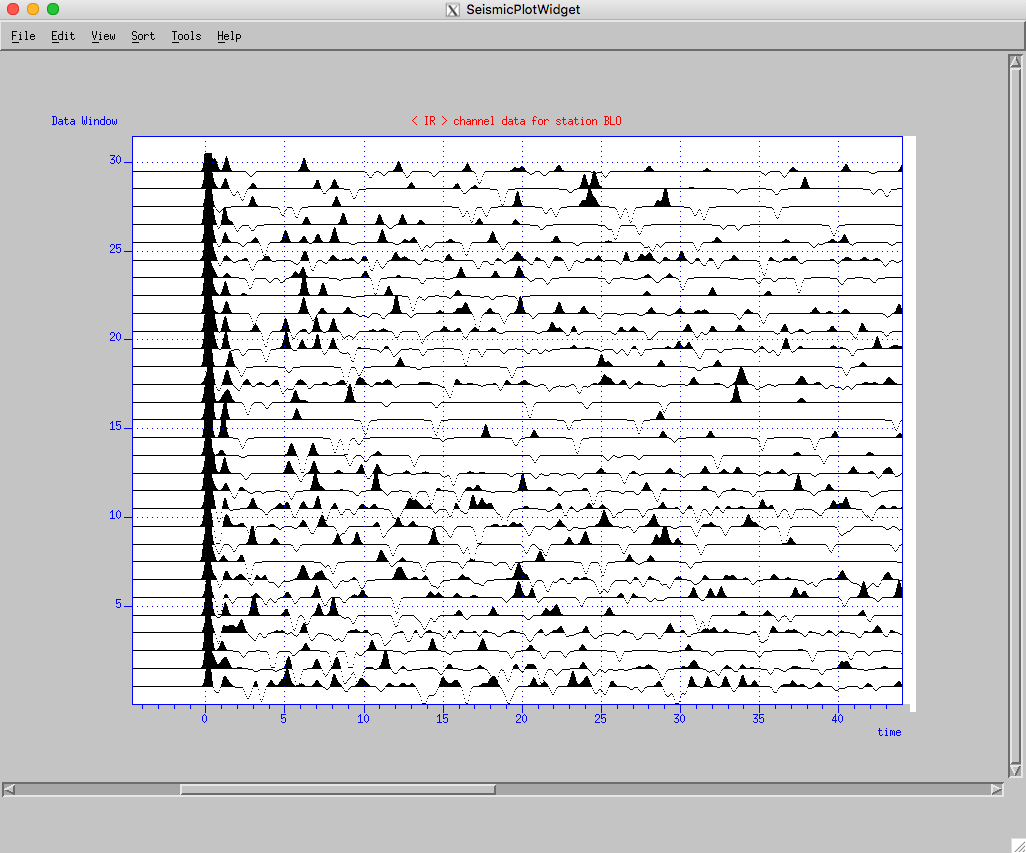
*RFeditor*

User Guide

v3.7.x



Xiaotao Yang[[1]](#footnote-1), Gary Pavlis, Yinzhi Wang

Indiana University

July 2018

# **Table of Content**

**Table of Content** i

**1. Introduction** 1

1.1 Download 1

1.2 Citation 1

1.3 Support 1

**2. Installation** 2

2.1 Operation Systems 2

2.2 Library dependency 2

2.3 Install *RFeditor* 4

**3. Command Line Options** 5

**4. Parameter File** 6

4.1 Global parameters driving the program 6

4.2 Parameters driving the QC in GUI-mode 8

4.3 Parameters driving the QC in Auto-mode 9

4.4 Other optional parameters 11

4.4.1 First arrival detection 11

4.4.2 Metadata lists for input and output tables 12

4.4.3 Keys of the deconvolution attributes 12

4.4.4 Plotting window parameters 12

**5. Data Preparation** 15

**5.1 Data for deconvolution** 15

**6. QC in GUI-Mode** 16

**6.1 Mouse operation conventions** 17

**6.2 Menus in the tool bar** 17

**7. QC in Auto-Mode** 23

**References** 24

# **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>. You can get the package by running in the terminal under the directory where you would like to save the code:

git clone https://github.com/xtyangpsp/RFeditor.git .

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. The source code for the required library libseisw.a is also included under libseisw.

*(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.

*(5) RFeditor\_demo\_portable.zip*

This data set should be used for testing purpose ONLY after the program is successfully installed. Please don’t use the data for research purpose. Accuracy of the data is NOT guaranteed.

## 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](http://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 contribpackage 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/. Another way to get contrib codes is to run: git clone https://github.com/antelopeusersgroup/antelope\_contrib.git src

From our experiences, change the ownership of the $ANTELOPE directory to be you, instead of the root user, if this is the case. After downloading/updating the contrib codes, please follow the instructions in README.md to compile the codes.

## 2.1 Operation Systems

This program has been fully tested under Mac OSX Mountain Lion (10.8), Mavericks (10.9), Yosemite (10.10), and Sierra (10.12. Newer Mac OS (e.g., High Sierra) should be working 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) X11

On MacOS, the library was part of the XQuartz package. The latest test in July 2018 was built against XQuartz 2.7.11.

(2) 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

(3) xmotif: xwindow libraries;

$fink list motif

$fink install motifpackagename

In addition to fink installation, xmotif libs (e.g., libXm.a) are required. The package can be installed manually from source code. The version installed using fink doesn’t include libXm.a. This package/source code could be downloaded from: http://motif.ics.com and the link there. The latest release is motif-2.3.8, available on sourceforge.net (accessed July 2018). The installation process is straightforward as documented in the downloaded motif-x.x.x package.

**<< Trouble shooting in installing motif-2.3.8>>**

You may need to install freetype2 libraries for font types. They can be installed using fink. If you get errors when installing motif, check the content in Makefile to make sure the libraries are pointing to the right directory. If not, try the following way to fun configure:

./configure CFLAGS='-I/sw/include -I/sw/include/freetype2 -I/opt/X11/include' CPPFLAGS='-I/opt/X11/include -I/sw/include' FREETYPE\_CFLAGS='-I/sw/include/freetype2' FONTCONFIG\_CFLAGS='-I/sw/include' FONTCONFIG\_LIBS='-L/sw/lib -lfontconfig' FREETYPE\_CFLAGS='-I/sw/include/freetype2'

<<< Configure Antelope localmake for boost and xmotif >>

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. For XMOTIF, if your fink directory is /sw, then the INCLUDE and LIB directories are: /sw/include and /sw/lib, respectively.

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

Note: under Antelope 5.8, seems now() function cannot be found. Commented out Makefile under libseispp for ArrivalUpdater.o in OBJS.

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

With the latest release (RFeditor v3.7.4), we added the source code for the required library libseisw.a under libseisw. When you compile RFeditor in the next step, this library will be automatically installed.

## 2.3 Install *RFeditor*

NOTE: In antelope 5.8 and later, antelope programs use TOOLCHAIN macro to define the compiler libraries. For now, RFeditor doesn't work under the antelope toolchain for OSX. Fix this issue by setting environment variable in ~/.bash\_profile (or similar other files):

**export TOOLCHAIN=native**

Then:

$source ~/.bash\_profile

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

$make Include

$make

If the above compiling procedure is successful, run:

$make install

This deposit tredit, wfprocess, 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/. wfprocess table was first released with dbxcor. We include it here for convenience since some of the editing procedure open database with wfprocess table. 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 v3.7.4 > 7/22/2018

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.

Xiaotao Yang & Gary Pavlis, Indiana University

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**

In addition to RFeditor.pf or other name for RFeditor parameter file, there is a very important PF file: seispp\_attribute\_maps.pf. This PF file is required, as it tells the program how to find tables and attributes that are not standard antelope built-in CSS3.0 attributes. This file also contains aliases of different attributes.

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 # if true, apply all procedures based on cutoffs of deconvolution

#parameters. Set to false if only some of the procedures are

#needed.

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

apply\_kdnitn

apply\_kdnspike

apply\_kdepsilon

apply\_kdpkamp

apply\_kdavamp

apply\_kdsnr

apply\_kldsi

*Table 4.1. The order of quality control procedures under Auto-mode and corresponding procedure names in the parameter file. See Yang et al. (2016) for details of the procedures.*

|  |  |  |  |
| --- | --- | --- | --- |
| **Order** | **Procedure** | **Name in parameter file** | **Category** |
| 1 | Cutoff of the number of iterations | *apply\_kdnitn* | *Decon-Procedures* |
| 2 | Cutoff of the number of spikes | *apply\_kdnspike* |
| 3 | Cutoff of epsilon (the ratio of residual energy to original energy) | *apply\_kdepsilon* |
| 4 | Cutoff of the maximum (peak) amplitude | *apply\_kdpkamp* |
| 5 | Cutoff of the average amplitude | *apply\_kdavamp* |
| 6 | Cutoff of the SNR of the original seismogram | *apply\_kdsnr* |
| 7 | Cutoff of the Deconvolution Success Index | *apply\_kldsi* |
| 8 | Discard Type-4 (negative first arrivals) traces | *apply\_knfa* | *Trace-Procedures* |
| 9 | Discard Type-5 (growing *P*-coda) traces | *apply\_kgpc* |
| 10 | Discard Type-6 (clustered *P*-coda) traces | *apply\_kca* |
| 11 | Cutoff of the Robust stack weight | *apply\_klsw* | *Stat-Procedures* |
| 12 | Cutoff of the correlation coefficient with reference trace | *apply\_klxcor* |
| 13 | Cutoff of the Receiver Function Quality Index | *apply\_klrfqi* |

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. These values are default values built-in in the program. Only add parameters that you want to change.

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 850

wbox 950

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

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.

# **5.1 Data for deconvolution**

Details on this step should be covered in deconvolution programs. We use trace\_decon by Wang and Pavlis (2016) based on generalized iterative deconvolution. trace\_decon saves deconvlution attributes that can be used in QC using RFeditor. The code for trace\_decon can be obtained through contacting Yinzhi Wang at Indiana University. See Wang and Pavlis (2016) for details on the method. The steps to get receiver functions can be summarized as:

1. Download waveform, either continuous or by earthquakes. Using miniseed2db to generate the wfdisc table.
2. If step 1) downloaded continuous waveforms, we need to build a teleseismic catalog here.
3. Generate predicted arrivals. trace\_decon doesn’t have to use hand-picked arrivals. We used predicted arrivals:

get\_predicted\_Parrivals catalog\_db station\_db outdb

Then, run:

dbassoc\_arrival –p 2 arrival\_db origin\_db

run: simple\_evid\_set (included in **Utilities**) to correct inconsistency of event table and origin table. Basically, this table use the new orid generated by dbassoc\_arrival to replace the evid and prefor and generates a new event table.

1. Link the wfdisc table to the catalog data base tables. Check the final data base to make sure the waveforms are correctly linked.
2. Run trace\_decon.
3. We call the resultant receiver functions database as: datadb.

5.2 Prepare for QC using RFeditor

Due to data gaps, in the receiver function db, i.e., datadb, there might be traces with only very few samples. Make sure exclude very short traces from the db. When getting errors, such as “stacking window outside the data window”, you need to come back to this step and clean up short traces.

# **6. QC in GUI-Mode**

Use the data in this release package for testing purpose. Unzip the file: RFeditor\_demo\_portable.zip. cd to the directory and following the instructions in README.txt to test the program and the instructions in this section and the next one.

GUI-mode is designed to let the user manually check the data quality and determine the optimized quality control parameters. Once the parameters are determined after working through 20-50 stations, they can be set in the parameter file, e.g., RFeditor.pf, for GUI-off mode. In RFeditor.pf, there are some GUI QC (Quality Control) parameters set as default values. All of those values and other unset parameters could be customized in GUI. **Specifically, as documented in Yang et al. (2016), we suggest the user conduct QC processing using *RFeditor* following the workflow below:**

*Step 1.* Test for appropriate filter or wavelet parameters used in viewing and editing the receiver functions;

*Step 2.* Set default GUI-mode parameters for QC procedures. The user may change these parameters later in GUI-mode;

*Step 3.* Start *RFeditor* under GUI-mode to test QC procedures and optimize parameters.

(a) Apply individual editing procedures under <Edit>. Examine the data as station gathers sorted by the attribute of interest (options in <Sort> menu) and then choose a cutoff based on this visual inspection. Exclude killed traces under <View> (shortcut: X).

(b) Inspect the data discarded by selected procedures under *View -> Review Killed Traces*. Check trace metadata to see the procedure that killed that trace under *View -> Show Trace Metadata -> For Picked Trace* (shortcut: M).

(c) Adjust parameters until the remaining station gather look clean and little to no good data are discarded.

(d) Repeat (b) to (c) for combinations of multiple QC procedures.

(e) Repeat (a) to (d) as needed for other station gathers to finalize QC procedures and parameters for Auto-mode. We have found that a sample of around 10 to 15 station gathers is sufficient to define a reasonable cutoff for most parameters.

*Step 4.* Run *RFeditor* under Auto-mode to process all station gathers based on the parameters determined from *Step 3*.

*Step 5.* Examine samples of station gather after QC under GUI-mode.

*Step 6.* Repeat Steps 1 to 5 as needed to adjust QC procedures and parameters.

To denote the percentage of receiver functions left after quality control (Yang et al., 2016), we define the Acceptance Rate (AR), , of applying QC on receiver functions as:

, (6.1)

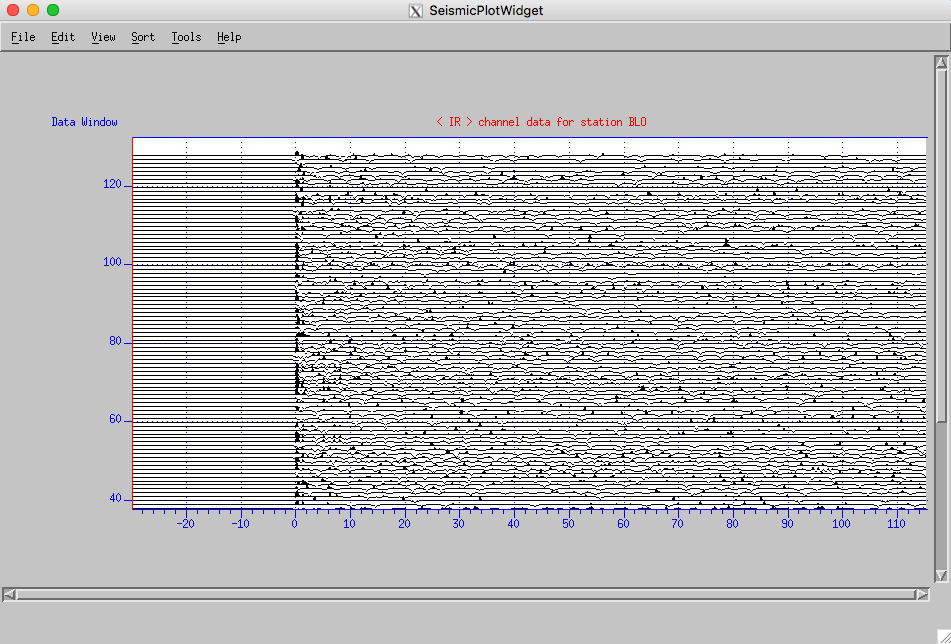
where *R* is the number of seismograms in the data set before editing, *R’* is the number of seismograms after applying the selected QC procedures. We use the factor of 100 to make a percentage measure. *RFeditor* computes the average for each station gather. For every example shown below, we estimated the average for the whole receiver function data set.

# **6.1 Mouse operation conventions**

In RFeditor, when selecting traces, please click middle-button of the mouse. Use left click as normal. Left-click, hold, and drag will zoom in on the selected area. Left-click again to go back to the original view after zooming in. The method that RFeditor uses to change parameter values is through clicking certain menus in GUI and, when inquired by the program, typing values in the terminal window, following the hints.

# **6.2 Menus in the tool bar**

Figure 6.1 is a screenshot of a typical RFeditor window after opening. Red title shows the channel code and the station name. In the Data Window, the positive lobes are filled in black. X-axis shows the time in seconds after first P, or from the beginning, depending on the definition of first arrival in RFeditor.pf. Y-axis is the order of the traces in the current view.



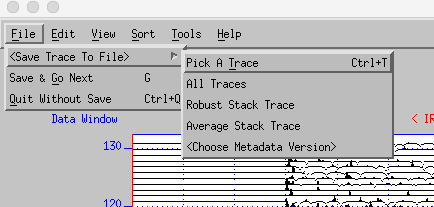
*Figure 6.1 A typical GUI window displaying the waveform traces. The traces, by default, are sorted by event ID. The order can be changed when sorting with different attributes.*

1. File

Currently, there are three functions (Figure 6.2): save picked trace, save edits and go next (edits are not saved under review mode with argument: -rm when starting the program), and quit the program without saving the QC result for the current station. On the right of each option is the keyboard shortcut, such as the Ctrl+Q to quit the program.

There are multiple options that you choose to save trace or all traces. From the submenu, you can save the picked trace (Pick A Trace), all traces in the current view (All Traces), the stack trace using Robust stack algorithm by Pavlis and Vernon (2010) (Robust Stack Trace), or the stack trace by simplying averaging all traces in the current view (Average Stack Trace). The trace is saved as plain text file. There is a MATLAB script under $RFEROOT/Utilities/readrf.m that can be used to read the saved trace data.

By default, the program saves trace metadata following version code=2. This can be changed by click: File -> <Save Trace To File> -> <Choose Metadata Version>. This choice will overwrite the default [2] for future stations, until changed again. There are currently three versions supported:



*Figure 6.2 <File> menu.*

Choose metadata version code for saved trace from below (1, 2, 3):

1 - Example below (basic version)

%Metadata version 1

station : BLO

start\_time : 4/01/2014 23:56:30.025

evid : 1169

samples : 6001

dt : 0.025000

t0 : -29.987570

stack\_weight : -9999.0000

RT\_xcorcoe : -9999.0000

RFQualityIndex : -9999.0000

DeconSuccessIndex : 0.4620

niteration : 3

nspike : 2

epsilon : 30.69890

peakamp : 0.40356

averamp : 0.00535

rawsnr : 2.50000

2 - Example below (basic+magnitude)

%Metadata version 2

station : BLO

start\_time : 4/01/2014 23:56:30.025

evid : 1169

magnitude : -9999.00

magtype : -

samples : 6001

dt : 0.025000

t0 : -29.987570

stack\_weight : -9999.0000

RT\_xcorcoe : -9999.0000

RFQualityIndex : -9999.0000

DeconSuccessIndex : 0.4620

niteration : 3

nspike : 2

epsilon : 30.69890

peakamp : 0.40356

averamp : 0.00535

rawsnr : 2.50000

3 - Example below (basic+magnitude+back\_azimuth)

%Metadata version 3

station : BLO

start\_time : 4/01/2014 23:56:30.025

evid : 1169

magnitude : -9999.00

magtype : -

back\_azimuth : 172.5

samples : 6001

dt : 0.025000

t0 : -29.987570

stack\_weight : -9999.0000

RT\_xcorcoe : -9999.0000

RFQualityIndex : -9999.0000

DeconSuccessIndex : 0.4620

niteration : 3

nspike : 2

epsilon : 30.69890

peakamp : 0.40356

averamp : 0.00535

rawsnr : 2.50000

1. Edit

All QC procedures and parameters are built in <Edit> menu, as shown in Figure 6.3. For options with a symbol of right-pointed triangle, there are sub-menus for additional QC options.

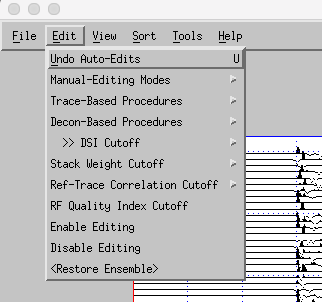
Undo Auto-Edits: This option can restore the traces killed by the LAST (only once) auto-editing procedure.

Manual-Editing Modes: Single Trace Edit mode lets you kill one trace at a time by picking the trace using middle button of the mouse. Cutoff Edit Mode allows you kill all trace below the picked trace on the screen. This is useful after sorting by certain attributes in <Sort>. You can reverse the trace order (Sort -> <Reverse Trace Order>) in case the bad traces are top of the sorted traces.

<Restore Ensemble>: This option allows you to restore all original traces before any editing for the current station.

Users are recommended to explore auto-editing procedures using the demo data package.

Table 6.1 lists a few shortcuts for quick QC:



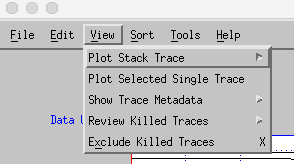
*Figure 6.3 <Edit> menu.*

*Table 6.1 Key shortcuts.*

|  |  |  |
| --- | --- | --- |
| ***Shortcut*** | ***Action*** | ***Category*** |
| *Ctrl+T* | *Pick a trace and save it to a text file.* | *File* |
| *G* | *Save edits for current station (not under review mode) and go to next station.* |  |
| *A* | *Apply all procedures based on trace charateristics.* | *Edit* |
| *D* | *Apply all procedures based on cutoffs of deconvolution parameters from trace\_decon.* |
| *I* | *Manually kill selected traces, one trace at a time* |
| *O* | *Manually kill traces below the picked trace in the current screen view* |
| *W* | *Sort the traces by robust-stack weight* | *Sort* |
| *R* | *Sort the traces by cross-correlation coefficent with picked reference trace* |
| *C* | *Compute decon-success index, defined in Yang et al. (2016)* |
| *L* | *Compute receiver function quality index, defined in Yang et al. (2016)* |
| *X* | *Exclude killed traces from the current screen view* | *View* |

1. View

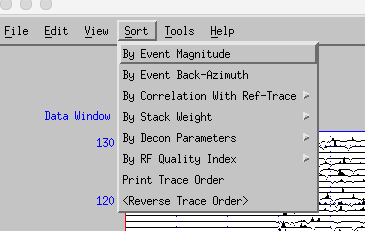
This menu includes, plot traces, view trace metadata, and review killed traces. Amonth these options, Exclude Killed Traces (shortcut: X) is useful to clean up the screen by only display the traces left after killing bad traces. Show Trace Metadata (shortcut: M) allows you to check the trace information. When under the mode of Review Killed Traces, the metadata for killed traces will show name of the procedure that this trace is killed by.



*Figure 6.4 <View> menu.*

1. Sort

This menu includes mainly the sorting methods. After applying any sorting method, click <Print Trace Order> to display the event ID and order of the traces in the terminal window.



*Figure 6.5 <Sort> menu.*

1. Tools

Click <Statistics> to print a summary of the current station, for example:

Total number of kills: 0>

--- Statistics from TraceEditPlot ---

Working on station BLO

Number of kills 0

Traces in current view 131

Traces in original data 131

Auto-killed traces 0

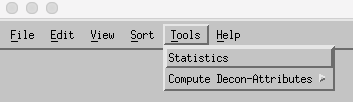
Manual-killed traces 0

Traces left 131

Acceptance Rate (%) 100

------ End of statistical data ------

The option <Compute Decon-Attributes> is under-development and will be implemented in future releases.



*Figure 6.6 <Tools> menu.*

1. Help

Print help message about major shortcuts and usage information.

# **7. QC in Auto-Mode**

For a large dataset, testing parameters in GUI-mode, Auto-mode is recommended to finish the QC process. In the Auto-mode, readers need to be aware of the order that QC procedures are implemented in *RFeditor*, as shown in Table 4.1.

# **References**

Wang, Y. Z., and G. L. Pavlis (2016), Generalized iterative deconvolution for receiver function estimation, Geophys. J. Int., 204(2), 1086-1099, doi:10.1093/gji/ggv503.

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.

Pavlis, G. L., and F. L. Vernon (2010). Array processing of teleseismic body waves with the USArray, *Comput. Geosci.* **36** 910-920.

1. Now at University of Massachusetts Amherst, xiaotaoyang@umass.edu. [↑](#footnote-ref-1)