cleanstrain+ – Simultaneously estimates tides, pressure response, offsets, rate changes, and power-law noise in strainmeter data

SYNOPSIS

cleanstrain+ -s *strain_data* **-m** *1, 2, or 3* [**-p** *pressure_data*] [**-f** *bot, wr, jl*] [**-u** *ns, ms, or cnts*] [**-b** *begin_date*] [**-e** *end_date*] [**-E** *edits_file*] [**-O** *offset_file*] [**-T** *trend_file*] [**-W** *window_length*] [**-R** *name_of_file_of_prior_results*] [**-B**] [**-v**] [**-h**] [**-i** *max_index*] [**-V** *suffix*]

DESCRIPTION

cleanstrain+ is a script that encompasses the FORTRAN program *est_noise6ac* which can simultaneously estimate various time-dependent trends in strain data, the pressure response, the earth tides, and the background noise which is assumed to be a combination of white noise and power-law noise. The results are output as several files:

1 *name_cl_suffix.fmt* strain data with bad data and instrumental caused offsets and trends removed

2 *name_pr_suffix.fmt* with pressure response and all spurious data removed

3 *name_pr_td_suffix.fmt* with pressure, tides and all spurious data removed

4 *name_pr_td_off_suffix.fmt* with pressure, tides, ALL trends, and ALL offsets and All spurious data removed

5 *name_pr_td_off_cl_suffix.fmt* with pressure, tides, instrumentally related trends, and instrumentally related offsets and instrumentally related spurious data removed

where *fmt* specifies the data format. The format is specified in **-f**. In addition, the following plot (Adobe illustrator postscript) files are made.

1 *pl_name_suffix.ai* plots of the original data plus the four sets of processed strain data

2 *psd_name_suffix.ai* plots of the power spectra and the periodogram of *strain_pr_td_off*

3 *pl_out_name_suffix.ai* shows times where there are outliers in the processed data *strain_pr_td_off*.

4 *bay_name_suffix.ai* plots the results from *baytap* analysis captured in output16.dat. The flag **-B** must be specified to run *baytap*. By default, *baytap* analysis is not run.

where *name* is the name of the strainmeter data and *suffix* is the identifier given with the **-V** option. Finally, the results of the analysis is summarized in plain text file called

- *name_suffix_results*

All of the calculations are made in a directory called *name_scr* located under the current, working directory. Upon finishing, the results are passed up to the current, working directory.

A SHORT INTRODUCTION

Observed strainmeter data is comprised of a combination of the earth tides, an atmospheric pressure response, and other signals that may be due to regional tectonics (long wavelength deformation greater than 1 km), deformation near the borehole, deformation of the borehole, deformation next to the instrument, electronic noise, and etc. The reality

is that it will be hard or impossible to distinguish between all of these signals. However, it is possible to successfully and reliably obtain the O1 and M2 tidal components and the atmospheric pressure response. Key to obtaining these three parameters is to model all of the other time-dependent terms whether they be due to electronics glitches or deformation near or far from the borehole. This script allows one to do all of this modeling. It also provides adjusted strainmeter data that have truly instrumental problems removed and a second set that has all of the model-able, time-dependence removed; those residuals should be close to that of a random-walk process.

Key to getting reliable results is to recognize that the spurious data, various trends and offsets are either due to known, instrumental causes, such as a site visit, or due to other causes. In tabulating files that make-up the *edits*, the *offsets*, and/or the *trend* files, the user will be required to make a judgment with each entry (or record) for each of the three files. If the entry is tagged with **N**, then that entry is assumed to have a, non-tectonic source such as an instrument failure or telemetry glitch. On the other hand, if the entry is tagged with **T**, then the entry is assumed to have an unknown cause but could be tectonic in nature. Unless each site has a carefully kept log of visits and maintenance, then, some of the signals that one might judge to be due to a problem in the instrument might actually be due to some other cause within or near the borehole.

When this script does the analysis, it deletes all time periods found in the *edits* files, estimates the sizes of all offsets for times found in the *offset* file, estimates all of the trends specified specified in the *trend* file, the earth tide constituents, the atmospheric pressure response, and the underlying noise model. On output, though, the script recognizes the differences between **T** and **N**; Those output files identified with *_cl* have only the instrumentally related adjustments, **N**, while the remaining files have had all, both **N** and **T** adjustments made.

ARGUMENTS AND OPTIONS

The following are required:

-m *1, 2, or 3*

For both **1** and **2**, a time-dependent model, and noise amplitudes are estimated. The time-dependent model can be a combination of a rate, rate changes, offsets, the earth tide, the pressure response, and exponentials. With *1*, *est_noise6ac* does everything, but this is very CPU intensive. If there are not too many gaps in the data, option *2* should give similar results but with less CPU effort. With option *2*, a nominal noise model is input to *est_noise6ac*. *est_noise6ac* is instructed only to estimate the time-dependent model only. Then, with the residual time series, a power spectra is estimated and, using least squares, a power-law plus white noise model is fit to the spectral estimates. This updated noise model is then input back into *est_noise6ac* which then re-estimates the time-dependent model using a better noise model (or data covariance matrix). For option *3*, the script uses results of prior analysis to remove the pressure and tidal periods from the data, and estimate offsets using the specified noise model. The results of the prior analysis (pressure response, tides, and noise model) were captured in *name_results*. This file must be identified as an input using **-R**

-s *strain_file*

the file of strainmeter data; the format description is described under **-f**. The default

format is the USGS *real bottles*. It is best that the file be named using the convention *name.xxx*. The script is designed to get the *name* for use in naming other files.

The following are optional. These are order relative to their importance.

-p *pressure_file*

Name of the pressure file data. This file must have the same format as the strainmeter data. (Comment; The script has not been debugged to determine whether everything works if the pressure data is missing).

-f *bot, wr, or jl*

Specifies the format of the data with USGS *real bottles (bot)* being the default. Currently, *integer* bottles will not work for now. Format *wr* consists of data with three columns; the first two being the time and the third being the observation. If there is no observation at specified time, the missing data symbol, *999999.0* is recorded as the datum. The time stamp (two columns) consist of *02/14/2007 19:10:00*, where the first column specifies month, day, and year, and the second column specifies hour (24 hour), minute, and seconds. The seconds information is ignored. Format *jl* consists of 4 columns for each observation. If there is a missing observation, it is simply not tabulated. A record of data looks like *2006 238.006944444 -459.75601 0.1*, where the first column is the year, the second column is the day of the year (1 January being Doy 1), the third column is the observation, and the forth column is ignored but needs to be present. This is the native format required to run *est_noise6ac* and is used throughout the script. On output from this script, however, the results, ie, the *strain* files, are converted back to the original format.

-u *cnts, ns, or ms*

Specifies the units of the strain data where *ns*, nanostrain, is default. If the input strain data is in microstrain, the *ms* should be used. This will cause the script to multiply the observations by *1000* to convert to nanostrain. The program *est_noise6ac* likes to have data that has a white noise value greater than 0.3 (or so) and the power-law amplitude greater than unity. This is because most of the format specifications in both *est_noise6ac* and *baytap* only give two units to the right of the decimal point. For units of digital counts, *cnts*, the multiplier is *1*. At this time, I have only tested this script on USGS strain data using 16-bits.

-V *suffix*

A text suffix that can help identify the segment of data that was analyzed. Use text with no spaces between letters or numbers.

-b *start_date*

Starting date Normally, the script will work on the entire data set. However, it might be advantageous to examine part of the data and that time period is specified with the **-b** and/or the **-e** options (start and stop times). The format of the date is either *year, month, day, hour, minute, 198702141930* or *year,* **j** *day of year, hour, minute 1987j0451930*. All digits must be used or else the script gracefully stops and gives an error message. For most strain data, it is best to provide between 30 to 180 days of data for analysis. If the original strain data consists of many years of

10-minute data, it is best to parse these data prior into shorter time periods prior to running this script as this script is not very efficient parsing the data.

-e *end_date*

End date; see **-b**.

-E *edits_file*

Name of the file containing the time periods of *spurious* data. This file has three columns for each interval of spurious data. Column 1 is either *N* or *T* which identifies whether the so-called spurious data could be *n*on-tectonic (instrumentally caused) or *t*ectonic. The next two columns are the time period of the spurious data using either the *year, month, day, hour minute* or the *year,* **j** *Doy, hour, minute* formats described in **-b**. The file can have a mix of formats. It is suggested that the time specification include a buffer time. For instance, if the data are sampled at 10-minute intervals, then specify the start time as 5 minutes prior to the spurious data and 5 minutes after the spurious observations have ceased. The flags in column 1, *N* and *T*, determine whether the so-called spurious observations are deleted from the output *strain_cl_fmt* file. For instance, obvious telemetry glitches should be identified as *N* and these are removed from the all of the output strain files including *strain_cl_fmt*. On the other hand, seismic waves might be evident in the data and their presence can bias the estimates of the noise model, tides, and pressure response. Consequently, by identifying these observations at *T*, they are removed from most of the strain files, but will be present in *strain_cl_fmt* file. Since the script only works on edit entries with *N* or *T*, it is convenient to make the value of the first column *S* (or some other letter) meaning that this entry is *s*kipped (and the other 3 columns are appended after the letter *S*). Additional information can be added after the last date to provide some documentation as to why the time-period is spurious. It is not always obvious whether the data are spurious due to either tectonic or non-tectonic causes; this requires individual judgment.

-O *offset_file*

Name of the file containing the times of offsets in the data. This file has two columns for each offset; the first is either *N* or *T* which identify whether the offset is *non-tectonic* (instrumentally caused) or *tectonic*. For Sacks strainmeters, they will have large offsets which are instrumental in cause and should identified as *N*. Offsets from earthquakes should be identified by *T*. Format of the date is the same as the edits file. It is suggested that the specified time be between the time of the actual offset and the time of the data immediately sampled after the offset. The effect of *N* or *T* is the same as with the edits file. Likewise, putting an *S* in the first column causes the script to skip that offset. If the date falls outside of the time interval of the data, the offset is ignored. Documentation to the cause of the offset can be added after the time stamp.

-T *trend_file*

Name of the file containing the type and times of time-dependent trends that are used to model the data. There are three types of trends; a rate change, an exponential or a logarithm. Each trend type is specified as a single line in the trend file. The first column, as described in **-E** is either *N* or *T*; an *S* causes the script to ignore the entry. The second column specifies the type of trend; *r* is a rate change, *e* is an

exponential, $A(1 - \exp(-t/\tau))$, and *m* is a logarithm, $A \log(1 + t/\tau)$. In all three cases, the amplitude is estimated. In most cases, the first column will be specified as *T* since the removal of these curves should reduce the long-period noise in the data which can affect the estimate of the pressure response. But, for the *strain_cl_fmt* file, these trends will remain.

For rate change, *r*, the time-interval of the rate change is specified in columns three and four using the same date format described in **-E**. If the last date falls beyond the end of the time-series, then, the script will internally set that date to be the ending time of the time series. If the first date falls before the beginning of the time series, the rate change is ignored.

For either exponential, *e*, or logarithm, *m*, the format of the input is the same for both, with the third column specifying the onset time of the function, $t = 0$, the forth column specifying τ, and the fifth column is either *float* or *fix* which specifies whether τ should be estimated (*float*) or not (*fix*). If the onset time is not within the interval of data under analysis, the entry in the trend file is skipped.

–R *results_file*
Required as an input when **-m** is *3*. Normally, this file is generated from a previous run of *cleanstrain+* using another time-period.

-B Does *baytap* analysis. The summary of that analysis is placed in the *name_results* file and an additional plot is made called *bay.pdf*. By default, *baytap* analysis is not run.

-i *maximum power law index*
The default is 2.4. In fitting a power law to the estimates of the power spectrum (**-m** *2*), the following is fit to the frequency dependent part of the spectrum; $P(f) = P_o/f^n$. In many cases, n might get too large and cause the estimates of various long-period trends to come close to a singularity. To prevent this, the maximum value of n can be restricted. Sometimes, to get visually good *fits* to the data, restricting n to 2.0, or random-walk, is required.

–W *window_length*
Window length in days to used to estimate the size of the offsets. By default, the script calculates the window length by determining the period in the power spectra where the power-law noise is equivalent to the white noise. That period is then multiplied by 5. However, the minimum window length is 0.33 days. This option allows the user to specify the window length needed to estimate the size of the offsets.

-h Outputs help and exits.

-v Outputs the version number and exits.

THE ALGORITHM

As background, the maximum likelihood method is employed to simultaneously estimate trends in the data and the background noise. This has been written as a FORTRAN

program called *est_noise6ac* and is discussed in several papers including:

Langbein and Johnson, *JGR 102*, 591-604, 1997

Langbein, *JGR 109*, doi:10.1029/2003JB002819, 2004

Williams *et al. JGR 109*, doi:1029/2003JB002741, 2004

Langbein et al., BSSA 96, doi:10.1785/0120050823, 2006

In addition, *est_noise6ac* had been modified to allow pressure as another input.

Many of the FORTRAN programs included in this package of scripts include subroutines from *Numerical Recipes; Press et al.*; **It is assumed that users have a license to use these subroutines.**

The compilation of *est_noise6ac* requires the *lapack* library. Nominally, this can be downloaded from netlib.org, but it may be already on your computer system. For the MacIntosh, these are included in the Apple Developer kit (freely downloadable). On my LINUX computer, I use Intel FORTRAN version 7 along with the Math Kernal Library (MKL) version 6. The Portland FORTRAN compiler has a similar distribution of *lapack*.

One may consider the *est_noise6ac* to be standard, least squares curve-fitting, which it is. In most implementations of least squares, it is assumed that the data are independent or their error model is Gaussian, white noise. But, power spectra of strainmeter data demonstrate that the error spectra is *red*, or that of a power law. So, unlike a data covariance (or data-weighting matrix) of diagonal elements commonly implemented in least squares, *est_noise6ac* uses a covariance matrix that incorporates the power-law dependence seen from the power spectra of strain data. One may use the results of fitting a power-law relation to the power spectrum and, through equations 11, 9, and 10 in *Langbein* (2004), construct the covariance directly. Or, one can let *est_noise6ac* estimate the parameters of the covariance; both of these options are offered in this script with the curve-fitting to the power spectrum being more efficient.

For all **-m** options, the following are done:

- Create a directory under the calling directory for the work to be done; change into the "scratch", (_scr), directory.
- Convert the data from its original format to the *jl* format
- Rescale the data. As described above, *est_noise6ac* (and *baytap*) work with formated data where there are only two digits present to the right of the decimal point.
- Remove data specified in the edits file. This uses a program called *getdata*.
- Remove data specified in the edits file identified as being "non-tectonic" or instrumental (N). This uses a program called *getdata*. Store results in strain_cl.dat.
- Collect the times of the offsets, rate changes, and exponential or logarithms specified in the offset or trend files.

For option 1 (-m *1*):

1 Data are decimated to one-hour samples. If after decimation there are more than 2000 observations, the original data are decimated such that the number of samples is less than 2000 after decimation. Decimation is required since *est_noise6ac* inverts the data covariance matrix many times in its search for the optimal noise parameters; the CPU required for inversion probably scales as N^3 where N is the number of observations. (Note, I may change 2000 to a smaller number for better efficiency; for tidal analysis, 3-hour samples should suffice). The program *decimate_2* does that decimation; option 3 is used in *decimate_2* which re-samples the

data at the specified interval; option 1 and 2 uses either running averages or medians to smooth the data prior to decimation (these are not used in the script).

2 Determine the tidal periods to fit to the data. This selection is based on the length (time) of the data set; a longer period of observations allows more tidal periods to be modeled. I use the program *tideline* where I lifted a portion of Duncan Agnew's program *tidhar* that determines the appropriate tidal periods to model. (The program *baytap* also does something similar, but gets slightly different tidal constituents; I don't know *baytap*'s algorithm).

3 Gather all of the information needed to run *est_noise6ac*. This includes rate, the times of rate changes, offsets, and exponentials; the pressure data, the strain data and an initial guess at the power law plus white noise model.

4 Run *est_noise6ac*. The downhill simplex method, as implemented in *Numerical Recipes* is used to find the optimal noise model and, if required, the time constants in the exponential functions used to model the time dependence.

5 Gather the output from *est_noise6ac*. This includes the parameters of curve-fitting and the noise model.

6 Convert the amplitude of the power-law noise estimated from the decimated data to an amplitude compatible with full sampling using equation 11 from *Langbein* (2004).

7 With the undecimated data, remove the pressure response using the pressure term determined by *est_noise6ac*. The removal is done with *diff_data_2*. Results stored strain_pr.dat

8 Remove the tidal model from strain_pr.dat using the program *off_rate_exp_gps_2*. Results are stored in strain_pr_td.dat

9 Remove the rates, rate changes, and exponential using *off_rate_exp_gps_2*; Results stored in strain_pr_td_off_1.dat.

10 Remove the offsets using *off_rate_exp_gps_2*; Results stored in strain_pr_td_off.dat.

11 From strain_pr_td_off.dat, take the first differences, $d_i - d_{i-1}$; Sort the differences and determine the range that includes 67% of the differences; divide by $2\sqrt{2}$ and use this value as the white noise amplitude; recall that the prior estimate of white noise by *est_noise6ac* was done with decimated data which could obscure the white noise.

12 Determine the period where the level of the power-law noise is equal to the white noise level in the power spectrum. Knowing this period will assist with precisely estimating the size of the offsets. Here, if there is indeed a white-noise component in the data, I want to make sure that I use enough data before and after the time of the offset for averaging; if, on the other hand, the white noise is much less than power law at the highest frequencies, the offset can in principal be estimated by differencing the two observations spanning the offset. Once the period of the cross-over point is determined, I arbitrarily multiply the cross-over period by 5 to set the window length used to precisely estimate the offsets. If the window length is still less than 0.33 days, I force the window length to be a third of a day. Testing indicates that the window length does not affect the estimate of the size the offset (or its standard error) since the proper data covariance will correctly "weight" the data.

13 At the time of each offset, use a short period of adjusted data in strain_pr_td_off_1.dat. The length of time used is twice the window length

determined in step 12 and centered on the time of the offset. If there are missing data before or after the offset, the window length is extended such that number of data equal the window length. If there are more offsets with that time period, the number of data are expanded to include the additional offsets.

14 For each offset, run *est_noise6ac*. The fully sampled data with all adjustments are used; in addition, the power law plus white noise model are used as input parameters; the program does not solve for the noise model, here. Tabulate the size of the offsets (and their standard errors). If the offset is less than twice the standard error of the offset, that offset is neglected in further adjustments.

15 Remove the offsets using *off_rate_exp_gps_2* from strain_pr_td_off_1.dat. Results stored in strain_pr_td_off_1.dat.

16 Calculate the periodogram of the fully adjusted data, strain_pr_td_off_1.dat, using the program *periodogram1*. This incorporates the *Numerical Recipes* subroutine *fasper* which implements the *Lomb* normalized periodogram for unevenly sampled data (data with gaps).

17 Fill in data gaps in strain_pr_td_off_1.dat using program *interpolgps_1*. Gaps are filled with white noise.

18 Compute the power spectra of the interpolated data using *powerJL*. This is Duncan Agnew's program *power* but modified to allow ASCII data as input. The window length for spectral smoothing is taken to by 1/5 of the time period of the data.

19 In the fully adjusted data, strain_pr_td_off_1.dat, look for gross outliers by taking the first differences and compare those differences with the expected wander for the interval of time between the observations. Tabulate the significant outliers.

20 Do *baytap* analysis if requested; tabulate the results. The script that does the analysis is *dobaytap+*

Once these steps are completed, the following are done for all options *1, 2,* and *3*.

- Create a *clean* strain data set. Tabulate the rate changes, exponentials, offsets, and outliers that are "non-tectonic" or instrumental (N). Apply those adjustments to strain_cl.dat using *off_rate_exp_gps_2*.
- Plot outliers (*Plot_outlier+*), the results from the steps in this analysis (*Plot_Strain+*), and the power spectrum and periodograms (*Plot_Spect+*). Results (pdf files) are placed in the calling directory
- Rescale the data to their original units.
- Convert the processed strain data back to their original format; these are placed in the calling directory. The script *jl2wr_bot+* does this conversion as times of missing data are identified and filled with the missing data symbol (999999). The program *jlmiss* does the identification.

For option 2 (-m *2*):

1-3 Run steps 1 through 3 from above.

4 At step 4, use white noise 1, power law index of 2, and power law amplitude of 200; these are "fixed" (not estimated). Run *est_noise6ac*.

5 Gather the results of estimating the time dependent model from the output of *est_noise6ac*.

6-18 Run steps 6 though 11 from above.

19 From the power spectra estimated from *powerJL* and the white noise determined by first differencing the adjusted data, least squares fit the power-law portion to the spectrum using the script *psd+* ($P(f) = P_o/f^n$). The estimates of the three longest periods are neglected. To give the longer periods more "weight" than the short periods, the spectral estimates are weighted according to the square-root of their period. Convert the power law amplitude to the appropriate values using equation 11 in *Langbein* (2004). If the power law index exceeds the prescribed limit (default is 2.4), repeat the least-square fitting of the power-law model to the spectral estimates but constraining the value of the index.

19.5 Use the new value of the power-law noise and, if this is the first past through this step, jump back to step 4; otherwise go to step 20.

20 Do *baytap* analysis if requested; tabulate the results. The script that does the analysis is *dobaytap+*

For option 3 (-m *3):*

1 Gather tidal model, pressure response, and the parameters of a power-law and white noise model from previous runs of other time periods of the time-series.

2 Adjust the data for pressure and tidal model per steps 7 and 8.

3 Remove offsets using steps 12 through 15.

4 Continue with steps 16 through 19.

RELATED SCRIPTS

glitch+ is useful to identify times of gross outliers (data that are truly beyond the expected observations or are *clipped*, smaller, spurious glitches, and offsets.

AUTHOR

John Langbein
US Geological Survey,
Menlo Park, CA

May 2007