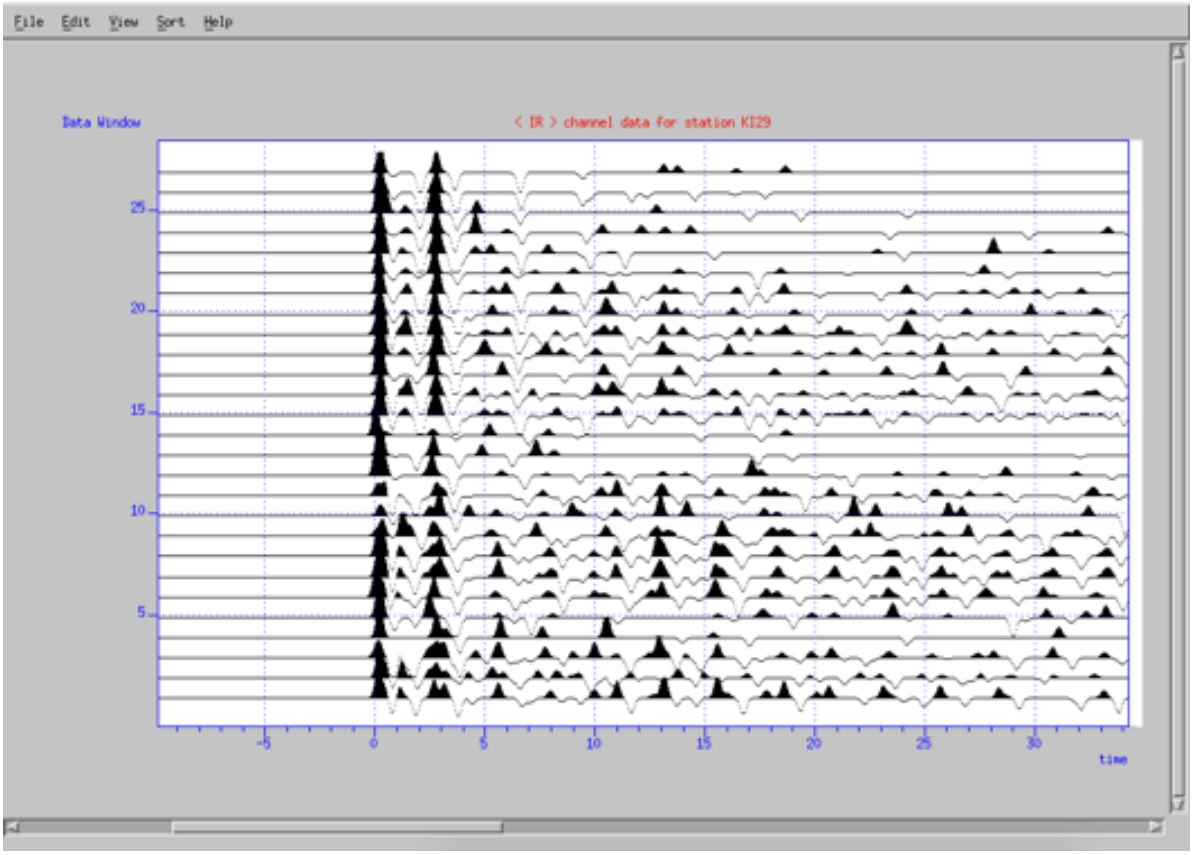
# RFeditor

User Guide

v3.7



Xiaotao Yang[*], Gary Pavlis, Yinzhi Wang

Indiana University

July 2017

[*] Now at University of Massachusetts Amherst, xiaotaoyang@umass.edu.

# Table of Content

# 1. Introduction

In this file, we use the symbol ‘`$command`’ to denote terminal inputs. We use $ANTELOPE to denote for the Antelope root path on your computer. We refer the path where you store the RFeditor source package to be RFEROOT.

## 1.1 Download

The package is available at: https://github.com/xtyangpsp/RFeditor. Assumming you have downloaded the package and saved it to: /Users/myhome/SOFT/src/RFeditor, this path is referred to as `RFEROOT` in this user guide. Under this directory, there are four subfolders:

(1) libtreditoperator

This directory contains the lib needed by RFeditor editing procedures. However, the library files are currently embedded in RFeditor core codes. Thus, compiling RFeditor does not require the compiling of this library. This library can be used by other utilities or extensional programs.

(2) RFeditor_core

This folder contains the RFeditor core source code.

(3) Utilities

This folder includes the utilities related to the use of *RFeditor*.

(4) Docs

This folder contains this user guide and other related materials.

## 1.2 Citation

Reference for the method: Yang, X. T., G. L. Pavlis, and Y. Wang (2016), A quality control method for teleseismic P-wave receiver functions, *Bull. Seismol. Soc. Am.*, *106*(5), 1948–1962, doi:10.1785/0120150347.

## 1.3 Support

Support is provided through GitHub platform. You can create an issue or ask questions at: https://github.com/xtyangpsp/RFeditor. Or you can send emails regarding questions to: xtyang@indiana.edu.

# 2. Installation

This program operates based on an Antelope Datascope database (version 5.5 and later). Go to: http://www.antelopeusersgroup.org to install `contrib` following the procedures there, if it is not installed along with the Antelope package. Please make sure that Antelope is working properly before installing this program. If you downloaded and installed the `contrib` package through `git` repository, you can update it, under your `$ANTELOPE/contrib`, by typing in terminal:

```
$ git pull
```

This will update your `contrib` package. After it finishes, recompile the updated programs. In Antelope 5.7, the `contrib` package is automatically downloaded into `/opt/antelope/src`. You can then copy or create a soft link of the `src` folder to `$ANTELOPE/`.

## 2.1 Operation Systems

This program has been fully tested under Mac OSX Mountain Lion (10.8), Mavericks (10.9), and Yosemite (10.10). Newer Mac OS should be fine but without heavy tests. For other linux-based systems, it should be working but please report any issues at: https://github.com/xtyangpsp/RFeditor.

## 2.2 Library dependency

The following libraries are required in order to compile RFeditor (current version 3.x). You may need to install these libraries by the order as shown here:

(1) boost (http://www.boost.org):

This is a C++ library. On Mac OSX, you can install it from fink. From our experience, please install the version tagged as “nophython”.

```
$fink list boost
$fink install boostpackagename
```

(2) xmotif: xwindow libraries;

On Mac OSX, you can install motif from fink. In terminal, type:

```
$fink list motif
```

Then, type:

```
$fink install motifpackagename
```

Or, you can compile the source code by yourself. This package/source code could be downloaded from: http://motif.ics.com and the link there.

<<< Configure Antelope localmake >>

After successfully installed boost and xmotif libraries, you have to enable boost and xmotif for Antelope. This can be done by running in any terminal:

```
$localmake_config
```

This will lead you to the interface where you can enable BOOST and XMOTIF capabilities. You will need to type the paths for LIB and INCLUDE for both BOOST and XMOTIF libraries and then: "File->Save and Quit" to save these configurations.

(3) libgclgrid (`$ANTELOPE/src/lib/utility/libgclgrid`):

This library should be installed automatically if you successfully compiled and installed the contrib package under `$ANTELOPE/src.` If not, cd to this directory and type:

```
$make Include
$make
```

(4) libseispp (`$ANTELOPE/src/lib/seismic/libseispp`):

This library is released along with the Antelope contrib package. Once you successfully compiled contrib software, libseispp should be already available. If not, cd to this directory and type:

```
$make Include
$make
```

(5) libseisw (`$ANTELOPE/src/lib/graphics/seisw`):

Seismic widget library for plotting seismic traces as wiggles. It is part of the Antelope contrib package. Once you successfully compiled contrib software, libseisw should be already available. If not, cd to this directory and type:

```
$make Include
$make
```

## 2.3 Install *RFeditor*

Once you have all of the required libraries installed and properly configured, cd to `$RFEROOT and` type in command line:

```
$make
```

If the above compiling procedure is successful, run:

```
$make install
```

This deposit tredit and decon tables to $ANTELOPE/contrib/data/css3.0.ext/, deposit RFeditor executable to $ANTELOPE/contrib/bin, and deposity RFeditor.pf to $ANTELOPE/contrib/data/pf/. Please make sure you have the permission to write, read, and execute programs!

If you the above compiling went through, congratulations, *`RFeditor`* is available to you.

<<< Trouble Shooting >>

If you get errors complaining that some libraries are not found, you may need to find the library and copy it (or create a symbolic link) to: `$ANTELOPE/lib`. For example, `lseispp` means library file `libseispp.a`.

## 3. Command Line Options

You can start RFeditor from the terminal by simply typing the program name: RFeditor. This will give you a brief usage information. For example:

```
$RFeditor
< version v3.7 > 5/4/2016
RFeditor dbin dbout [-d outdir][-tredit filename][-rm][-go][-
continue]
[-fa fa_filename][-pf pffile][-laststa xx][-ss
subset_condition][-v|V][-h|H]
** Use -h to print out detailed explanations on the options.
```

However, to start the program on an Antelope database, it requires two arguments: the input database and the output database, where the edited data will be stored.

-d outdir: directory saving the output data;

-tredit filename: the editing summary will be saved into a plain text file with the name of filename. By default (without this option), the program only saves edit summary to Antelope table “tredit”;

-rm: turn on review mode. Under review mode, edits will NOT be saved. In this case, type dash (-) as the outdb name;

-go: GUI-off mode. This is designed for automated QC of large dataset;

-continue: turn on continue mode. When this option is used, the program automatically continues onto the next station after the last station in the existing outdb. The program checks tredit table for the last working station;

-fa fa_filename: the program will save first arrival (first peak) information into the plain text file specified by fa_filename;

-pf pffile: parameter file. By default, the pffile is: $PFPATH/RFeditor.pf in the current working directory. $PFPATH is specified by the Antelope enrironment.

-laststa xx: jump to the station after xx;

-ss subset_condition: apply subset condition to the database before editing;

-v|V: run program in verbose mode;

-h|H: get help information on command options and version history.

*RFeditor* has two running modes: interactive mode with Graphical User Interface (GUI-mode) and automated mode (Auto-mode). GUI-mode allows the user to interactively work on each station gather, select quality control procedures and customize parameters for each metric. For a large dataset, we recommend this mode for testing parameters before running in Auto-mode.

By default, the program starts in GUI-mode. To run the program in Auto-mode, use option: `--gui-off` or `-go`.

# 4. Parameter File

The following are main parameters driving the program (see RFeditor.pf in RFEROOT for example parameter values).

## 4.1 Global parameters driving the program

`minimum_number_receiver_functions`

Stations with number of events less than this will be skipped automatically with a log message.

`FA_reference_time`

First arrival time shift for plot (default displays First Arrival at time 0). Signs: +, shift to positive time axis; -, shift left to negative time axis. When `use_arrival_data` is set to false, this is the FA time in the data and the trace will be plotted starting from 0 seconds.

`use_arrival_data`

This is a Boolean parameter (true/false). If this is true, the arrival and assoc tables must be provided and will be used in converting the time frame from absolute to relative.

`use_decon_in_editing`

This is a Boolean parameter (true/false). If this is true, the decon table must be provided. This table is generated by the program associated with the Generalized Iterative Deconvolution method by Wang and Pavlis (2016). When this parameter is set to true, the QC process will enable deconvolution attributes related procedures.

`use_netmag_table` true

This is a Boolean parameter (true/false). If this is true, the netmag table must be provided. The program will read in magnitude information from this table. Currently, magnitude information can be used in sorting the receiver functions (a station gather).

`radial_channel_key`
`transverse_channel_key`
`vertical_channel_key`

The above three parameters are channel codes for the three component of the receiver function.

`no_vertical_data`

This is a Boolean parameter (true/false). If this is true, the program will only process radial and transverse data. If your data includes vertical data, this parameter is not required. The default value is false. The program only tries to read this parameter after it fails to read the `vertical_channel_key` from the parameter file.

`edit_on_channel`

This parameter (radial, transverse, or vertical) gives the option to choose which component to edit.

`use_wfdisc_in`

This is a Boolean parameter (true/false). If this is true, the program will read waveform data following wfdisc table. Otherwise, wfprocess table is used.

`apply_prefilter`

This is a Boolean parameter (true/false). If this is true, the receiver function data will be filtered before processing and plotting (in GUI-mode). We recommend turn the prefilter on for pure-spike RF data. Otherwise, QC processing will not be working correctly.

`wavelet_type`

Available types: filter, gaussian, ricker. Ricker wavelet should NOT be used for the purpose of QC.

`filter`

Only if `wavelet_type` is filter, the filter parameters should be specified here. For example, a Butterworth band-pass filter with 2 poles: BW 0.2 2 2 2. Ignore this parameter if the `wavelet_type` is set to gaussian or ricker.

`data_sample_interval`
`wavelet_width_parameter`
`wavelet_length`
`wavelet_normalization_method`

The above four parameters are needed by gaussian or ricker `wavelet_type`. These are ignored if the `wavelet_type` is filter. The `data_sample_interval` is the sample interval in time domain, e.g., 0.025. The `wavelet_width_parameter` is the width of the wavelet. It is the sigma for Gaussian wavelet and the central frequency for Ricker wavelet. The `wavelet_length` is length of the wavelet. We suggest that the wavelet is at least 3 times the width parameter in time domain. `wavelet_normalization_method` is recommended to be PEAK.

`save_wfdisc_table`

This is a Boolean parameter (true/false). If this is true, when reading input from wfprocess table, a wfdisc table will be generated containing the receiver functions after QC. The program will use channel keys define above for different channels. This parameter will be ignored if reading input waveform from wfdisc (i.e., `use_wfdisc_in` is true). If `use_wfdisc_in` is false and this parameter is true, then the program saves the output to both wfdisc and wfprocess tables.

`save_wfprocess_table`

This is a Boolean parameter (true/false). If this is true, when reading input from wfdisc table, a wfprocess table will be generated containing the receiver functions after QC. This parameter will be ignored if reading input waveform from wfprocess (i.e., `use_wfdisc_in` is false), when wfprocess table will be saved by default.

`save_3C_data`

This is a Boolean parameter (true/false). It is applicable only if the input is from wfprocess table and with 3c datatype, which means that each row in wfprocess table is a three-component seismogram. If this is true, the waveforms after QC will be saved in 3c datatype. datatype is an entry of the wfprocess and wfdisc table.

`save_decon_table`

This is a Boolean parameter (true/false). When this is true, the decon table will be saved after QC. However, ONLY one component of the decon table will be saved. This component is specified by `edit_on_channel` and the associated channel key will be used in saving the table. This parameter will be ignored if `use_decon_in_editing` is false.

`save_vertical_channel`

This is a Boolean parameter (true/false). When it is true, vertical component of the three component seismogram will be saved after QC. This is automatically set to true if the input datatype is 3c (i.e., ThreeComponentSeismogram). We use this parameter because there are some imaging programs that only use radial and transverse components.

`save_metadata_only`

This is a Boolean parameter (true/false). When it is true, the program only saves database tables but not the real waveform data. This works for the case when the same type of input and out tables are used. For example, both `use_wfdisc_in` and `save_wfdisc_table` are true. When `use_wfdisc_in` is false, which means use wfprocess as input table, the datatype is 3c, and `save_wfdisc_table` is true, the program currently is unable to handle this combination and will throw errors and exist.

`save_filtered_data`

The user could choose to save the filtered data by tuning this on (or set to true). If this is true, the data after QC will be filtered before saving using the filter defined above by `apply_prefilter` and other related parameters.

`output_dfile_base`

Base of the output data file name, e.g., RFedited. The file will be named as: RFedited_KF28.R for radial, *.T for transverse, and *.Z for vertical. When saving 3C data, the file extension will be *.3C for wfprocess table, and *.w for wfdisc table.

## 4.2 Parameters driving the QC in GUI-mode

The parameter array, `gui_edit_parameters &Arr{},` contains all of the parameters used by GUI-mode RFeditor.

```
#stacking window params for robustSNR stacking.
#stacktype RobustSNR
robust_window_start -1
robust_window_end 10

NFA_tolerance_TW_start -2
NFA_tolerance_TW_end 5
PCoda_search_TW_start 5
```

```
PCoda_search_TW_end 35.0
PCoda_grow_tolerance  0.0
max_trace_abs_amplitude 100    #true amplitude (as stored in the data).
CodaCA_search_TW_start 2.0
CodaCA_search_TW_end 20.0
CodaCA_tolerance_twin_length 5   #recommend: 5*(filter width in time-domain).
RefXcor_search_TW_start -1
          RefXcor_search_TW_end 10

#decon parameter threshold: default values.
niteration_min 20
niteration_max 1000
nspike_min 20
nspike_max 1000
epsilon_min 0.0
epsilon_max 50.0
peakamp_min 0.001
peakamp_max 1
averamp_min 0.0
averamp_max 1
rawsnr_min 1
rawsnr_max 1000
```

### 4.3 Parameters driving the QC in Auto-mode

```
apply_klat false
apply_decon_ALL true
apply_kdnitn false
apply_kdnspike false
apply_kdepsilon false
apply_kdpkamp false
apply_kdavamp false
apply_kdsnr false
apply_kldsi true

apply_knfa true
apply_kgpc true
apply_kca true

apply_klsw true
apply_klxcor true
apply_klrfqi true

NFA_tolerance_TW_start -2
```

```
NFA_tolerance_TW_end 5
PCoda_search_TW_start 5.0
PCoda_search_TW_end 35.0
CodaCA_search_TW_start 2.0
CodaCA_search_TW_end 20.0
RefXcor_search_TW_start -1
RefXcor_search_TW_end 10
max_trace_abs_amplitude 100
```

This is the threshold for maximum true amplitude (as stored in the data).

```
CodaCA_tolerance_twin_length 5
PCoda_grow_tolerance 0.0
RFQI_weigth_stackweight 0.2
RFQI_weigth_refxcorcoe  0.3
RFQI_weigth_successindex  0.5
rfqi_min  0.6

niteration_min 20
niteration_max 1000
nspike_min 20
nspike_max 1000
epsilon_min 0.0
epsilon_max 50.0
peakamp_min 0.001
peakamp_max 1
averamp_min 0.0
averamp_max 1
rawsnr_min 1
rawsnr_max 1000

stackweight_min 0.2
xcorcoe_min  0.5
dsi_min 0.3
stacktype RobustSNR
robust_window_start -1
robust_window_end 10
```

## 4.4 Other optional parameters

### 4.4.1 First arrival detection

First Arrival (FA) detection parameters are not required if FA detection is not turned on. Use -fa option when running RFeditor to turn it on. Here we list these parameters and the default values. The user can change the default values by specifying new values in the parameter file.

```
FA_sensitivity 10e-4
```

This is the sensitivity in amplitude: turn on detection only if the amplitude is above this value. this is used in TraceEditOperator object.

```
FA_detect_length 0.8
FA_search_TW_start -5
FA_search_TW_end 5
```

Time window length for FA detection. The program will only detect FA within this window specified by the start and end time stamps. When detecting FAs, the moving window length is specified the the `FA_detect_length` parameter above. The program searches for FA within this length after first non-zero values (>=FA_sensitivity). The empirical length value for reference: >=4*gaussian_sigma for gaussian or 1.5*ricker_side_lope_distance for ricker.

```
data_shaping_wavelet_type GAUSSIAN
```

This is referenced only when detecting the first arrivals. This is the wavelet type used to generate the receiver functions, which is input of the RFeditor program. Use RICKER if ricker was used in either deconvolution or the pre-filtering process. There are three available types: SPIKE, GAUSSIAN, RICKER (CASE SENSITIVE). This parameter is read-in just once when the TraceEditOperator object is initiated.

### 4.4.2 Metadata lists for input and output tables

In parameter file, metadata lists are specified by a table of metadata tags/attributes and metadata types. In the current version of RFeditor, it uses built-in metadata lists unless the user specifies them in the parameter file.

```
mdlist_ensemble
```

Metadata list read in for each time series ensemble (station gather).

```
mdlist_wfdisc_in
```

Metadata list read in for each receiver function trace when using wfdisc table as input.

```
mdlist_wfdisc_out
```

Metadata list of the output wfdisc table.

```
mdlist_wfprocess_in
```

Metadata list read in for each receiver function trace when using wfprocess table as input. Adjust the lists of attributes when using decon table, arrival data or netmag table.

```
mdlist_wfprocess_out
```

Metadata list of the output wfprocess table.

### 4.4.3 Keys of the deconvolution attributes

The following are default values for decon keys. If you want to use other values, please make sure they are consistent with those in the mdlist_wfprocess_in metadata list/table when `use_decon_in_editing` is true. Those are defined in the parameter arrays, `gui_edit_parameters &Arr{}` and `auto_edit_parameters &Arr{}`.

Defaults definitions:

```
decon_nspike_key decon.nspike
```

```
decon_rawsnr_key decon.rawsnr
decon_averamp_key decon.averamp
decon_epsilon_key decon.epsilon
decon_niteration_key decon.niteration
decon_peakamp_key decon.peakamp
```

### 4.4.4 Plotting window parameters

Parameters for the plotting window shown here are the built-in default values. PLEASE DO NOT change them unless necessary! You can specify the values for these parameters in the parameter array, `gui_edit_parameters &Arr{}`. Here are list the default values for all of these parameters.

Default values:

```
SUVariableArea_grey_value 1
VariableArea true
WiggleTrace  true
blabel   Data Window
blabel2  Data Window
clip_data    true
clip_percent 99.5
clip_wiggle_traces   false
d1num    0.0
d2num    0.0
default_curve_color  black
editing_mode single_trace
f1num    0.0
f2num    0.0
first_trace_offset   0.0
grid1    1
grid2    1
gridcolor    blue
hbox 5000
interpolate  true
label1   time
label2   index
labelcolor   blue
labelfont    Rom14
labelsize    18.0
n1tic    5
n2tic    1
plot_file_name   SeismicPlot.ps
style    normal
time_axis_grid_type  solid
time_scaling auto
title    Receiver Function Data
titlecolor   red
titlefont    Rom22
titlesize    36.0
trace_axis_attribute assoc.delta
```

```
trace_axis_grid_type none
trace_axis_scaling   auto
trace_spacing    1.0
trim_gap_edges   true
use_variable_trace_spacing    false
verbose  true
wbox 950
windowtitle  RFeditor
x1beg    0.0
x1end    120.0
x2beg    0.0
x2end    24.0
xbox 50
xcur 1.0
ybox 50
beam_hbox 250
beam_clip_data false
beam_trace_spacing 1.0
beam_xcur 1.0
beam_trace_axis_scaling auto
```

## 5. Data Preparation

The waveforms (receiver functions) need to be saved in Antelope Datascope Database format using either wfdisc table or wfprocess table for indexing.

# 6. QC in GUI-Mode

# 7. QC in Auto-Mode

## References

Yang, X. T., G. L. Pavlis, and Y. Wang (2016), A quality control method for teleseismic P-wave receiver functions, *Bull. Seismol. Soc. Am.*, *106*(5), 1948–1962, doi:10.1785/0120150347.