> **This notebook will detail the complete process for exactly how a single station-event is downloaded, corrected by ATaCR, corrected by HPS, and compared with eachother. It can serve as a precise reference (a cookbook) on the specific actions all executed code takes at any stage in the download and corrections to a single station-event and the subsequent comparison of those results.**

## Data

For this example station-event correction and method comparison,

The station will be:

- Station: M07A
- Network: 7D
- Latitude: 44.898701
- Longitude: -125.116798
- Experiment: CASCADIA INITIATIVE
- Instrument_Design: AB
- Seismometer: Trillium Compact
- Environment: North Pacific
- Pressure_Gauge: DPG
- Water_Depth_m: 1356.5

The event will be:

- Event: 2012.181.21.07
- M: 6.3
- depth: 17.1
- ev_lla: 43.4335
- ev_lon: 84.6125

> Nothing in this correction and comparison process is designed to any specific station or event properties.

## Procedure Overview

This entire process is done, in sequence, in four phases:

> Phase One: Download data
>
> Phase Two: ATaCR Correction
>
> Phase Three: HPS Correction

> Phase Four: Compare Results

## Procedure

### Phase One: Download data

#### Step 1.1: Days randomly selected for ATaCR noise download

> Using the station's deployment times recorded at IRIS DMC, an external administrator function called ObsQA.TOOLS.io.DownloadDayNoise randomly selects 15 days then calls ATaCR's atacr_download_data script to download these days. To ensure the requested noise population is always satisfied and protected from errors caused by data gaps, atacr_download_data is called recursively until atleast 15 days of noise from each channel (Z,P,1,2) is downloaded and saved to SAC. If any day fails to download, a new one is randomly selected.
>
> > [Note] 10 days is the minimum target used in the noise spectra calculations. Thus, we download 15 days of noise in case some are removed in the QC steps.
> >
> > [Note] the random selection is made with a pre-defined seed so the same selections can be repeated for a given station.

#### Step 1.2: Noise downloaded, pre-processed, then saved to SAC.

> Within atacr_download_data, a standard download procedure is implimented for 24-hr four-component seismic/pressure data.
>
> Data is downloaded and processed in this exact sequence:

1. 24-hr seismic (Z,1,2) traces are downloaded by the IRIS client.
2. In a seperate request to IRIS, 24-hr pressure data is downloaded by the client.
3. All channels detrended (linear & demean).
4. A 2-corner zero-phase anti-alias filter is applied to each channel at 2.5hz.
5. Each channel is resampled to 5hz.
6. Timing is checked on all channels to verify perfect overlap. The code is vulnerable to data gaps without this QC step.
7. Lastly, instrument response is removed from all four channels then saved to seperate SAC files.

#### Step 1.3: 2-hr Event (used by ATaCR) downloaded, pre-processed, then saved to SAC.

> Following the exact same set of pre-processing steps but for 2-hr event traces, atacr_download_event is called for the chosen event.

### Step 1.4: 24-hr Event (used by HPS) downloaded, pre-processed, then saved to SAC.

> To ensure the exact same pre-processing steps used in ATaCR are always applied to the 24-hr event traces used by HPS, the same script used by ATaCR to download 24-hr noise in step 1.2 (atacr_download_data) is directed to a 24-hr window that covers the complete day the chosen event occured. The SAC outputs are saved to a separate folder.
>
> > [Note] a reasonable alternative would have been to have ATaCR download its events in 24-hr traces by default and trim them at some later step. However, this would have required changing parts of the download, daily-spectra, and event correction scripts and significantly slowing down ATaCR overall. My approach is simpler and gaurantees identical pre-processing between ATaCR/HPS while still being non-invasive to the ATaCR mechanics and run speed.

Stages:

1. Download data
2. ATaCR
   A. Download noise
   B. Download event
   C. Quality-control of noise spectra
   D. Average spectra
   E. Calculate tilt corrective transfer functions
   F. Calculate compliance corrective transfer functions
   G. Correct event
3. HPS
4. Compare results

## Phase Two: ATaCR Correction

Phase Two is implimented in three steps: 2.1: atacr_daily_spectra | Calculation of noise spectra. 2.2: atacr_clean_spectra | Quality control of noise spectra. 2.3: atacr_transfer_functions | Calculate tilt corrective transfer functions

### Step 2.1: atacr_daily_spectra | Calculate noise spectra

> Using atacr_daily_spectra, this step yields the daily spectra for each of the 10 days of downloaded noise.

> For each day, stft's are made of each of the four channels in 2-hr windows with a 30% overlap. Using the log-PSDs for each, windows with an std-norm greater than 1.5 times the day's std are labeled for removal in the next step (2.2). If less than 10 good windows exist, the day is not used.
>
> A QC figure of all fft segments (good and bad) is saved to show this.

# Figure: QC of 2-hr STFT segments (7D.M07A.2012.164.QC)

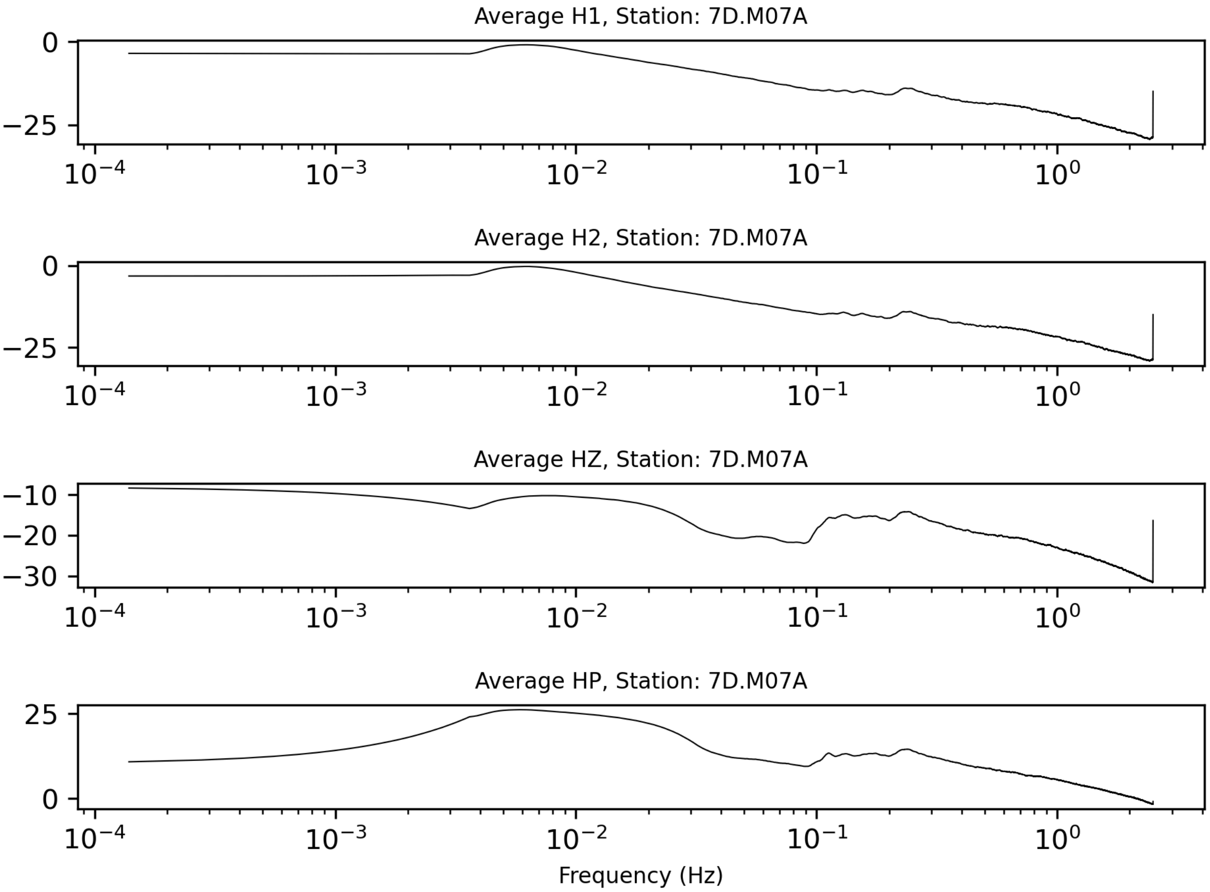
Figure

# Figure: QC of 2-hr STFT segments (7D.M07A.2012.164.QC)

Figure

> Using the good windows, the day's average of power and cross channel spectra are calculated and stored.
>
> The day's average of power and cross channel spectra for Z and P are then calculated with respect to the horizontal channels rotated parallel to the direction of maximum coherence with Z.
>
> Lastly, the day's average power spectra (with bad windows removed) is calculated and saved to a figure output.
>
>> [Note] The day's average coherence/phase/admittance are also plotted but are not shown here for the sake of brevity.
>>
>> [Note] these ZH and PH spectra and the subsequent related transfer functions are only used in the day-average event corrections. This example will use station averaged transfer functions, so these are never used. Nonetheless, the ATaCR code still calculates and stores everything necessary for either (day or station average) event correction approach.

# Figure: A single day's average power spectra (7D.M07A.2012.164.average)

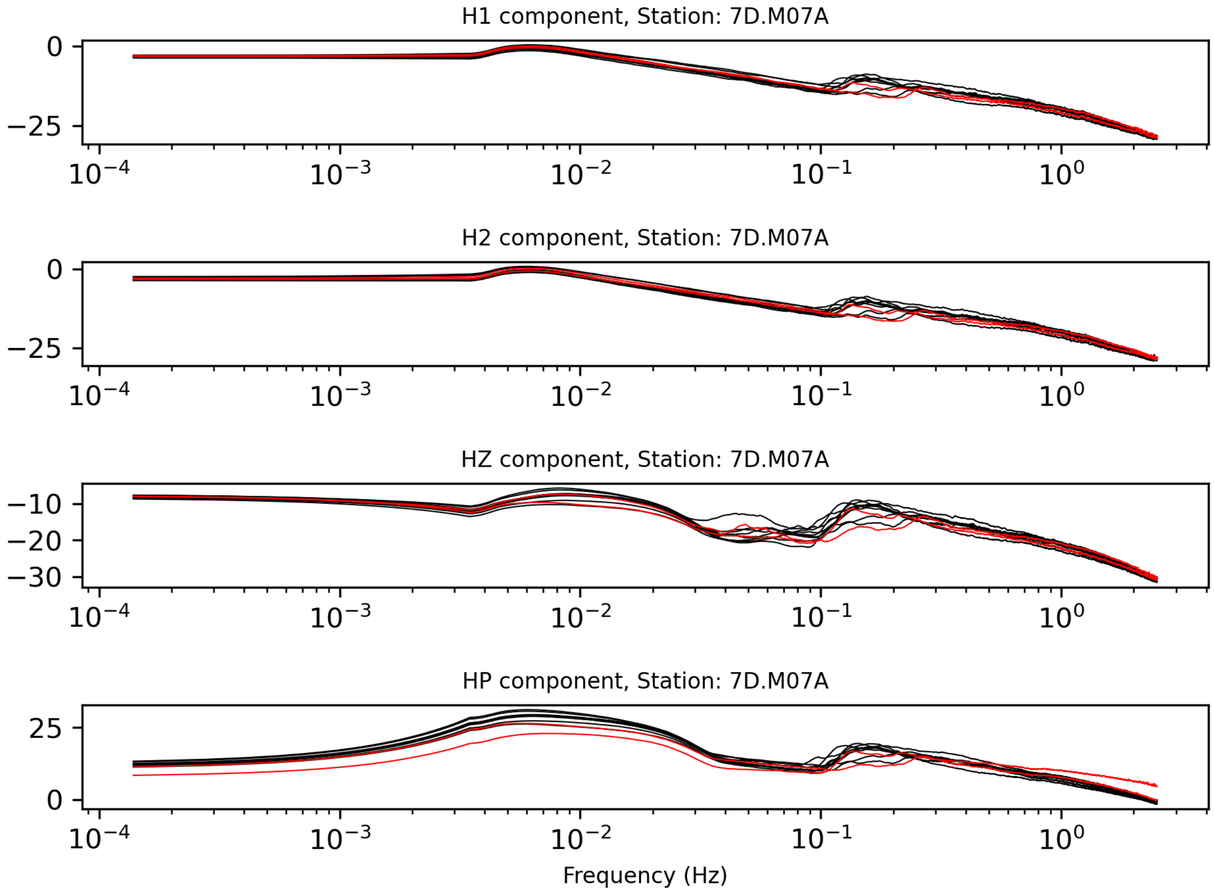


`2.2: atacr_clean_spectra | Quality control of noise spectra.`

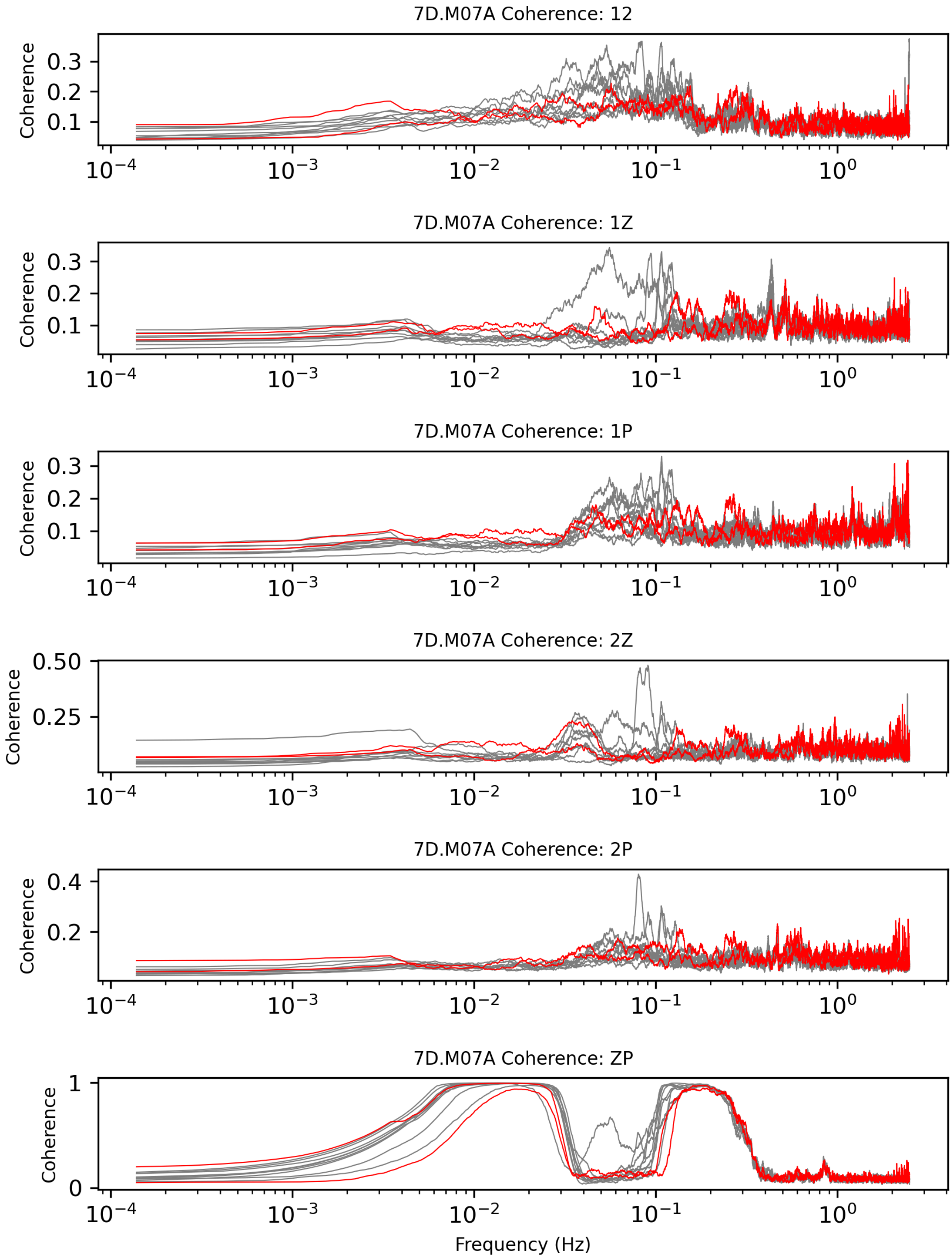
> Using atacr_clean_spectra, the station's average noise spectra are modeled.
>
> First, all 15 days of day-averaged noise for this station are loaded.
>
> Similar to the QC stage in 2.1, days with component PSDs that have an std>1.5x the station average are labeled for removal. Additionally, an f-test is made againstall day-noise variances to test for an equal distribution of power on the PSDs. Days with less than a 95% confidence interval are also labeled for removal.
>
> With the good days defined, the station-averaged power and cross spectra are calculated and stored. As mentioned with 2.1, HH/HZ/HP are also calculated though never used with station-average corrections.
>
> Lastly, figures are made for station's average admittance, phase, and coherence.

# Figure: QC of station daily power specta used to make the station average

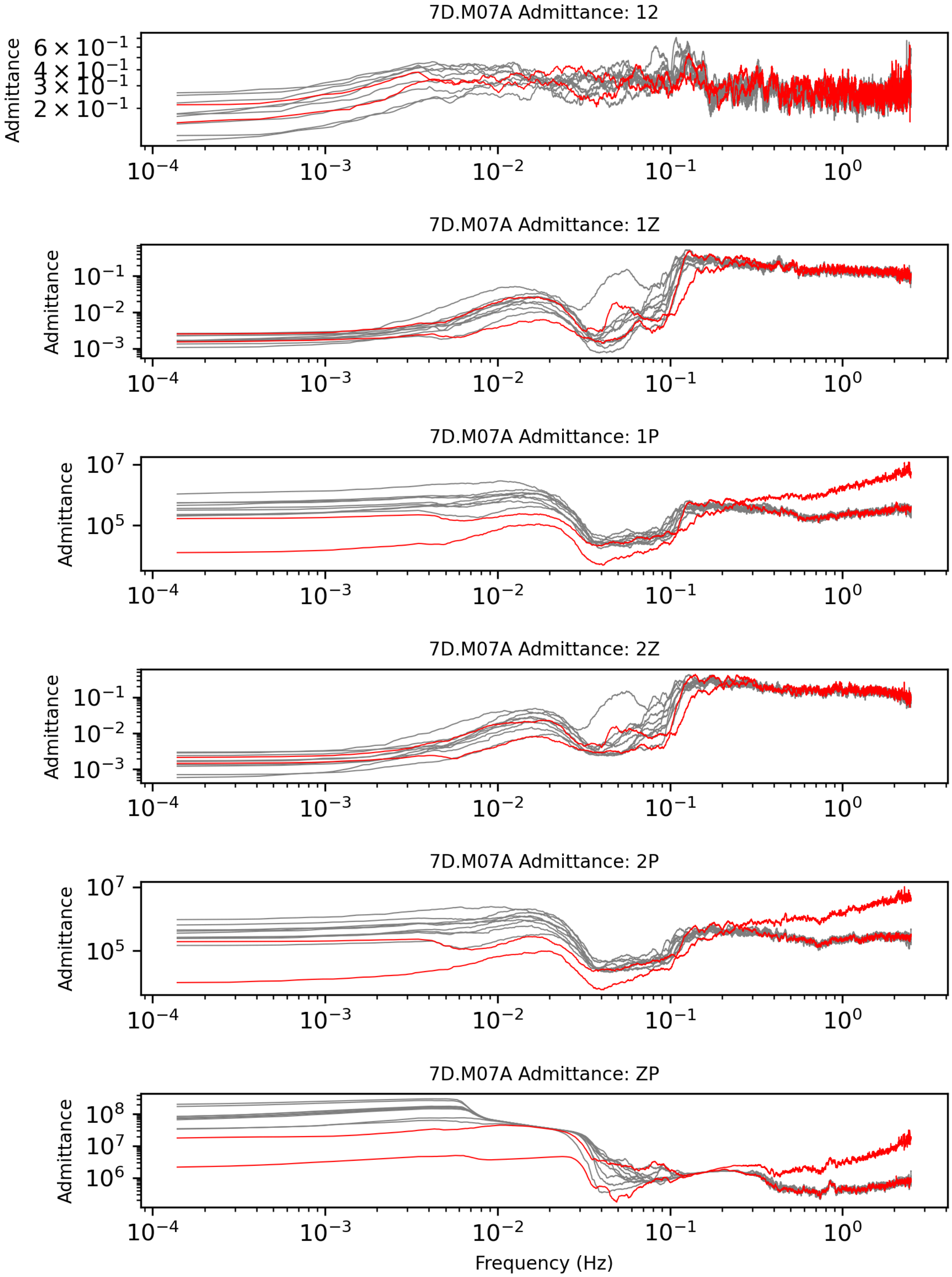
# Red are bad days.



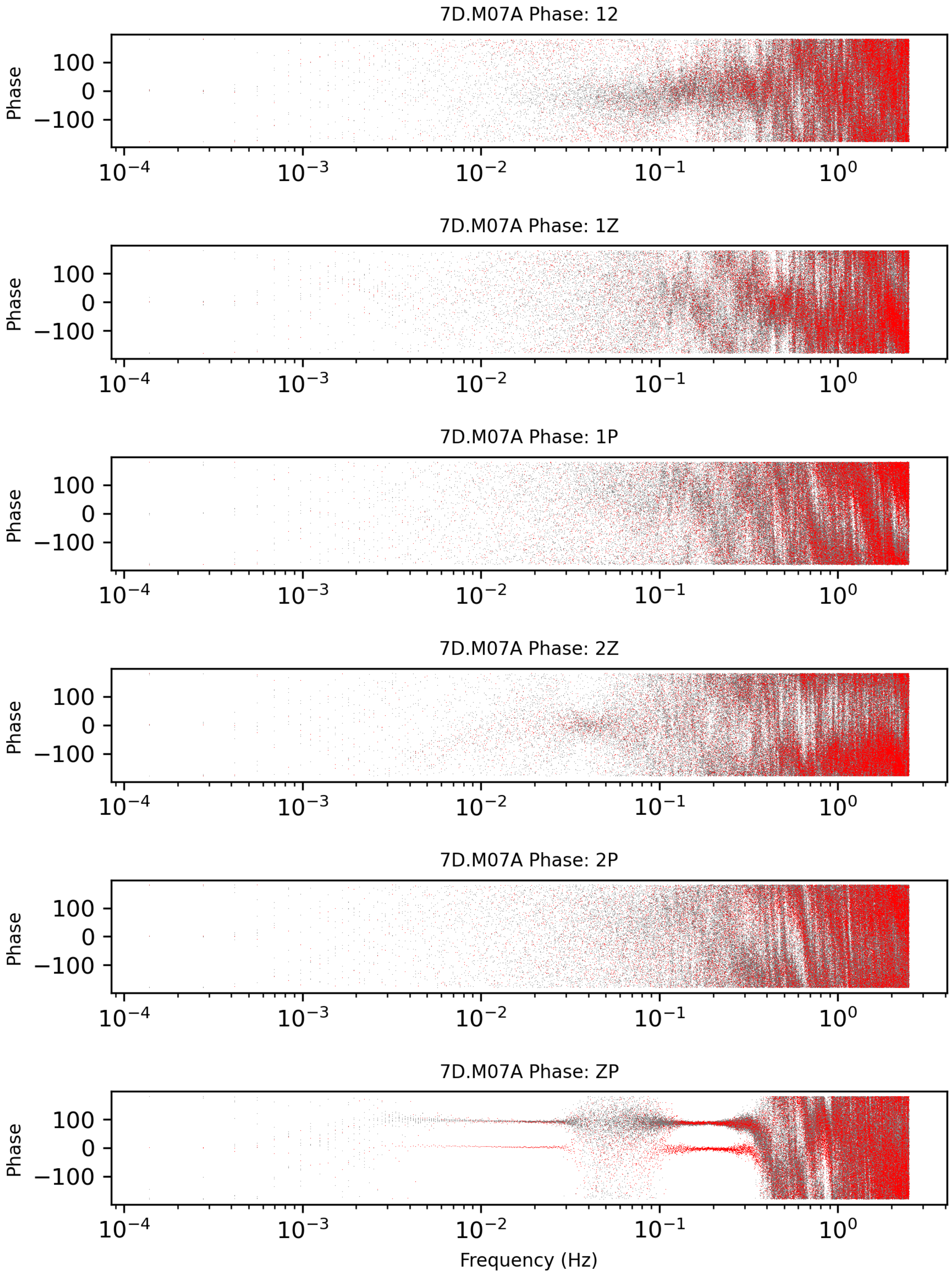
# Figure: Station Averaged Coherence



# Figure: Station Averaged Admittance



# Figure: Station Averaged Phase


7D.M07A Phase: 12

7D.M07A Phase: 1Z

7D.M07A Phase: 1P

7D.M07A Phase: 2Z

7D.M07A Phase: 2P

7D.M07A Phase: ZP

### 2.3: atacr_transfer_functions | Calculate the tilt and compliance corrective transfer functions.

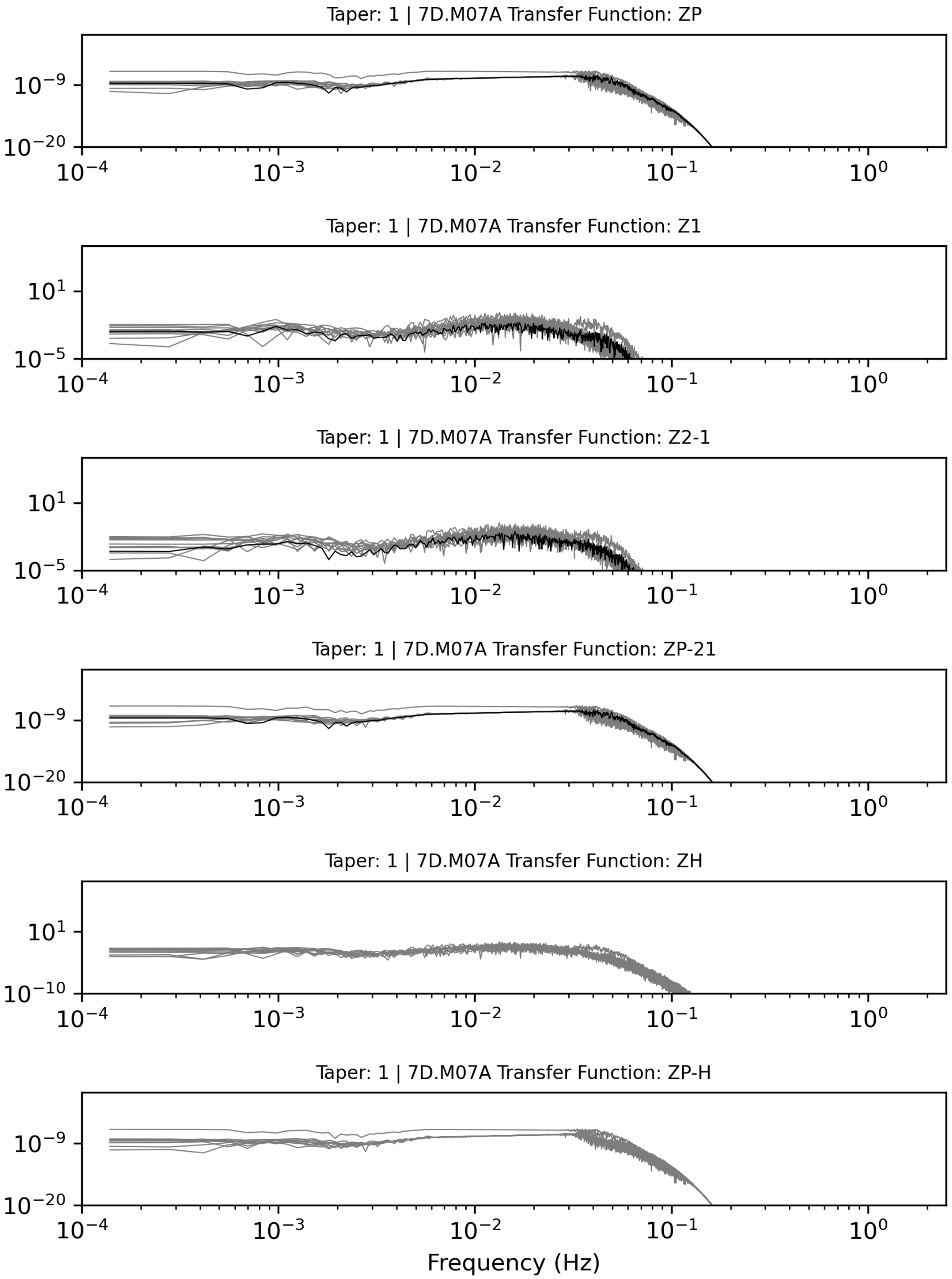
> Using atacr_transfer_functions, we build the library of transfer functions used in the tilt and/or compliance correction of the vertical event trace.

> In total, there are six primary transfer functions modeled by ATaCR: ZP, Z1, ZH, ZP-H, Z2-1, and ZP-21. The last three are sets of multiple transfer functions.
>
> Before calculating each transfer function, the power and cross spectra are tapered to zero for all frequencies above the station-depth dependent f-notch.
>
> For the station-averaged event correction used in this example, the ZP-21 transfer is used.
>
> ZP-21 is a set of six sub-transfer (denoted by a 'TF_') functions:

1. TF_Z1 = c1Z* / c11
2. TF_21 = c12* / c11
3. TF_P1 = c1P* / c11
4. TF_P2-1 = c2P - *(c12/*c11 x c1P)* / c22 x (1-coh12)
5. TF_Z2-1 = c2Z - *(c12/*c11 x c1Z)* / c22 x (1-coh12)
6. TF_ZP-21 = cZP - (TF_P1 x c1Z) - *(TF_P2-1* x x2Z - *(TF_21 x c12)*) / cPP x (1-coh_1P) x (1-coh_c2cP_c1)

> '*' denotes a complex conjugate, 'c' denotes power/cross spectra, 'coh_' denotes a coherence function.
>
> Lastly, the tranfser functions are stored and plotted to figures.
>
> > [Note] Multiple styles of tapering were explored but we ultimately decided to use the same f-notch dependent taper for both compliance and tilt correction transfer functions.
> >
> > [Note] each are calculated at this step for daily and station-averaged spectra with the exception of ZH and ZP-H which are only made for the daily average correction.

# Figure: All Transfer Functions (daily and averaged)



### 2.3: atacr_correct_event | Correct the event for tilt and compliance noise.

> The final step in ATaCR is correcting the event. For this station-averaged correction, we use the ZP-21 transfer function.

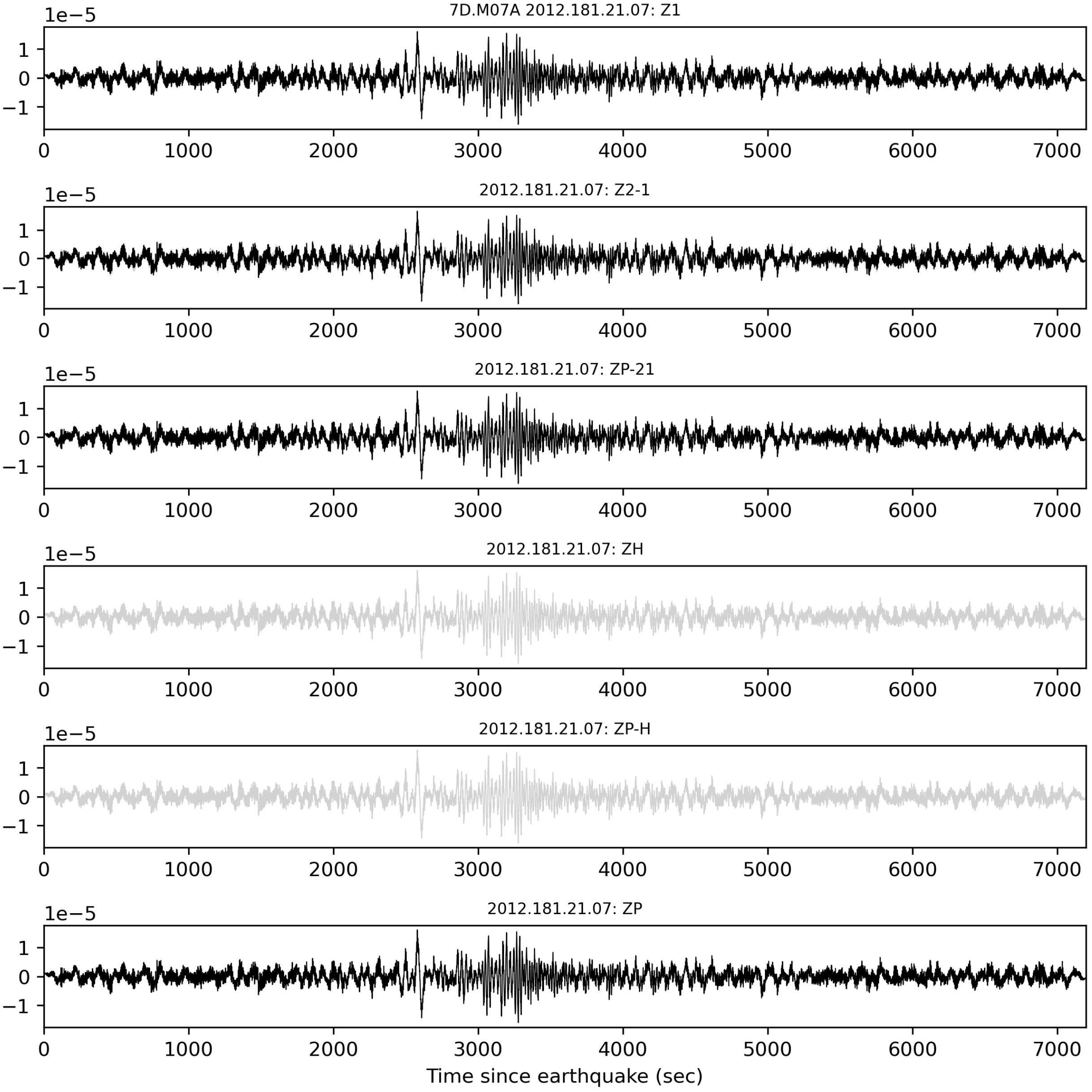
> After calculating the 4-component event spectra using fft (ftZ, ftP, ft1, and ftP),
>
> The ZP-21 transfer function deconvolves tilt and compliance noise from the vertical event spectra (ftZ) as follow:

Corrected = ftZ - TF_Z1*ft1 - (ft2-ft1*TF_21)*TF_Z2_1 - (ftP - ft1*TF_P1 -(ft2 - ft1*TF_21)*TF_P2_1)*TF_ZP_21

> The corrected spectra is returned to the time-domain, plotted, and saved.

# Figure: Corrected event traces plot by ATaCR

## For the four transfer functions available to a station-averaged spectra.



## Phase Three: HPS Correction

HPS is much simpler to impliment compared to ATaCR. After loading the 24-hr event SAC file for the vertical trace downloaded and pre-processed in, run_noisecut pushes the vertical trace to Noisecut.

> Noisecut has multiple parameters that can be adjusted.
>
> To remain consistent with Zali et al. (2023), all parameters defined below are hard-set to their published setup.
>
> Within Noisecut, a power spectrogram is calculated using:

win_length = 163.84 #seconds for the raw spectrogram

hop_length = win_length_sampled // 4 #This corresponds to a 75% overlap.

> All subsequent stages in Noisecut run on partitioned (masked) components of the raw spectrogram.
>
> Using only the spectrogram from 0.1 to 1hz, a cosine match-filter is then ran to produce an inverted mask of recurrent noise. Combined with the 0.1-1hz partitioned spectrogram yields a spectrogram in this band containing only the observed recurrent noises sources, S_background.
>
> Using only the frequency outside of that same 0.1 to 1hz range, the median filter algorithm (hpss) is then used to produce the spectogram model, D_harmonic, containing all observed harmonic noise in this range.
>
> S_background and D_harmonic are then stacked to yield a full band spectogram of ONLY the observed noise.
>
> This noise spectogram is then convolved with its original phase and returned to the time-domain, creating a seismic trace of just the noise.
>
> Finally, this noise trace is subtracted from the raw time-domain trace to yield a noise corrected output.

## Phase Four: Compare Results

> My own library of helper functions and classes are used to make the subsequent analysis much quicker, scalable, and mutable for comparisons of any ATaCR/HPS output.

Comparative_Analysis has been the primary controller for producing the plots and analysis used in the method comparison.

First step in the method comparison for a single station-event is simply to plot the corrected outputs side-by-side.

> Note, when loading the comparisons in for this plot, Comparative_Analysis sends a copy of the ATaCR vertical trace output to run_noisecut to combine the two methods for this third column of traces.

# Figure: A qualitative comparison of the corrected vertical trace when using either or both of the methods

# Figure: Vertical PSD's for each method with the uncorrected trace PSD for reference.

> Next, ZZ, ZP, Z1, and Z2 coherences.
>
> ZZ coherence is done between the corrected and uncorrected traces for each method, yielding three coherences in the subplot.
>
> ZP, Z1, and Z2 coherences are done between the corrected traces and the raw P,1,2 channels. The station averaged ZP, Z1, and Z2 noise coherence is plotted in gray.

.