hvsrpy - A Python module for horizontal-to-vertical spectral ratio (H/V) calculations

> Joseph Vantassel, University of Texas at Austin

About *hvsrpy*

`hvsrpy` is a Python module for performing horizontal-to-vertical spectral ratio (H/V) calculations. `hvsrpy` was developed by Joseph P. Vantassel with contributions from Dana M. Brannon under the supervision of Professor Brady R. Cox at the University of Texas at Austin. The fully-automated frequency-domain rejection algorithm implemented in `hvsrpy` was developed by Tianjian Cheng under the supervision of Professor Brady R. Cox at the Univesity of Texas at Austin and detailed in Cox et al. (in review).

Comparison of *hvsrpy* with *geopsy*

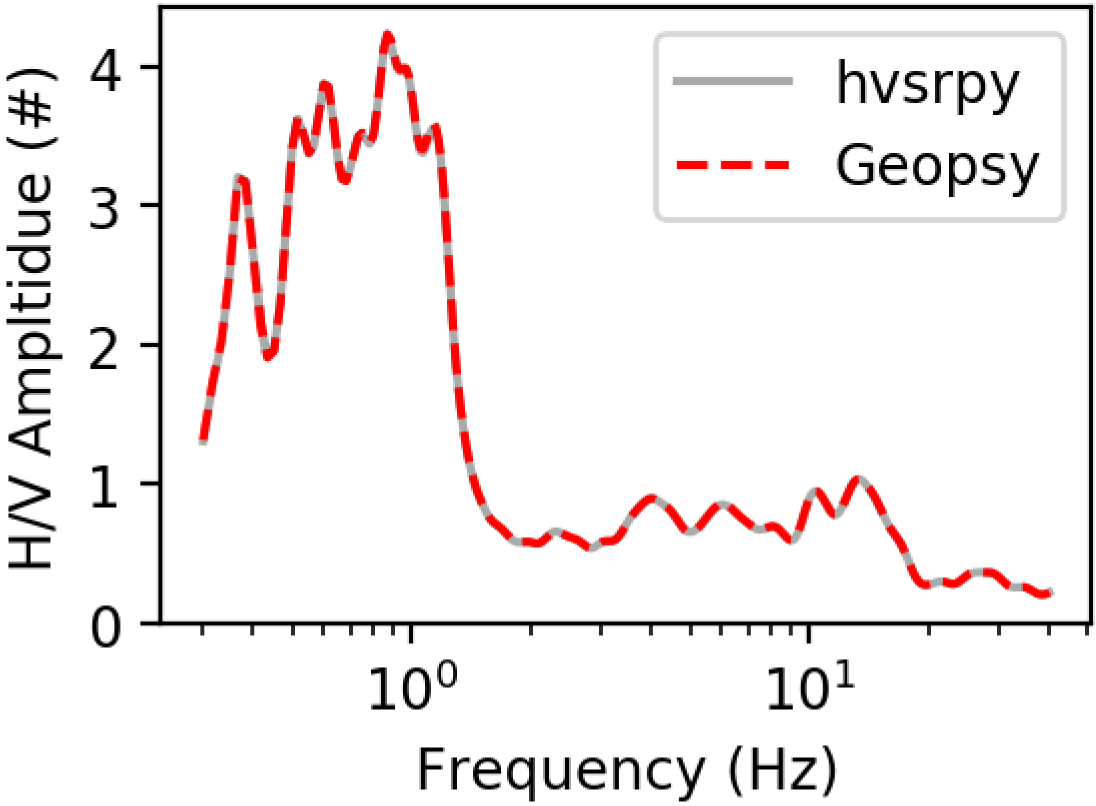
While `hvsrpy` can be used to produce results exactly equal to the popular open-source program geopsy , `hvsrpy` contains additional functionality including:

- A fully-automated frequency-domain rejection algorithm.
- An option to use a log-normal distribution for the mean `f0` from the time windows so H/V uncertainty can be represented consistently in terms of frequency and period.
- Combination of the two orthoginal horizontal components using the geometric mean.
- Direct access to the H/V data from each time window, not only the mean/median curve.

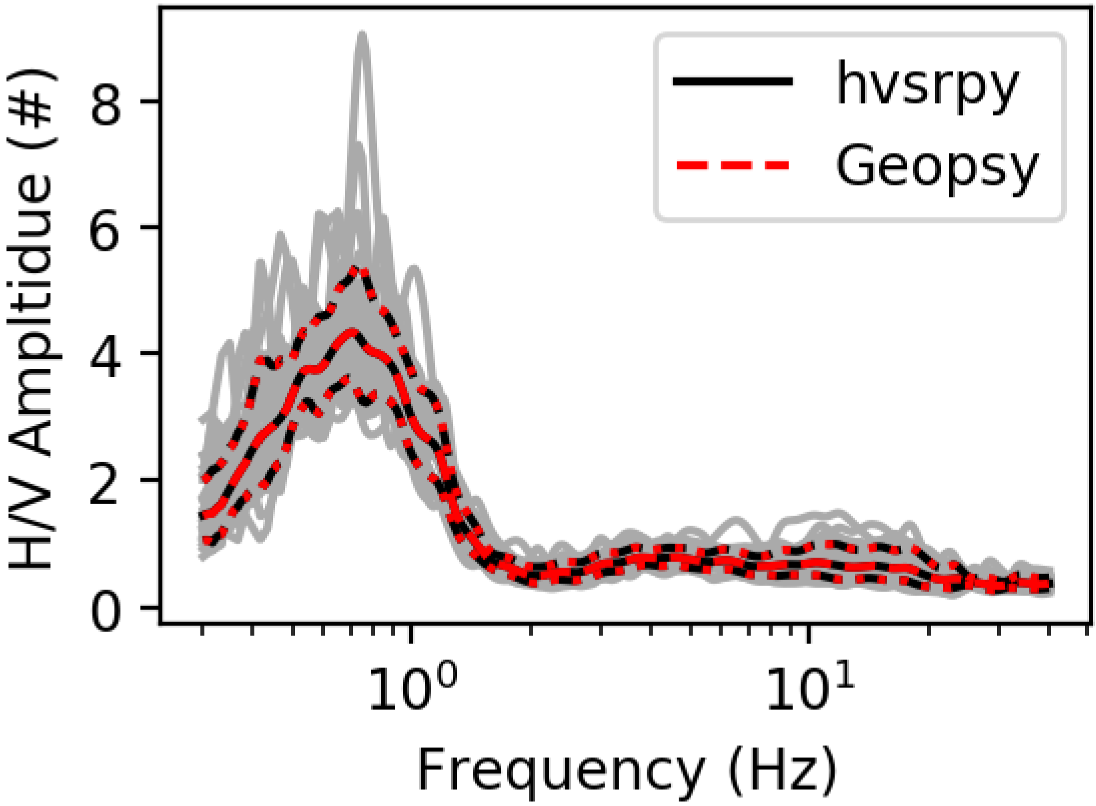
After completing the **Getting Started** section below, use the provided examples to explore all of these new features.

To illustrate that `hvsrpy` can exactly reproduce the results from Geopsy two comparisons are shown below. One for a single time window and one for multiple time windows. More examples and the necessary information to reproduce them if so desired are provided at the end of this document.

Single Time Window



Multiple Time Windows



Getting Started

Installing *sigpropy* (a dependency of *hvsrpy*)

1. Download and unzip the provided zip file named `hvsrpy_v0.0.1`.

2. Move the directory `sigpropy` and its contents to the root directory of your main hardrive, this is typically the `C` drive on Windows.

3. Open a Windows Powershell (recommended) or Command Prompt window inside the `sigpropy` directory. If using Windows Powershell you can do this with `shift + right click` on the directory and selecting the option `open PowerShell window here`. If using Command Prompt you will need to navigate to that directory using the console.

4. Ensure you are in the correct directory by confirming it contains a sub-directory call `sigpropy` and a file named `setup.py`. You can see the contents of the current directory by using the command `ls` in Windows Powershell or `dir` in Command Prompt.

5. If in the correct directory, install the module's dependencies with `pip install -r requirements.txt`. Note that an internet connection is required for the installation to be successful.

6. And install the module with `pip install .`.

7. Confirm that `sigpropy` was built successfully by reading the last few lines printed to the console.

Installing *hvsrpy* and its dependencies

1. Move the directory `hvsrpy` and its contents to the root directory of your main hard drive, this is typically the `C` drive on Windows.

2. Open a Windows Powershell (recommended) or Command Prompt window inside the `hvsrpy` directory.

3. Ensure you are in the correct directory by confirming it contains a sub-directory call `hvsrpy` and a file named `setup.py`.

4. If in the correct directory, install the module's dependencies with `pip install -r requirements.txt`.

5. And install the module with `pip install .`.

6. Confirm that `hvsrpy` was built successfully by reading the last few lines printed to the console.

Begin using *hvsrpy*

1. Copy the directory `examples` and its contents out of the directory `hvsrpy` which is now located on your main hard drive (recall *Step 1.* of the section **Installing *hvsrpy* and its dependencies**) and move to any location of your choice.

2. Navigate to the copy of the `examples` directory and open the Jupyter notebook titled `simple_hvsrpy_interface.ipynb`.

3. Follow the instructions in the notebook for a no-coding-required introduction to the `hvsrpy` module.

Reproducible Comparisons between *hvsrpy* and *geopsy*

Multiple Windows

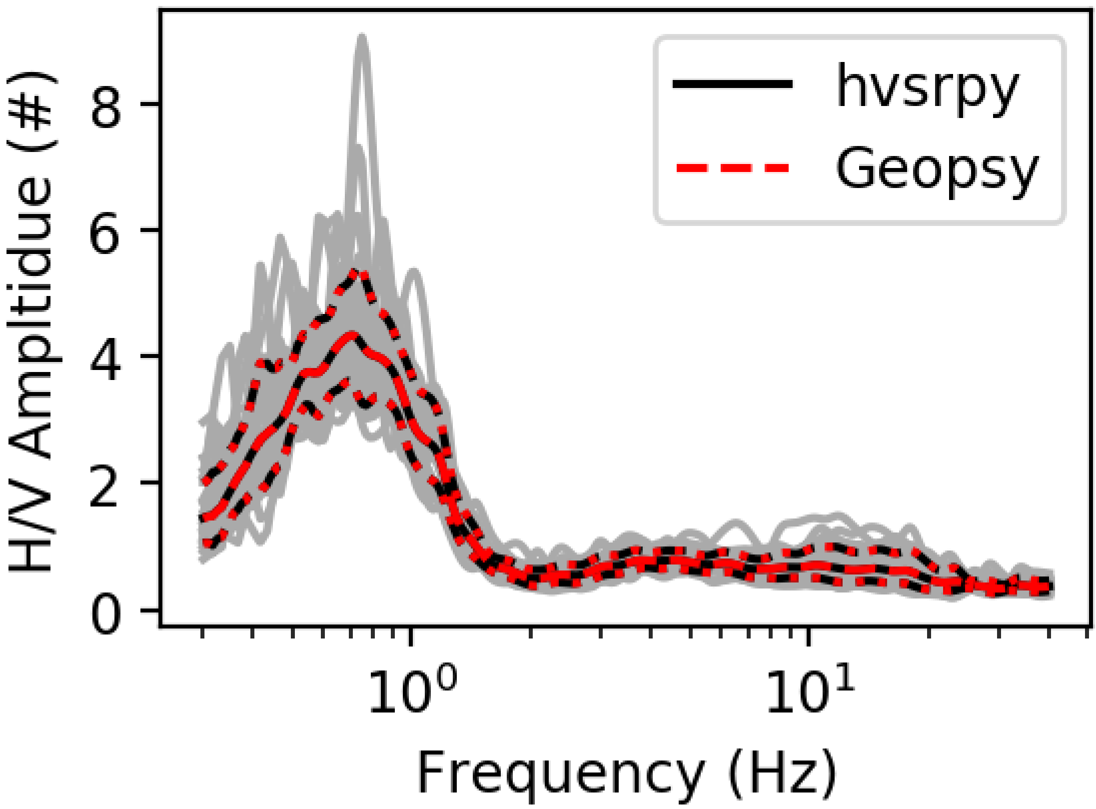
All of the following examples utilize the same settings applied to different noise records. The settings are provided below with the name of each file provided above the corresponding figure. The noise records themselves are provided in the `examples` directory included as part of this module, see previous section for details.

Settings

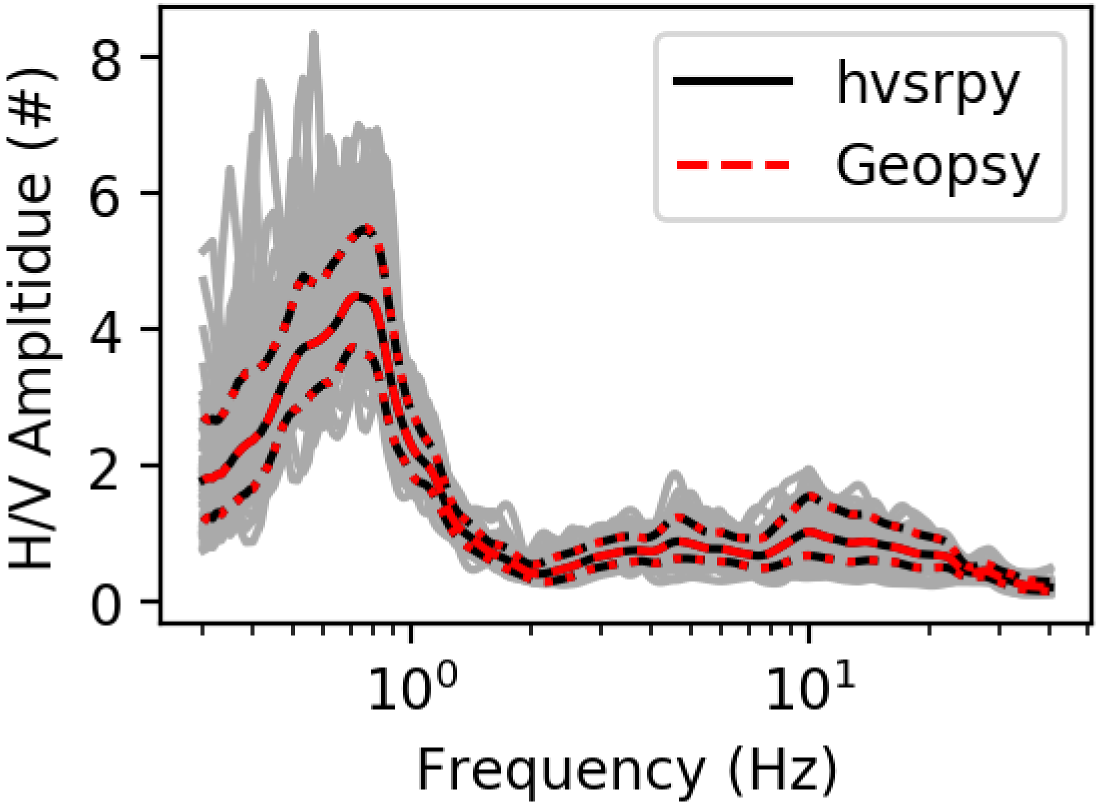
- **Window Length:** 60 seconds
- **Bandpass Filter Boolean:** False
- **Cosine Taper Width:** 10% (i.e., 5% in geopsy)
- **Konno and Ohmachi Smoothing Coefficient:** 40
- **Resampling:**
  - **Minimum Frequency:** 0.3 Hz
  - **Maximum Frequency:** 40 Hz
  - **Number of Points:** 2048
  - **Sampling Type:** 'log'
- **Method for Combining Horizontal Components:** 'squared-average'
- **Distribution for f0 from Time Windows:** 'normal'
- **Distribution for Mean Curve:** 'log-normal'

Results

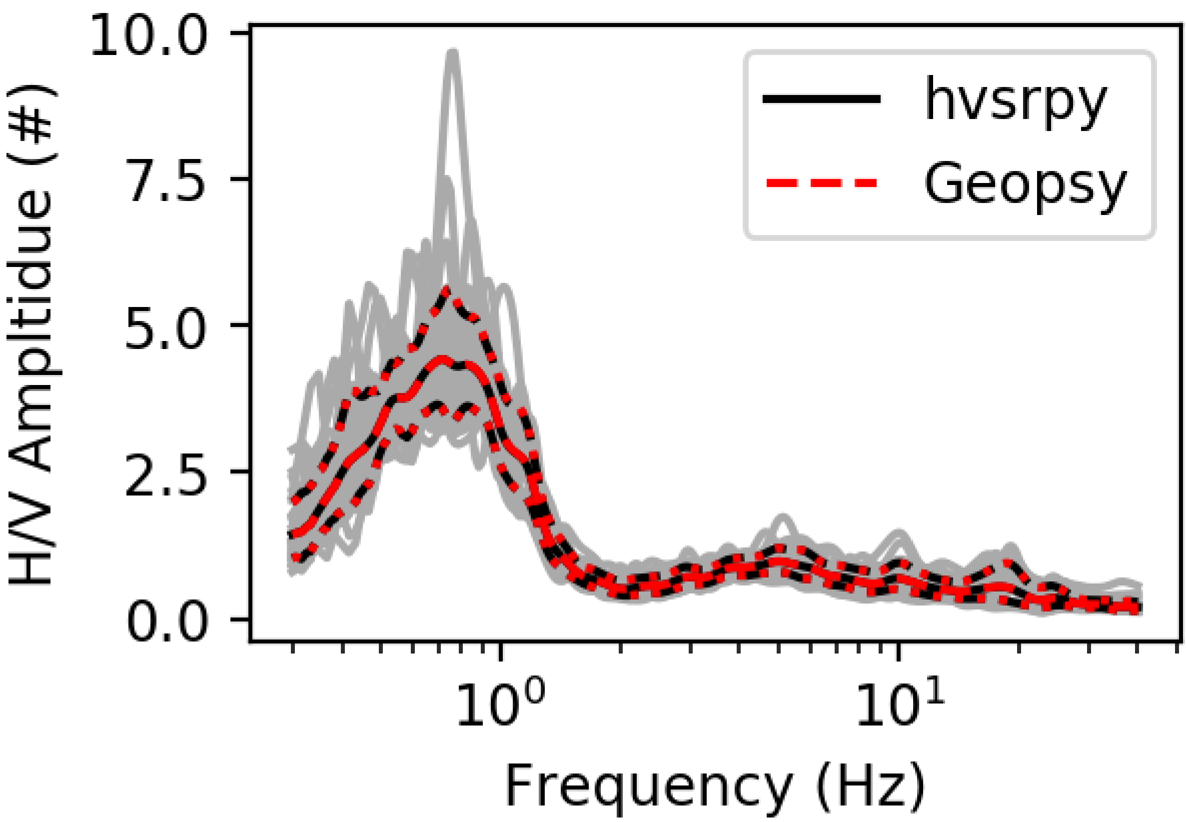
File Name: *UT.STN11.A2_C50.miniseed*



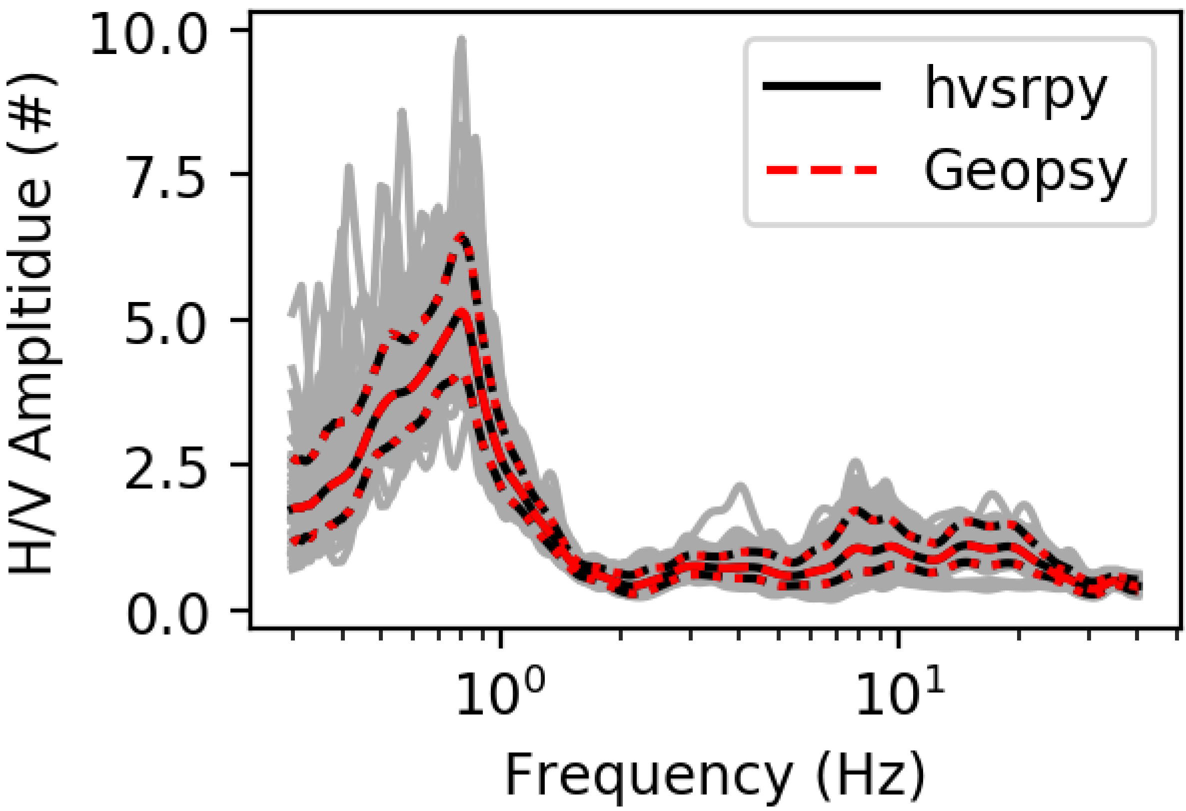
File Name: *UT.STN11.A2_C150.miniseed*



File Name: *UT.STN12.A2_C50.miniseed*



File Name: *UT.STN12.A2_C150.miniseed*



Single Window

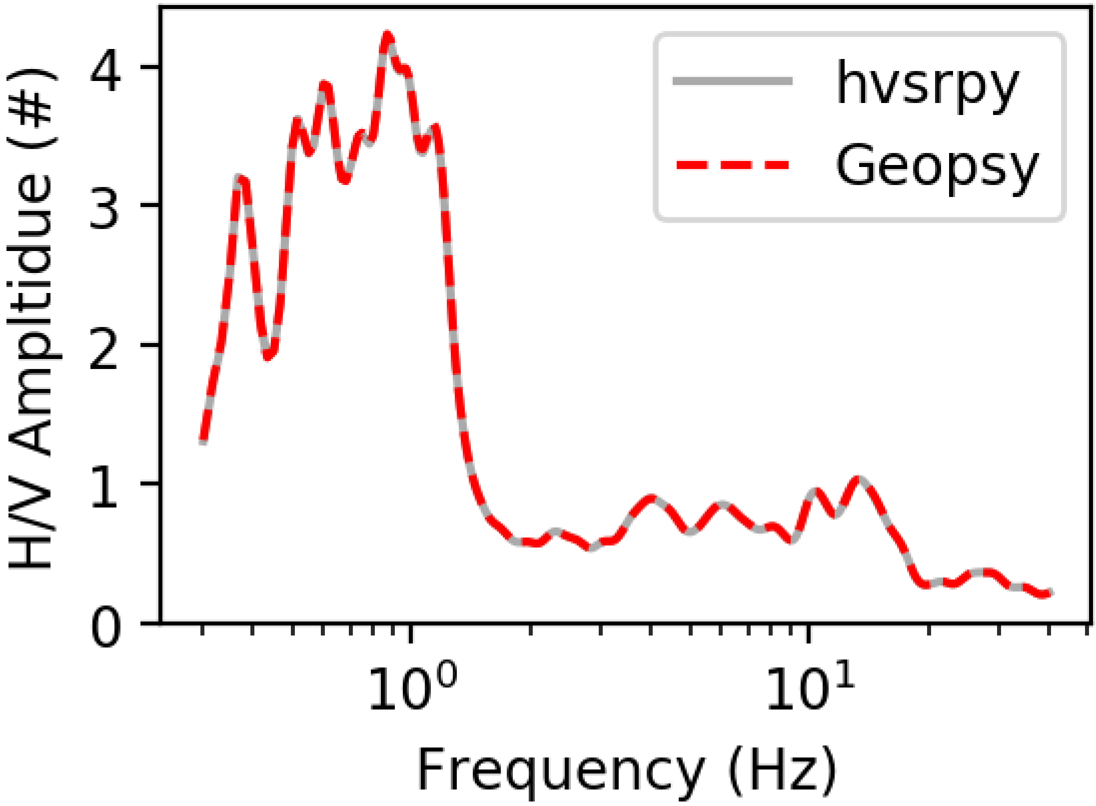
The following examples use different processing settings to the same noise record (*UT.STN11.A2_C50.miniseed*). The default settings are listed below such that only the changes from these setting are noted for each example.

Default Settings

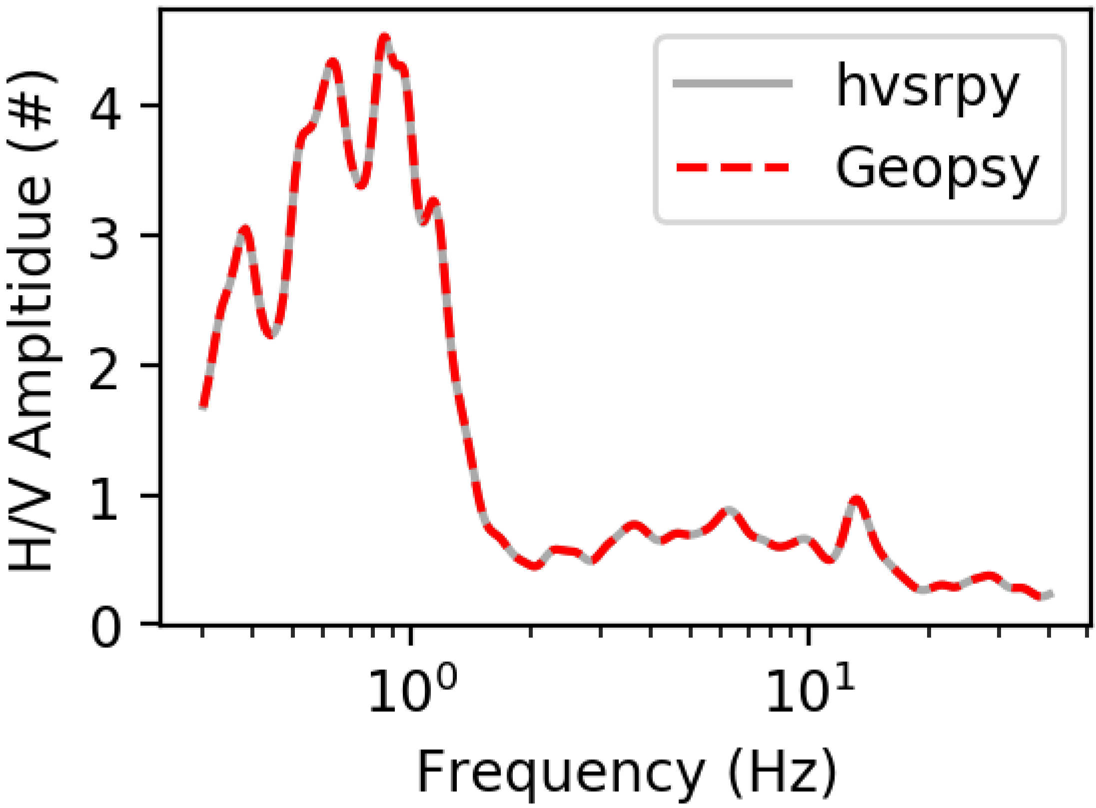
- **Window Length:** 60 seconds
- **Bandpass Filter Boolean:** False
- **Cosine Taper Width:** 10% (i.e., 5% in geopsy)
- **Konno and Ohmachi Smoothing Coefficient:** 40
- **Resampling:**
    - **Minimum Frequency:** 0.3 Hz
    - **Maximum Frequency:** 40 Hz
    - **Number of Points:** 2048
    - **Sampling Type:** 'log'
- **Method for Combining Horizontal Components:** 'squared-average'
- **Distribution for f0 from Time Windows:** 'normal'
- **Distribution for Mean Curve:** 'log-normal'

Results

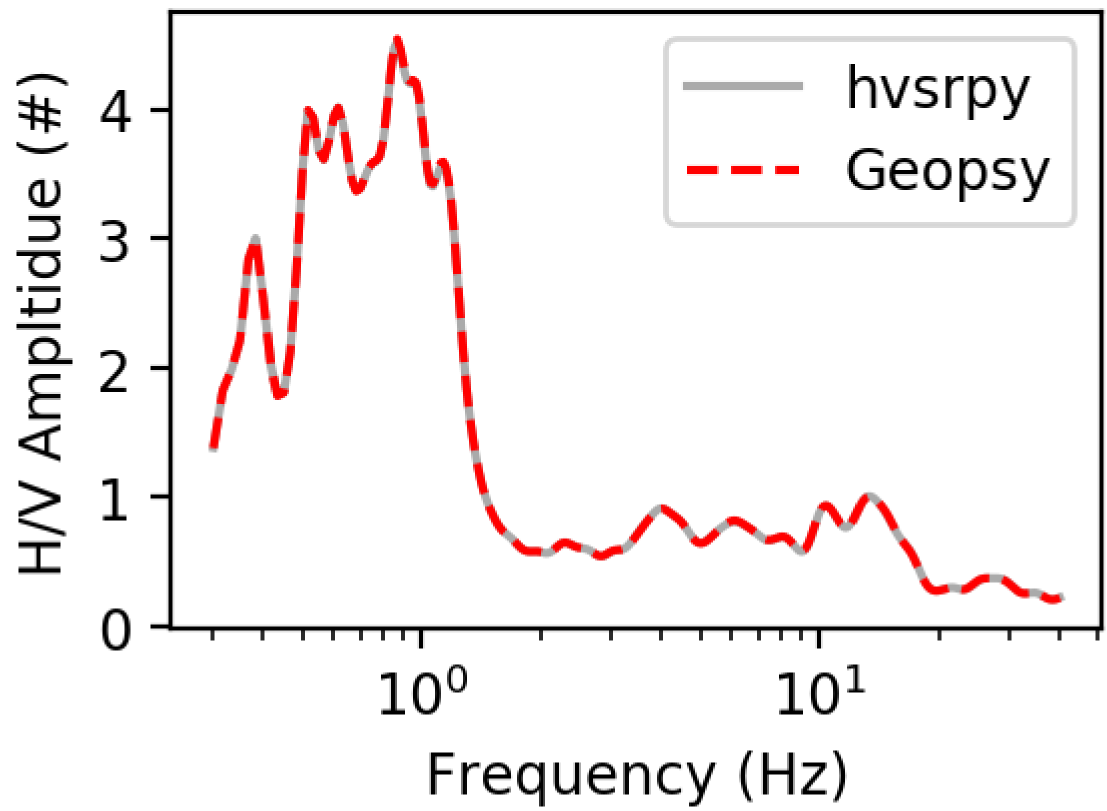
Default Case: No deviation from those settings listed above.



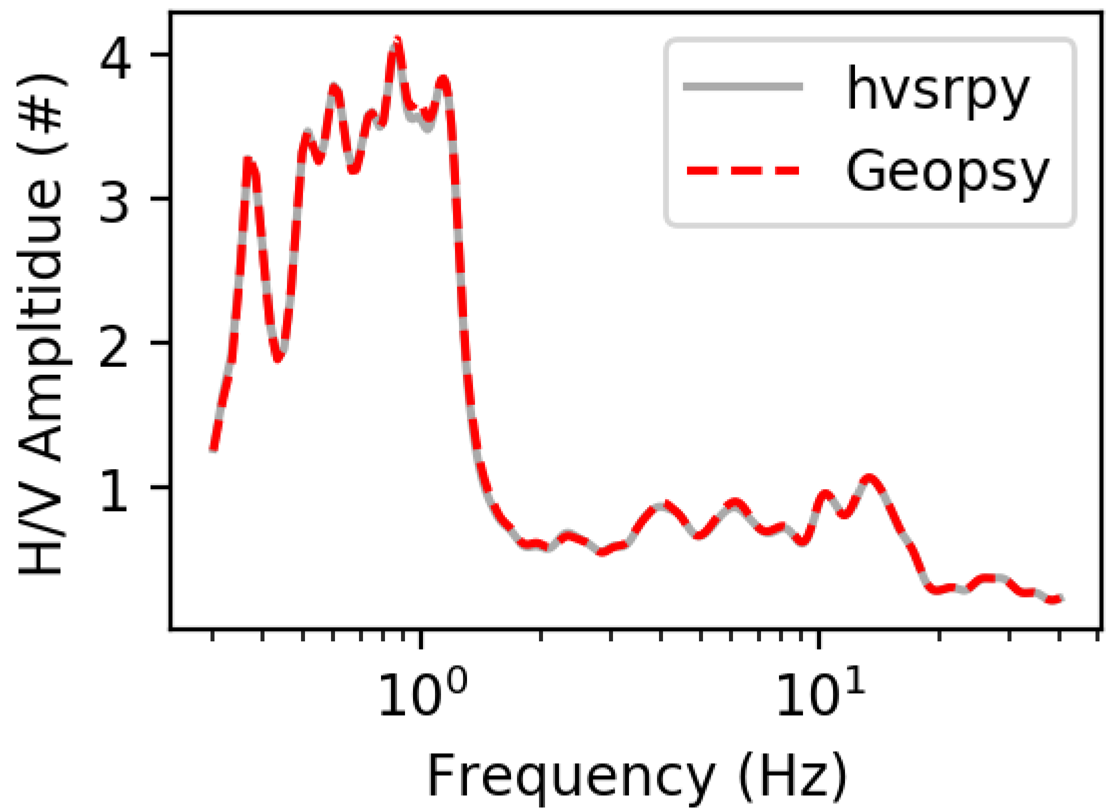
Window Length: 120 seconds.



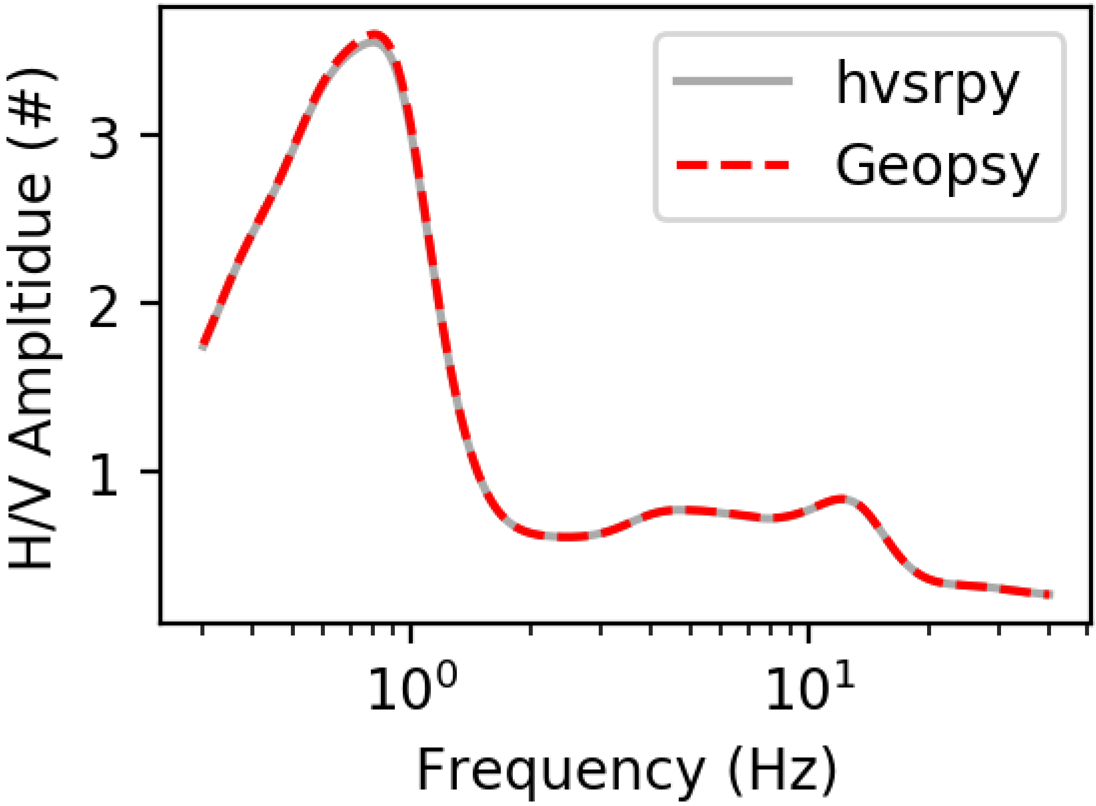
Cosine Taper Width: 20 % (i.e., 10% in geopsy)



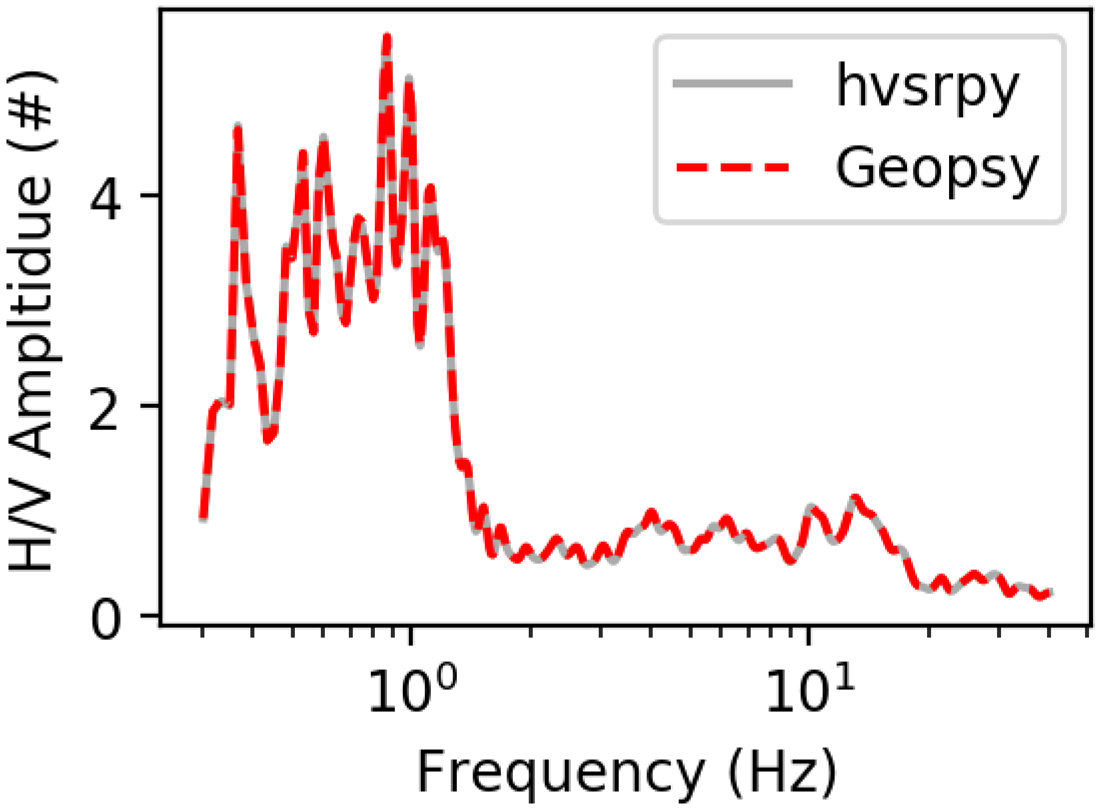
Cosine Taper Width: 0.2 % (i.e., 0.1% in geopsy)



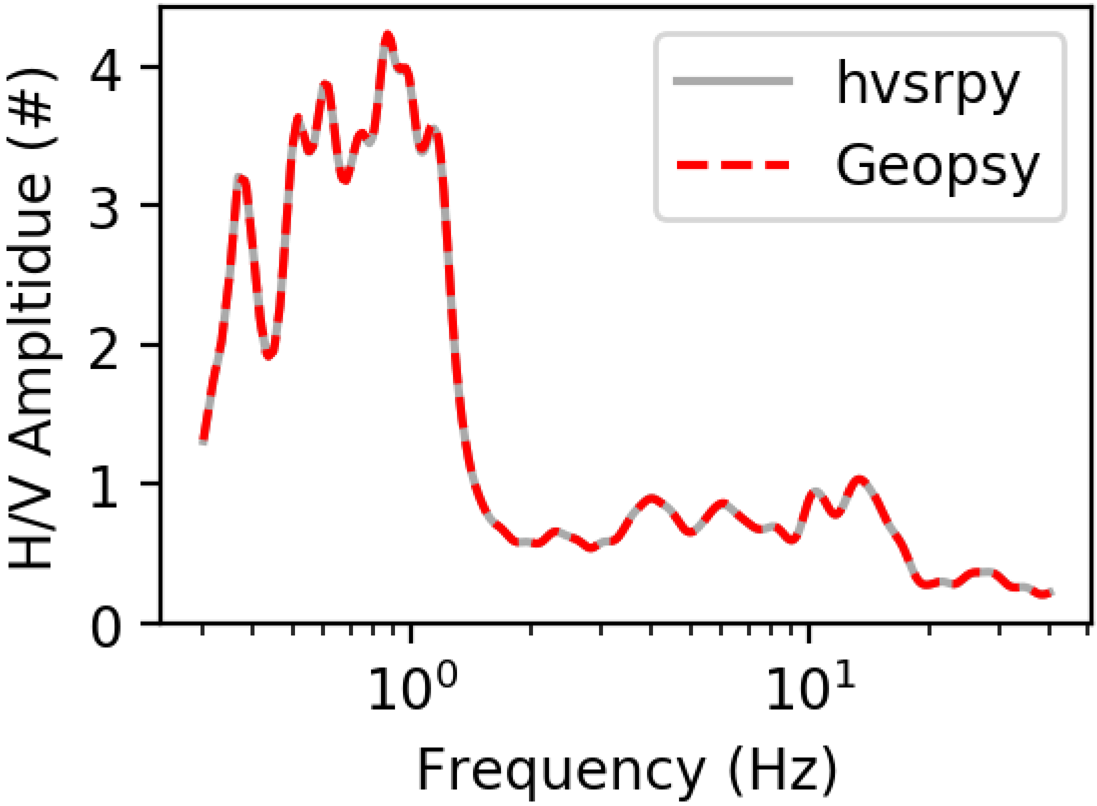
Konno and Ohmachi Smoothing Coefficient: 10



Konno and Ohmachi Smoothing Coefficient: 80



Number of Points: 512



Number of Points: 4096