows, but the method is the same on Linux.

Expand BIDO2.0.tgz using decompression software. You will find the following files inside it:

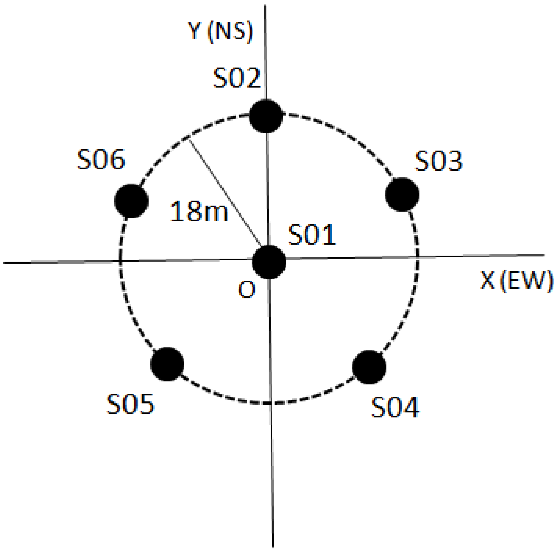


The folder demo\synth_SN100_18mGamR0.8RV0.1N6 contains synthetic data for demo analysis.

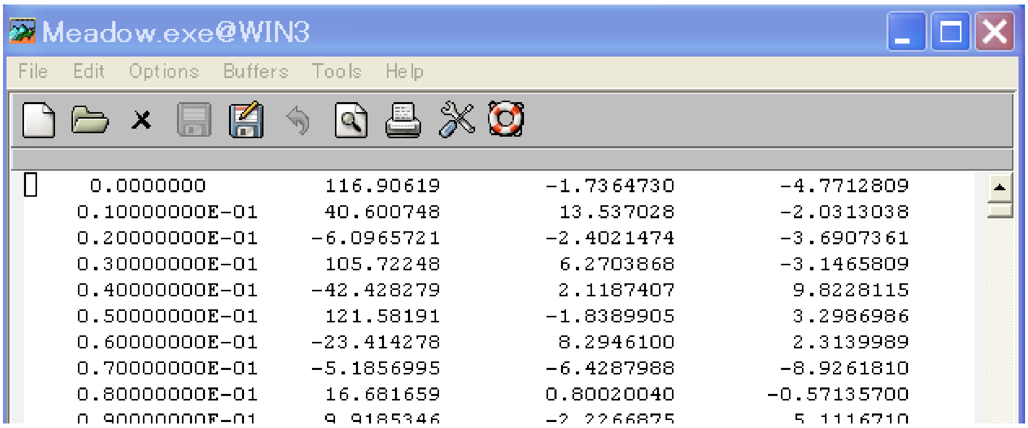


名前	更新日時	種類	サイズ
info	2009/08/12 9:51	ファイル フォルダー	
S01.d	2009/06/11 14:06	D ファイル	4,278 KB
S02.d	2009/06/11 14:06	D ファイル	4,278 KB
S03.d	2009/06/11 14:06	D ファイル	4,278 KB
S04.d	2009/06/11 14:06	D ファイル	4,278 KB
S05.d	2009/06/11 14:06	D ファイル	4,278 KB
S06.d	2009/06/11 14:06	D ファイル	4,278 KB
seism.d	2010/10/18 4:44	D ファイル	1 KB
seism.d.1station	2010/10/16 19:17	1STATION ファイル	1 KB
seism.d.1station.h	2010/10/16 19:45	H ファイル	1 KB
seism.d.1station.ud	2010/10/16 19:45	UD ファイル	1 KB
seism.d.2station	2010/10/16 19:17	2STATION ファイル	1 KB
seism.d.2station.h	2010/10/16 19:52	H ファイル	1 KB
seism.d.2station.ud	2010/10/16 19:52	UD ファイル	1 KB
seism.d.3station_cca	2010/10/16 19:18	3STATION_CCA ...	1 KB
seism.d.3station_cca.h	2010/10/16 19:48	H ファイル	1 KB
seism.d.3station_cca.ud	2010/10/16 19:48	UD ファイル	1 KB
seism.d.3station_spac	2010/10/16 19:18	3STATION_SPAC...	1 KB
seism.d.3station_spac.h	2010/10/16 20:03	H ファイル	1 KB
seism.d.3station_spac.ud	2010/10/16 20:03	UD ファイル	1 KB
seism.d.6station	2010/10/16 19:16	6STATION ファイル	1 KB
seism.d.6station.h	2010/10/16 19:44	H ファイル	1 KB
seism.d.6station.ud	2010/10/16 19:44	UD ファイル	1 KB

The folder Info contains information on the synthetic data. There are six files named in the format S0x.d. These are microtremor waveform data obtained by assuming six seismic sensors, named S01 to S06, in the configuration shown below. These are, however, synthetic data created numerically for demonstration purposes. The data were synthesized by simulating three-component observation at a sampling time interval of 0.01 sec and a duration of 5 minutes. See A6-1-1 for details on the microtremor waveform synthesis.



Open S01.d with any editor, and you will find:



```
 0.0000000          116.90619        -1.7364730       -4.7712809
0.10000000E-01      40.600748        13.537028       -2.0313038
0.20000000E-01     -6.0965721       -2.4021474       -3.6907361
0.30000000E-01      105.72248        6.2703868       -3.1465809
0.40000000E-01     -42.428279        2.1187407        9.8228115
0.50000000E-01      121.58191       -1.8389905        3.2986986
0.60000000E-01     -23.414278        8.2946100        2.3139989
0.70000000E-01     -5.1856995       -6.4287988       -8.9261810
0.80000000E-01      16.681659       0.80020040      -0.57135700
0.90000000E-01      9.9185346       -2.2266875        5.1116710
```

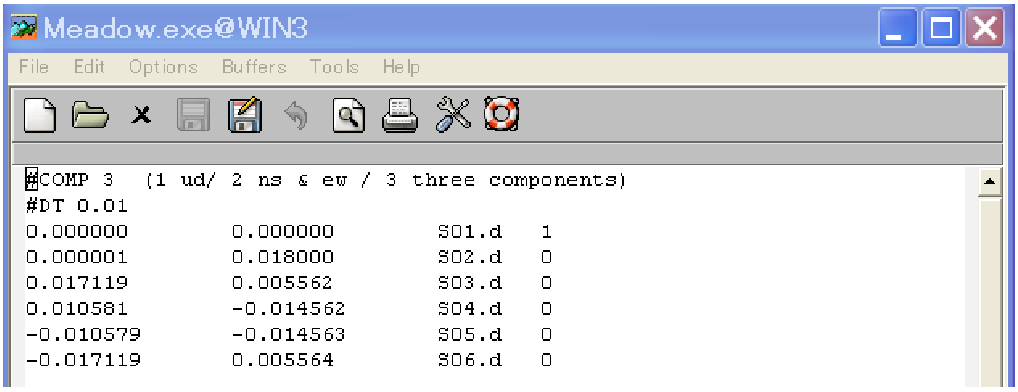
The left-hand side column stands for time, and the rest stand for the amplitudes of the vertical (z), EW (x) and NS (y) components, from left to right.

When you analyze your own measurement data, format them in four columns if they are equipped with all three components. If they have the two horizontal components alone, format them in the same way by inserting zeroes or any dummy data where the vertical component should be. If they have the vertical component alone, let there be just two columns for time and the vertical component (there can be four columns as in the case of three-component data, but there is no need for inserting dummy data). This is because my own measurements were often for the vertical component alone but were never for the horizontal components alone (because of the seismometer setup, measurement of the horizontal components was always accompanied by measurement of the vertical component).

The analysis method of this program presupposes that waveforms are sampled at equal time intervals. The time column should therefore be unnecessary as long as the start time and the sampling time interval are given (you can even do without the start time in practical array data processing). If you are already familiar with array analysis of microtremors, you may wonder why the first column is necessary. In fact, the string of time data in the first column are dummy and are actually not read (whatever figures you may put in the first column have no influence on the analysis results). I created this column simply because the graphics tool, which I was using to verify the measurement data, required a format of time and amplitude pairs (analyzing waveforms, plotted and verified on the spot, was the most efficient way). As I will mention below, the sampling time interval should be given in a file named seism.d (A6-1-3).

Anyway, as long as the data are aligned in this way, the data type can be anything like velocity or acceleration, and can be of any unit. There is no constraint on the format of the numbers (be it with an exponent part or with a floating point). Separators between the numbers can either be spaces, tabs or commas. There is no particular rule on the naming of data files (they can even lack the extension *.d). When you have created data files, one for each seismic sensor, place them all within a single folder. Let all waveforms in the data files start at time zero.

You will find a file named seism.d in the same folder. Open it to find:



```
#COMP 3   (1 ud/ 2 ns & ew / 3 three components)
#DT 0.01
0.000000        0.000000        S01.d   1
0.000001        0.018000        S02.d   0
0.017119        0.005562        S03.d   0
0.010581        -0.014562       S04.d   0
-0.010579       -0.014563       S05.d   0
-0.017119       0.005564        S06.d   0
```

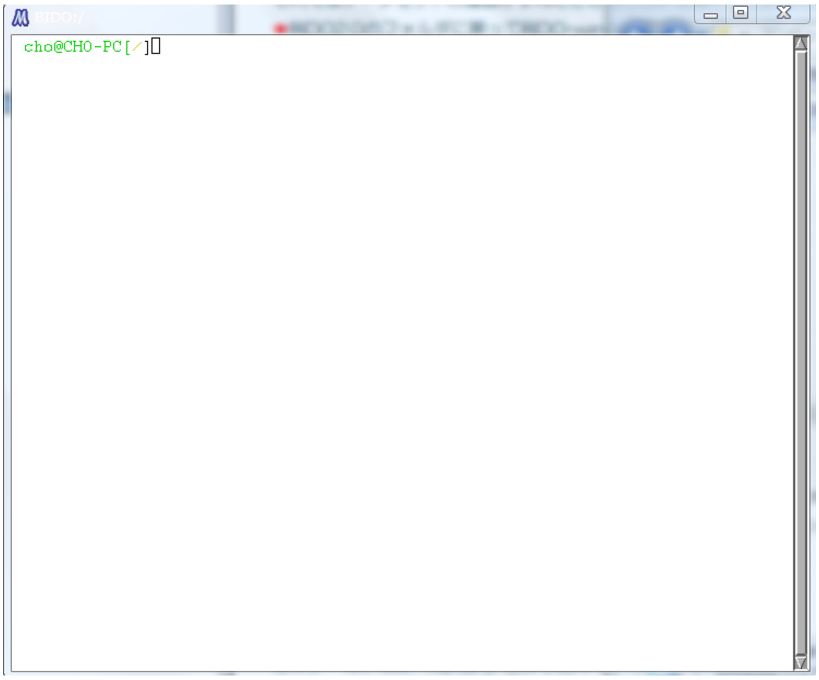
The number in the first row indicates the components to be used in the analysis. In the second row is the sampling time interval of the waveform records (#COMP and #DT are a sort of spells and cannot be omitted). In the third and following rows are the x- and y-coordinates (km), data file names and center/periphery IDs (1 for center and 0 for periphery). The file name seism.d cannot be modified, and it should be stored in the same folder as the measurement data. This file need not, however, necessarily have been created beforehand. When the data folder lacks this file, the program simply asks for necessary information and automatically generates one.

In this example, the origin of the x- and y-coordinates is set at the array center, but this is not obligatory. The program is written in such a way that it calculates information necessary for the array analysis, or the center location and the radius, by using information on the first three circumferential stations in the stations list. It also verifies whether the other stations lie precisely at the center or around the circle. The stations around the circle do not need to be equidistant (see Reference [2] for theory) but, to warrant precision, please make sure that the array station distances are as even as possible. When a station does not lie on the circle (has an error in the radial direction), it is interpreted as lying around the circle as long as the error is less than the default tolerance limit of 5%, because there is no theory for correction.

With BIDO 2.0, it is now possible to conduct huddle tests (tests to confirm the agreement of all instrumental properties by huddling all seismic sensors at a single location) (A6-3). Please assign identical x- and y-coordinates to all data points when you wish to conduct a huddle test. In that case, the flow of analysis is different from what is explained below. Power-spectral densities, magnitude-squared coherences and phase differences are the quantities to be calculated.

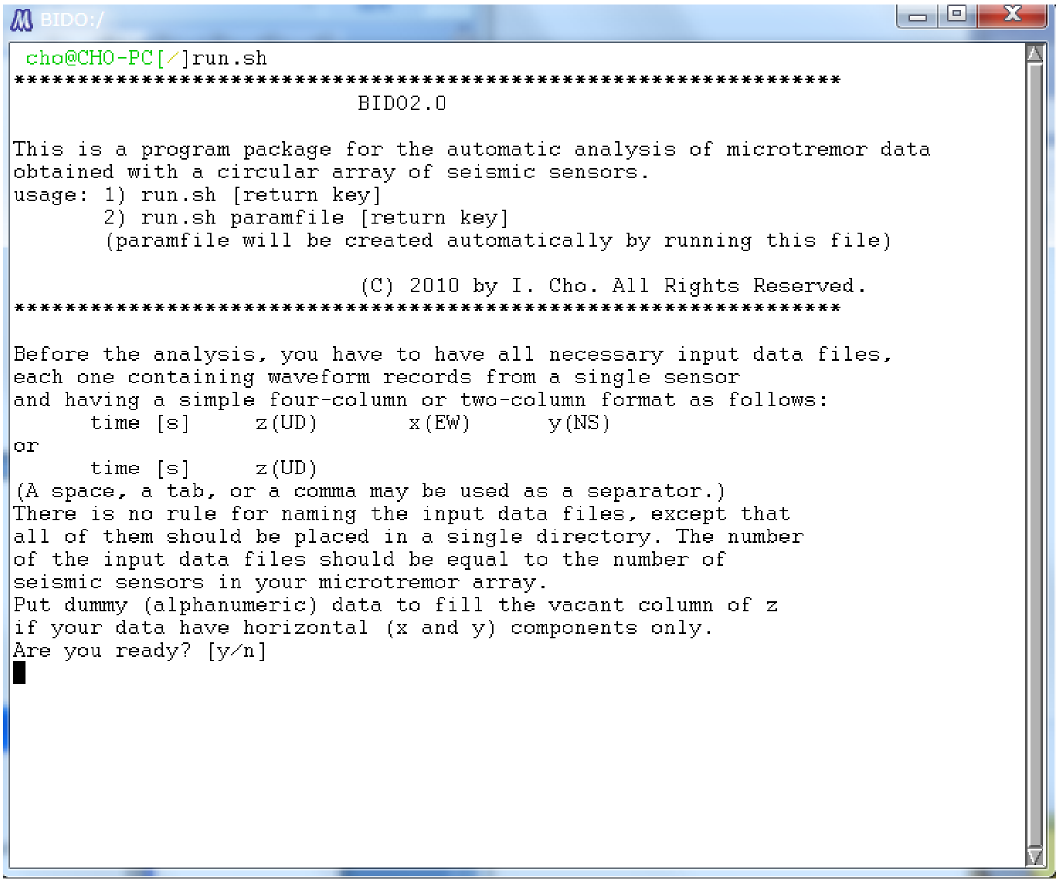
See A6-1-3 for a description of seism.d at a glance.

Now that you know what data set there is, let us start the analysis. Go back to the BIDO 2.0 folder and double-click BIDO-win.bat. This opens up a window (terminal) like the following (this action is not necessary on Linux).



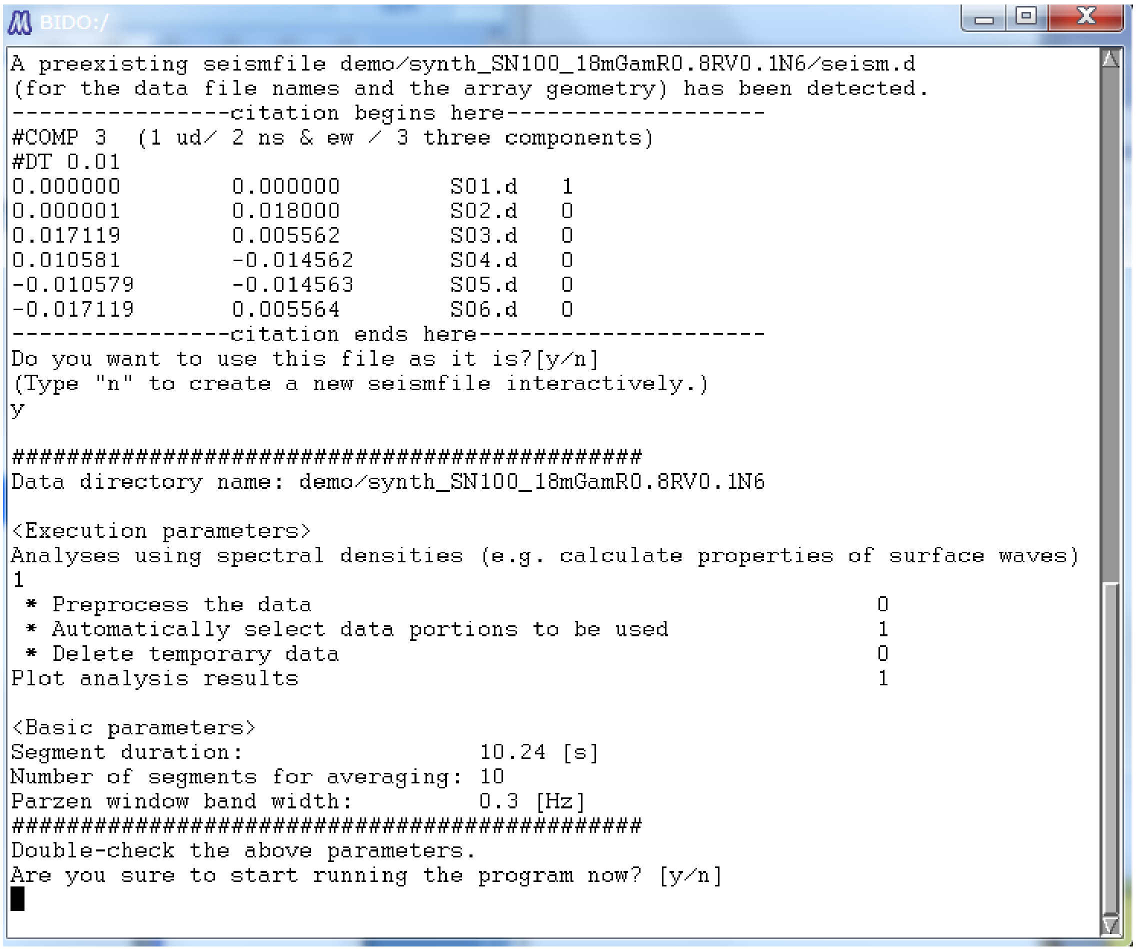
```
cho@CHO-PC[/]
```

Type "run.sh" on this terminal and press the return key, and you will see



```
 cho@CHO-PC[/]run.sh
*****************************************************************
                           BIDO2.0

This is a program package for the automatic analysis of microtremor data
obtained with a circular array of seismic sensors.
usage: 1) run.sh [return key]
       2) run.sh paramfile [return key]
       (paramfile will be created automatically by running this file)

                           (C) 2010 by I. Cho. All Rights Reserved.
*****************************************************************

Before the analysis, you have to have all necessary input data files,
each one containing waveform records from a single sensor
and having a simple four-column or two-column format as follows:
      time [s]     z(UD)       x(EW)      y(NS)
or
      time [s]     z(UD)
(A space, a tab, or a comma may be used as a separator.)
There is no rule for naming the input data files, except that
all of them should be placed in a single directory. The number
of the input data files should be equal to the number of
seismic sensors in your microtremor array.
Put dummy (alphanumeric) data to fill the vacant column of z
if your data have horizontal (x and y) components only.
Are you ready? [y/n]
█
```

In the above and all following screens, simply type "y" or press the return key (to use default parameters). You will finally come down to the following screen:

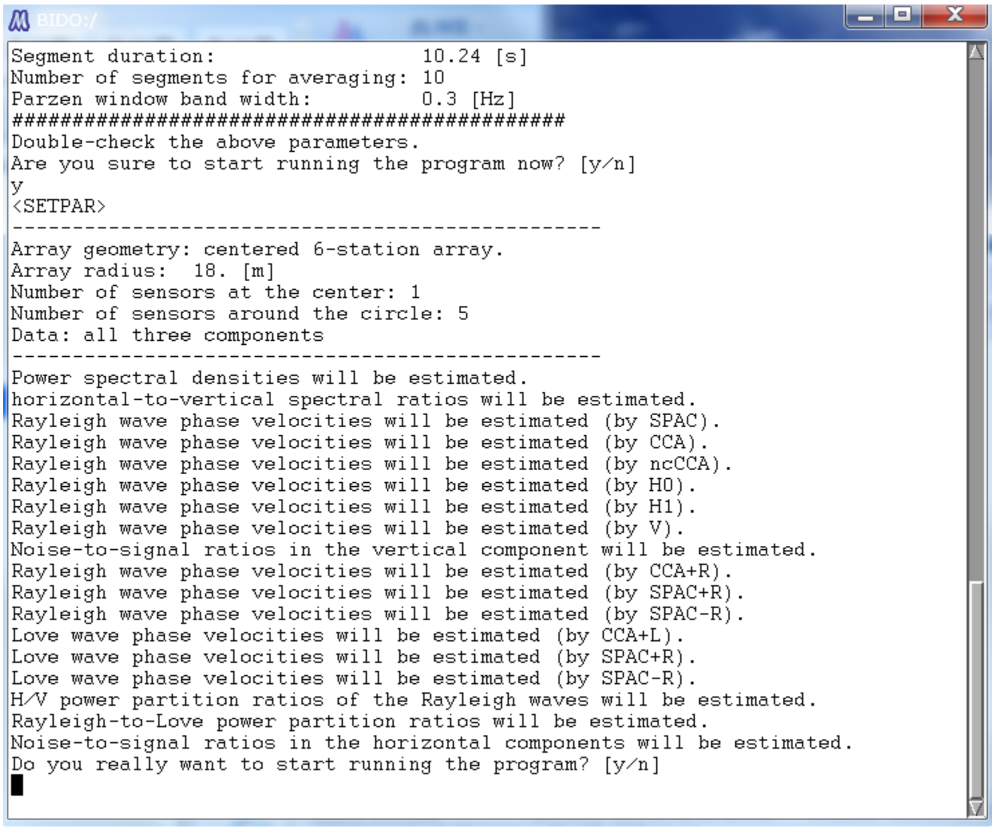


```
A preexisting seismfile demo/synth_SN100_18mGamR0.8RV0.1N6/seism.d
(for the data file names and the array geometry) has been detected.
----------------citation begins here-------------------
#COMP 3  (1 ud/ 2 ns & ew / 3 three components)
#DT 0.01
0.000000        0.000000        S01.d   1
0.000001        0.018000        S02.d   0
0.017119        0.005562        S03.d   0
0.010581        -0.014562       S04.d   0
-0.010579       -0.014563       S05.d   0
-0.017119       0.005564        S06.d   0
----------------citation ends here---------------------
Do you want to use this file as it is?[y/n]
(Type "n" to create a new seismfile interactively.)
y

##############################################
Data directory name: demo/synth_SN100_18mGamR0.8RV0.1N6

<Execution parameters>
Analyses using spectral densities (e.g. calculate properties of surface waves)
1
 * Preprocess the data                                              0
 * Automatically select data portions to be used                    1
 * Delete temporary data                                            0
Plot analysis results                                               1

<Basic parameters>
Segment duration:                 10.24 [s]
Number of segments for averaging: 10
Parzen window band width:         0.3 [Hz]
##############################################
Double-check the above parameters.
Are you sure to start running the program now? [y/n]
█
```

You will find, between the two rows of "###...," the parameters you entered. When you are analyzing your own measurement data, be sure to take a look at A6-1-8 and A6-1-9.

In the present demo, the default answer to the third question "Preprocessing the data?" is "n". It is sometimes effective to preprocess the data when you are using measurement data, but this is not the case in this demo because it uses synthetic data. I skip explanations for the sake of simplicity. See A6-2 for more details.

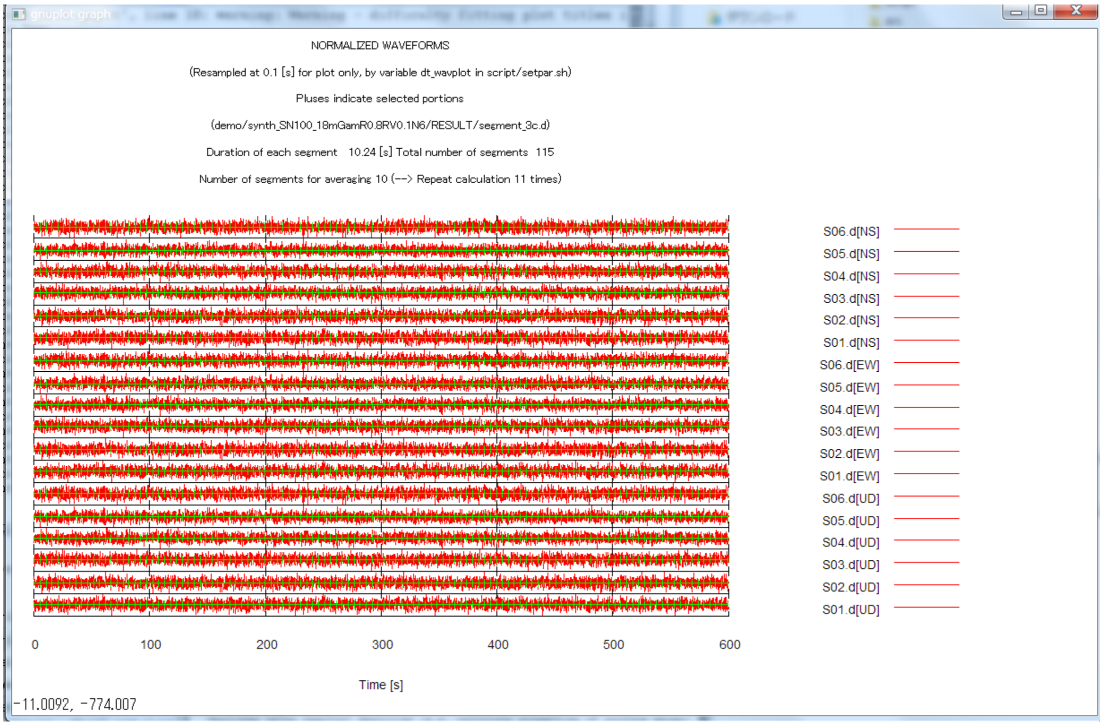
At any rate, you were being asked, in the above screen, whether is it okay to proceed with analysis with the values shown. Let us type "y" anyway.



```
Segment duration:                  10.24 [s]
Number of segments for averaging:  10
Parzen window band width:          0.3 [Hz]
###############################################
Double-check the above parameters.
Are you sure to start running the program now? [y/n]
y
<SETPAR>
--------------------------------------------------
Array geometry: centered 6-station array.
Array radius:  18. [m]
Number of sensors at the center: 1
Number of sensors around the circle: 5
Data: all three components
--------------------------------------------------
Power spectral densities will be estimated.
horizontal-to-vertical spectral ratios will be estimated.
Rayleigh wave phase velocities will be estimated (by SPAC).
Rayleigh wave phase velocities will be estimated (by CCA).
Rayleigh wave phase velocities will be estimated (by ncCCA).
Rayleigh wave phase velocities will be estimated (by H0).
Rayleigh wave phase velocities will be estimated (by H1).
Rayleigh wave phase velocities will be estimated (by V).
Noise-to-signal ratios in the vertical component will be estimated.
Rayleigh wave phase velocities will be estimated (by CCA+R).
Rayleigh wave phase velocities will be estimated (by SPAC+R).
Rayleigh wave phase velocities will be estimated (by SPAC-R).
Love wave phase velocities will be estimated (by CCA+L).
Love wave phase velocities will be estimated (by SPAC+R).
Love wave phase velocities will be estimated (by SPAC-R).
H/V power partition ratios of the Rayleigh waves will be estimated.
Rayleigh-to-Love power partition ratios will be estimated.
Noise-to-signal ratios in the horizontal components will be estimated.
Do you really want to start running the program? [y/n]
█
```

The program then lays out, as shown above, what analysis output there will be if you conduct the analysis under the current array geometry. Type "y", and the rest of the analysis will proceed automatically.

Soon after the analysis begins, prior to data processing for the estimation of spectra, there appears a plot of the waveforms, and information on which parts of them will be used in the spectral analysis (see the figure below. Waveforms are shown in red, and the parts of them used in the spectral analysis are indicated in green plus signs).



NORMALIZED WAVEFORMS

(Resampled at 0.1 [s] for plot only, by variable dt_wavplot in script/setpar.sh)

Pluses indicate selected portions

(demo/synth_SN100_18mGamR0.8RV0.1N6/RESULT/segment_3c.d)

Duration of each segment 10.24 [s] Total number of segments 115

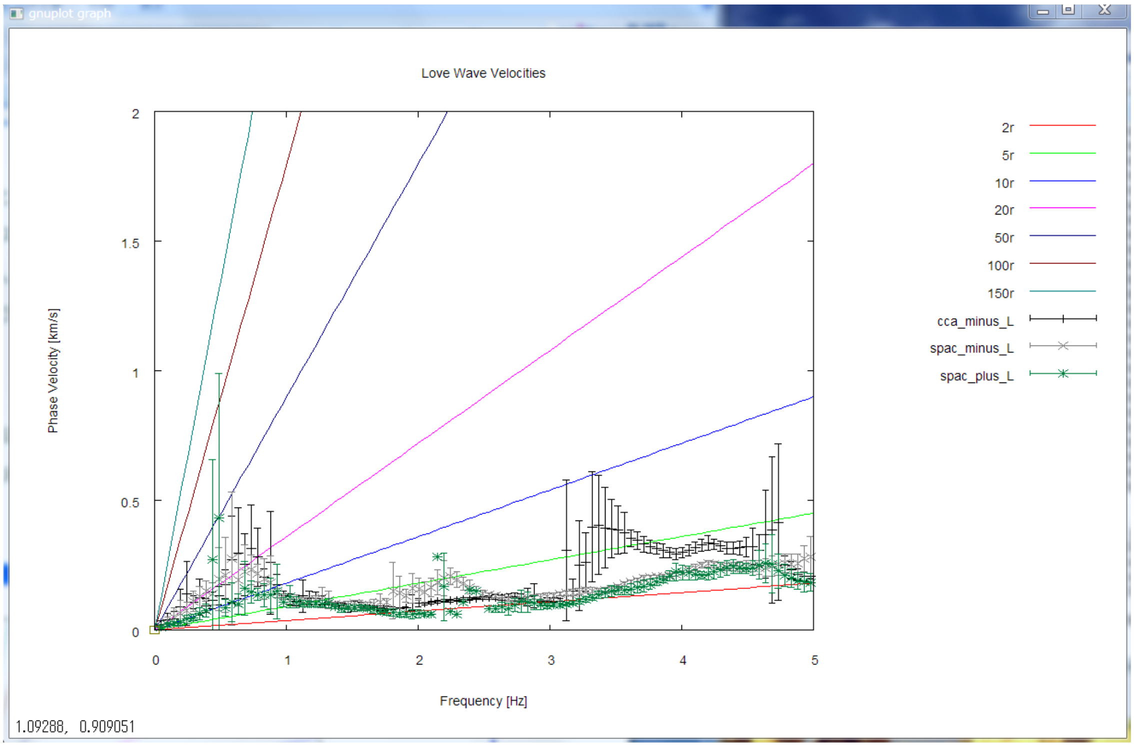
Number of segments for averaging 10 (--> Repeat calculation 11 times)

Go back to the interactive terminal screen and press the return key to erase this plot and move on to the spectral analysis.
The execution log of the analysis is in Appendix 2. This log is generated automatically in the folder where the data files are stored.

When the analysis is over, the
- Phase velocities of Love waves,
- Phase velocities of Rayleigh waves,
- R/V spectrum of Rayleigh waves,
- Comparison of the R/V spectrum of Rayleigh waves and the H/V spectrum,
- Shares of Rayleigh waves in the total power of horizontal motion,
- NS ratios (inverse of the SN ratios)
- H/V spectrum, and
- Power-spectral densities

are plotted. Below I explain them one after another.

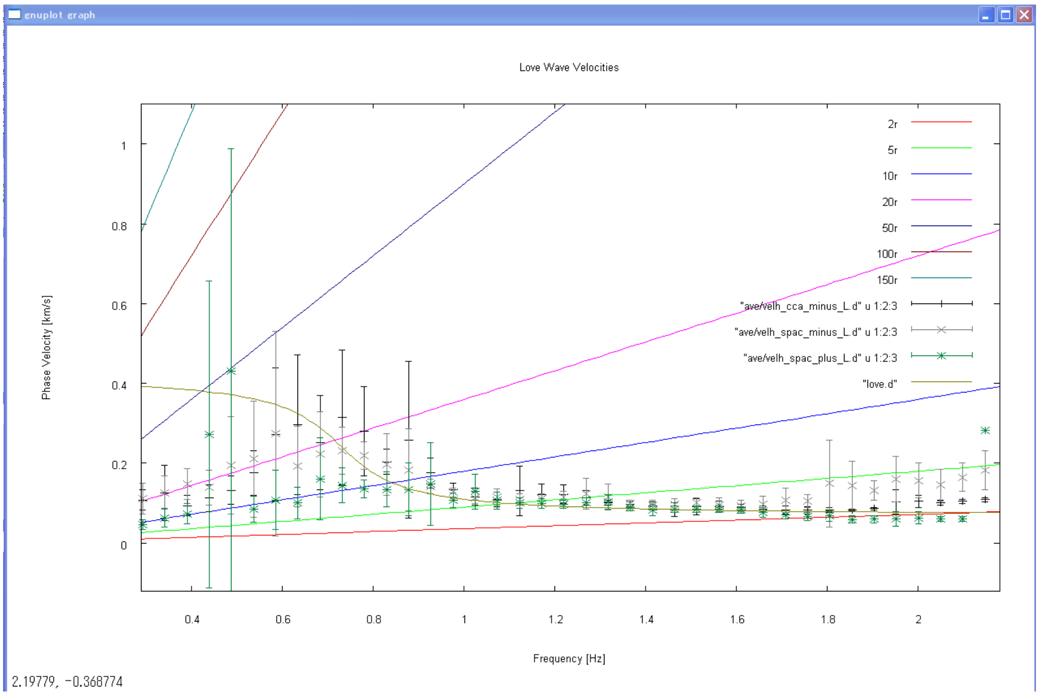
1) Phase velocities of Love waves are shown in data points with error bars (standard deviations). The horizontal axis stands for the frequency, and the vertical axis stands for the phase velocity.



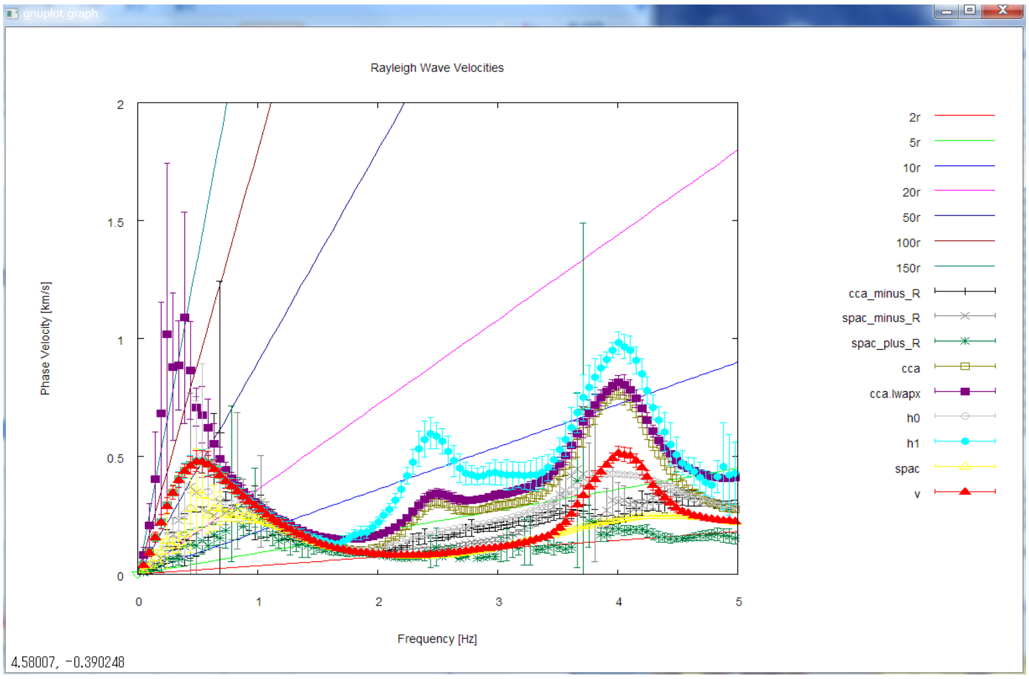
Analysis results from three methods, called the CCA-L, SPAC-L and SPAC+L methods, are shown superimposed on one another. The straight lines that emanate from the origin point stand for wavelengths of 2, 5, etc. times the array radius *r* (for reference). The prescribed Love-wave phase velocities (see A6-1-1) have seemingly been reproduced within a frequency range of about 1-2 Hz.

Incidentally, the range (upper limit) of phase velocities plotted has been adjusted automatically in the graph above. If you wish to adjust it on your own (prescribe a fixed value to it), revive the commented-out parameter ymax_for_gnuplot in \script\setpar.sh (delete the # at the head of the line) and assign an appropriate value (phase velocity [km/s]) to it. Likewise, if you wish to prescribe the range (upper limit) of frequencies on your own, revive the commented-out parameter xmax_for_gnuplot in \script\setpar.sh (delete the # at the head of the line) and assign an appropriate value (frequency [Hz]) to it.

Below is an enlarged view, for reference, of the comparison, between approximately 0.4 and 2.2 Hz, of the phase velocities prescribed during the computation of the synthetic data (solid curve) and the analysis results. With gnuplot, it is fairly easy to enlarge specific parts of a plot. See gnuplot manual pages for more details.

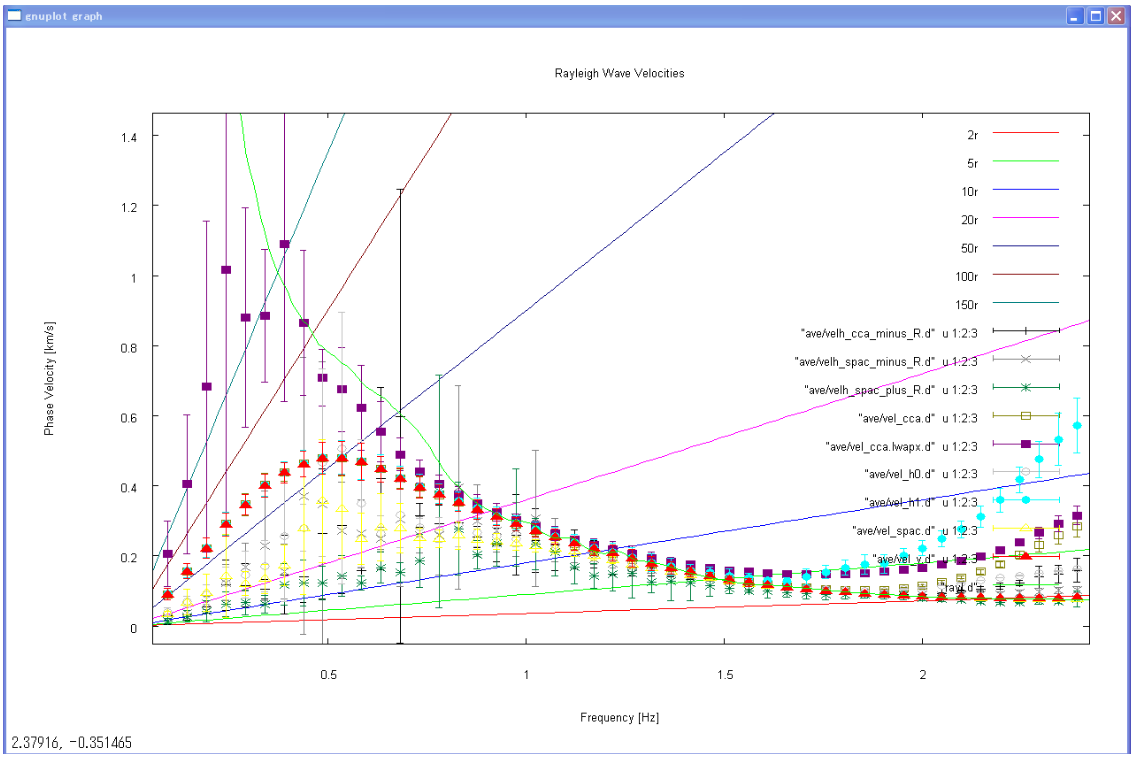


2) Go back to the terminal and press the return key. This will take you to the following plot of the analysis results for phase velocities of Rayleigh waves.

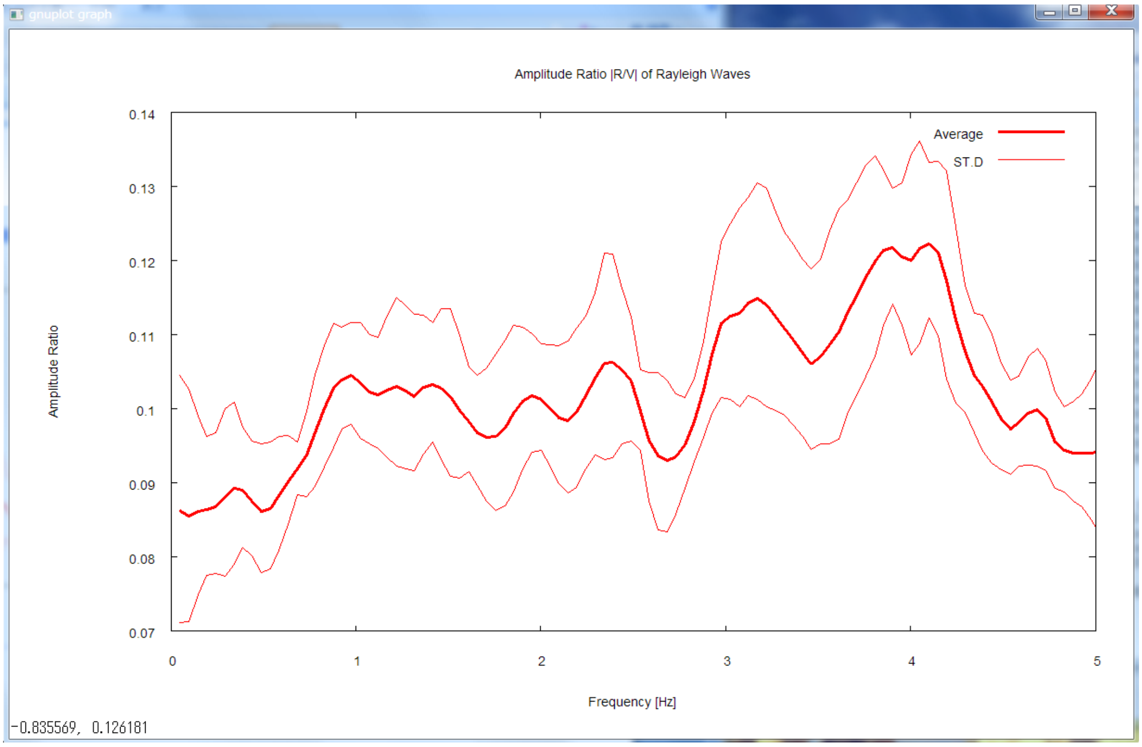


Analysis results from the CCA-R, SPAC-R and SPAC+R methods (using horizontal-motion records) and the CCA, nc-CCA (denoted as CCA-lwapx), H0, H1 and V methods (using vertical-motion records) are plotted simultaneously. The prescribed Rayleigh-wave phase velocities (see A6-1-1) seem to have been reproduced within a frequency range of about 0.5-2 Hz.

For reference, below is an enlarged view of the comparison, up to about 2 Hz, of the phase velocities prescribed during the computation of the synthetic data (solid curve) and the analysis results. The nc-CCA seems to retain analysis capabilities down to the lowest frequency among all methods.

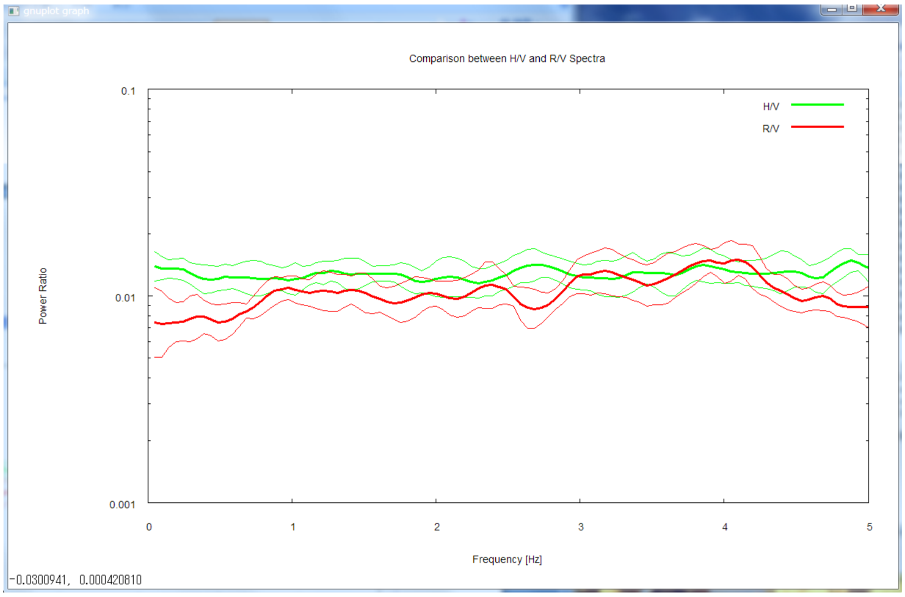


3) Go back to the terminal again and press the return key. This plots the R/V spectrum (amplitude ratios between the horizontal motion and the vertical motion) of Rayleigh waves (heavy curve, mean; light-colored curves, standard deviations).



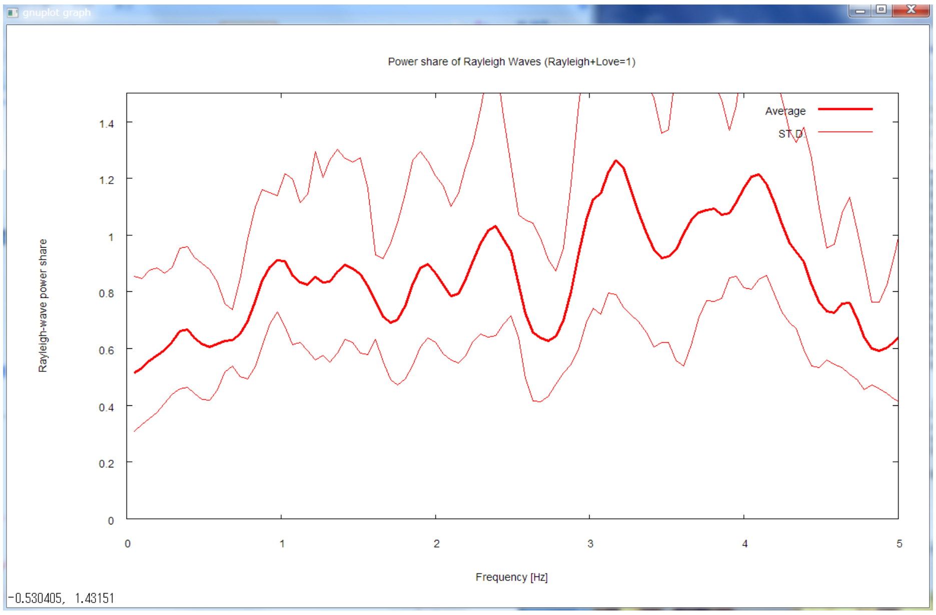
During the calculation of the synthetic data, I set the R/V spectrum at 0.1 irrespective of the frequency (see A6-1-1). The analysis results returned comparable values within a frequency range centered on 1-2 Hz.

4) Pressing the return key again takes you to a comparative plot of the R/V spectrum of Rayleigh waves and the H/V spectrum. Please note that, in this plot, the R/V spectrum is represented in terms of power-spectral ratios (not in terms of the amplitude ratios between the radial and vertical components). As I have stated in 3), the R/V equals 0.1 in terms of the amplitude ratio, so it should be equal to its square, 0.01, in terms of the power ratio. The prescribed R/V and H/V were 0.01 and 0.0125, respectively, a difference well represented in the analysis results given here.



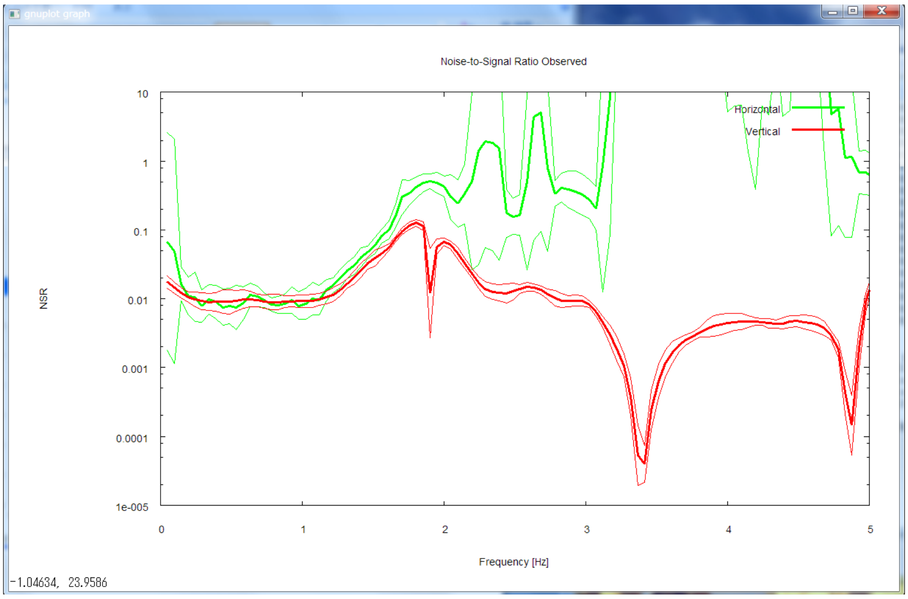
5) Pressing the return key again takes you to a plot of g_R, the power shares of Rayleigh waves in horizontal motion (g_R is defined as the fraction of the power of Rayleigh waves, with the total power of horizontal motion taken as unity. Defining g_L, the power of Love waves, in a similar way, you have the relationship g_R +g_L = 1).

I prescribed g_R =0.8 in the synthetic data (see A6-1-1). The analysis results returned comparable values in a frequency range centered on 1-2 Hz.



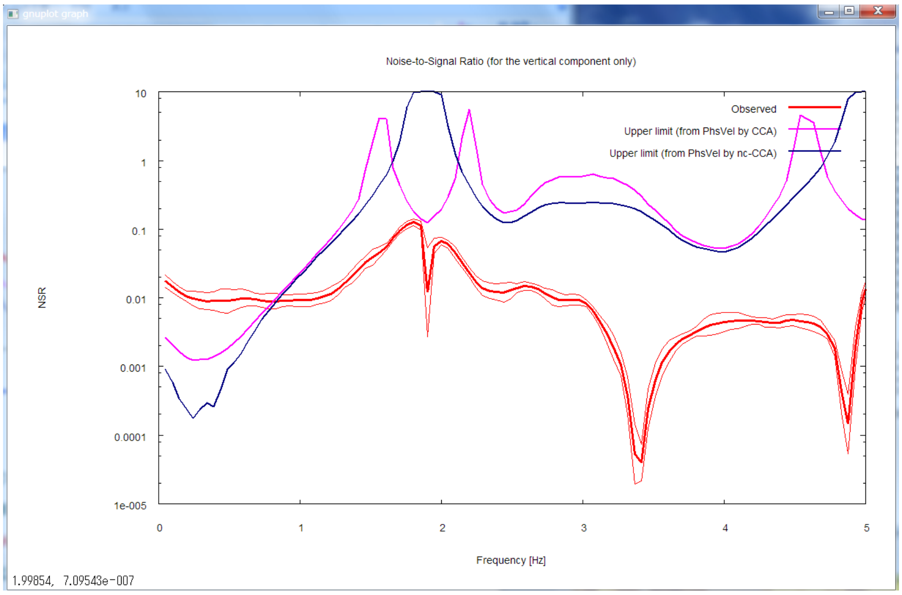
6) Pressing the return key next time takes you to a plot of the NS ratios (inverse of the SN ratios. Ratios of the power of incoherent noise to the power of signals) of horizontal motion (green) and of vertical motion (red).

I set the SN ratio at 100 in the synthetic data (see A6-1-1). This corresponds to an NS ratio of 0.01 because it is the inverse of the SN ratio. In fact, the analysis results returned comparable values in low-frequency ranges below 1 Hz.



```
-1.04634, 23.9586
```

Pressing the return key again leads you to a plot of the NS ratios of vertical motion (red) alone, accompanied by a blue and a pink curve denoted "Upper limit."


gnuplot graph
Noise-to-Signal Ratio (for the vertical component only)
Observed
Upper limit (from PhsVel by CCA)
Upper limit (from PhsVel by nc-CCA)
NSR
10
1
0.1
0.01
0.001
0.0001
1e-005
0
1
2
3
4
5
Frequency [Hz]

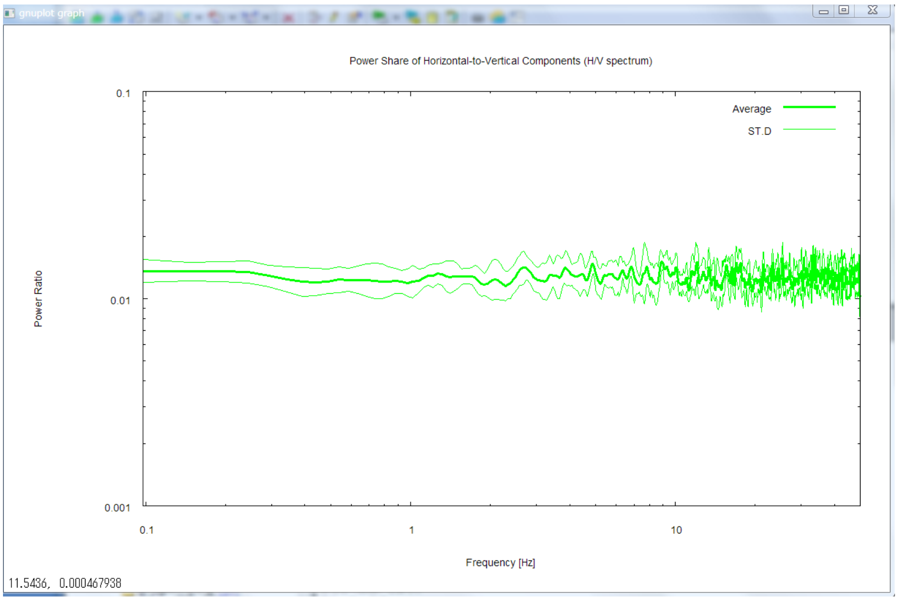
```
1.99854, 7.09543e-007
```

The pink and blue curves in the above figure denote the threshold NS ratios, calculated using phase velocity estimates from the CCA and nc-CCA methods, respectively. **A threshold NS ratio is a reference value to be used to assess the reliability of phase velocity estimates from the**

CCA method (not the nc-CCA method). When the observed NS ratios exceed the threshold NS ratios, the phase velocity estimates from the CCA method tend to be underestimates. In the above figure, red > pink (or red > blue) below about 0.7 Hz, so there is anxiety that the phase velocities inferred by the CCA method may be underestimates in that frequency range. A look at the analysis results in step 2) confirms that this is in fact the case. On the other hand, the nc-CCA method is a method to correct for underestimations due to noise (nc stands for noise-compensated) (Reference [4]). As you can see in the plot in step 2), the nc-CCA method is able to retain analysis capabilities down to lower-frequency ranges as long as the NS ratios are evaluated accurately.

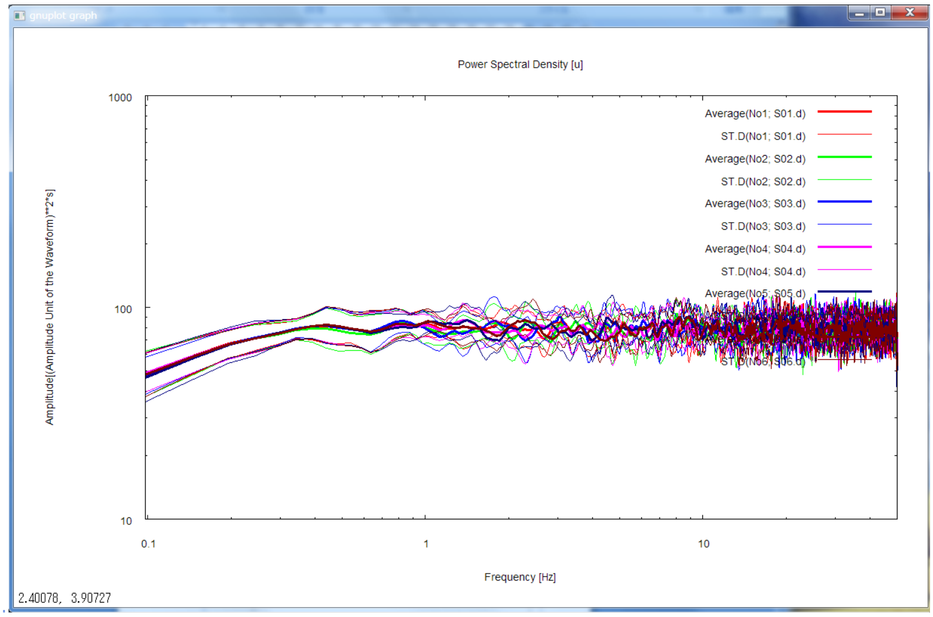
The threshold NS ratios have thus been plotted for vertical motion, but not for horizontal motion. This is because, for vertical motion, a method to calculate the applicability limit of the CCA method has been derived theoretically (Reference [3]) whereas, for horizontal motion, the applicability limits of relevant analysis methods have unfortunately not been elucidated.

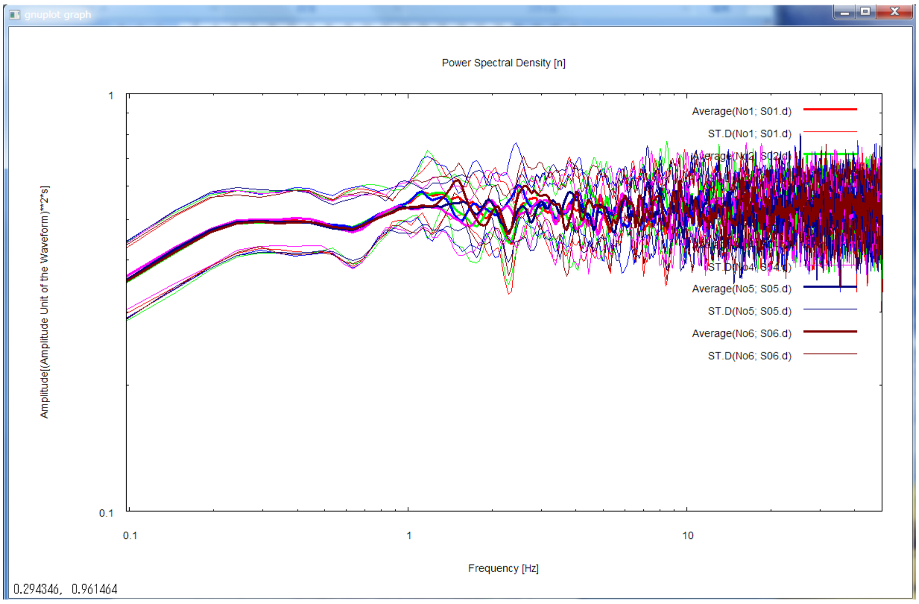
7) Pressing the return key next time plots the H/V spectrum again. This time, the frequency axis is scaled logarithmically and is shown up to the maximum frequency. The maximum frequency refers to the smaller one between the value set by the parameter freqmax_ave in \script/\setpar.sh (the purpose of this parameter is to reduce the computation time by letting the averages calculated only up to the set frequency value. 50 Hz by default, but modifiable on your own) and the Nyquist frequency. If you do not prefer logarithmic scaling, comment out the line containing the parameter autologscale_x in \script\setpar.sh by appending # to the head of the line.

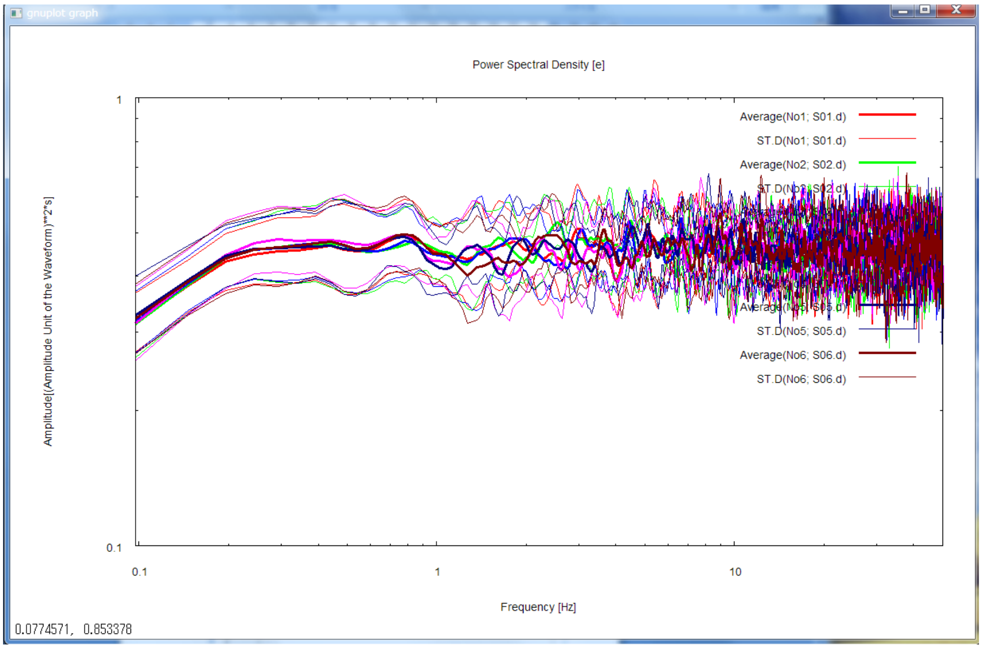


8) Pressing the return key once again plots the power-spectral densities of all records. Please note that the frequency axis is again scaled

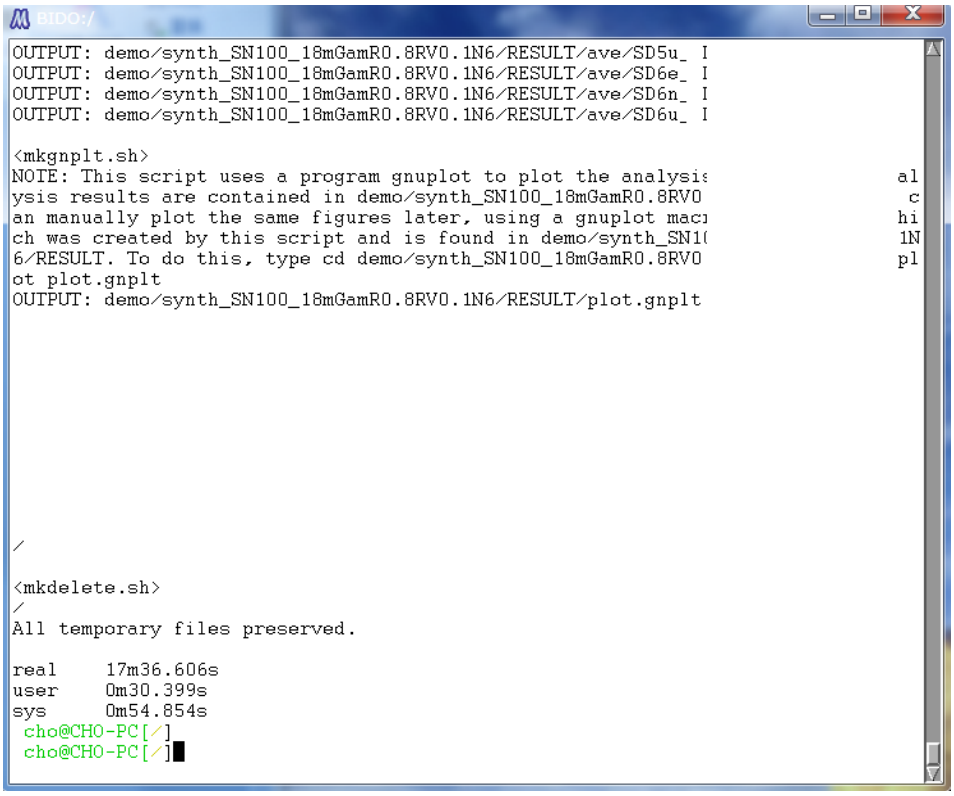
logarithmically and is shown up to the maximum frequency. The letters u, n and e in parentheses ([ ]) on the right-hand side of the headline above the figure represents the up-down, north-south and east-west components, respectively. The numbers in parentheses to the right of the letters indicating the mean and standard deviation in the legends refer to the data described in seism.d and are numbered No. 1, 2, and so forth from top to bottom. To make this point sure, these numbers are followed by data file names like S01.d.







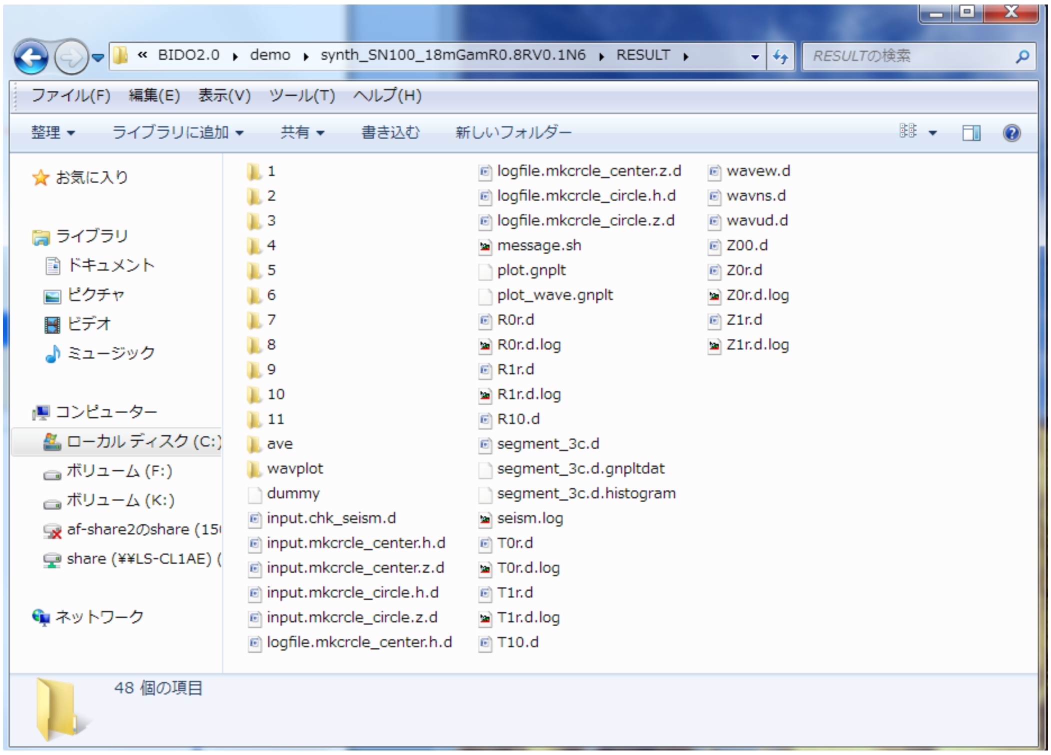
This completes the plotting of the analysis results. When you press the return key, the plot screen disappears, and on the terminal you see:



```
OUTPUT: demo/synth_SN100_18mGamR0.8RV0.1N6/RESULT/ave/SD5u_ I
OUTPUT: demo/synth_SN100_18mGamR0.8RV0.1N6/RESULT/ave/SD6e_ I
OUTPUT: demo/synth_SN100_18mGamR0.8RV0.1N6/RESULT/ave/SD6n_ I
OUTPUT: demo/synth_SN100_18mGamR0.8RV0.1N6/RESULT/ave/SD6u_ I

<mkgnplt.sh>
NOTE: This script uses a program gnuplot to plot the analysis              al
ysis results are contained in demo/synth_SN100_18mGamR0.8RV0                c
an manually plot the same figures later, using a gnuplot mac               hi
ch was created by this script and is found in demo/synth_SN1(              1N
6/RESULT. To do this, type cd demo/synth_SN100_18mGamR0.8RV0               pl
ot plot.gnplt
OUTPUT: demo/synth_SN100_18mGamR0.8RV0.1N6/RESULT/plot.gnplt









/

<mkdelete.sh>
/
All temporary files preserved.

real    17m36.606s
user    0m30.399s
sys     0m54.854s
 cho@CHO-PC[/]
 cho@CHO-PC[/]
```

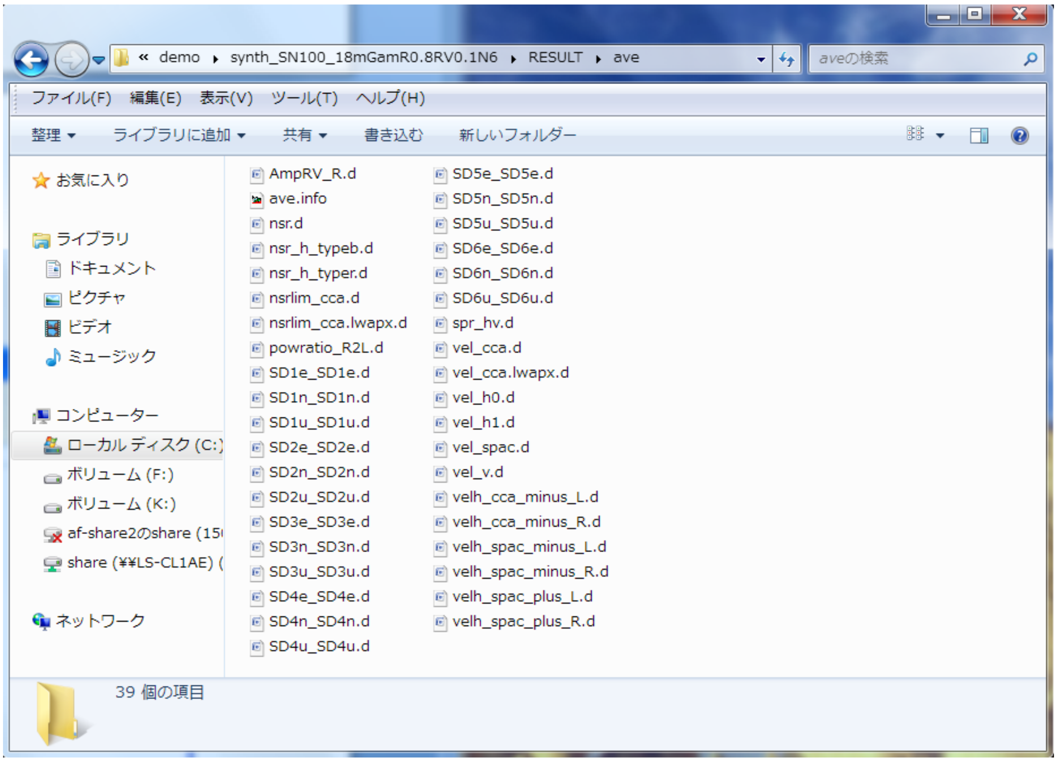
This shows that the temporary files used in the analysis have been deleted and that the whole procedure has ended. The figures following the letters real, user and sys in the last lines indicate the time it took to do the calculations (in this case, it took 17 minutes in real time after you started the calculations and until you arrived at this screen). Typing "exit" at the command prompt or pressing C-Z (simultaneously pressing the control key and the z key) finishes the terminal screen.

The entire analysis is over now. All analysis results are stored in a folder, named RESULT, which has been created beneath the data folder. The following is a look into RESULT (See A6-1-5 for descriptions of the individual files):



1
2
3
4
5
6
7
8
9
10
11
ave
wavplot
dummy
input.chk_seism.d
input.mkcrcle_center.h.d
input.mkcrcle_center.z.d
input.mkcrcle_circle.h.d
input.mkcrcle_circle.z.d
logfile.mkcrcle_center.h.d
logfile.mkcrcle_center.z.d
logfile.mkcrcle_circle.h.d
logfile.mkcrcle_circle.z.d
message.sh
plot.gnplt
plot_wave.gnplt
R0r.d
R0r.d.log
R1r.d
R1r.d.log
R10.d
segment_3c.d
segment_3c.d.gnpltdat
segment_3c.d.histogram
seism.log
T0r.d
T0r.d.log
T1r.d
T1r.d.log
T10.d
wavew.d
wavns.d
wavud.d
Z00.d
Z0r.d
Z0r.d.log
Z1r.d
Z1r.d.log

All data files used in the above plots are stored in the folder named ave (short for average). The following is a look into ave (See A6-1-6 for descriptions of the individual files):

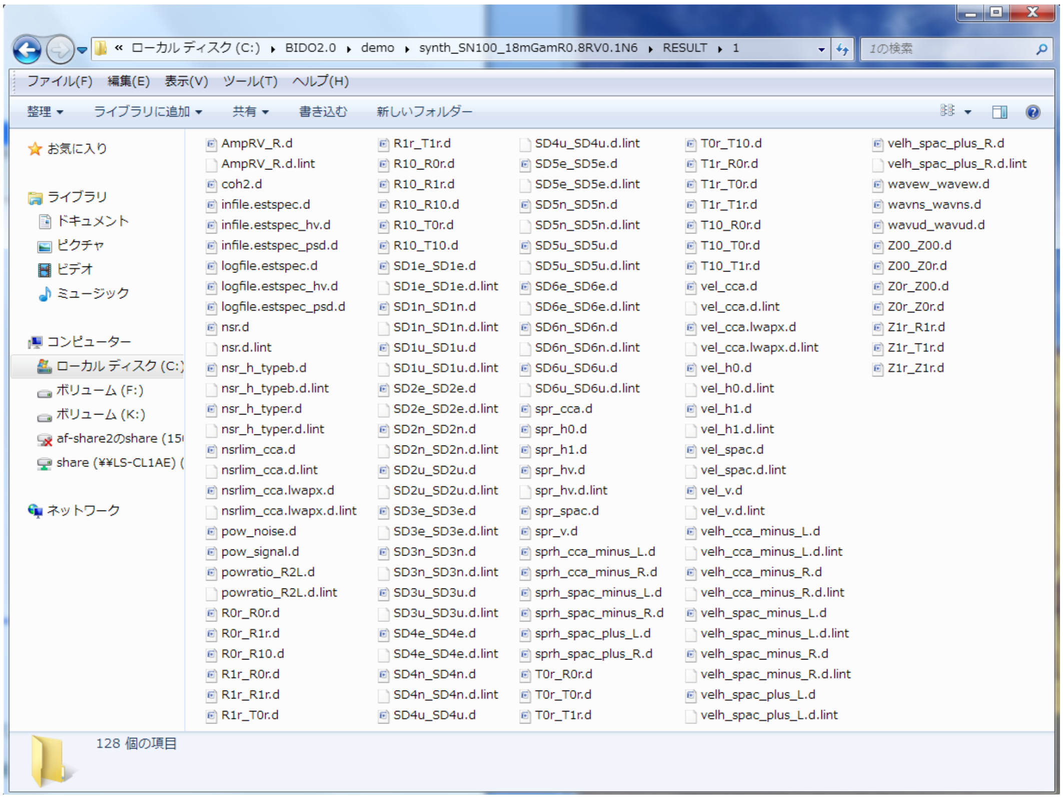


« demo ▸ synth_SN100_18mGamR0.8RV0.1N6 ▸ RESULT ▸ ave

AmpRV_R.d
ave.info
nsr.d
nsr_h_typeb.d
nsr_h_typer.d
nsrlim_cca.d
nsrlim_cca.lwapx.d
powratio_R2L.d
SD1e_SD1e.d
SD1n_SD1n.d
SD1u_SD1u.d
SD2e_SD2e.d
SD2n_SD2n.d
SD2u_SD2u.d
SD3e_SD3e.d
SD3n_SD3n.d
SD3u_SD3u.d
SD4e_SD4e.d
SD4n_SD4n.d
SD4u_SD4u.d
SD5e_SD5e.d
SD5n_SD5n.d
SD5u_SD5u.d
SD6e_SD6e.d
SD6n_SD6n.d
SD6u_SD6u.d
spr_hv.d
vel_cca.d
vel_cca.lwapx.d
vel_h0.d
vel_h1.d
vel_spac.d
vel_v.d
velh_cca_minus_L.d
velh_cca_minus_R.d
velh_spac_minus_L.d
velh_spac_minus_R.d
velh_spac_plus_L.d
velh_spac_plus_R.d

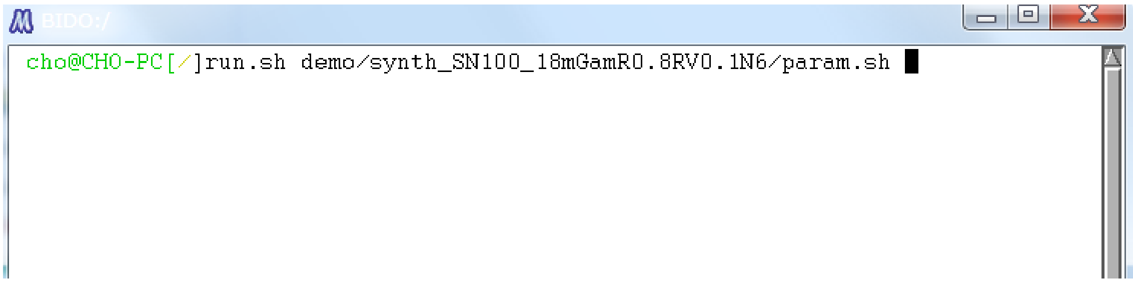
39 個の項目

All files in the ave folder are statistical processings of individual analysis results that are stored in folders with numerical names like 1, 2 and so forth. The numerically named folders contain not just the analysis results intended for statistical processing and storage in ave, but also a variety of analysis results (such as temporary work files and spectral densities) that are normally not used unless you want to do in-depth analysis. For example, the folder named 1 contains the following analysis results (See A6-1-6 for descriptions of the individual files).



```
AmpRV_R.d
AmpRV_R.d.lint
coh2.d
infile.estspec.d
infile.estspec_hv.d
infile.estspec_psd.d
logfile.estspec.d
logfile.estspec_hv.d
logfile.estspec_psd.d
nsr.d
nsr.d.lint
nsr_h_typeb.d
nsr_h_typeb.d.lint
nsr_h_typer.d
nsr_h_typer.d.lint
nsrlim_cca.d
nsrlim_cca.d.lint
nsrlim_cca.lwapx.d
nsrlim_cca.lwapx.d.lint
pow_noise.d
pow_signal.d
powratio_R2L.d
powratio_R2L.d.lint
R0r_R0r.d
R0r_R1r.d
R0r_R10.d
R1r_R0r.d
R1r_R1r.d
R1r_T0r.d
R1r_T1r.d
R10_R0r.d
R10_R1r.d
R10_R10.d
R10_T0r.d
R10_T10.d
SD1e_SD1e.d
SD1e_SD1e.d.lint
SD1n_SD1n.d
SD1n_SD1n.d.lint
SD1u_SD1u.d
SD1u_SD1u.d.lint
SD2e_SD2e.d
SD2e_SD2e.d.lint
SD2n_SD2n.d
SD2n_SD2n.d.lint
SD2u_SD2u.d
SD2u_SD2u.d.lint
SD3e_SD3e.d
SD3e_SD3e.d.lint
SD3n_SD3n.d
SD3n_SD3n.d.lint
SD3u_SD3u.d
SD3u_SD3u.d.lint
SD4e_SD4e.d
SD4e_SD4e.d.lint
SD4n_SD4n.d
SD4n_SD4n.d.lint
SD4u_SD4u.d
SD4u_SD4u.d.lint
SD5e_SD5e.d
SD5e_SD5e.d.lint
SD5n_SD5n.d
SD5n_SD5n.d.lint
SD5u_SD5u.d
SD5u_SD5u.d.lint
SD6e_SD6e.d
SD6e_SD6e.d.lint
SD6n_SD6n.d
SD6n_SD6n.d.lint
SD6u_SD6u.d
SD6u_SD6u.d.lint
spr_cca.d
spr_h0.d
spr_h1.d
spr_hv.d
spr_hv.d.lint
spr_spac.d
spr_v.d
sprh_cca_minus_L.d
sprh_cca_minus_R.d
sprh_spac_minus_L.d
sprh_spac_minus_R.d
sprh_spac_plus_L.d
sprh_spac_plus_R.d
T0r_R0r.d
T0r_T0r.d
T0r_T1r.d
T0r_T10.d
T1r_R0r.d
T1r_T0r.d
T1r_T1r.d
T10_R0r.d
T10_T0r.d
T10_T1r.d
vel_cca.d
vel_cca.d.lint
vel_cca.lwapx.d
vel_cca.lwapx.d.lint
vel_h0.d
vel_h0.d.lint
vel_h1.d
vel_h1.d.lint
vel_spac.d
vel_spac.d.lint
vel_v.d
vel_v.d.lint
velh_cca_minus_L.d
velh_cca_minus_L.d.lint
velh_cca_minus_R.d
velh_cca_minus_R.d.lint
velh_spac_minus_L.d
velh_spac_minus_L.d.lint
velh_spac_minus_R.d
velh_spac_minus_R.d.lint
velh_spac_plus_L.d
velh_spac_plus_L.d.lint
velh_spac_plus_R.d
velh_spac_plus_R.d.lint
wavew_wavew.d
wavns_wavns.d
wavud_wavud.d
Z00_Z00.d
Z00_Z0r.d
Z0r_Z00.d
Z0r_Z0r.d
Z1r_R1r.d
Z1r_T1r.d
Z1r_Z1r.d
```

Coming back to the top data folder, you will find that an execution log, named run.log, and a parameter file, named param.sh, have been generated alongside the RESULT folder for the analysis results. The parameter file (param.sh) can be conveniently used as an argument for run.sh when you rerun the analysis with slightly modified parameters, like in the following panel:



```
cho@CHO-PC[/]run.sh demo/synth_SN100_18mGamR0.8RV0.1N6/param.sh
```

You may be concerned about having to type a long path name or mistyping it. Please take it easy, because there is a typing-aid feature called auto-complete. Auto-complete allows you to type alphabetic keys only halfway down a word and press the TAB key so that the rest of the spelling turns up automatically. For example, if you press

run.sh d [TAB]

it automatically turns into

run.sh demo/

Auto-complete is valid for the rest of the text. So, if you execute run.sh with a parameter file as the argument, the dialogue appearing at the start of the program asks you, by default, about the parameters of the last execution (when the parameters are not given explicitly like this, default_param.sh beneath the script folder is automatically read). Note that, when you rerun the analysis, any folders with numerical names such as 1, 2, and so forth and the ave folder beneath RESULT are automatically deleted. If you do not wish these folders to be deleted, you have to move them somewhere else or rename them before rerun.

Finally, in the present example we have obtained analysis results for the

- Phase velocities of Love waves,
- Phase velocities of Rayleigh waves,
- R/V spectrum of Rayleigh waves,
- Comparison of the R/V spectrum of Rayleigh waves and the H/V spectrum,
- Shares of Rayleigh waves in the total power of horizontal motion,
- NS ratios (inverse of the SN ratios)
- H/V spectrum, and
- Power-spectral densities,

because we used three-component array data with a central station. In general, however, the analysis output (and graphic output) depend on the array geometry (the number of stations around the circle and whether there is a station at the center) and the number of components in the data waveforms (vertical motion alone, horizontal motion alone or all three components). Take a try by editing the seism.d file to set #COMP at 1, 2 and 3 and by commenting out the line concerning the central station (in seism.d, appending # at the head of a line comments that line out, except in the lines for #DT and #COMP). As described in a correspondence table (Appendix 1), the analysis output differs according to the array geometry and the record components. The analysis output is, however, selected automatically by the program, so the user can rely on an identical procedure to carry out the analysis. By way of a sample, I have bundled a few different patterns of seismfile (including seism.d.1.station) in the folder \demo\synth_SN100_18mGamR0.8RV0.1N6. Please rename each one of them to seism.d and try rerun of the analysis.

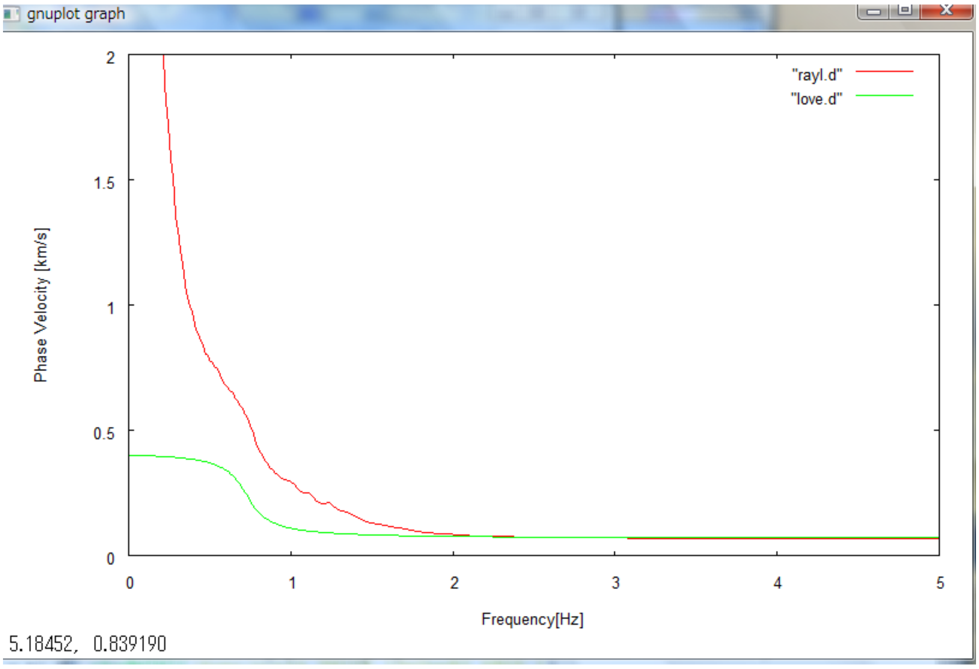
A6-1-1 Details of the Synthetic Data

The synthetic data were generated under the assumption that the field of microtremors satisfied the following conditions.

1) Chacateristics of the microtremor wavefield (signals)
- The field of microtremors is dominated by surface waves (Rayleigh and Love waves).

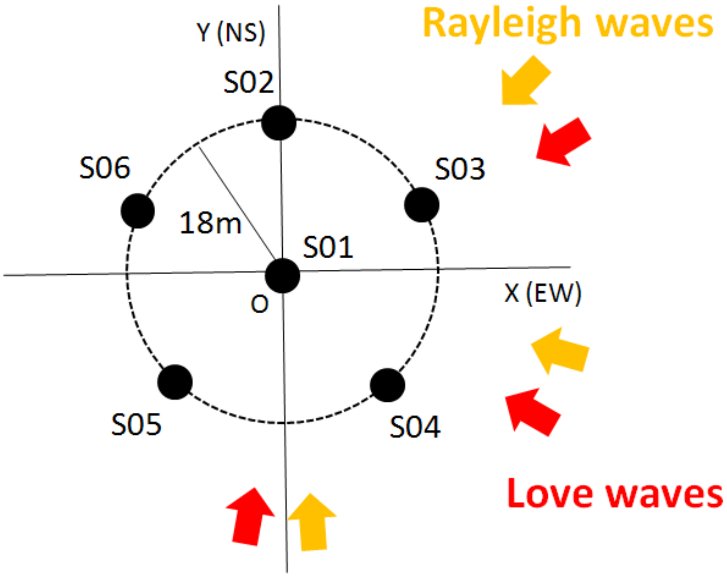
- The waves arrive as plane waves in the array.
- The wavefield is stationary in both time and space.
- The Rayleigh and Love waves are mutually uncorrelated.

2) Phase velocities of Rayleigh and Love waves
Phase velocities as illustrated in the figure below were prescribed to the Rayleigh and Love waves (red and green, respectively).



3) Arrival directions and intensities of Rayleigh and Love waves
Rayleigh and Love waves were made to arrive as plane waves in the array as illustrated in the figure below. We assumed that both Rayleigh and Love waves arrived from all directions with equal intensities, but that Rayleigh and Love waves had different intensities. More specifically, we set the Rayleigh-to-Love power ratio at 4:1. The horizontal-to-vertical amplitude ratio of the Rayleigh waves (which we call R/V in this manual) was set at 1:10. The ratios were fixed at these values for all frequencies.

4) Noise
On top of the Rayleigh- and Love-wave signals we added incoherent noise with the following properties.
- Records of noise at different stations are mutually uncorrelated.
- Noise is stationary in time.
- Noise is uncorrelated with the signals.
We calculated the noise intensity corresponding to an SN ratio of 100 for all UD, NS and EW components, and added noise on top of the signals composed of Rayleigh and Love waves. The term NS ratio, which we use repeatedly in this manual, refers to the inverse of the SN ratio.



A6-1-2 The dialogue that appears on activating the program asks about the following parameters.

Enter appropriate values according to the instructions written in red.

Data directory name [demo/synth_SN100_18mGamR0.8RV0.1N6] >← Enter the name of the folder where the data files are stored. Separate the folder path with slashes (/).

* Execution parameters
Do spectral analysis [e.g., Calculate phase velocities and other properties of surface waves]? (y/n=1/0) [1] ← Set this option to 1 to execute spectral analysis and phase velocity analysis.
Preprocess the data? [This includes elimination of the trend (including subtraction of the mean as a special case), application of tapers, bandpass filtering, decimation and correction for differences in instrumental response] (y/n=1/0) [0] ← Set this option to 1 to execute the data preprocessing.
Automatically select data portions to be used? (y/n=1/0)[1] ← Setting this option to 1 means that data portions to be used in the estimation of spectra are selected automatically from the waveform data (**click here for the algorithm of automatic selection**) and that a segment file is generated (**click here for the segment file names and their format**). Create a segment file on your own and set this option to 0 if you wish to make your own selection of the portions of the waveform to be used in the analysis.
Delete temporary data files? (y/n=1/0)[0] ← Setting this option to 1 ensures that all intermediary data, generated on the way, are deleted when the data processing is over, except for the final analysis results and a small

number of input files and log files. Setting this to 0 allows all intermediary data to be preserved.
Plot analysis results? (y/n=1/0)[1] ← Set this option to 1 to plot the analysis results.

* Basic parameters
Take a look at **Estimating Spectral Densities** and **A Knack for Setting Parameters** before proceeding to enter the following parameters.
Duration of data segments for the evaluation of spectra [s] [10.24] ← The segment duration [seconds] used in the estimation of spectra (segment averaging method). When the number of data points, calculated by (segment duration) / (sampling time interval), is not equal to a power of two, it is automatically zero-padded to a power of two during FFT (for the sake of efficiency, it is recommendable to adjust values to produce a power of two).
Number of data segments over which averages are taken. Enter 0 or a very large number if you wish to use all segments simultaneously [10] ← The number of segments (integer) used in the estimation of spectra (segment averaging method). Entering a large integer, which exceeds the total number of segments catalogued in the segment file, or zero means "segment averaging using a single segment (virtually no segment averaging)."
Band width of the Parzen spectral window [Hz] [0.3] ← The width of the spectral (Parzen) window [Hz] used in the estimation of spectra (smoothing method). Entering zero means "a zero window width (virtually no use of a spectral window)."

* Data file and array geometry
A preexisting seismfile demo/synth_SN100_18mGamR0.8RV0.1/seism.d (for the data file names and the array geometry) has been detected. ← Prescribe the array geometry, waveform components and the sampling time interval in \(data folder)\seism.d. When no file is found under this name, interactive questions generate one automatically.
----------------citation begins here------------------
#COMP 3 (1 ud / 2 ns & ew / 3 three components) ← Enter the waveform components (type 1, 2 or 3 following a space or a tab after #COMP).
#DT 0.01 ← Enter the time interval of waveform sampling (type a figure following a space or a tab after #DT).
0.000000   0.000000   S01.d   1 ← From left to right, the x coordinate [km], y coordinate [km], data file name, and integer 1 or 0 (1 if center).

0.000001	0.018000	S02.d	0
0.017119	0.005562	S03.d	0
0.010581	-0.014562	S04.d	0
-0.010579	-0.014563	S05.d	0
-0.017119	0.005564	S06.d	0

Notes

(*1) Appending "#" at the head of a line comments that line out (except for #COMP and #DT). There is no rule on the order of arrangement of these data. There is no special format for the numbers.
(*2) When only two sensors constitute the array, think of either one of them as being at the center and the other as lying around the circle, and accordingly assign them integers of 1 and 0 to the right of the data file names.
(*3) When there is only one sensor constituting the array, the integer to the right of the data file name can be either 0 or 1 (the H/V spectrum alone is calculated, and there will be no array processing).
(*4) When the records have three components (#COMP 3), the H/V spectrum is calculated using the file indicated in the top data line.

A6-1-3 The seismfile

Description
A file prescribing the array geometry, data components and the sampling time interval.

File Name
\(data folder)\RESULT\seism.d

Format

#COMP 3 (1 ud / 2 ns & ew / 3 three components) ←Enter the waveform components (type 1, 2 or 3 following a space or a tab after #COMP).

#DT 0.01 ←Enter the time interval of waveform sampling (type a figure following a space or a tab after #DT).

0.000000   0.000000   S01.d 1 ←From left to right, the x coordinate [km], y coordinate [km], data file name, and integer 1 or 0 (1 if center).

```
0.000001     0.018000     S02.d  0
0.017119     0.005562     S03.d  0
0.010581     -0.014562    S04.d  0
-0.010579    -0.014563    S05.d  0
-0.017119    0.005564     S06.d  0
```

Notes
(*1) Appending "#" at the head of a line comments that line out (except for #COMP and #DT). There is no rule on the order of arrangement of these data. There is no special format for the numbers.
(*2) When only two sensors constitute the array, think of either one of them as being at the center and the other as lying around the circle, and accordingly assign them integers of 1 and 0 to the right of the data file names. When three sensors constitute the array, there are two options: either you think of it as an array of two stations around the circle plus

another at the center and apply the SPAC and H0 methods, or you think of all three stations as lying around the circle and apply the CCA method.
(*3) When there is only one sensor constituting the array, the integer to the right of the data file name can be either 0 or 1 (the power spectral densities and the H/V spectrum alone are calculated, and there will be no array processing).
(*4) When the records have three components (#COMP 3), the H/V spectrum is calculated using the file indicated in the top data line.
(*5) Assigning identical x and y coordinates to all stations automatically activates the huddle test mode, where the power spectral densities, the magnitude-squared coherences and phase differences are calculated. The coherences and phase differences calculated are relative to the data indicated in the top line.

Example: How to write the seismfile in the case of a huddle test

```
#COMP 3
#DT 0.01
0.000000     0.000000     S01.d  1
0.000000     0.000000     S02.d  0
0.000000     0.000000     S03.d  0
```

A6-1-4 The segment File

Description
A file indicating which segment portions should be extracted from the waveform data.

File Name
The file is named differently according to what components are contained in the waveform data.
- Vertical motion alone → \(data folder)\RESULT\segment_z.d
- Horizontal motion alone → \(data folder)\RESULT\segment_h.d
- All three components → \(data folder)\RESULT\segment_3c.d

Format

```
115 ←Total number of segments
10.24 ←Segment duration
0.01 ←Time interval of waveform sampling
0. ←Start time of the first segment
5.12 ←Start time of the second segment
10.24 ←Start time of the third segment
...
```

Numbers should be laid out like this, one in each line. There is no particular format for the numbers (except that an integer alone is allowed in the first line).

A6-1-5 Analysis Results (folder RESULT)

1, 2, ..., *n*, ... Folder containing analysis results relevant to the *n*th segment cluster

ave Folder containing data files used in gnuplot plots (files with statistics of the analysis results in folders 1, 2, ...)

R0r.d Zeroth-order Fourier coefficients for the radial component of measurement data around the circle

R0r.d.log Execution log of the program to generate R0r.d

R10.d First-order Fourier coefficients for the radial component of measurement data at the center

R1r.d First-order Fourier coefficients for the radial component of measurement data around the circle

R1r.d.log Execution log of the program to generate R1r.d

T0r.d Zeroth-order Fourier coefficients for the tangential component of measurement data around the circle

T0r.d.log Execution log of the program to generate T0r.d

T10.d First-order Fourier coefficients for the tangential component of measurement data at the center

T1r.d First-order Fourier coefficients for the tangential component of measurement data around the circle

T1r.d.log Execution log of the program to generate T1r.d

Z00.d Zeroth-order Fourier coefficients for the UD component of measurement data at the center

Z0r.d Zeroth-order Fourier coefficients for the UD component of measurement data around the circle

Z0r.d.log Execution log of the program to generate Z0r.d

Z1r.d First-order Fourier coefficients for the UD component of measurement data around the circle

Z1r.d.log Execution log of the program to generate Z1r.d

wavud.d Copy of the data file indicated first in the seism file. Used to estimate the H/V spectrum

wavns.d Copy of the data file indicated first in the seism file. Used to estimate the H/V spectrum

wavew.d Copy of the data file indicated first in the seism file. Used to estimate the H/V spectrum

dummy Used in gnuplot plots. A dummy data file that contains nothing

input.mkcrcle_center.h.d Input data for the calculation of Fourier coefficients at the center (horizontal components)

input.mkcrcle_center.z.d Input data for the calculation of Fourier coefficients at the center (UD component)

input.mkcrcle_circle.h.d Input data for the calculation of Fourier coefficients around the circle (horizontal components)

input.mkcrcle_circle.z.d Input data for the calculation of Fourier coefficients around the circle (UD component)

logfile.mkcrcle_center.h.d Execution log of the program to calculate Fourier coefficients at the center (horizontal components)

logfile.mkcrcle_center.z.d Execution log of the program to calculate Fourier coefficients at the center (UD component)
logfile.mkcrcle_circle.h.d Execution log of the program to calculate Fourier coefficients around the circle (horizontal components)
logfile.mkcrcle_circle.z.d Execution log of the program to calculate Fourier coefficients around the circle (UD component)
message.sh Work file that contains messages outputted during execution of the calculations plus a number of parameters
input.chk_seism.d Work (input) file for a program that checks the coordinates of the array data
plot.gnplt Macro to activate gnuplot (to plot analysis results)
wavplot Work directory to store data that are decimated to plot waveform data using gnuplot
plot_wave.gnplt Macro to activate gnuplot (to plot waveforms)
segment_3c.d Segment file (A6-1-4) describing which segments were extracted from the measurement data
segment_3c.d.histogram Histogram of RMSs used in the automatic selection of the segments
segment_3c.d.gnpltdat Work data file for drawing waveform data using gunplot

A6-1-6 Analysis Results (folder RESULT/ave)

AmpRV_R.d Horizontal-to-vertical amplitude ratios of Rayleigh waves (logarithmic mean)
SD1e_SD1e.d Power-spectral density function (logarithmic mean). The data described in the seismfile are assigned the numbers 1, 2, ... from top to bottom. The letters e, n and u denote the east-west. north-south, and up-down components, respectively.
ave.info Rough explanation of how the analysis results were averaged
nsr.d NS ratios of vertical motion (logarithmic mean)
nsr_h.d NS ratios of horizontal motion (logarithmic mean)
nsrlim_cca.d Upper limits on NS ratios calculated using phase velocity estimates from the CCA method (5% tolerance for relative errors) (logarithmic mean)
nsrlim_cca.lwapx.d Upper limits on NS ratios calculated using phase velocity estimates from the nc-CCA method (5% tolerance for relative errors) (logarithmic mean)
powratio_R2L.d Power shares of Rayleigh waves in the total power of horizontal motion (logarithmic mean)
spr_hv.d Ratios of the power of horizontal motion (sum of the two horizontal-component powers) to the power of vertical motion (logarithmic mean)
vel_cca.d Phase velocities of Rayleigh waves according to the CCA method

vel_cca.lwapx.d Phase velocities of Rayleigh waves according to the nc-CCA method

vel_h0.d Phase velocities of Rayleigh waves according to the H0 method

vel_h1.d Phase velocities of Rayleigh waves according to the H1 method

vel_spac.d Phase velocities of Rayleigh waves according to the SPAC method (with spatial autocorrelation coefficients calculated according to their original definition)

vel_v.d Phase velocities of Rayleigh waves according to the V method

velh_cca_minus_L.d Phase velocities of Love waves according to the CCA-L method

velh_cca_minus_R.d Phase velocities of Rayleigh waves according to the CCA-R method

velh_spac_minus_L.d Phase velocities of Love waves according to the SPAC-L method

velh_spac_minus_R.d Phase velocities of Rayleigh waves according to the SPAC-R method

velh_spac_plus_L.d Phase velocities of Love waves according to the SPAC+L method

velh_spac_plus_R.d Phase velocities of Rayleigh waves according to the SPAC+R method

* In the above data files, the frequency, the mean of segment-by-segment analysis results and the standard deviation are laid out in the first, second and third columns, respectively. For data with a note "(logarithmic mean)" in parentheses, the frequency, the mean, the mean minus a standard deviation and the mean plus a standard deviation are laid out in the first, second, third and fourth columns, respectively.

A6-1-7 Analysis Results (folders with alphanumeric names)

infile.estspec.d Input data for the estimation of spectra (array analysis)

infile.estspec_hv.d Input data for the estimation of spectra (H/V)

infile.estspec_psd.d Input data for the estimation of spectra (power spectral densities)

logfile.estspec.d Execution log of the program to estimate spectra (array analysis)

logfile.estspec_hv.d Execution log of the program to estimate spectra (H/V)

logfile.estspec_psd.d Execution log of the program to estimate spectra (power spectral densities)

The following are spectral estimates for the array analysis. For example, Z1r_R1r.d represents cross-spectral densities between the Z (UD)-

component, first-order, circumferential data and the radial (R)-component, zeroth-order, circumferential data.
R0r_R0r.d R0r_R10.d R0r_R1r.d R10_R0r.d R10_R10.d R10_R1r.d
R10_T0r.d R10_T10.d R1r_R0r.d R1r_R1r.d R1r_T0r.d R1r_T1r.d
T0r_R0r.d T0r_T0r.d T0r_T10.d T0r_T1r.d T10_R0r.d T10_T0r.d
T10_T1r.d T1r_R0r.d T1r_T0r.d T1r_T1r.d Z00_Z00.d Z00_Z0r.d
Z0r_Z00.d Z0r_Z0r.d Z1r_R1r.d Z1r_T1r.d Z1r_Z1r.d
The following are temporary files for the estimation of H/V ratios (power-spectral densities). For example, wavew_wavew.d represents power-spectral densities of the east-west component. The data indicated in the top line of the seismfile are used in the estimation of H/V ratios.
wavew_wavew.d wavns_wavns.d wavud_wavud.d
The following are power-spectral density estimates. For example. SD1e_SD1e.d represents power-spectral densities of the east-west component of the data indicated in the top line of the seismfile.
SD1e_SD1e.d SD1n_SD1n.d SD1u_SD1u.d SD2e_SD2e.d SD2n_SD2n.d
SD2u_SD2u.d SD3e_SD3e.d SD3n_SD3n.d SD3u_SD3u.d SD4e_SD4e.d
SD4n_SD4n.d SD4u_SD4u.d SD5e_SD5e.d SD5n_SD5n.d SD5u_SD5u.d
SD6e_SD6e.d SD6n_SD6n.d SD6u_SD6u.d

coh2.d Magnitude-squared coherences between the vertical-motion, zeroth-order, circumferential data and the central data
nsr.d NS ratios of vertical motion
nsrlim_cca.d Upper limits on NS ratios calculated using phase velocity estimates from the CCA method (5% tolerance for relative errors)
nsrlim_cca.lwapx.d Upper limits on NS ratios calculated using phase velocity estimates from the nc-CCA method (5% tolerance for relative errors)
nsr_h.d NS ratios of horizontal motion
pow_noise.d Power of noise at the center, vertical motion
pow_signal.d Power of signals at the center, vertical motion
AmpRV_R.d Horizontal-to-vertical amplitude ratios of Rayleigh waves
powratio_R2L.d Power shares of Rayleigh waves in the total power of horizontal motion

spr_cca.d Spectral ratios used to infer phase velocities of Rayleigh waves with the CCA method
spr_cca.lwapx.d Spectral ratios used to infer phase velocities of Rayleigh waves with the nc-CCA method
spr_h0.d Spectral ratios used to infer phase velocities of Rayleigh waves with the H0 method
spr_h1.d Spectral ratios used to infer phase velocities of Rayleigh waves with the H1 method
spr_hv.d Ratios of the power of horizontal motion (sum of the two horizontal-component powers) to the power of vertical motion
spr_spac.d Spectral ratios used to infer phase velocities of Rayleigh waves with the SPAC method (with spatial autocorrelation coefficients calculated according to their original definition)

spr_v.d Spectral ratios used to infer phase velocities of Rayleigh waves with the V method

sprh_cca_minus_L.d Spectral ratios used to infer phase velocities of Love waves with the CCA-L method

sprh_cca_minus_R.d Spectral ratios used to infer phase velocities of Rayleigh waves with the CCA-R method

sprh_spac_minus_L.d Spectral ratios used to infer phase velocities of Love waves with the SPAC-L method

sprh_spac_minus_R.d Spectral ratios used to infer phase velocities of Rayleigh waves with the SPAC-R method

sprh_spac_plus_L.d Spectral ratios used to infer phase velocities of Love waves with the SPAC+L method

sprh_spac_plus_R.d Spectral ratios used to infer phase velocities of Rayleigh waves with the SPAC+R method

vel_cca.d Phase velocities of Rayleigh waves according to the CCA method

vel_cca.lwapx.d Phase velocities of Rayleigh waves according to the nc-CCA method

vel_h0.d Phase velocities of Rayleigh waves according to the H0 method

vel_h1.d Phase velocities of Rayleigh waves according to the H1 method

vel_spac.d Phase velocities of Rayleigh waves according to the SPAC method (with spatial autocorrelation coefficients calculated according to their original definition)

vel_v.d Phase velocities of Rayleigh waves according to the V method

velh_cca_minus_L.d Phase velocities of Love waves according to the CCA-L method

velh_cca_minus_R.d Phase velocities of Rayleigh waves according to the CCA-R method

velh_spac_minus_L.d Phase velocities of Love waves according to the SPAC-L method

velh_spac_minus_R.d Phase velocities of Rayleigh waves according to the SPAC-R method

velh_spac_plus_L.d Phase velocities of Love waves according to the SPAC+L method

velh_spac_plus_R.d Phase velocities of Rayleigh waves according to the SPAC+R method

*.lint Data resampled for statistical processing

A6-1-8 One Approach to Make the Most of the Method's Potential

The CCA method, one of the analysis methods adopted in our codes, allows one to analyze waves of very long wavelengths when the SN ratios (signal-to-noise power ratios) are sufficiently good. Presence of noise sources near the seismic array lowers SN ratios, but it is difficult to control microtremor noise sources in urban areas where humans live and industries are active. The only solution would be to use "miniature arrays" with very small sizes (if the array lies within the reach of arms, it should be easy to ensure a calm environment in its surroundings, at least while the measurement is active [about 15 minutes]). The pictures in the top page show how we realized large SN ratios by using an array with a radius of just 30 cm. This array realized SN ratios in excess of 10,000, which made it possible to analyze wavelengths larger than 500 times the array radius (reference [6]).

A6-1-9 A Knack for Setting Parameters

One major feature of the analysis methods adopted in our codes lies in their ability to analyze long wavelengths. A knack for setting analysis parameters to make the most of this feature is to make the segment duration as small as possible within the tolerable range that depends on the frequency band of interest you wish to look at. This measure is effective to extracting a large number of segments from observed waveforms that are contaminated by non-stationary noise, and helps to make as narrow as possible the bandwidth of smoothing with a spectral window. Spectral windowing can cause biases, which becomes a particularly critical problem in the analysis of long wavelength ranges (reference [3]).

6-2 Demo of Preprocessing

BIDO can perform the following sorts of data preprocessing as part of array data (or huddle data) analysis.

1) Removal of trends (direct-current components)
   In case the waveform baseline drifts linearly with time, removing its influence may help to extract a larger number of segments for use in spectral analysis, and to do so more appropriately.
2) Tapering of waveforms
   A preprocessing to be applied prior to bandpass filtering that makes the waveform amplitudes converge more smoothly toward zero near both ends.
3) Bandpass filtering

Cutting off noise-rich frequency bands by filtering may help to extract a larger number of segments for use in spectral analysis, and to do so more appropriately.

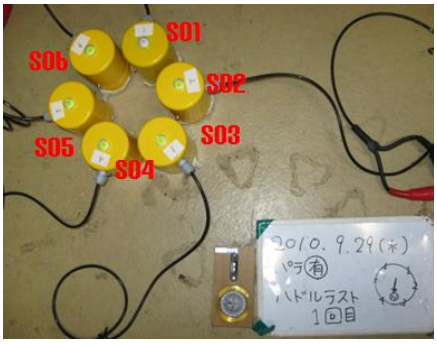
4) Data decimation

This helps to reduce the time it takes to do spectral analysis. Decimation cuts high-frequency bands off the data, but it does not affect the accuracy of analysis in the analyzable frequency bands.

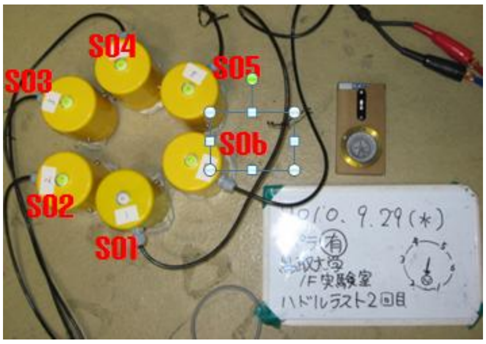
5) Correction for differences in instrumental characteristics

Data from a recording system, where instrumental response characteristics differ from channel to channel, are not immediately suitable for array analysis. They may turn into usable array data when the differences in instrumental characteristics are corrected for.

Those who are using BIDO for the first time are requested to read Demo of array data analysis (6-1) first in order to get familiar with its general usage. Here we explain the preprocessing part by using the demo data in \demo\HDL0002, which were made available by courtesy of Dr Tatsuya Noguchi of Tottori University, just as was the case with \demo\HDL0001 (see Demo of a huddle test (6-3) for descriptions of HDL0001). They were recorded on the same day and under the same environment as HDL0001, except for differences in the sensor locations as shown in the photos below.



HDL0001

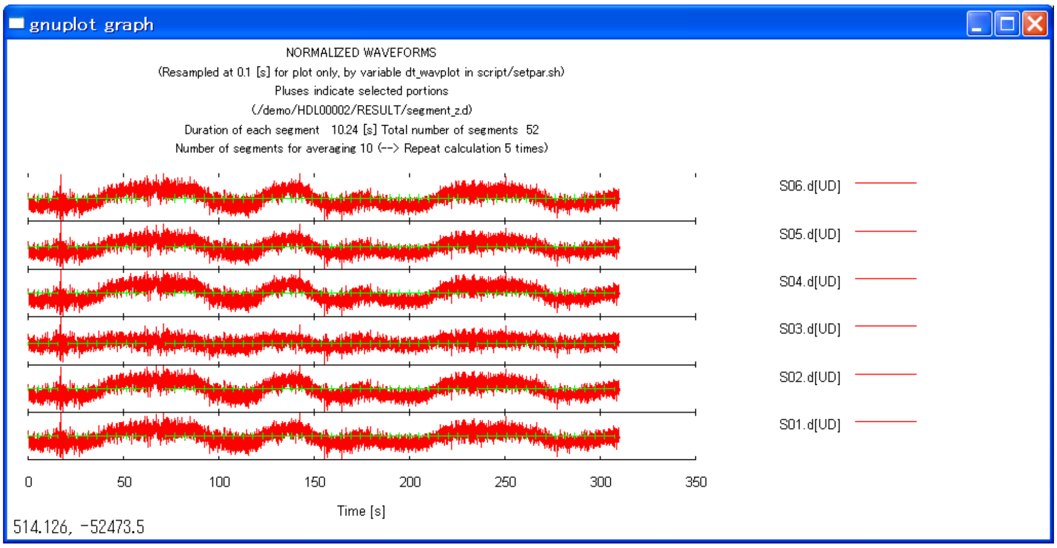


HDL0002

Please start the analysis by typing

```
run.sh  demo/HDL0002/param.sh [RETURN KEY].
```

You will be asked in a dialogue, "Preprocessing the data?" Answering "y" (for yes) allows you to apply preprocessing to the data. If you use the default answers to all the questions in this demo, you will first encounter the following plot of waveforms. These are the original data before preprocessing.



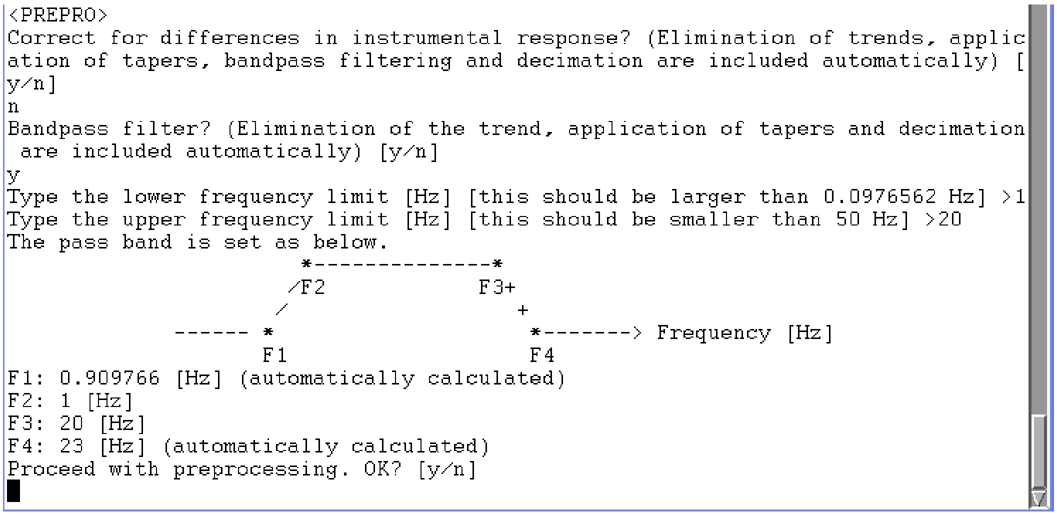
NORMALIZED WAVEFORMS
(Resampled at 0.1 [s] for plot only, by variable dt_wavplot in script/setpar.sh)
Pluses indicate selected portions
(/demo/HDL00002/RESULT/segment_z.d)
Duration of each segment 10.24 [s] Total number of segments 52
Number of segments for averaging 10 (--> Repeat calculation 5 times)

Please be sure to take a look at important messages that will be given at this stage in the dialogue:

-------------------------Citation begins here------------------------------
Do you want to preprocess the waveforms? [y/n]
[This includes elimination of the trend, application of tapers (and bandpass filtering and decimation if necessary)]

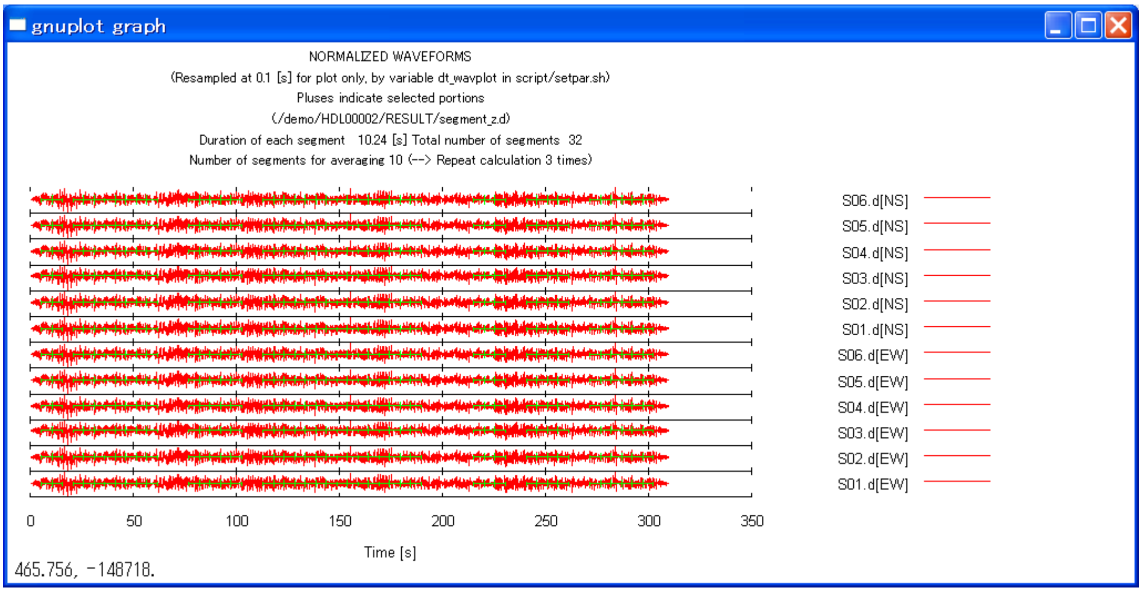
NOTE: The original data will be moved to a directory named "originaldata", which will be automatically created by BIDO under the directory where the original data files are currently stored. Instead, new files with the same names as those of the original data will be created to store the preprocessed data. (The seismfile will also be moved to the directory "originaldata" as well, and newly created by BIDO under the directory where the original seismfiles are currently stored. This is necessary because the preprocessing possibly involves decimation.) The original data files will not be overwritten (destroyed) by BIDO. It is strongly recommended, however, to create backup of the original data files to avoid their accidental destruction.
Type "n" to skip preprocessing.
-------------------------Citation ends here------------------------------

Typing "y" here launches a dialogue for implementing the preprocessing. Answer "n" to "Correct for the difference in instrumental response?" and "y" to "Bandpass filter?" (corrections for the differences in instrumental response will be explained later). Setting the cutoff limits on the low- and high-frequency sides at 1 and 20 [Hz] respectively, you will see a message, asking for final confirmation, as follows:



```
<PREPRO>
Correct for differences in instrumental response? (Elimination of trends, applic
ation of tapers, bandpass filtering and decimation are included automatically) [
y/n]
n
Bandpass filter? (Elimination of the trend, application of tapers and decimation
 are included automatically) [y/n]
y
Type the lower frequency limit [Hz] [this should be larger than 0.0976562 Hz] >1
Type the upper frequency limit [Hz] [this should be smaller than 50 Hz] >20
The pass band is set as below.
                          *---------------*
                         /F2             F3+
                        /                   +
             ------ *                        *-------> Frequency [Hz]
                    F1                       F4
F1: 0.909766 [Hz] (automatically calculated)
F2: 1 [Hz]
F3: 20 [Hz]
F4: 23 [Hz] (automatically calculated)
Proceed with preprocessing. OK? [y/n]
█
```

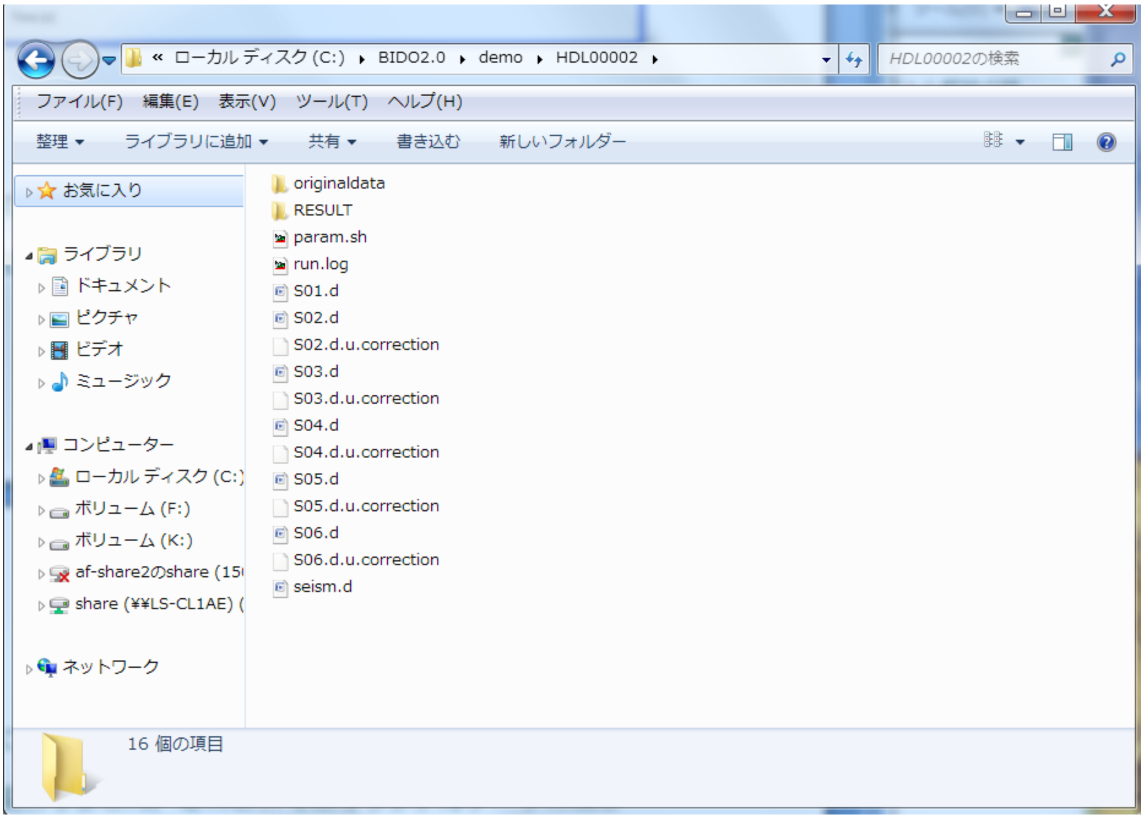
Type "y", and you will see the preprocessed waveforms as follows. Bandpass filtering has eliminated the undulations (components of long periods on the order of tens of seconds) that the original data contained.



NORMALIZED WAVEFORMS
(Resampled at 0.1 [s] for plot only, by variable dt_wavplot in script/setpar.sh)
Pluses indicate selected portions
(/demo/HDL00002/RESULT/segment_z.d)
Duration of each segment 10.24 [s] Total number of segments 32
Number of segments for averaging 10 (--> Repeat calculation 3 times)

The application of a bandpass filter automatically involves the elimination of trends, tapering and decimation. The taper is of a cosine type and is applied, by default, to 5% parts on both ends of the waveforms. The length percentage of tapering can be modified through the variable tpend in \script\setpar.sh. Bandpass filtering uses a Chebyshev filter I with an equiripple passband as described by Saito (1978). After filtering, the data are decimated automatically (to a maximal extent) by considering the cutoff on the high-frequency side. In the present analysis, the sampling time interval is 0.01 sec in the original data, but high-frequency ranges in excess of 20-23 Hz have been discarded through filtering. Therefore, the data are decimated so as to reset the sampling time interval at 0.02 sec, or to reset the Nyquist frequency at 25 Hz.

Let us check out what files there are beneath \demo\HDL0002. The folder is made up as follows:



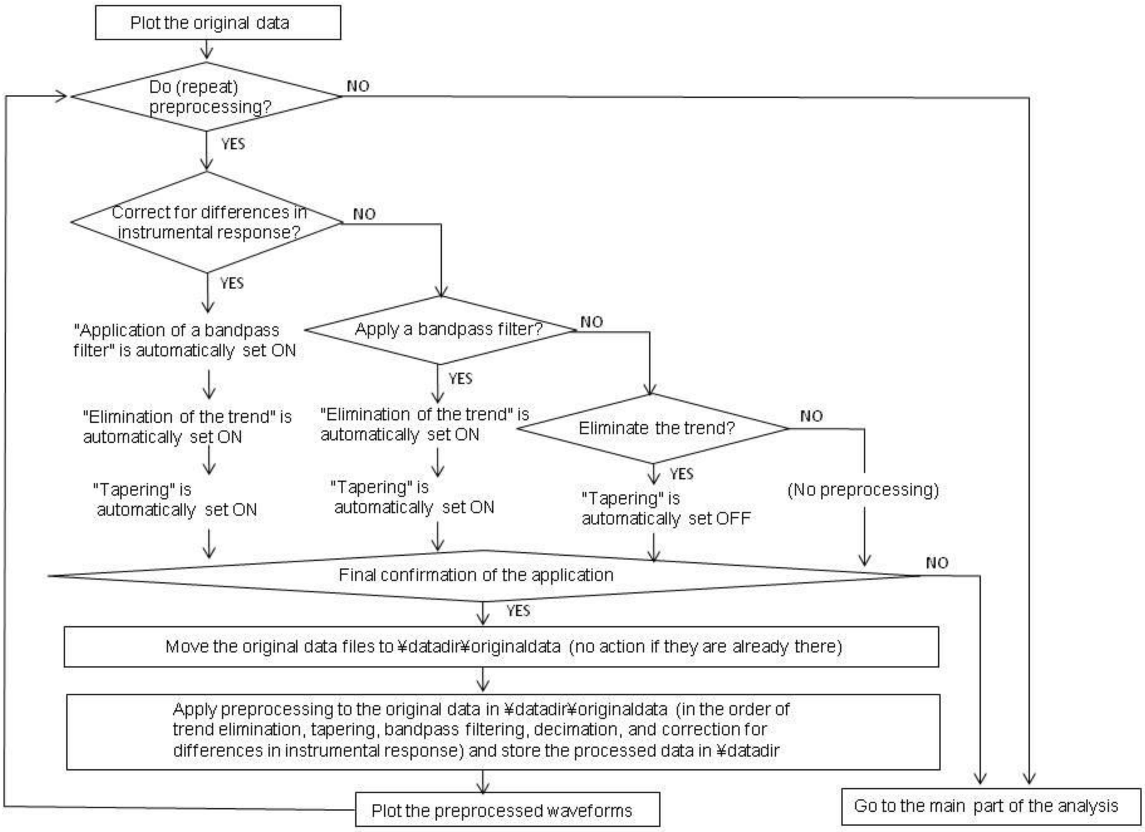
```
originaldata
RESULT
param.sh
run.log
S01.d
S02.d
S02.d.u.correction
S03.d
S03.d.u.correction
S04.d
S04.d.u.correction
S05.d
S05.d.u.correction
S06.d
S06.d.u.correction
seism.d
```

16 個の項目

You will see that a folder, named originaldata, has been generated as was so indicated in the dialogue message. Please note that all original data files have been moved there, whereas all data files that you find here, named S0X.d, are preprocessed data (the same thing is true for seism.d).

When the preprocessing is over, you will again see the message
Do you want to preprocess the waveforms? [y/n]
in addition to graphic output of the waveforms. Type "n", and you can proceed to the next stage, or the main part of the analysis. You can repeat preprocessing as many times as you like by typing "y". All repetitions that follow proceed along the line: i) Reading of the original data stored in the folder originaldata; ii) Preprocessing, and iii) Output to the data folder (the preprocessed waveform data files are overwritten). Therefore, the data will return to their original state if you answer "n" (not to apply) to all preprocessing options during the dialogue.

The flow of preprocessing by dialogue can be summarized as follows:


Plot the original data
Do (repeat) preprocessing?
NO
YES
Correct for differences in instrumental response?
NO
YES
"Application of a bandpass filter" is automatically set ON
Apply a bandpass filter?
NO
YES
"Elimination of the trend" is automatically set ON
"Elimination of the trend" is automatically set ON
Eliminate the trend?
NO
YES
"Tapering" is automatically set ON
"Tapering" is automatically set ON
"Tapering" is automatically set OFF
(No preprocessing)
Final confirmation of the application
NO
YES
Move the original data files to ¥datadir¥originaldata (no action if they are already there)
Apply preprocessing to the original data in ¥datadir¥originaldata (in the order of trend elimination, tapering, bandpass filtering, decimation, and correction for differences in instrumental response) and store the processed data in ¥datadir
Plot the preprocessed waveforms
Go to the main part of the analysis

Correction for differences in instrumental characteristics

In the analysis results for HDL0002, you may have noticed that the phase characteristics of channel no. 3 (data file S03.d) behave differently than those of the other channels, just like in the case of the analysis results for HDL0001 (see the analysis results in 6-3). It appears that channel no. 3 tends to respond differently than the other sensors. When, like in this case, the recording characteristics of a particular channel always demonstrate an identical bias (or peculiarity) irrespective of who installs it (or slight differences in the circumstances of installation), it appears sensible to correct for the difference before proceeding to the data analysis. BIDO is capable of implementing such corrections.

Create a file, in the folder where the data files are stored, which helps to correct for differences in the instrumental response characteristics, one for each seismic sensor component. The files should be named like:
(data file name).{e, n, u}.correction
where e, n and u correspond to the east-west, north-south and up-down components, respectively. The five files in the HDL0002 archive, named "*.u.correction," are the files for correction. Each file contains the following data strings:

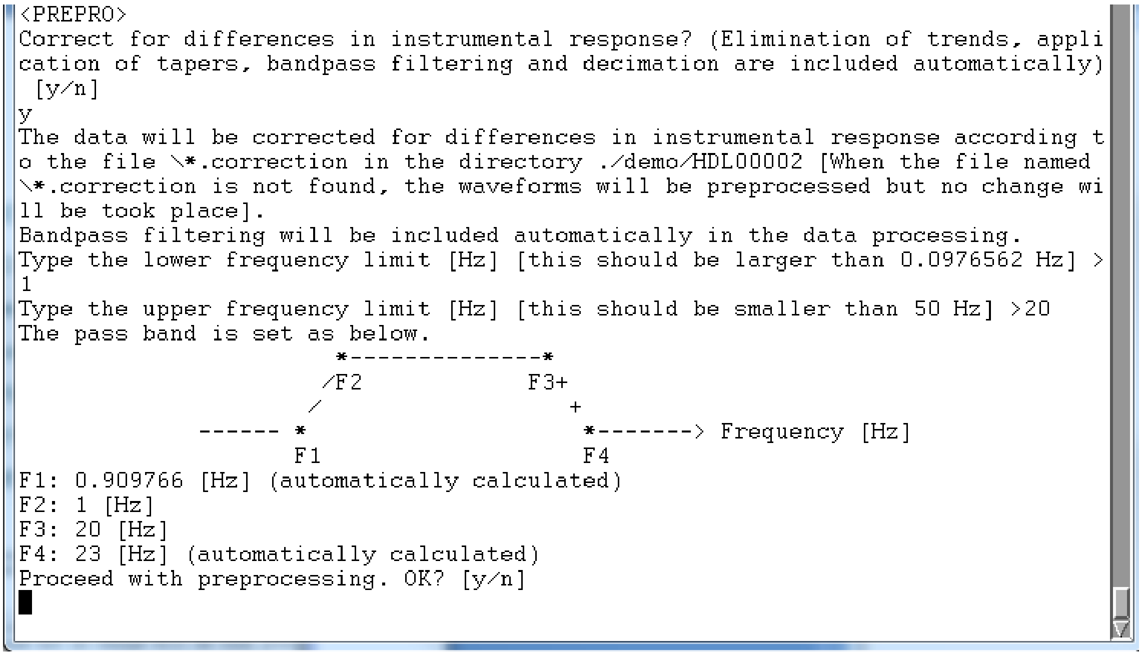
  Frequency F [Hz}   Amplitude ratio R [non-dimensional]   Phase difference P [deg]
If the FFT spectrum of the pre-correction data is given by A exp(iq), the post-correction spectrum will be (A/R) exp(i(q-P)). The frequency steps in the correction files can be anything (they are interpolated linearly). R=1 and P=0 are postulated when no correction file is found in the same folder even though the instrumental response characteristics correction option is set ON.

Correction files are already bundled together in the folder \demo\HDL0002. These correction files are the analysis results from HDL0001, or \RESULT\ave\DIFINSTRES1 Xu.d, copied and renamed \S0X.d.u.correction. Let us use these HDL0001 results to correct the HDL0002 data before proceeding to analysis.

Restart the analysis by typing

run.sh demo/HDL0002/param.sh [RETURN KEY].

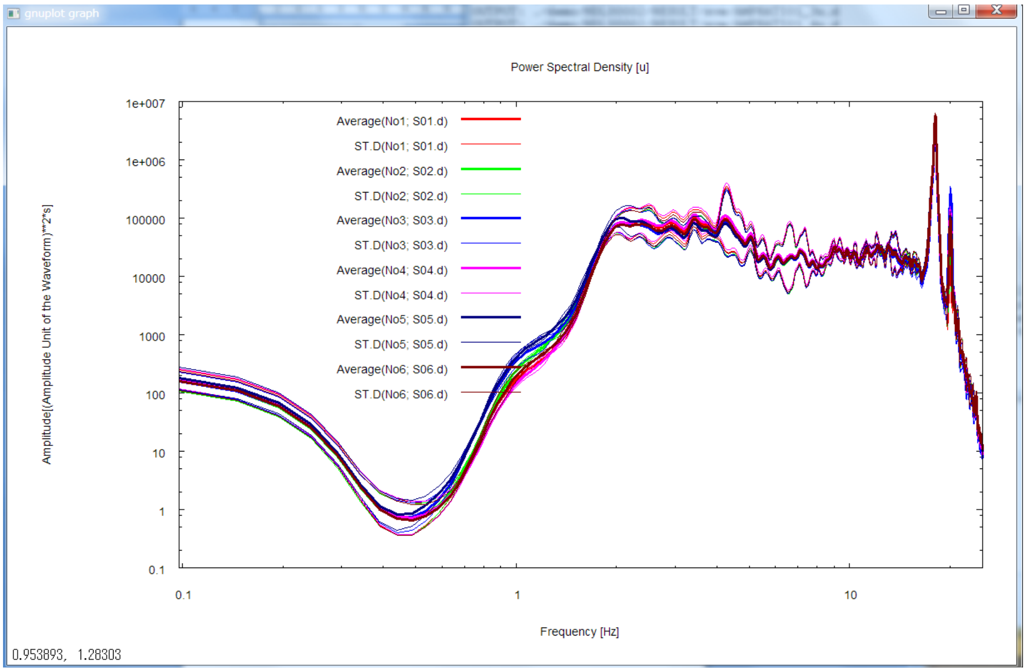
Continue with the dialogue, and answer "y" to "Do you want to preprocess the original waveforms anew?" and "y" to "Correct for the difference in instrumental response?" Preprocessing automatically involves bandpass filtering. Therefore, if you set the cut-off limits on the low- and high-frequency sides at 1 and 20 [Hz] respectively, you will see the following message asking for final confirmation:



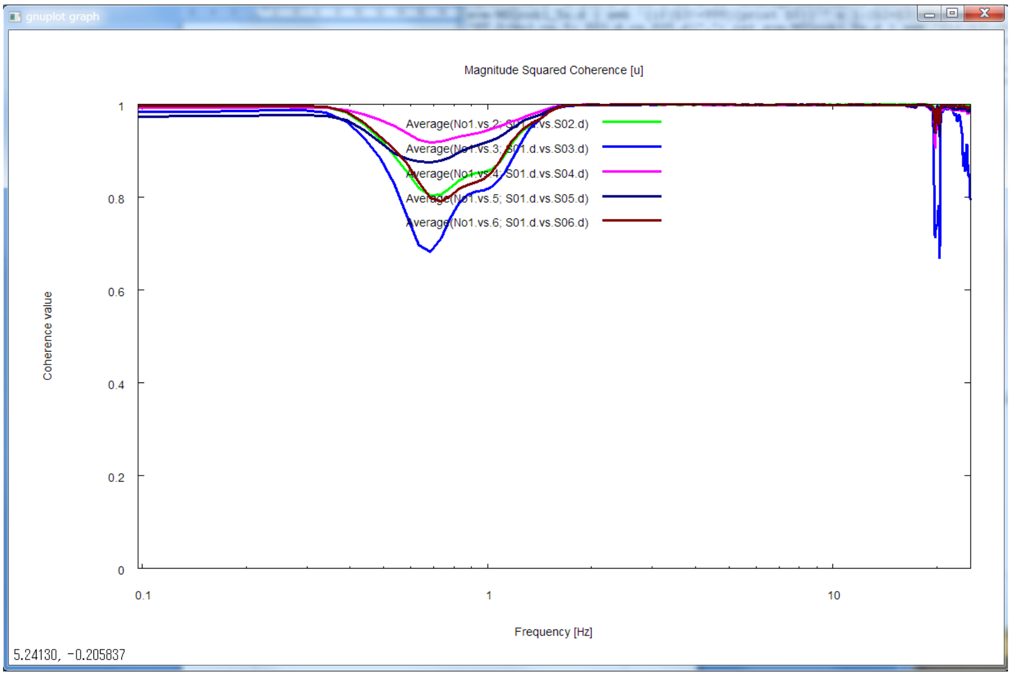
```
<PREPRO>
Correct for differences in instrumental response? (Elimination of trends, appli
cation of tapers, bandpass filtering and decimation are included automatically)
 [y/n]
y
The data will be corrected for differences in instrumental response according t
o the file \*.correction in the directory ./demo/HDL00002 [When the file named
\*.correction is not found, the waveforms will be preprocessed but no change wi
ll be took place].
Bandpass filtering will be included automatically in the data processing.
Type the lower frequency limit [Hz] [this should be larger than 0.0976562 Hz] >
1
Type the upper frequency limit [Hz] [this should be smaller than 50 Hz] >20
The pass band is set as below.
                       *---------------*
                      /F2            F3+
                     /                  +
             ------ *                    *-------> Frequency [Hz]
                    F1                   F4
F1: 0.909766 [Hz] (automatically calculated)
F2: 1 [Hz]
F3: 20 [Hz]
F4: 23 [Hz] (automatically calculated)
Proceed with preprocessing. OK? [y/n]
█
```

You can proceed with preprocessing by typing "y". After a plot of the waveforms, you will be asked again, "Do you want to preprocess the original waveforms anew?" The differences in instrumental response have already been corrected for, so type "n" to proceed to the main part of the analysis. You will get the following final analysis results:

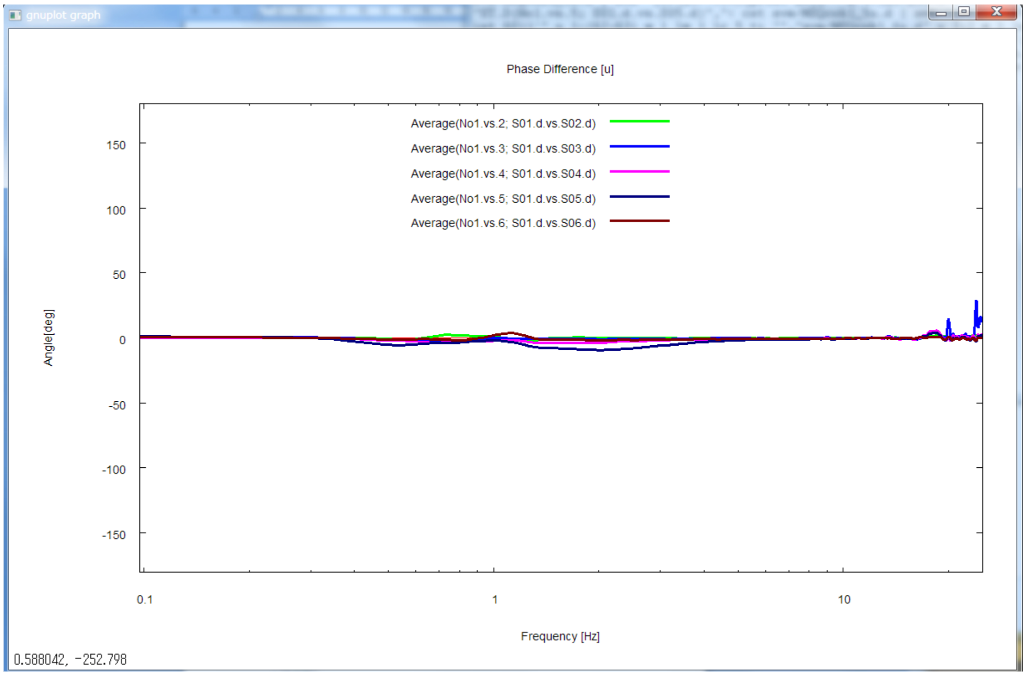
- Power-spectral densities



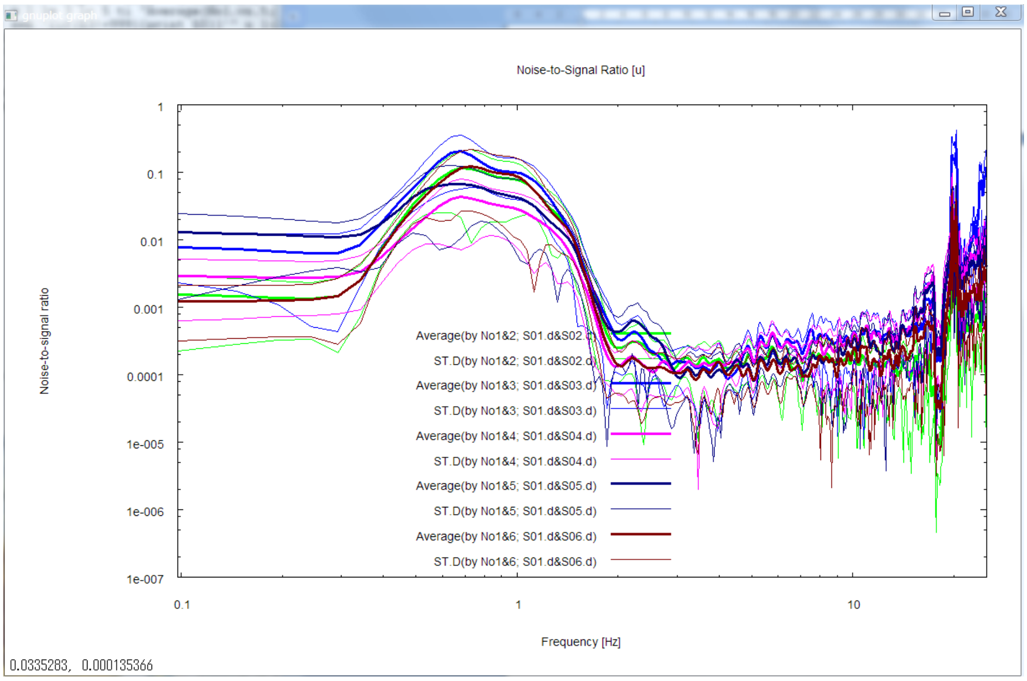
- Magnitude-squared coherences


gnuplot graph
Magnitude Squared Coherence [u]
Coherence value
1
0.8
0.6
0.4
0.2
0
0.1
1
10
Frequency [Hz]
Average(No1 vs.2; S01.d vs.S02.d)
Average(No1 vs.3; S01.d vs.S03.d)
Average(No1 vs.4; S01.d vs.S04.d)
Average(No1 vs.5; S01.d vs.S05.d)
Average(No1 vs.6; S01.d vs.S06.d)
5.24130, -0.205837

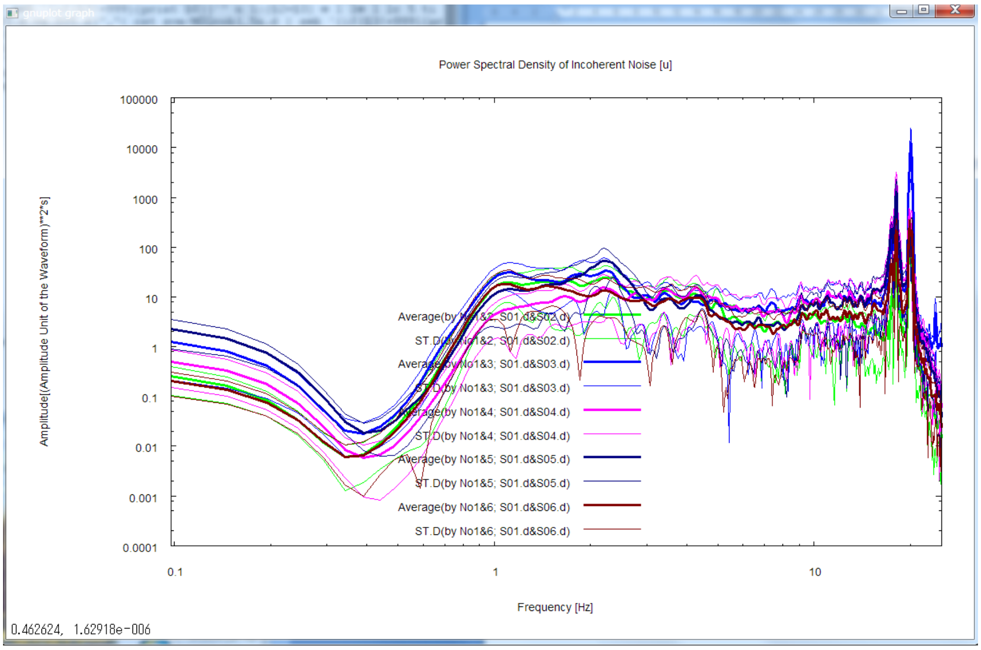
- Phase differences



- Noise-to-signal ratios



- Power-spectral densities of incoherent noise



You will notice the near-total disappearance of the peculiarity in phase characteristics of channel no. 3 (data file S03.d) thanks to the correction for differences in instrumental characteristics.

Saito, M., 1978, An automatic design algorithm for band selective recursive digital filters (in Japanese), Butsuri-Tanko (Geophysical Exploration), 31, 112-135.

6-3 Demo of a Huddle Test

A huddle test refers to a test where you huddle all seismic sensors at a single location and record data simultaneously for the purpose of confirming the agreement of the instrumental properties of all recording systems. In a huddle test, you can use exactly the same analysis method that is described in the demo of array data analysis (6-1), so please read that page first if you are using BIDO for the first time. You can conduct analysis following exactly the same procedure as in array analysis if only you assign identical locations to all seismic sensors in the seismfile.

For example, please download the demo data meant for huddle data analysis, decompress it beneath the BIDO 2.0 folder, and analyze it following the same procedure that you would use to analyze array data. The demo data have been made available by courtesy of Dr Tatsuya Noguchi at Tottori University. They were obtained by six vertical motion sensors (HS-1 Geophones) of Oyo Geospace Corporation installed on a concrete laboratory floor on the premises of Tottori University (see photos), and were recorded by an es8 data recorder via SA16 amplifiers and a low-pass filter (cutoff frequency 30 Hz).

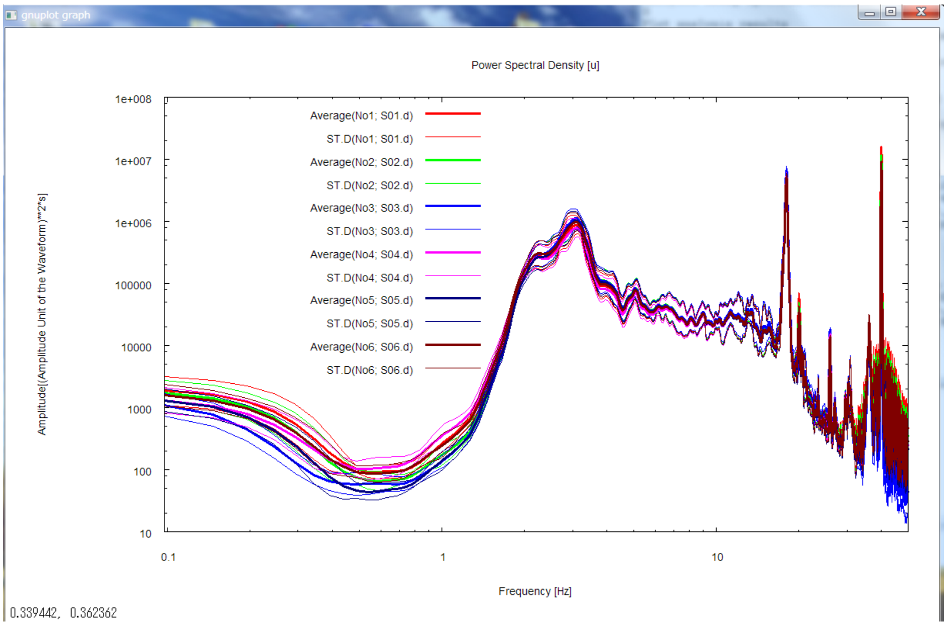
Let us move on to data analysis by typing

```
run.sh demo/HDL0001/param.sh [RETURN KEY].
```

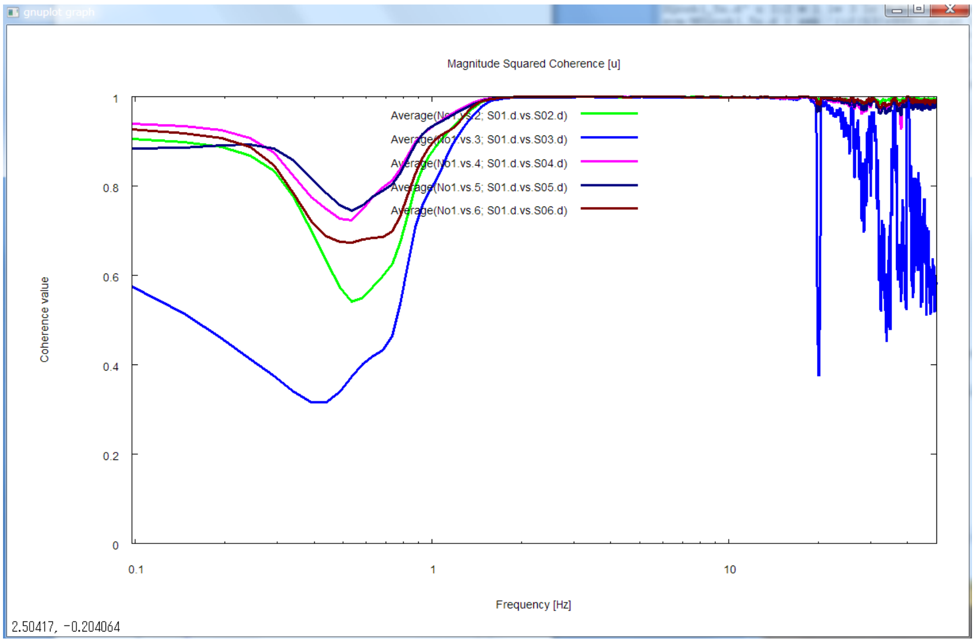
Just like in array analysis, the analysis results are stored in a folder, named RESULT, that is generated beneath the data folder.

The graphic output of the huddle test includes:

- Power-spectral densities

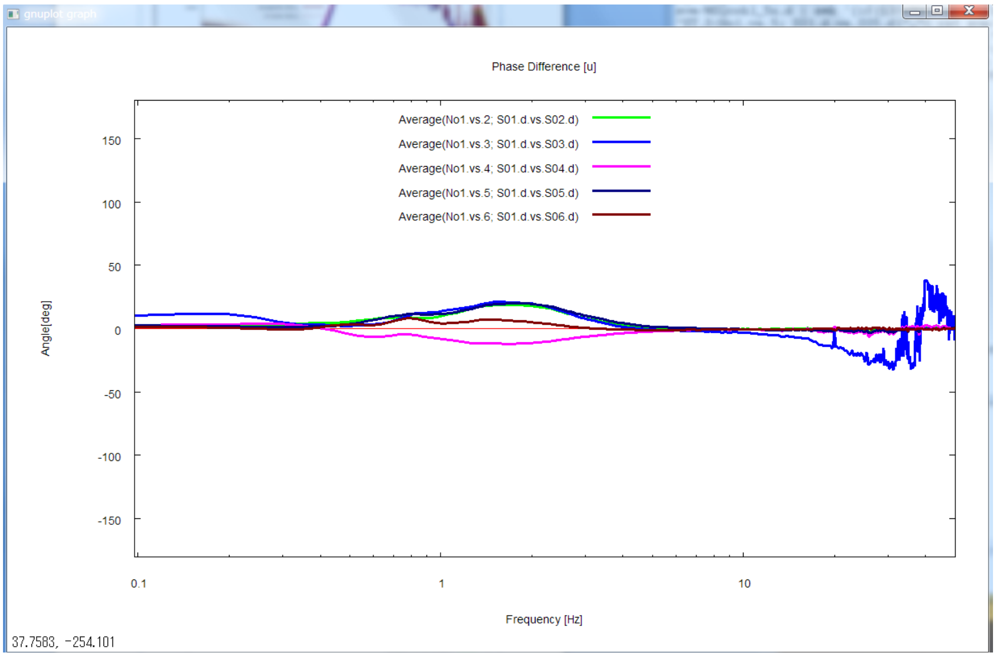


- Magnitude-squared coherences



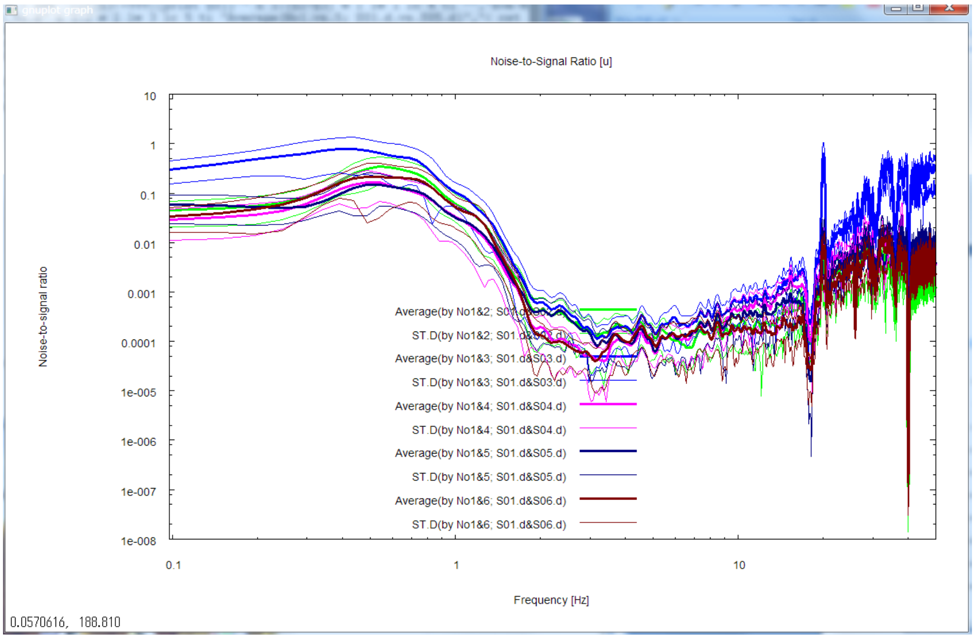
2.50417, -0.204064

- Phase differences



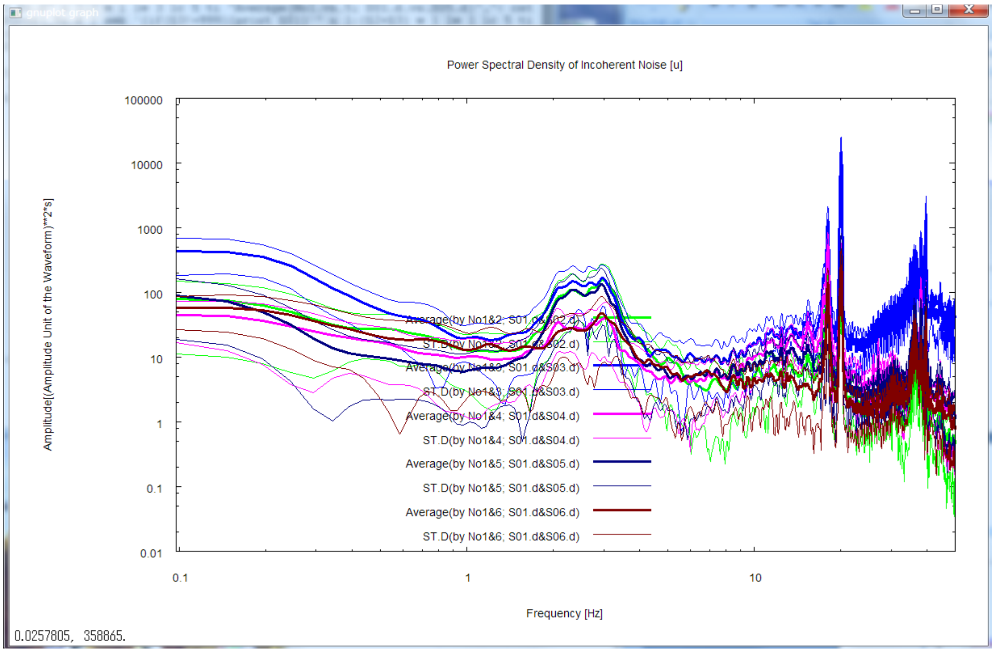
37.7583, -254.101

- Noise-to-signal ratios



and

- Power-spectral densities of incoherent noise.



All graphic output is shown up to the maximum frequency with the frequency axis scaled logarithmically. The maximum frequency here refers to the smaller one between the value set by the parameter freqmax_ave in \script/\setpar.sh (50 Hz by default but modifiable on your own) and the Nyquist frequency. If you do not prefer logarithmic scaling, comment out the line containing the parameter autologscale_x in \script/\setpar.sh by appending # to the head of the line.

The legends for the magnitude-squared coherences and phase differences are denoted like Average (by No 1 .vs. 2: S01.d .vs. S02.d). This refers to the mean of coherences between record numbers 1 and 2, of their phase differences (positive when 2 is more advanced than 1 in phase), and of NS ratios and noise intensities calculated on their basis. Record number 1 refers to the data indicated at the top of the seismfile. The records are numbered 2, 3 and so forth in the descending order of indication in the seismfile from top to bottom. The letters "by No 1 .vs. 2" in the graph legends indicate that record numbers 1 and 2 are concerned. To make this point sure, these numbers are followed by data file names like "S01.d .vs. S02.d." Please refer to ave.info in the ave folder for a table of correspondence between the record numbers and file names. ST. D. means standard deviation. The above analysis results show that record number 3 (records of S03.d; blue) has distinctly low coherences and has large phase differences with respect to the other records.

Bendat and Piersol (1971) and Carter et al. (1973) are useful references for the estimation and physical meaning of magnitude-squared coherences. The NS ratios are the inverse of the SN ratios calculated by substituting the magnitude-squared coherences (coh^2) into the equation

$$\mathrm{S/N} = (\text{signal intensity})/(\text{noise intensity}) = \sqrt{\mathrm{coh}^2} \Big/ \left(1 - \sqrt{\mathrm{coh}^2}\right)$$

(Carter et al., 1973). The power-spectral densities of noise are calculated by multiplying the power-spectral densities of the records by the NS ratios.

These plot data are stored, as in the case of array analysis, in a folder named RESULT/ave. Please refer to a list (A6-3-1) of the file names and descriptions of the plot data. There is a file named DIFINSTRES1_2e.d. This file name is short for Differences in instrumental response. It lays out the amplitude ratios and phase differences of record number 2 with respect to record number 1 in the format

Frequency F [Hz]   Amplitude ratio R [non-dimensional]   Phase difference P [deg]

for each frequency. This file, when renamed, can be used directly for the purpose of correcting for instrumental characteristics in array data analysis (see Demo of data preprocessing, 6-2).

Bendat, J. S., and A. G. Piersol, Random Data: Analysis and Measurement Procedures: John Wiley & Sons, 1971.

Carter, G. C., C. H. Knapp, and A. H. Nuttall, 1973, Estimation of the magnitude-squared coherence function via overlapped Fast Fourier

Transform processing: IEEE Transactions on Audio Electroacoustics, **AU-21**, 337–344.

A6-3-1 Analysis Results (folder RESULT/ave)

ave.info Gives rough explanation of how the analysis results were averaged, and describes the correspondence between the numbers in the names of files stored in this folder (e.g. "1" and "2" in MSQcoh1_2e.d) and the data files.

MSQcoh1_2e.dMagnitude-squared coherences. 1_2 means the coherences between sensors 1 and 2. The letter "e" stands for the east-west component.

MSQcoh1_2n.dMagnitude-squared coherences. 1_2 means the coherences between sensors 1 and 2. The letter "n" stands for the north-south component.

MSQcoh1_2u.dMagnitude-squared coherences. 1_2 means the coherences between sensors 1 and 2. The letter "u" stands for the up-down component.

NSR1_2e.d Noise-to-signal ratios. 1_2 means that the values were calculated using the coherences between sensors 1 and 2. The letter "e" stands for the east-west component. (logarithmic mean)

NSR1_2n.d Noise-to-signal ratios. 1_2 means that the values were calculated using the coherences between sensors 1 and 2. The letter "n" stands for the north-south component. (logarithmic mean)

NSR1_2u.d Noise-to-signal ratios. 1_2 means that the values were calculated using the coherences between sensors 1 and 2. The letter "u" stands for the up-down component. (logarithmic mean)

PHASE1_2e.d Phase differences. 1_2 means that the values were calculated using the cross-spectral densities between sensors 1 and 2. The letter "e" stands for the east-west component.

PHASE1_2n.d Phase differences. 1_2 means that the values were calculated using the cross-spectral densities between sensors 1 and 2. The letter "n" stands for the north-south component.

PHASE1_2u.d Phase differences. 1_2 means that the values were calculated using the cross-spectral densities between sensors 1 and 2. The letter "u" stands for the up-down component.

PSD_NOISE1_2e.d Power-spectral densities of incoherent noise (electrical noise). 1_2 means that the values were calculated using the coherences between sensors 1 and 2. The letter "e" stands for the east-west component. (logarithmic mean)

PSD_NOISE1_2n.d Power-spectral densities of incoherent noise (electrical noise). 1_2 means that the values were calculated using the coherences between sensors 1 and 2. The letter "n" stands for the north-south component. (logarithmic mean)

PSD_NOISE1_2u.d Power-spectral densities of incoherent noise (electrical noise). 1_2 means that the values were calculated using the coherences between sensors 1 and 2. The letter "u" stands for the up-down component. (logarithmic mean)

SD1e_SD1e.d Power-spectral densities of waveform records. 1 stands for the sensor number. The letter "e" stands for the east-west component. (logarithmic mean)
AMPRATIO1_2e.d Ratios of the power-spectral densities of the records of sensors 1 and 2. The letter "e" stands for the east-west component. (logarithmic mean)
AMPRATIO1_2n.d Ratios of the power-spectral densities of the records of sensors 1 and 2. The letter "n" stands for the north-south component. (logarithmic mean)
AMPRATIO1_2u.d Ratios of the power-spectral densities of the records of sensors 1 and 2. The letter "u" stands for the up-down component. (logarithmic mean)
DIFINSTRES1_2e.d The file name is short for Difference in instrumental response. The data from PHASE1_2e.d and AMPRATIO1_2e.d are laid out in two columns. Can be utilized to correct for differences in instrumental response characteristics. The letter "e" stands for the east-west component.
DIFINSTRES1_2n.d The file name is short for Difference in instrumental response. The data from PHASE1_2n.d and AMPRATIO1_2n.d are laid out in two columns. Can be utilized to correct for differences in instrumental response characteristics. The letter "n" stands for the north-south component.
DIFINSTRES1_2u.d The file name is short for Difference in instrumental response. The data from PHASE1_2u.d and AMPRATIO1_2u.d are laid out in two columns. Can be utilized to correct for differences in instrumental response characteristics. The letter "u" stands for the up-down component.

** In the above data files, the frequency, the mean of segment-by-segment analysis results and the standard deviation are laid out in the first, second and third columns, respectively. For data with a note "(logarithmic mean)" in parentheses, the frequency, the mean, the mean minus a standard deviation and the mean plus a standard deviation are laid out in the first, second, third and fourth columns, respectively.*

** A phase difference is denoted in terms of the angle formed between the x-axis and a vector whose x- and y-components are given by the real part and the imaginary part, respectively, of the arithmetic mean of the cross-spectral densities between records.*

7. Warnings / Download

<Warnings>
- This program is distributed free with source codes.
- Do not give this program to a third party (a third party should download it from this Web site).
- We assume no responsibility for any problems arising from the use of this program.

- Consult with us in advance if you wish to use this program for commercial purposes.
- Be sure to make appropriate citations when you publish research achievements benefiting from this program.

If you agree to the above provisions, register your e-mail address (to be used for possible information of bugs and for statistical purposes only) and click OK to proceed to the download screen.

<Download>
This archive is common to both Windows and Linux.
Latest version (2010.11)

zip format

BIDO2.0.zip Source files + executable files + demo data (21 MBytes)

demo_obsz.zip Sample data for vertical motion (worth two arrays at two sites—AIST and KSKB(reference [3,6])) (42 MBytes)

demo_obs3c.zip Sample data for three components (worth six arrays at two sites—KSKB and KIBA(reference [5])) (74 MBytes)

tar+gzip format

BIDO2.0.tgz Source files + executable files + demo data (21 MBytes)

demo_obsz.tgz Sample data for vertical motion (worth two arrays at two sites—AIST and KSKB(reference [3,6])) (42 MBytes)

demo_obs3c.tgz Sample data for three components (worth six arrays at two sites—KSKB and KIBA(reference [5])) (74 MBytes)

Depending on the setting environment, it sometimes happens that the downloaded archive file does not have the extension .tgz on your computer (e.g. the extension can be .tar). In such a case, change the extension to .tgz manually so that you can decompress it normally.

8. Citation

When you publish analysis results obtained by using the BIDO software, we request that you cite reference [7] and state in your publication that you downloaded the software from this Web site (http://staff.aist.go.jp/ikuo-chou). Cite the following papers with respect to specific aspects:

[2] for the general theory;
[1, 3] for the CCA method;
[4] for the H0, H1, V and nc-CCA methods;

[5] for the CCA-L (R), SPAC-L (R) and SPAC+L (R) methods and for Rayleigh-to-Love power ratios;
[6] for miniature arrays;
[3] for NS ratios of vertical motion;
[8] for NS ratios of horizontal motion;
[2] for horizontal-to-vertical amplitude ratios of Rayleigh waves; and
[5,7] for Rayleigh-to-Love power ratios.

[1] Cho, I., T. Tada, and Y. Shinozaki, 2004, A new method to determine phase velocities of Rayleigh waves from microseisms, *Geophysics*, **69**, 1535-1551.

[2] Cho, I., T. Tada, and Y. Shinozaki, 2006, A generic formulation for microtremor exploration methods using three-component records from a circular array, *Geophysical Journal International,* **165**, 236-258*.*

[3] Cho, I., T. Tada, and Y. Shinozaki, 2006, Centerless circular array method: Inferring phase velocities of Rayleigh waves in broad wavelength ranges using microtremor records, *Journal of Geophysical Research*, **111**, B09315, doi:10.1029/2005JB004235.

[4] Tada, T., I. Cho, and Y. Shinozaki, 2007, Beyond the SPAC method: Exploiting the wealth of circular-array methods for microtremor exploration, *Bulletin of the Seismological Society of America*, **97**, 2080-2095, doi:10.1785/0120070058.

[5] Tada, T., I. Cho, and Y. Shinozaki, 2009, New circular-array microtremor techniques to infer Love-wave phase velocities, *Bulletin of the Seismological Society of America*, **99**, 2912-2926, doi:10.1785/0120090014.

[6] Cho, I., T. Tada, and Y. Shinozaki, 2008, A new method of microtremor exploration using miniature seismic arrays: Quick estimation of average shear velocities of the shallow soil (in Japanese with English abstract), *Butsuri-Tansa*, **61**, 457-468.

[7] Tada, T., I. Cho, and Y. Shinozaki, 2010, Analysis of Love-wave components in microtremors, *Joint Conference Proceedings*, *7th International Conference on Urban Earthquake Engineering* (*7CUEE*) & *5th International Conference on Earthquake Engineering* (*5ICEE*), Center for Urban Earthquake Engineering, Tokyo Institute of Technology, 115-124. (http://www.cuee.titech.ac.jp/Japanese/Publications/Doc/conference_7th.pdf)

[8] Tada, T., 2010, Calculating noise-to-signal ratios in the horizontal-motion components of circular-array microtremor data, *unpublished manuscript***.**
http://staff.aist.go.jp/ikuo-chou/BIDO/2.0/bidodl_SN_horizontal.pdf

[9] Cho, I., T. Tada, and Y. Shinozaki, 2008, Assessing the applicability of the spatial autocorrelation method: A theoretical approach, J. Geophys. Res., **113**, B06307, doi:10.1029/2007JB005245.

Appendix 1 Array Geometries and the Analyzable Quantities

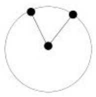
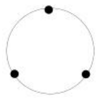
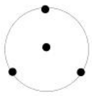
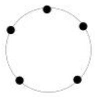
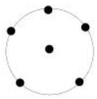
What Array Geometries Is BIDO Good for?

BIDO's array analysis theory is based on the assumption of a circular array installed on the ground surface (with sensors at all locations around a circle). In practice, however, problems arise as to how many sensors should be installed around the circle and whether or not a sensor should be installed at the center. To make a shortcut to the conclusion, BIDO can be applied to arrays of 0, 1, 2, 3 or 5 (or odd numbers of) sensors around a circle. An array can consist of a maximum number of 6 sensors if one of them is installed at the center. The range of analyzable quantities differ for different array geometries and record components.

The following describes the correspondence relations between the array geometries, record components and the analyzable quantities. The table lists the names of the quantities that can be analyzed and of the analysis methods that can be implemented.

As shown in the above table, BIDO allows one not only to analyze array data but also data from single-station measurements, linear arrays of two stations and L-shaped arrays of three stations. (It can also cope with huddle tests where more than one seismic sensors are assigned to a single location). What we had in mind during the early phase of development was the analysis of circular-array data, but we later came to realize that, at real measurement sites (especially in urban areas), simpler array geometries are preferred because of their ease of installation. It is coming to be known that linear arrays and L-shaped arrays can be applied to the analysis of vertical motion in a number of cases, and so we decided to incorporate these into BIDO. The third and fourth columns on the left-hand side of the table both refer to arrays of three seismic sensors, but the range of applicable methods differ according to whether one of the sensors is regarded as assigned to the center of the circular array or whether all of them are regarded as lying around the periphery of the circle.

It goes without saying, though, that larger numbers of sensors around the circle produce higher precision and more stable analysis results. We recommend the installation of six-sensor arrays, as shown in the right-hand column of the table, when small arrays (10 m or less in radius) are to be used. Particularly, when using miniature arrays of around 1 m in radius (see photos in the top page), six-station arrays are expected to an effective design not only in terms of high precision but also in terms of efficiency (see A6-1-8).

Array geometry (Black circles: sensors)								
Total number of sensors		1	2	3	3	4	5	6
At the center		1	1	1	0	1	0	1
Around the circle		0	1	2	3	3	5	5
Vertical motion alone	Phase velocities of Rayleigh waves	None	SPAC	SPAC H0	CCA	SPAC CCA nc-CCA H0, H1, V	CCA	SPAC CCA nc-CCA H0, H1, V
	Phase velocities of Love waves	None	None	None	None	None	None	None
	Other quantities	PSD	PSD	PSD	PSD	PSD NSR(V)	PSD	PSD NSR(V)
Horizontal motion alone	Phase velocities of Rayleigh waves	None	None	None	CCA-R	CCA-R SPAC-R SPAC+R	CCA-R	CCA-R SPAC-R SPAC+R
	Phase velocities of Love waves	None	None	None	CCA-L	CCA-L SPAC-L SPAC+L	CCA-L	CCA-L SPAC-L SPAC+L
	Other quantities	PSD	PSD	PSD	PSD	PSD NSR(H)	PSD	PSD NSR(H)
Three-component	Phase velocities of Rayleigh waves	None	SPAC	SPAC H0	CCA CCA-R	SPAC CCA nc-CCA H0, H1, V CCA-R SPAC-R SPAC+R	CCA CCA-R	SPAC CCA nc-CCA H0, H1, V CCA-R SPAC-R SPAC+R
	Phase velocities of Love waves	None	None	None	CCA-L	CCA-L SPAC-L SPAC+L	CCA-L	CCA-L SPAC-L SPAC+L
	Other quantities	PSD H/V	PSD H/V	PSD H/V	PSD H/V	PSD H/V R/V, R/L NSR(V) NSR(H)	PSD H/V	PSD H/V R/V, R/L NSR(V) NSR(H)

(Notes)

- SPAC: Spatial Autocorrelation method・(its more generalized variant is applicable to horizontal motion, but here, the term is used to denote "the method to analyze phase velocities of Rayleigh waves using vertical motion)
- CCA: A method to analyze phase velocities of Rayleigh waves using vertical motion (reference [1, 3])
- nc-CCA, H0, H1, V: Methods to analyze phase velocities of Rayleigh waves using vertical motion (reference [4])
- CCA-R, SPAC-R, SPAC+R: Methods to analyze phase velocities of Rayleigh waves using horizontal motion (reference [5])
- CCA-L, SPAC-L, SPAC+L: Methods to analyze phase velocities of Love waves using horizontal motion (reference [5])
- PSD: Power-spectral densities
- H/V: H/V spectra (Power ratios of horizontal motion to vertical motion)
- NSR(V): Noise-to-signal ratios of the vertical component (reference [3])
- NSR(H): Noise-to-signal ratios of the horizontal components (reference [8])
- R/V: Horizontal-to-vertical amplitude ratios (NOT power ratios) of the Rayleigh waves (reference [2])
- R/L: Rayleigh-to-Love power ratios (reference [5, 7])

Notes on the Array Analysis of Horizontal Motion

Array analysis of horizontal motion is currently still on the level of basic research. It has been shown that Love wave velocities and other properties can be analyzed using an array of five sensors around a circle and another at its center (references [5, 7]), but the extent of reliability of the analysis results from arrays with only three sensors around a circle remains yet to be studied. BIDO is designed in such a way that the analysis of horizontal motion is possible even when the array has only three sensors around the circle, but this feature is intended principally for the developer, and considerable caution is required in interpreting the analysis output. Caution is also necessary when S/N ratios are low (noise is large), because this considerably impairs the reliability of analysis results even when five sensors are installed around the circle. Unfortunately, there is, so far, no established theoretical rationale for assessing the reliability of analysis results for horizontal motion on the basis of S/N ratio estimates.

On Deviations from a "Circular" Array

When using circular arrays with three or more sensors around a circle, the stations do not necessarily have to be installed at equal intervals around the circle (reference [2]), but equidistant arrays are preferred to ensure high precision. There is no theory of correction for stations that do not lie on the circle (location errors in the radial direction), so the program sets a default allowance limit of 5% for such errors (modifiable by program recompilation) and regard the stations as lying on the circle.

Appendix 2 Execution log

```
OS windows
paramfile:./param.sh
segment_duration=10.24
smoothband=0.3
nseg_segave=10
idprepro=0
idmksegment=1
idcalc=1
idgnplt=1
delete_level=0
<SETPAR>
------------------------------------------------
Array geometry: centered 6-station array.
Array radius:  18. [m]
Number of sensors at the center: 1
Number of sensors around the circle: 5
Data: all three components
------------------------------------------------
Power spectral densities will be estimated.
horizontal-to-vertical spectral ratios will be estimated.
Rayleigh wave phase velocities will be estimated (by SPAC).
Rayleigh wave phase velocities will be estimated (by CCA).
Rayleigh wave phase velocities will be estimated (by ncCCA).
Rayleigh wave phase velocities will be estimated (by H0).
Rayleigh wave phase velocities will be estimated (by H1).
Rayleigh wave phase velocities will be estimated (by V).
Noise-to-signal ratios in the vertical component will be estimated.
Rayleigh wave phase velocities will be estimated (by CCA+R).
Rayleigh wave phase velocities will be estimated (by SPAC+R).
Rayleigh wave phase velocities will be estimated (by SPAC-R).
Love wave phase velocities will be estimated (by CCA+L).
Love wave phase velocities will be estimated (by SPAC+R).
Love wave phase velocities will be estimated (by SPAC-R).
H/V power partition ratios of the Rayleigh waves will be estimated.
Rayleigh-to-Love power partition ratios will be estimated.
Noise-to-signal ratios in the horizontal components will be estimated.

Do you really want to start running the program? [y/n]
<MKSEGMENT>
Output demo/synth_SN100_18mGamR0.8RV0.1N6/RESULT/segment_3c.d
[Segments with RMS values between 1.000000 and 1.200000 are selected]
NOTE: 57 segments have been selected.

<mkgnplt_wave.sh>
NOTE: This script uses a program gnuplot to plot the waveforms. You can manually plot the same figures later, using a gnuplot
macro, plot_wave.gnplt, which was created by this script and is found in demo/synth_SN100_18mGamR0.8RV0.1N6/RESULT. To
do this, type cd demo/synth_SN100_18mGamR0.8RV0.1N6/RESULT & gnuplot plot_wave.gnplt
OUTPUT: demo/synth_SN100_18mGamR0.8RV0.1N6/RESULT/plot_wave.gnplt
/
<MKCIRCLE>
seismfile: demo/synth_SN100_18mGamR0.8RV0.1N6/seism.d
OUTPUT: demo/synth_SN100_18mGamR0.8RV0.1N6/RESULT/Z0r.d
OUTPUT: demo/synth_SN100_18mGamR0.8RV0.1N6/RESULT/Z1r.d
OUTPUT: demo/synth_SN100_18mGamR0.8RV0.1N6/RESULT/Z00.d
<MKCIRCLE_H>
seismfile: demo/synth_SN100_18mGamR0.8RV0.1N6/seism.d
OUTPUT: demo/synth_SN100_18mGamR0.8RV0.1N6/RESULT/R0r.d
OUTPUT: demo/synth_SN100_18mGamR0.8RV0.1N6/RESULT/R1r.d
OUTPUT: demo/synth_SN100_18mGamR0.8RV0.1N6/RESULT/T0r.d
OUTPUT: demo/synth_SN100_18mGamR0.8RV0.1N6/RESULT/T1r.d
OUTPUT: demo/synth_SN100_18mGamR0.8RV0.1N6/RESULT/R10.d
OUTPUT: demo/synth_SN100_18mGamR0.8RV0.1N6/RESULT/T10.d
<MKWAV3C>
seismfile: demo/synth_SN100_18mGamR0.8RV0.1N6/seism.d
demo/synth_SN100_18mGamR0.8RV0.1N6/S01.d used to estimate horizontal-to-vertical spectral ratios
OUTPUT: demo/synth_SN100_18mGamR0.8RV0.1N6/RESULT/wavud.d
OUTPUT: demo/synth_SN100_18mGamR0.8RV0.1N6/RESULT/wavew.d
OUTPUT: demo/synth_SN100_18mGamR0.8RV0.1N6/RESULT/wavns.d

* Preexisting directori(es) detected (created in a previous calculation?):
demo/synth_SN100_18mGamR0.8RV0.1N6/RESULT/1 demo/synth_SN100_18mGamR0.8RV0.1N6/RESULT/ave
NOTE: These directories are removed to proceed with the current analysis.

<CPWAV>
```

```
Copying original waveforms to a temporary dierectory demo/synth_SN100_18mGamR0.8RV0.1N6/RESULT/cpwav to estimate the
power spectral densities.

### Step 1 (Total 5) ###
<SETPAR>
-----------------------------------------------
Array geometry: centered 6-station array.
Array radius:  18. [m]
Number of sensors at the center: 1
Number of sensors around the circle: 5
Data: all three components
-----------------------------------------------
Power spectral densities will be estimated.
horizontal-to-vertical spectral ratios will be estimated.
Rayleigh wave phase velocities will be estimated (by SPAC).
Rayleigh wave phase velocities will be estimated (by CCA).
Rayleigh wave phase velocities will be estimated (by ncCCA).
Rayleigh wave phase velocities will be estimated (by H0).
Rayleigh wave phase velocities will be estimated (by H1).
Rayleigh wave phase velocities will be estimated (by V).
Noise-to-signal ratios in the vertical component will be estimated.
Rayleigh wave phase velocities will be estimated (by CCA+R).
Rayleigh wave phase velocities will be estimated (by SPAC+R).
Rayleigh wave phase velocities will be estimated (by SPAC-R).
Love wave phase velocities will be estimated (by CCA+L).
Love wave phase velocities will be estimated (by SPAC+R).
Love wave phase velocities will be estimated (by SPAC-R).
H/V power partition ratios of the Rayleigh waves will be estimated.
Rayleigh-to-Love power partition ratios will be estimated.
Noise-to-signal ratios in the horizontal components will be estimated.
<ESTSPEC>
NOICE:idrimag=1, read columns 1-3
[column 1: time; 2: real part; 3: imaginary part]
OUTPUT DIR: demo/synth_SN100_18mGamR0.8RV0.1N6/RESULT/1
 No. of Seismograms: 9
DATAFILE  1) demo/synth_SN100_18mGamR0.8RV0.1N6/RESULT/R10.d
DATAFILE  2) demo/synth_SN100_18mGamR0.8RV0.1N6/RESULT/R0r.d
DATAFILE  3) demo/synth_SN100_18mGamR0.8RV0.1N6/RESULT/R1r.d
DATAFILE  4) demo/synth_SN100_18mGamR0.8RV0.1N6/RESULT/T10.d
DATAFILE  5) demo/synth_SN100_18mGamR0.8RV0.1N6/RESULT/T0r.d
DATAFILE  6) demo/synth_SN100_18mGamR0.8RV0.1N6/RESULT/T1r.d
DATAFILE  7) demo/synth_SN100_18mGamR0.8RV0.1N6/RESULT/Z00.d
DATAFILE  8) demo/synth_SN100_18mGamR0.8RV0.1N6/RESULT/Z0r.d
DATAFILE  9) demo/synth_SN100_18mGamR0.8RV0.1N6/RESULT/Z1r.d

Cross-spectral density calculations (1:yes; 2: no)
   |  1  2  3  4  5  6  7  8  9
------------------------------
  1|  1  1  1  1  1  0  0  0  0
  2|  1  1  1  0  0  0  0  0  0
  3|  0  1  1  0  1  1  0  0  0
  4|  0  1  0  0  1  1  0  0  0
  5|  0  1  0  1  1  1  0  0  0
  6|  0  1  0  0  1  1  0  0  0
  7|  0  0  0  0  0  0  1  1  0
  8|  0  0  0  0  0  0  1  1  0
  9|  0  0  1  0  0  1  0  0  1

No. of segments:    10
Data segment duration: 10.240 s
dt          :  0.010 s
Total duration (zero-padded):   20.480s (    2048 pts)
Ratio of data length to total length (zero-padded):  0.500
(Power reduction is corrected based on this value)
Frequency interval:0.488E-01 Hz
Nyquist frequency : 50.0     Hz
Use data window  [Hanning window with taper rate: 0.500 (  513 pts)]
Ratio of tapered power to raw power:  0.50
(Power reduction is corrected based on this value)
Use spectral window [Parzen window with band width:  0.30000 Hz (Total   13 pts)]
Resulting degree of freedom:    65
Standard error:     0.176777 (the effects of cos taper & zero-padding considered)
OUTPUT (CSD): demo/synth_SN100_18mGamR0.8RV0.1N6/RESULT/1/R10_R10.d
OUTPUT (CSD): demo/synth_SN100_18mGamR0.8RV0.1N6/RESULT/1/R10_R0r.d
OUTPUT (CSD): demo/synth_SN100_18mGamR0.8RV0.1N6/RESULT/1/R10_R1r.d
OUTPUT (CSD): demo/synth_SN100_18mGamR0.8RV0.1N6/RESULT/1/R10_T10.d
OUTPUT (CSD): demo/synth_SN100_18mGamR0.8RV0.1N6/RESULT/1/R10_T0r.d
OUTPUT (CSD): demo/synth_SN100_18mGamR0.8RV0.1N6/RESULT/1/R0r_R10.d
OUTPUT (CSD): demo/synth_SN100_18mGamR0.8RV0.1N6/RESULT/1/R0r_R0r.d
OUTPUT (CSD): demo/synth_SN100_18mGamR0.8RV0.1N6/RESULT/1/R0r_R1r.d
OUTPUT (CSD): demo/synth_SN100_18mGamR0.8RV0.1N6/RESULT/1/R1r_R0r.d
OUTPUT (CSD): demo/synth_SN100_18mGamR0.8RV0.1N6/RESULT/1/R1r_R1r.d
OUTPUT (CSD): demo/synth_SN100_18mGamR0.8RV0.1N6/RESULT/1/R1r_T0r.d
OUTPUT (CSD): demo/synth_SN100_18mGamR0.8RV0.1N6/RESULT/1/R1r_T1r.d
OUTPUT (CSD): demo/synth_SN100_18mGamR0.8RV0.1N6/RESULT/1/T10_R0r.d
OUTPUT (CSD): demo/synth_SN100_18mGamR0.8RV0.1N6/RESULT/1/T10_T0r.d
OUTPUT (CSD): demo/synth_SN100_18mGamR0.8RV0.1N6/RESULT/1/T10_T1r.d
OUTPUT (CSD): demo/synth_SN100_18mGamR0.8RV0.1N6/RESULT/1/T0r_R0r.d
OUTPUT (CSD): demo/synth_SN100_18mGamR0.8RV0.1N6/RESULT/1/T0r_T10.d
OUTPUT (CSD): demo/synth_SN100_18mGamR0.8RV0.1N6/RESULT/1/T0r_T0r.d
OUTPUT (CSD): demo/synth_SN100_18mGamR0.8RV0.1N6/RESULT/1/T0r_T1r.d
OUTPUT (CSD): demo/synth_SN100_18mGamR0.8RV0.1N6/RESULT/1/T1r_R0r.d
OUTPUT (CSD): demo/synth_SN100_18mGamR0.8RV0.1N6/RESULT/1/T1r_T0r.d
OUTPUT (CSD): demo/synth_SN100_18mGamR0.8RV0.1N6/RESULT/1/T1r_T1r.d
OUTPUT (CSD): demo/synth_SN100_18mGamR0.8RV0.1N6/RESULT/1/Z00_Z00.d
OUTPUT (CSD): demo/synth_SN100_18mGamR0.8RV0.1N6/RESULT/1/Z00_Z0r.d
OUTPUT (CSD): demo/synth_SN100_18mGamR0.8RV0.1N6/RESULT/1/Z0r_Z00.d
OUTPUT (CSD): demo/synth_SN100_18mGamR0.8RV0.1N6/RESULT/1/Z0r_Z0r.d
OUTPUT (CSD): demo/synth_SN100_18mGamR0.8RV0.1N6/RESULT/1/Z1r_R1r.d
OUTPUT (CSD): demo/synth_SN100_18mGamR0.8RV0.1N6/RESULT/1/Z1r_T1r.d
OUTPUT (CSD): demo/synth_SN100_18mGamR0.8RV0.1N6/RESULT/1/Z1r_Z1r.d
<ESTSPEC>
NOICE:idrimag=    0 read only columns 1 and 2
No other columns are read
OUTPUT DIR: demo/synth_SN100_18mGamR0.8RV0.1N6/RESULT/1
 No. of Seismograms: 3
DATAFILE  1) demo/synth_SN100_18mGamR0.8RV0.1N6/RESULT/wavud.d
DATAFILE  2) demo/synth_SN100_18mGamR0.8RV0.1N6/RESULT/wavew.d
DATAFILE  3) demo/synth_SN100_18mGamR0.8RV0.1N6/RESULT/wavns.d

Cross-spectral density calculations (1:yes; 2: no)
   |  1  2  3
-------------
  1|  1  0  0
  2|  0  1  0
  3|  0  0  1

No. of segments:    10
Data segment duration: 10.240 s
dt           :  0.010 s
Total duration (zero-padded):   20.480s (    2048 pts)
Ratio of data length to total length (zero-padded):  0.500
(Power reduction is corrected based on this value)
Frequency interval:0.488E-01 Hz
Nyquist frequency : 50.0     Hz
Use data window  [Hanning window with taper rate: 0.500 (  513 pts)]
Ratio of tapered power to raw power:  0.50
(Power reduction is corrected based on this value)
Use spectral window [Parzen window with band width:  0.30000 Hz (Total   13 pts)]
Resulting degree of freedom:    65
Standard error:     0.176777 (the effects of cos taper & zero-padding considered)
OUTPUT (CSD): demo/synth_SN100_18mGamR0.8RV0.1N6/RESULT/1/wavud_wavud.d
OUTPUT (CSD): demo/synth_SN100_18mGamR0.8RV0.1N6/RESULT/1/wavew_wavew.d
OUTPUT (CSD): demo/synth_SN100_18mGamR0.8RV0.1N6/RESULT/1/wavns_wavns.d
<ESTSPEC>
NOICE:idrimag=    0 read only columns 1 and 2
No other columns are read
OUTPUT DIR: demo/synth_SN100_18mGamR0.8RV0.1N6/RESULT/1
 No. of Seismograms: 18
DATAFILE  1) demo/synth_SN100_18mGamR0.8RV0.1N6/RESULT/cpwav/SD1u.d
DATAFILE  2) demo/synth_SN100_18mGamR0.8RV0.1N6/RESULT/cpwav/SD2u.d
DATAFILE  3) demo/synth_SN100_18mGamR0.8RV0.1N6/RESULT/cpwav/SD3u.d
DATAFILE  4) demo/synth_SN100_18mGamR0.8RV0.1N6/RESULT/cpwav/SD4u.d
DATAFILE  5) demo/synth_SN100_18mGamR0.8RV0.1N6/RESULT/cpwav/SD5u.d
DATAFILE  6) demo/synth_SN100_18mGamR0.8RV0.1N6/RESULT/cpwav/SD6u.d
DATAFILE  7) demo/synth_SN100_18mGamR0.8RV0.1N6/RESULT/cpwav/SD1n.d
DATAFILE  8) demo/synth_SN100_18mGamR0.8RV0.1N6/RESULT/cpwav/SD2n.d
DATAFILE  9) demo/synth_SN100_18mGamR0.8RV0.1N6/RESULT/cpwav/SD3n.d
DATAFILE 10) demo/synth_SN100_18mGamR0.8RV0.1N6/RESULT/cpwav/SD4n.d
DATAFILE 11) demo/synth_SN100_18mGamR0.8RV0.1N6/RESULT/cpwav/SD5n.d
DATAFILE 12) demo/synth_SN100_18mGamR0.8RV0.1N6/RESULT/cpwav/SD6n.d
DATAFILE 13) demo/synth_SN100_18mGamR0.8RV0.1N6/RESULT/cpwav/SD1e.d
DATAFILE 14) demo/synth_SN100_18mGamR0.8RV0.1N6/RESULT/cpwav/SD2e.d
DATAFILE 15) demo/synth_SN100_18mGamR0.8RV0.1N6/RESULT/cpwav/SD3e.d
DATAFILE 16) demo/synth_SN100_18mGamR0.8RV0.1N6/RESULT/cpwav/SD4e.d
DATAFILE 17) demo/synth_SN100_18mGamR0.8RV0.1N6/RESULT/cpwav/SD5e.d
DATAFILE 18) demo/synth_SN100_18mGamR0.8RV0.1N6/RESULT/cpwav/SD6e.d

Cross-spectral density calculations (1:yes; 2: no)
   |  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18
-------------------------------------------------------------
  1|  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
  2|  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
  3|  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
  4|  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0
  5|  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0
  6|  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0
  7|  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0
  8|  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0
  9|  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0
 10|  0  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0
 11|  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0
 12|  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0
 13|  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0  0
 14|  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0
 15|  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0
 16|  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0
 17|  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0
 18|  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1

No. of segments:    10
Data segment duration: 10.240 s
dt           :  0.010 s
Total duration (zero-padded):   20.480s (   2048 pts)
Ratio of data length to total length (zero-padded):  0.500
(Power reduction is corrected based on this value)
Frequency interval:0.488E-01 Hz
Nyquist frequency : 50.0     Hz
Use data window  [Hanning window with taper rate: 0.500 (  513 pts)]
Ratio of tapered power to raw power:  0.50
(Power reduction is corrected based on this value)
Use spectral window [Parzen window with band width:  0.30000 Hz (Total   13 pts)]
Resulting degree of freedom:    65
Standard error:     0.176777 (the effects of cos taper & zero-padding considered)
OUTPUT (CSD): demo/synth_SN100_18mGamR0.8RV0.1N6/RESULT/1/SD1u_SD1u.d
OUTPUT (CSD): demo/synth_SN100_18mGamR0.8RV0.1N6/RESULT/1/SD2u_SD2u.d
OUTPUT (CSD): demo/synth_SN100_18mGamR0.8RV0.1N6/RESULT/1/SD3u_SD3u.d
OUTPUT (CSD): demo/synth_SN100_18mGamR0.8RV0.1N6/RESULT/1/SD4u_SD4u.d
OUTPUT (CSD): demo/synth_SN100_18mGamR0.8RV0.1N6/RESULT/1/SD5u_SD5u.d
OUTPUT (CSD): demo/synth_SN100_18mGamR0.8RV0.1N6/RESULT/1/SD6u_SD6u.d
OUTPUT (CSD): demo/synth_SN100_18mGamR0.8RV0.1N6/RESULT/1/SD1n_SD1n.d
OUTPUT (CSD): demo/synth_SN100_18mGamR0.8RV0.1N6/RESULT/1/SD2n_SD2n.d
OUTPUT (CSD): demo/synth_SN100_18mGamR0.8RV0.1N6/RESULT/1/SD3n_SD3n.d
OUTPUT (CSD): demo/synth_SN100_18mGamR0.8RV0.1N6/RESULT/1/SD4n_SD4n.d
OUTPUT (CSD): demo/synth_SN100_18mGamR0.8RV0.1N6/RESULT/1/SD5n_SD5n.d
OUTPUT (CSD): demo/synth_SN100_18mGamR0.8RV0.1N6/RESULT/1/SD6n_SD6n.d
OUTPUT (CSD): demo/synth_SN100_18mGamR0.8RV0.1N6/RESULT/1/SD1e_SD1e.d
OUTPUT (CSD): demo/synth_SN100_18mGamR0.8RV0.1N6/RESULT/1/SD2e_SD2e.d
OUTPUT (CSD): demo/synth_SN100_18mGamR0.8RV0.1N6/RESULT/1/SD3e_SD3e.d
OUTPUT (CSD): demo/synth_SN100_18mGamR0.8RV0.1N6/RESULT/1/SD4e_SD4e.d
OUTPUT (CSD): demo/synth_SN100_18mGamR0.8RV0.1N6/RESULT/1/SD5e_SD5e.d
OUTPUT (CSD): demo/synth_SN100_18mGamR0.8RV0.1N6/RESULT/1/SD6e_SD6e.d
<SPECRATIO>
OUTPUT (for CCA)  demo/synth_SN100_18mGamR0.8RV0.1N6/RESULT/1/spr_cca.d
OUTPUT (for SPAC)  demo/synth_SN100_18mGamR0.8RV0.1N6/RESULT/1/spr_spac.d
OUTPUT (for H0)  demo/synth_SN100_18mGamR0.8RV0.1N6/RESULT/1/spr_h0.d
OUTPUT (for MSC)  demo/synth_SN100_18mGamR0.8RV0.1N6/RESULT/1/coh2.d
Array radius: 0.018000 [km][from demo/synth_SN100_18mGamR0.8RV0.1N6/RESULT/Z0r.d.log]
OUTPUT (for NSR_Z)  demo/synth_SN100_18mGamR0.8RV0.1N6/RESULT/1/nsr.d
OUTPUT (for signal power)  demo/synth_SN100_18mGamR0.8RV0.1N6/RESULT/1/pow_signal.d
OUTPUT (for noise power)   demo/synth_SN100_18mGamR0.8RV0.1N6/RESULT/1/pow_noise.d
OUTPUT (for H1)  demo/synth_SN100_18mGamR0.8RV0.1N6/RESULT/1/spr_h1.d
OUTPUT (for V)  demo/synth_SN100_18mGamR0.8RV0.1N6/RESULT/1/spr_v.d
<SPECRATIO for horizontal components>
OUTPUT (for CCA-L)  demo/synth_SN100_18mGamR0.8RV0.1N6/RESULT/1/sprh_cca_minus_L.d
OUTPUT (for CCA-R)  demo/synth_SN100_18mGamR0.8RV0.1N6/RESULT/1/sprh_cca_minus_R.d
OUTPUT (for R/V)  demo/synth_SN100_18mGamR0.8RV0.1N6/RESULT/1/AmpRV_R.d
OUTPUT (for SPAC-L)  demo/synth_SN100_18mGamR0.8RV0.1N6/RESULT/1/sprh_spac_minus_L.d
OUTPUT (for SPAC+L)  demo/synth_SN100_18mGamR0.8RV0.1N6/RESULT/1/sprh_spac_plus_L.d
OUTPUT (for SPAC-R)  demo/synth_SN100_18mGamR0.8RV0.1N6/RESULT/1/sprh_spac_minus_R.d
OUTPUT (for SPAC+R)  demo/synth_SN100_18mGamR0.8RV0.1N6/RESULT/1/sprh_spac_plus_R.d
OUTPUT (for R/(R+L))  demo/synth_SN100_18mGamR0.8RV0.1N6/RESULT/1/powratio_R2L.d
OUTPUT (for NSR_H)  demo/synth_SN100_18mGamR0.8RV0.1N6/RESULT/1/nsr_h.d
<SPECRATIO HV>
```

```
OUTPUT (for HV spectral ratio） demo/synth_SN100_18mGamR0.8RV0.1N6/RESULT/1/spr_hv.d
<SPEC2PV>
Array radius: 0.018000 [km][from demo/synth_SN100_18mGamR0.8RV0.1N6/RESULT/Z0r.d.log]
OUTPUT (for CCA） demo/synth_SN100_18mGamR0.8RV0.1N6/RESULT/1/vel_cca.d
OUTPUT (for SPAC） demo/synth_SN100_18mGamR0.8RV0.1N6/RESULT/1/vel_spac.d
OUTPUT (for H0） demo/synth_SN100_18mGamR0.8RV0.1N6/RESULT/1/vel_h0.d
OUTPUT (for H1） demo/synth_SN100_18mGamR0.8RV0.1N6/RESULT/1/vel_h1.d
OUTPUT (for V） demo/synth_SN100_18mGamR0.8RV0.1N6/RESULT/1/vel_v.d
OUTPUT (for nc-CCA） demo/synth_SN100_18mGamR0.8RV0.1N6/RESULT/1/vel_cca.lwapx.d
OUTPUT (for NSR limit by CCA） demo/synth_SN100_18mGamR0.8RV0.1N6/RESULT/1/nsrlim_cca.d
OUTPUT (for NSR limit by nc-CCA） demo/synth_SN100_18mGamR0.8RV0.1N6/RESULT/1/nsrlim_cca.lwapx.d
<SPEC2PV for horizontal components>
Array radius: 0.018000 [km][from demo/synth_SN100_18mGamR0.8RV0.1N6/RESULT/R0r.d.log]
OUTPUT (for CCA-L） demo/synth_SN100_18mGamR0.8RV0.1N6/RESULT/1/velh_cca_minus_L.d
OUTPUT (for CCA-R） demo/synth_SN100_18mGamR0.8RV0.1N6/RESULT/1/velh_cca_minus_R.d
OUTPUT (for SPAC-L） demo/synth_SN100_18mGamR0.8RV0.1N6/RESULT/1/velh_spac_minus_L.d
OUTPUT (for SPAC+L） demo/synth_SN100_18mGamR0.8RV0.1N6/RESULT/1/velh_spac_plus_L.d
OUTPUT (for SPAC-R） demo/synth_SN100_18mGamR0.8RV0.1N6/RESULT/1/velh_spac_minus_R.d
OUTPUT (for SPAC+R） demo/synth_SN100_18mGamR0.8RV0.1N6/RESULT/1/velh_spac_plus_R.d

### Step 2 (Total 5) ###

. . . （省略） . . .

Averaging over 5 segment clusters...
For the sake of robustness, the maximum and minimum values (plus non-numbers (ex. NaN)) are eliminated from the averaging
process when the number of segment clusters exceeds NROBUST4AVERAGE_INC (defined in the file PARAM.h). Read the file
demo/synth_SN100_18mGamR0.8RV0.1N6/RESULT/ave/ave.info for details.

OUTPUT: demo/synth_SN100_18mGamR0.8RV0.1N6/RESULT/ave/vel_cca.d
OUTPUT: demo/synth_SN100_18mGamR0.8RV0.1N6/RESULT/ave/vel_cca.lwapx.d
OUTPUT: demo/synth_SN100_18mGamR0.8RV0.1N6/RESULT/ave/vel_h0.d
OUTPUT: demo/synth_SN100_18mGamR0.8RV0.1N6/RESULT/ave/vel_h1.d
OUTPUT: demo/synth_SN100_18mGamR0.8RV0.1N6/RESULT/ave/vel_spac.d
OUTPUT: demo/synth_SN100_18mGamR0.8RV0.1N6/RESULT/ave/vel_v.d
OUTPUT: demo/synth_SN100_18mGamR0.8RV0.1N6/RESULT/ave/velh_cca_minus_L.d
OUTPUT: demo/synth_SN100_18mGamR0.8RV0.1N6/RESULT/ave/velh_cca_minus_R.d
OUTPUT: demo/synth_SN100_18mGamR0.8RV0.1N6/RESULT/ave/velh_spac_minus_L.d
OUTPUT: demo/synth_SN100_18mGamR0.8RV0.1N6/RESULT/ave/velh_spac_minus_R.d
OUTPUT: demo/synth_SN100_18mGamR0.8RV0.1N6/RESULT/ave/velh_spac_plus_L.d
OUTPUT: demo/synth_SN100_18mGamR0.8RV0.1N6/RESULT/ave/velh_spac_plus_R.d
OUTPUT: demo/synth_SN100_18mGamR0.8RV0.1N6/RESULT/ave/AmpRV_R.d
OUTPUT: demo/synth_SN100_18mGamR0.8RV0.1N6/RESULT/ave/powratio_R2L.d
OUTPUT: demo/synth_SN100_18mGamR0.8RV0.1N6/RESULT/ave/nsr.d
OUTPUT: demo/synth_SN100_18mGamR0.8RV0.1N6/RESULT/ave/nsr_h.d
OUTPUT: demo/synth_SN100_18mGamR0.8RV0.1N6/RESULT/ave/nsrlim_cca.d
OUTPUT: demo/synth_SN100_18mGamR0.8RV0.1N6/RESULT/ave/nsrlim_cca.lwapx.d
OUTPUT: demo/synth_SN100_18mGamR0.8RV0.1N6/RESULT/ave/spr_hv.d
OUTPUT: demo/synth_SN100_18mGamR0.8RV0.1N6/RESULT/ave/SD1e_SD1e.d
OUTPUT: demo/synth_SN100_18mGamR0.8RV0.1N6/RESULT/ave/SD1n_SD1n.d
OUTPUT: demo/synth_SN100_18mGamR0.8RV0.1N6/RESULT/ave/SD1u_SD1u.d
OUTPUT: demo/synth_SN100_18mGamR0.8RV0.1N6/RESULT/ave/SD2e_SD2e.d
OUTPUT: demo/synth_SN100_18mGamR0.8RV0.1N6/RESULT/ave/SD2n_SD2n.d
OUTPUT: demo/synth_SN100_18mGamR0.8RV0.1N6/RESULT/ave/SD2u_SD2u.d
OUTPUT: demo/synth_SN100_18mGamR0.8RV0.1N6/RESULT/ave/SD3e_SD3e.d
OUTPUT: demo/synth_SN100_18mGamR0.8RV0.1N6/RESULT/ave/SD3n_SD3n.d
OUTPUT: demo/synth_SN100_18mGamR0.8RV0.1N6/RESULT/ave/SD3u_SD3u.d
OUTPUT: demo/synth_SN100_18mGamR0.8RV0.1N6/RESULT/ave/SD4e_SD4e.d
OUTPUT: demo/synth_SN100_18mGamR0.8RV0.1N6/RESULT/ave/SD4n_SD4n.d
OUTPUT: demo/synth_SN100_18mGamR0.8RV0.1N6/RESULT/ave/SD4u_SD4u.d
OUTPUT: demo/synth_SN100_18mGamR0.8RV0.1N6/RESULT/ave/SD5e_SD5e.d
OUTPUT: demo/synth_SN100_18mGamR0.8RV0.1N6/RESULT/ave/SD5n_SD5n.d
OUTPUT: demo/synth_SN100_18mGamR0.8RV0.1N6/RESULT/ave/SD5u_SD5u.d
OUTPUT: demo/synth_SN100_18mGamR0.8RV0.1N6/RESULT/ave/SD6e_SD6e.d
OUTPUT: demo/synth_SN100_18mGamR0.8RV0.1N6/RESULT/ave/SD6n_SD6n.d
OUTPUT: demo/synth_SN100_18mGamR0.8RV0.1N6/RESULT/ave/SD6u_SD6u.d

<mkgnplt.sh>
NOTE: This script uses a program gnuplot to plot the analysis results (all analysis results are contained in
demo/synth_SN100_18mGamR0.8RV0.1N6/RESULT). You can manually plot the same figures later, using a gnuplot macro,
plot.gnplt, which was created by this script and is found in demo/synth_SN100_18mGamR0.8RV0.1N6/RESULT. To do this, type cd
demo/synth_SN100_18mGamR0.8RV0.1N6/RESULT & gnuplot plot.gnplt
OUTPUT: demo/synth_SN100_18mGamR0.8RV0.1N6/RESULT/plot.gnplt

<mkdelete.sh>
All temporary files preserved.
```