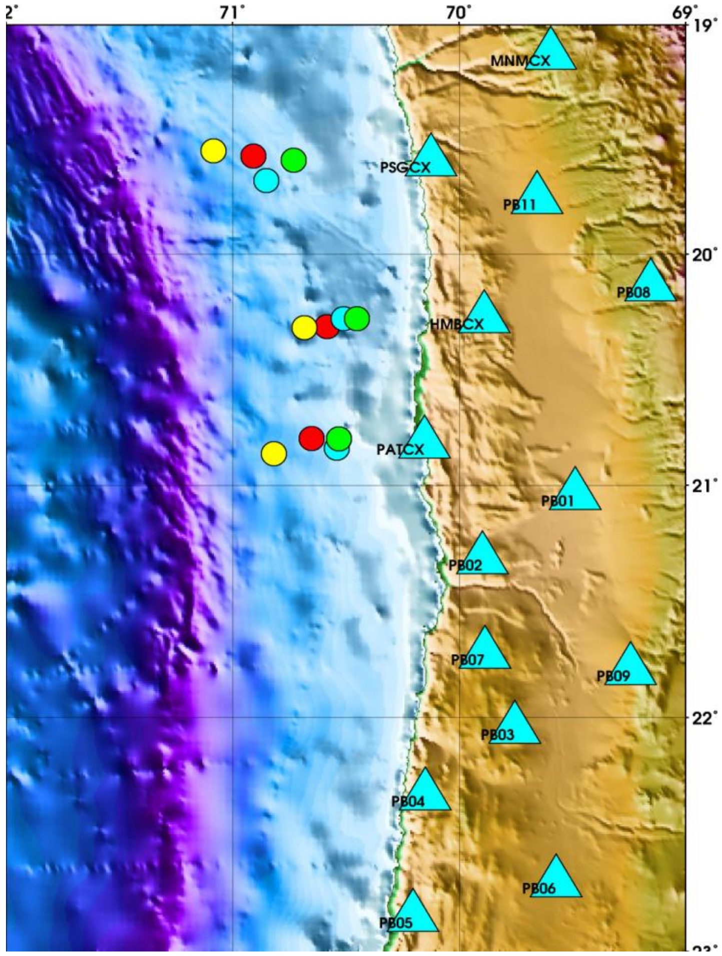
RTQUAKE

A Real-Time Earthquake Detection System Integrated with SEISAN

Version 1.2



Terje Utheim and Jens Havskov

Department of Earth Science
University of Bergen
Allegaten 41, 5007 Bergen
Norway

Ph. +47 55583408
Email : terje.utheim@geo.uib.no , terjeu@hotmail.com

July 2014

Cover

The map shows 3 events recorded by the example configuration described in this manual. The yellow marker is the automatic location done by RTQUAKE. The red, green and cyan markers are locations done by other institutions.

The events are:
2014/04/01 23:46 M 8.2
2014/04/03 01:58 M 6.3
2014/04/03 05:26 M 6.3

Questions and suggestions

Any questions or suggestions concerning the software can be sent to the email addresses on the front page or to rtquake@gmail.com

1 INTRODUCTION

RTQUAKE is a system for monitoring, triggering and recording of data coming from one or several SeedLink servers or digitizing units providing data according to the SeedLink protocol. The system is intended for routine operation of local and regional networks. RTQUAKE is written in C and uses OpenGL and GD (Boutell) for graphics.
The system functionality is similar to both Earthworm and SeisComp3 in the sense that it detects events and records them. It does not have the many utilities and advanced features for automatic processing of these systems, but has the advantage of recording events and s-files (event parameter files) directly into the SEISAN database (Havskov and Ottemöller (1999)) ready for processing without further steps.
The installation and configuration however, is simple and the manual processing through SEISAN of recorded events and continuous data in the Seiscomp3 ring buffer system is very easy.
The SEISAN system is mainly working with event data, so for each event there is one ASCII file (so called S-file) containing all parameter data for the event as well as a link to the corresponding waveform file(s) or position in the SeisComp archive. The S-files are organized in a data base like structure which can be accessed through a main processing program. The main task for a real time system is then to create this S-file and the corresponding waveform files and put them into the correct location in the data base.
RTQUAKE has several independent modules of which the trigger-recording module RTDET is the core module. The user can chose to run several other modules depending on the degree of monitoring that is desired. Common for most modules is that they read incoming streams from a SeedLink server (SeedLink clients).
RTQUAKE has an option for doing automatic location of events that works reasonably well when the phase-picker is able to find well defined phases on a sufficient number of stations. In general the automatic location option works better for events with a magnitude from 2.0 and above. The calculated locations should be used as indicators and by no means as a final determination of an event location.
The automatic magnitude that is calculated is based on the events coda, in this case from the event onset until the de-trigger of the event.
RTQUAKE has an option for computing local magnitude (Ml and Mw from spectra) automatically (using a SEISAN module) provided that the response-files for the stations are present. For the test configuration the response files for the stations are supplied in the distribution and should be copied into the SEISAN CAL catalogue.
RTQUAKE can be configured to run single-network, subnets and to read data from different SeedLink servers. The SeedLink servers can be digitizers that support the SeedLink protocol, local or remote SeedLink servers that provide data from a single seismic network or from international SeedLink servers that provide data from a lot of international seismic stations.

In the case where the seismic stations are spread over a geographically big area it would make sense to configure subnets from for example the northern part, the southern part, the eastern and western part. Events would then be recorded from the specified regions. The subnets can overlap in the sense that several stations from one region also are defined in another region.

Several subnets can be defined within one instance of RTQUAKE

Several instances of RTQUAKE (up to 10) can run simultaneously with different parameter sets.

In chapter 3 several different configurations are discussed in detail.

The figure below explains some possible configurations:

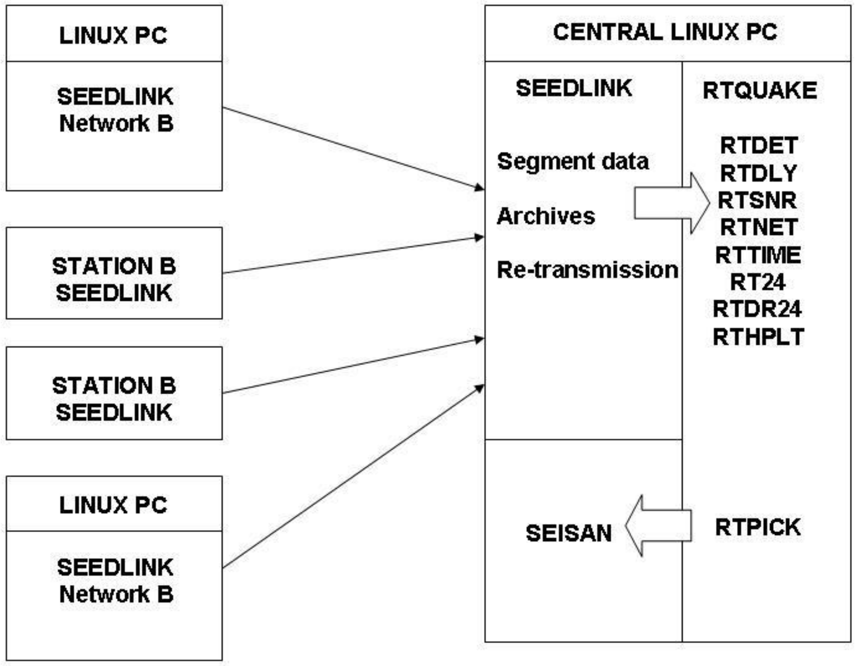


Figure 1 A typical RTQUAKE configuration using input-data from one or more SeedLink servers. The data enters a local SeedLink server before being processed by RTQUAKE in order to have direct access from SEISAN to the archive with continuous data. See chapter 13-17 for documentation on the different modules.

- In this configuration RTQUAKE runs on the same computer as the local SeedLink server receiving data and SEISAN.
- Data from different SeedLink servers and stations are fed into the local SeedLink server and RTQUAKE connects to the local SeedLink server as a client, selecting the components that will be used for detection.
- Detections are recorded directly in the SEISAN database with the corresponding S-file.
- The events can be processed manually immediately.
- The software includes an automatic phase picking option to include phases in the S-file. Optionally automatic location and magnitude can be done based on these readings.
- SEISAN has direct access to the SeedLink server archive.

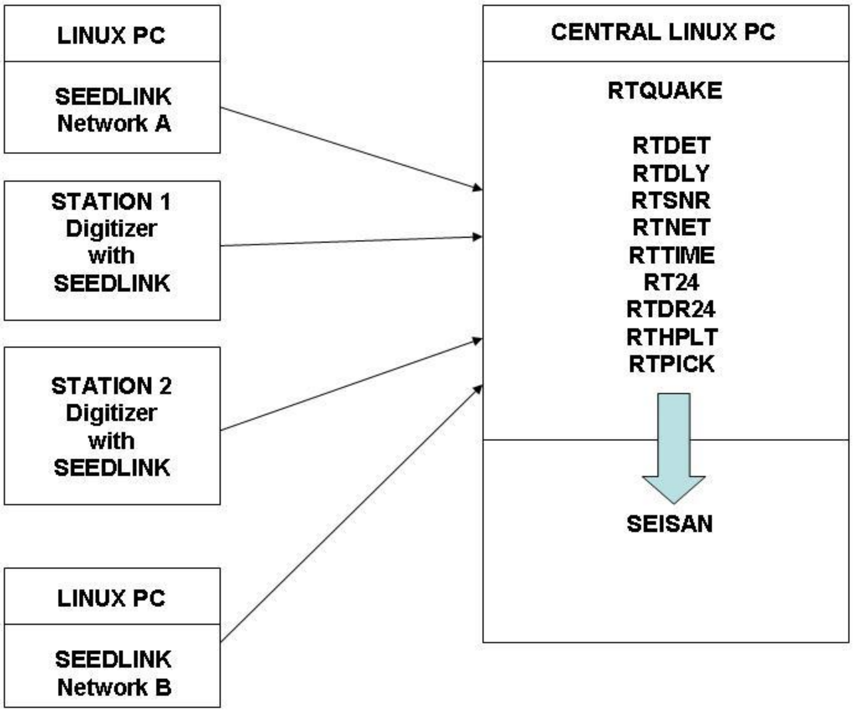


Figure 2. An alternative configuration is that RTQUAKE is configured to read directly from external SeedLink servers or digitizers that support the SeedLink protocol. You then do not need a local SeedLink server installed locally, but you will lose the direct access to the SeedLink archive from SEISAN.

The distribution comes with a test setup so immediately after installation, the system can be tested with real data.

2 INSTALLATION

Pre-requisites:

Before installing RTQUAKE, some third party free software must be installed.

2.1 SeedLink

RTQUAKE routines will only work when there is access to a SeedLink server locally or remotely. If a local SeedLink server will be used, it must be installed. The SeedLink server is part of SeisComp 2.5 or SeisComp3. Version 2.5 is public software and is included with RTQUAKE in the SeedLink catalogue in the distribution together with the user manual. SeisComp 3 can be found at: http://www.seiscomp3.org . A local SeedLink server is not needed if you only want to run the test example.

2.2 Graphics libraries

All graphics modules use OpenGL and/or the GD library by Thomas Boutell.
The following libraries must be installed:

GD library (In Ubuntu: search for gd with the Synaptic Package Manager or with the Ubuntu Software Centre and look for: Generate graphs using the GD library). Select "**libgdchart-gd2-xpm-dev**" Generate graphs using the GD library (development version). When you select this package, other needed packages will automatically be installed.

OpenGL (In Ubuntu: search for glut or freeglut with the Synaptic Package Manager or Ubuntu Software centre and look for: glut. Select "**freeglut3-dev**" OpenGL Toolkit development files. When you select this package, other needed packages will be automatically installed.

GD library (In Centos: yum search gd)
OpenGL (In Centos: yum search glut)

2.3 SEISAN

SEISAN for data analysis must be installed before using RTQUAKE as recorded events are stored in the SEISAN database and SEISAN programs are used for the manual and automatic processing. SEISAN is found at www.seisan.info

2.4 STEP-BY-STEP installation of RTQUAKE

RTQUAKE can be installed on a standard installation of Linux. It has been developed and tested under Linux Centos, Ubuntu and Fedora.

STEP 1:

It is assumed there exist a user account in which to install RTQUAKE. If not or you want to use a separate account, then first create a user account with a username and directory name. An account **seismo** will be used throughout this manual, but any account name will work.

Username: **seismo**
Password: selected by the user.

This will create a home directory: /home/seismo.

Log into this account to start the installation.

STEP 2:

Make a directory for the RTQUAKE installation. Can be any legal directory name.

mkdir mydir
cd mydir

STEP 3:

RTQUAKE is distributed as rtquakeddmmmyy.tar or rtquakeddmmmyy.tar.gz file, where dd is day, mmm is month and yy is year, for example: rtquake06feb12.tar. The distribution can be found at:

ftp://ftp.geo.uib.no/pub/seismo/SOFTWARE/RTQUAKE

Download the distribution file to the directory you just created and uncompress the file and unpack the distribution:

gunzip rtquakeddmmmyy.tar.gz
tar –xvf rtquakeddmmmyy.tar

A directory structure has now been installed with programs, parameter files, data files, temporary files etc. For details see chapter 8. The most important for the user operation are:

/home/seismo/mydir/par	Parameter files for the different modules. Each setup of parameters is in a named subdirectory which contains several parameter files for the particular setup. An example is the DEMO1 directory with the data for the test run..
/home/seismo/mydir/par/DEMO1	Test configuration (Test run example)
/home/seismo/mydir/wrk	Work catalogue for testing of software
/home/seismo/mydir/map	File containing locations, station file, html file

showing last location.

/home/seismo/mydir/loc Links to static Google map showing locations.

STEP 4

Set environment for where RTQUAKE is installed:

In the /home/seismo/mydir/com directory there is a setup file that must be sourced. This can be done from the command line or from the .cshrc or .bashrc file depending on the shell used in your account. Check what shell is used with the command:

env | grep SHELL

Edit the setup_rt.csh or setup_rt.bash file in the /home/seismo/mydir/com catalogue before sourcing it so that it corresponds to your environment !!
Modify the line that define the RTQUAKE_TOP to fit your RTQUAKE catalogue.
RTQUAKE expects to find SEISAN installed on the system and the environment variable SEISAN_TOP defined, see SEISAN manual.

for csh:
Include the following line at the end of your /home/seismo/.cshrc file:
source /home/seismo/mydir/com/setup_rt.csh

for bash:
Include the following line at the end of your /home/seismo/.bashrc file:
source /home/seismo/mydir/com/setup_rt.bash

When you now open a new terminal window the correct environment will be active.
Continue with STEP 5 to compile the software.

STEP 5:

Installation and compilation of complete RTQUAKE package:

cd /home/seismo/mydir
make clean
make rtquake
make install

Change to the RTQUAKE work directory or to a working directory in your home directory. This is to avoid temporary output files to be mixed with the RTQUAKE software:

rtwrk

RTQUAKE is now ready for operation.

3 TEST RUN INCLUDING MONITORING

A set of parameter files has been prepared to test the installation of the RTQUAKE package. The SeedLink server at GFZ Potsdam, Germany is used. The server has both public and restricted data.
To demonstrate the use of the software, the non-restricted data from the Plate Boundary Project (IPOC), GFZ Potsdam, Germany in northern Chile is used.
As this is a very active seismic area, new events will normally be detected and recorded within a few minutes.

To test the software, the setup files and parameter files have been prepared. The user can use these as recipes for setting up a configuration for an actual network. For details of the test configuration see chapter 4.

In SEISAN 10.1 and later, the IPOC stations are included in the STATION0.HYP file. If you are using a different STATION0.HYP file, the IPOC stations must be added if you want to locate events from the test run. The file IPOC.TST file (in SEISAN format) in /home/seismo/mydir/par must then be included in your STATION0.HYP file.

To start the test, type:

rtstart
This command will start RTQUAKE

rtstop
This command will stop the data acquisition and the graphic monitoring.

Output from test run

After executing the start command, you will after some seconds see a plot showing the signals in real time (Figure 3) and a plot showing indication of the trigger times and duration of triggers (Figure 4).

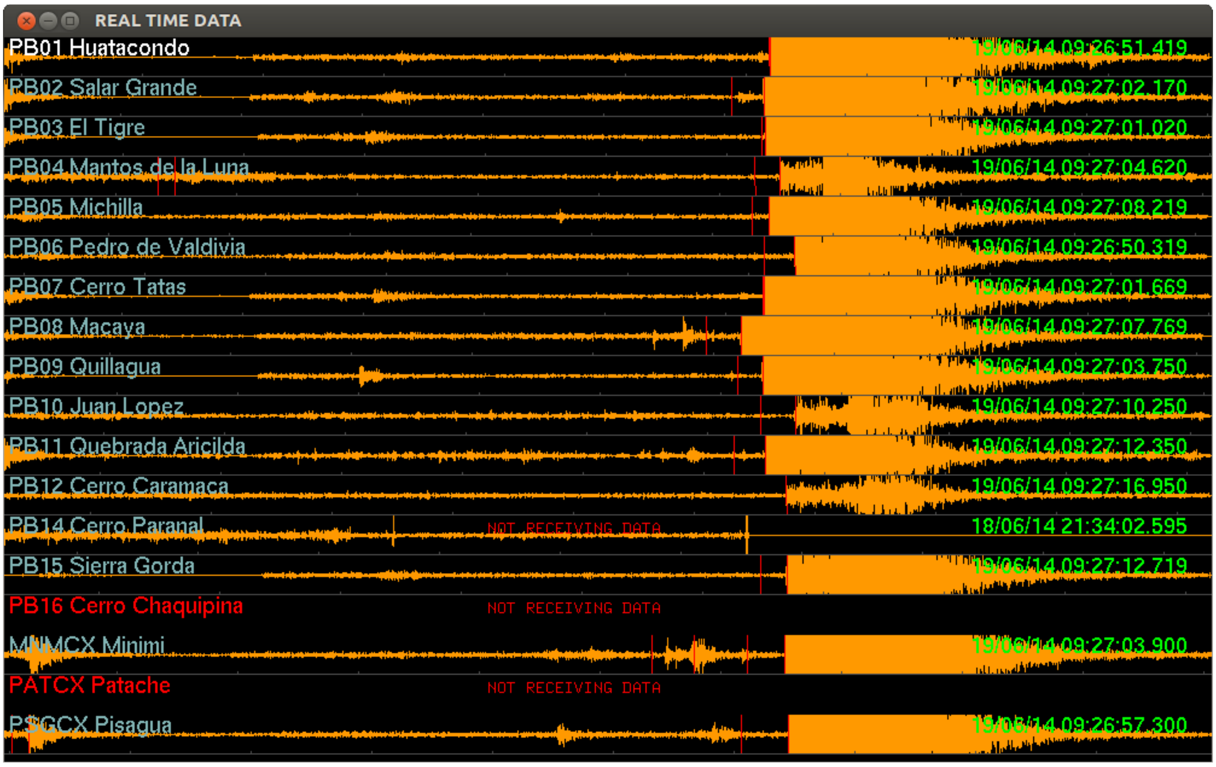
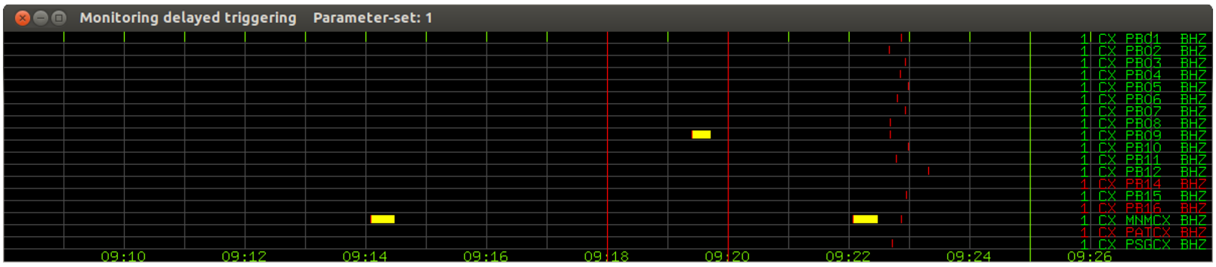
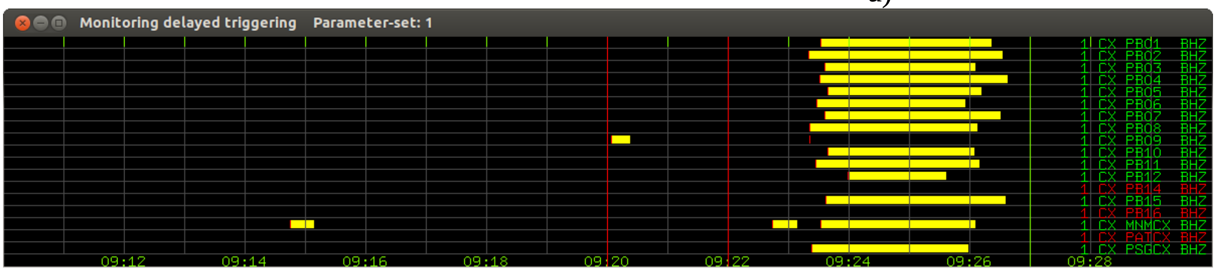


Figure 3 RTNET shows the signal from selected channels in "near-real-time". It also indicates when channels are not transmitting data as for station PB14, PB16 and PATCX in this case. The red vertical lines indicate possible triggers, and are inserted when the traces are filtered. These triggers are not the RTQUAKE triggers computed by the RTDET module. Several instances of the program can be executed to show different stations, to apply different band pass filters, different color schemes, different window sizes and different positioning on the screen.



a)



b)

Figure 4 RTDLY shows the onset and duration of triggers (yellow lines) for individual channels. When a trigger starts on a particular channel, this is indicated with a small red vertical lines at the trigger time. When the trigger is turned off, the duration of the trigger is indicated with a yellow line. The green vertical line to the right indicates the current time. The two vertical red lines indicate the array-propagation-window (APW) within which the network detection is performed. The text on the right, for example 1 CX PB01 BHZ, displays the subnet number, network id, station name and component respectively. Figure 4 top shows the onsets of triggers at a) while the bottom figure shows the situation a minute later at b). Most triggers are now turned off and the duration of the triggers are marked in yellow. The triggers will finally reach the Array-Propagation-Window (between the two vertical red lines) and a network trigger will be declared if sufficient triggers are flagged. Station names marked with red color indicates that the station is not receiving data.

If the default test run records some events and manage to do a location, you may open the system browser and enter the address:
/home/seismo/mydir/map/LAST_TRIG.html and a map will show on the screen with the stations and the suggested location marked. Clicking on the station markers will show the signal for that station if generated. This link will only show the last located event. See Figure 5a and 5b below. A parameterfile /home/seismo/mydir/map/map.par control the zoom-level, latitude/longitude grid, maptype id, number of previous event location to plot and if the red marker should be blinking or not. Figure 5c is a webpage that shows only the google map with the red marker (blinking or not) with the last located events and the x number of the last events if specified in the map.par.

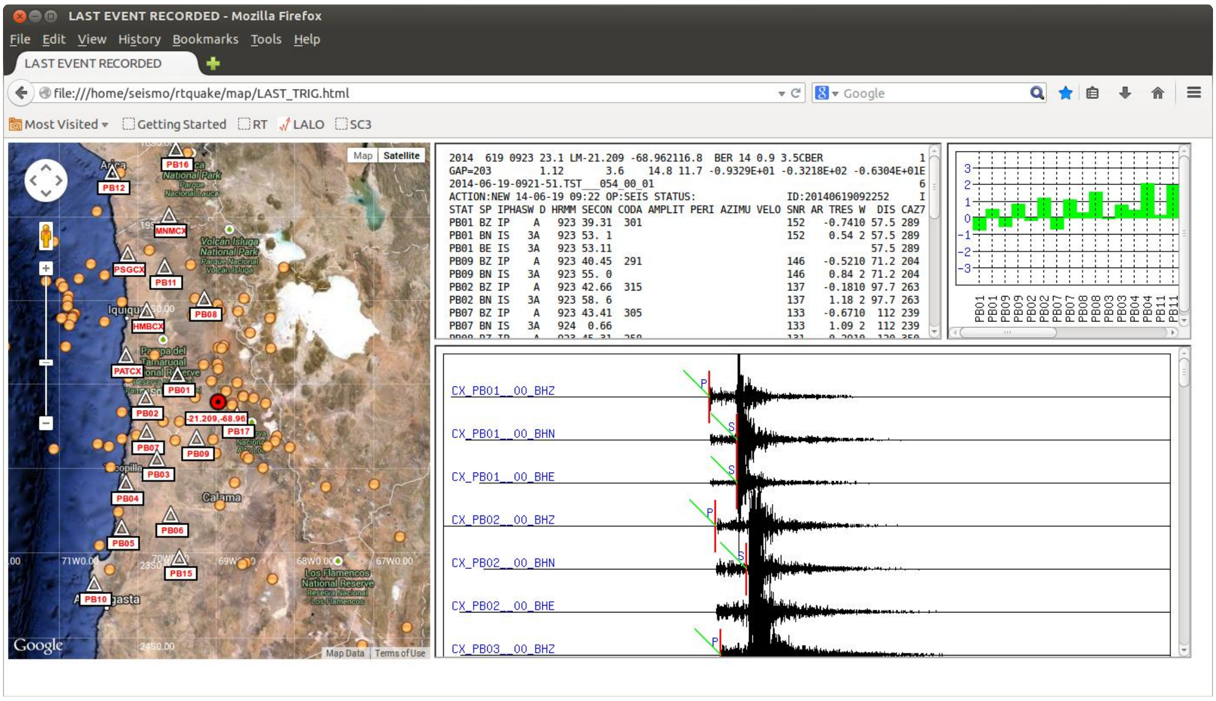


Figure 5a Web page showing location of last located event. Maptype set to HYBRID.

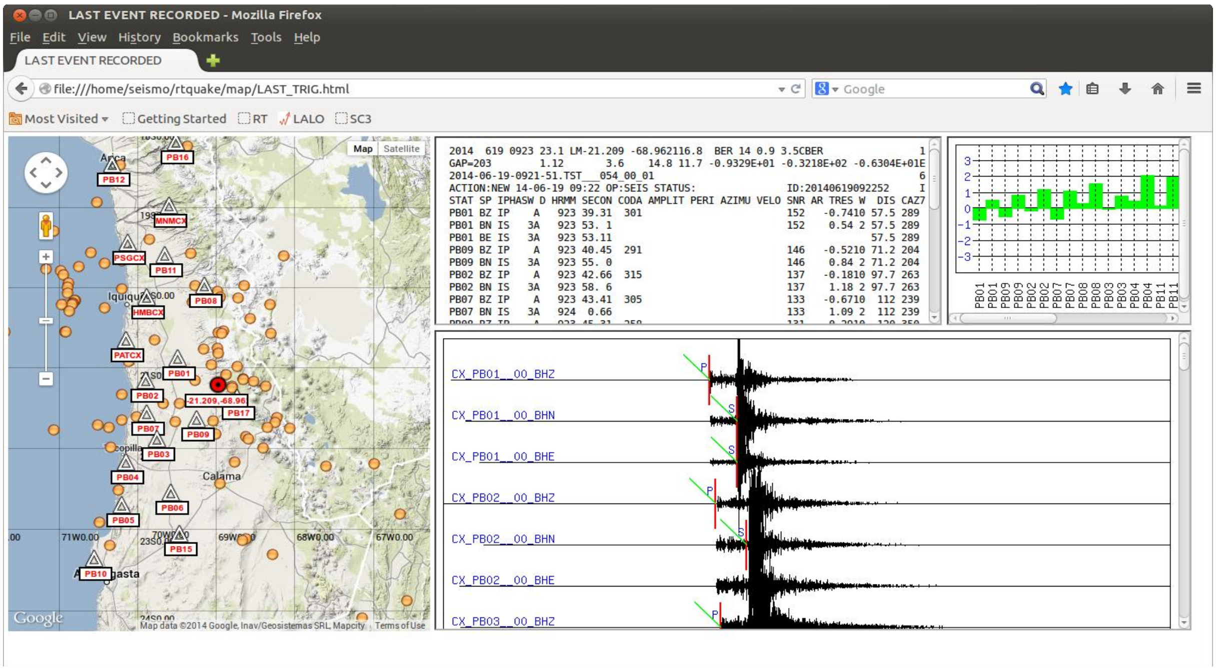


Figure 5b Web page showing location of last located event. Maptype set to TERRAIN.

The left window in the browser shows the map with the stations show as triangles and name. The calculated location is marked in the center of the map with the red circle and the coordinates below.
Initially the window right-below shows the recorded signals with phases marked. Clicking on one of the stations on the map will show the signals from that station only if existing, with the suggested phases.
The window center-top shows a listing of the s-file for this event.
The window right-top shows a plot with the residuals each component.

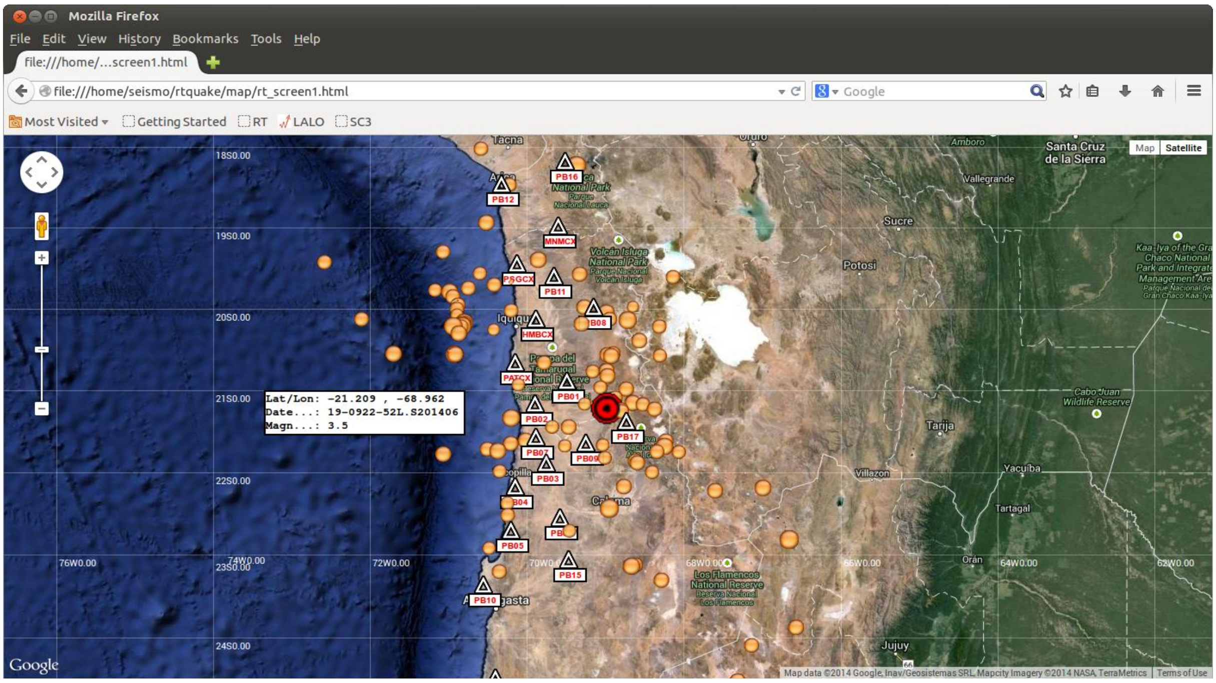


Figure 5c Web page showing location of last located event. Maptype set to SATELITE.

The parameter file for the web pages is described in detail in 4.6

In the catalog /home/seismo/mydir/loc you can find links to all locations up to the current time. Entering a link in your browser will show a static map as in the Figure 6 below with the suggested automatic location.

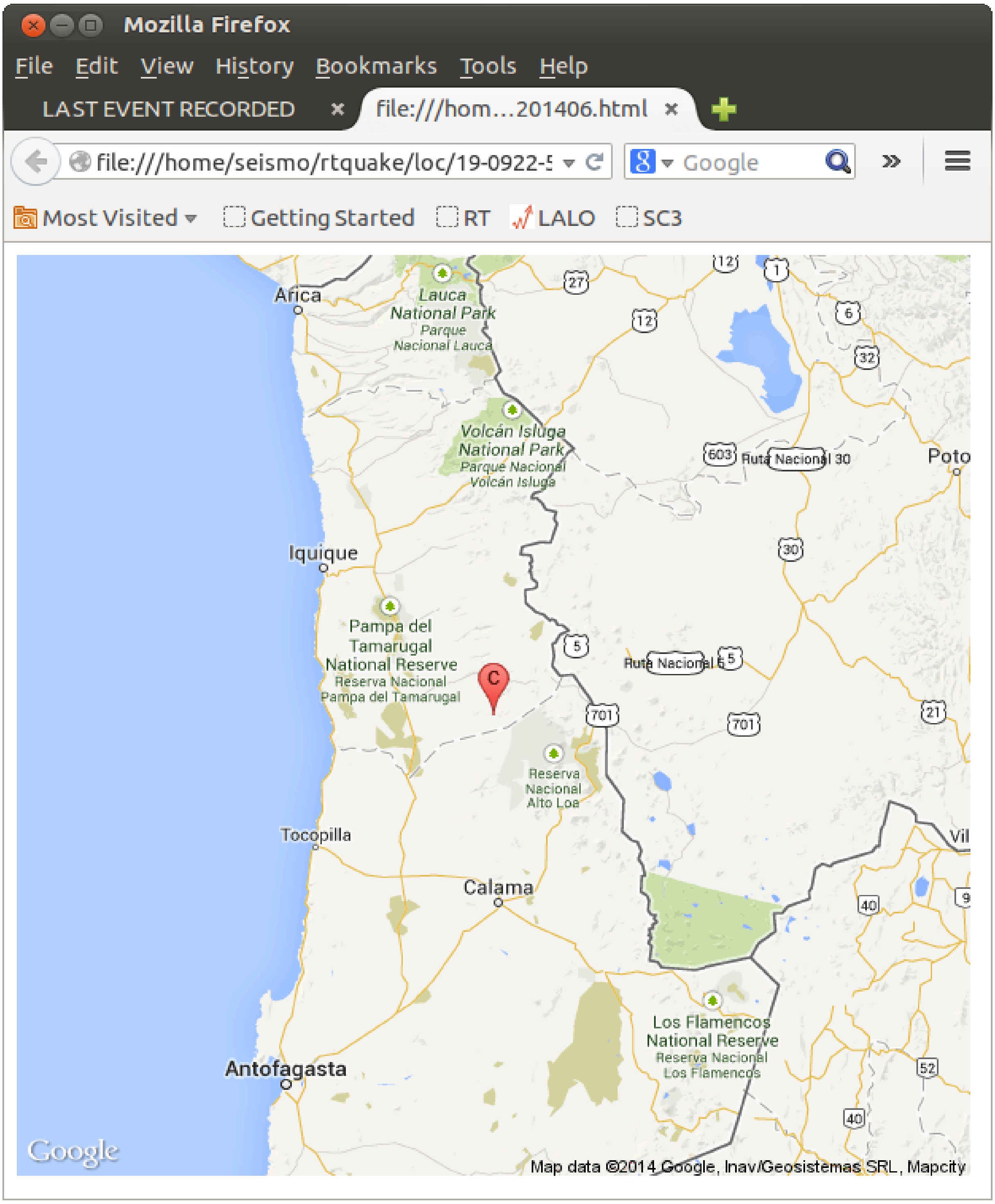


Figure 6 Static Google map generated by RTQUAKE.

The detected events are in the SEISAN data base TST. In order to check the events, use SEISAN command eev 201406 TST (for events in June 2014), find the event and write "po" to e.g. plot the event (Figure 7). For more details see SEISAN manual.

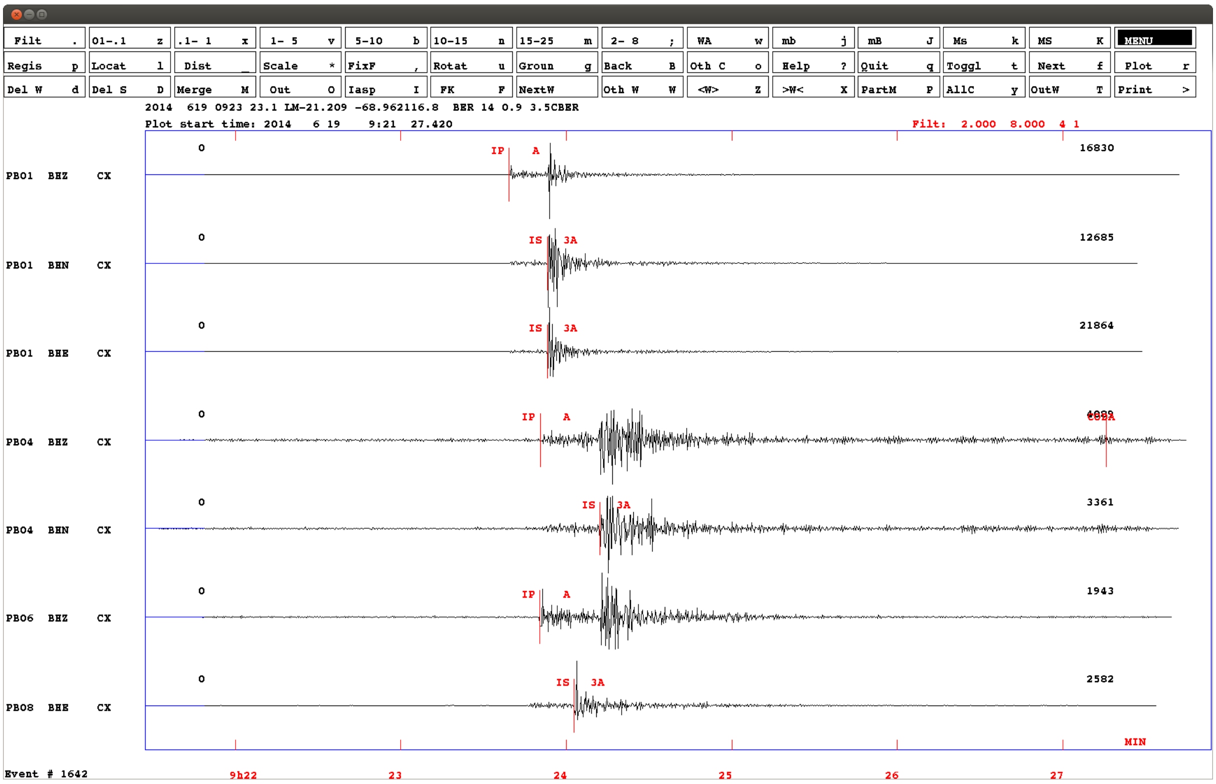


Figure 7 Recorded event plotted by SEISAN

The recorded event can now be plotted and processed by SEISAN. The test run is configured to pick phases and they are shown on the plot as in Figure 7.

In the test-run, the events waveform files are stored in the SEISAN WAV catalog structure (e.g. /home/seismo/WAV/TST__/2014/06/xxx and the database is called TST__ as set up in the test parameters (see chapter 4).
For the test run, the IPOC.TST file in /home/seismo/mydir/par must be included in the STATION0.HYP file to facilitate location of events. The IPOC.TST file contains the coordinates for the IPOC stations used in the test. If SEISAN10.1 or later is used, the stations are already installed.

To process an event change to the WOR catalog (wo) (or any other catalog you want to work in) and run eev for the actual date, for example: eev 20130110 TST

The test setup can also generate helicorder plots (see the examples at in chapter 18) and send out mail (see chapter 12).

4 PARAMETER FILES, SCRIPTS and ALIASES

4.1 Overview of configuration and parameter files, scripts and aliases

The following is a description of the different configuration and parameter files in RTQUAKE that the user has to adjust to the actual environment and network.

In /home/seismo/mydir/COM:

rtquake.par
Configuration file where the user can adjust some RTQUAKE system parameters such as where to write event files, if the system should do auto-location or not, if the system should calculate Ml and Mw or not, if the system should send a mail when detections occur etc. The file is described in 4.2

In /home/seismo/mydir/par/user_created_subdirectory:

Parameters for one particular setup is in a directory called user_created_subdirectory (name decided by user, an example was TEST1)

rt_config: This file defines the channels and SeedLink servers for one of several subnets using the same SeedLink servers, trigger parameters etc. See 4.3

streams_plot: streams to input from SeedLink server for continuous plot (RTNET module). See 4.4

stations_plot: selected components of streams, station description for continuous plot (RTNET module). See 4.4

streams_heli: streams to input from SeedLink server for heliplots. See 4.5

stations_heli: stations to plot, factor to amplify signals, filters. See 4.5

Aliases defined in the /home/seismo/mydir/com/setup_rt.bash and setup_rt.csh:

`alias rtstart='$RTQUAKE_TOP/com/rtquake_start'`
Start the rtquake_start script See 4.6. . **`$RTQUAKE_TOP is set in rt_config`**.

`alias rtstop='$RTQUAKE_TOP/com/rtquake_stop'`
Start the rtquake_stop script. See 4.6

`alias rtheli1='$RTQUAKE_TOP/com/rtquake_heli_tst1'`
Start the rtquake_heli_tst1 script. See 4.6

`alias rtheli2='$RTQUAKE_TOP/com/rtquake_heli_tst2'`
Start the rtquake_heli_tst2 script. See 4.6

`alias rthom='cd $RTQUAKE_TOP'`

Change directory to /home/seismo/mydir

alias rtcom='cd $RTQUAKE_TOP/com'
Change directory to /home/seismo/mydir/com

alias rtrtdet='cd $RTQUAKE_TOP/rtdet'
Change directory to /home/seismo/mydir/rtdet

alias rtpar='cd $RTQUAKE_TOP/par'
Change directory to /home/seismo/mydir/par

alias rtut1='cd $RTQUAKE_TOP/utils1'
Change directory to /home/seismo/mydir/utils1

alias rtut2='cd $RTQUAKE_TOP/utils2'
Change directory to /home/seismo/mydir/utils2

alias rtwrk='cd $RTQUAKE_TOP/wrk'
Change directory to /home/seismo/mydir/wrk

4.2 rtquake.par

In this file you specify if you want s-files created and how. You can also specify if you want auto-location, some parameters for the Filterpicker and how the delayed trigger should work. An example file is included below where the parameters are explained in some more detail.
The FilterPicker routine process each component of recorded data trying to identify p and s phases and their onset time.
The parameters marked 'FilterPicker' are default parameters for the FilterPicker module and should not be altered. The parameters are described in Lomax et al. (2012).
For the automatic location option to work, the coordinates of the stations must be included in the SEISAN STATION0.HYP file. For the test configuration, the coordinates can be found in /home/seismo/mydir/par/IPOC.TST file. If SEISAN10.1 or later is used, the stations are already included.
For the automatic computation of local magnitude, the response files for the configured stations must be present in the SEISAN CAL catalogue. For the test configuration the response files are stored in the /home/seismo/mydir/cal catalogue and should be copied into SEISAN CAL directory.

For some parameters, see the respective programs

Example of file

```
The following is an overview of /home/seismo/mydir/com/rtquake.par:


This file is parameter file for rtquake.
Only the lines with recognized keyword under Par 1 will be read.
The comments have no importance.
Columns Par 1-Par 2 start in columns 41,51.

keep   locate Action
```

```
----------------------------
  -1      0/1 A new s-file is created with no phase-picks.
              No location. This option is used for RTQUAKE: detection + no picks + no
              location

   0       0/1 A new s-file is created with the detection phase-picks only. No location.

   1        0  A new s-file is created with all phase-picks from FilterPicker.
              No location. This option is used for RTQUAKE: detection + NO location

   1        1  A new s-file is created with all phase-picks from FilterPicker.
              Automatic location. Phases causing high residuals will be removed automatically
              until MAX RESIDUAL (see below) and /or MINSTALOC (see below) is reached.
              The s-file will contain the location and the phase-picks that are left.
              This option is used for RTQUAKE: detection + autoloc

All keywords in capital letters. Now follows the actual parameters.

KEYWORD...........Comments............Par 1.....Par 2
----------------------------------------how to record s-files--------------------------------
KEEP                 1:sfile,-1:no sfile  1
----------------------------------------automatic location or not-----------------------------
LOCATION             1:Locate,0:No Locate 1
----------------------------------------automatic local magnitude or not----------------------
AUTOMAG              1 compute Ml,Mw      0
----------------------------------------name of SEISAN database-------------------------------
DBASENAME            For SEISAN           TST__
----------------------------------------SEISAN catalogue for waveforms------------------------
WAVEDIR              For SEISAN           WAV
----------------------------------------store waveforms in database or not--------------------
WAVE_DB_ACTIVE       For SEISAN           1
----------------------------------------max number of iterations discarding phases------------
ITERATION            Number of iterations 200.0
----------------------------------------maximum acceptable residual to do location------------
MAX_RESIDUAL         Maximum residual     2.5
----------------------------------------min no of stations with phase reading to do location--
MINSTALOC            Min stat to locate   5
----------------------------------------separate sub networks or all as one-------------------
ALLSUBNETS           0-sep.net >0 one net 0
----------------------------------------mail or not-------------------------------------------
MAIL1                0-no mail,1-mail     0         terjeu@hotmail.com
MAIL2                0-no mail,1-mail     0         receiver2@gmail.com
MAIL3                0-no mail,1-mail     0         whatever@mail.com
MAIL4                0-no mail,1-mail     0         any@mail.com
MAIL5                0-no mail,1 mail     0         to_you@yahoo.com
----------------------------------------total delay buffer trigger---------------------------
DELAY_BUFFER         Minutes delaybuffer  20.0
----------------------------------------where to set current time in delay buffer-------------
MINUT_NOW            Minute current data  17.0
----------------------------------------delay for trigger window------------------------------
DET_DELAY            Detection delay      7.0
----------------------------------------array-propagation-window-----------------------------
APW                  Array prop. window   120.0
----------------------------------------seconds to shuffle buffer don't change----------------
SECONDS2SHUFLE       Seconds to shift     4.0
----------------------------------------pre-event in seconds---------------------------------
PRE_EVENT            Pre-event (seconds)  60.0
----------------------------------------post-event in seconds--------------------------------
POST_EVENT           Post-event (seconds) 60.0
----------------------------------------no of days to save heliplots-------------------------
HELI_DAYS            No of days to save   5.0
----------------------------------------filterpicker don't change-----------------------------
FILTERWINDOW         FilterPicker         300.0
LTWINDOW             FilterPicker         500.0
THRESHOLD1           FilterPicker         10.0
THRESHOLD2           FilterPicker         10.0
TUPEVENT             FilterPicker         20.0
----------------------------------------sound on or off when trigger--------------------------
SOUND                1-sound, 0-nosound   1.0
----------------------------------------printing or not--------------------------------------
PRINTING             Debug printing       0
```

KEEP	How to record s-files. -1: Record the s-file in the database, but with no phases. 1 : Record the s-file with phases 0 : Record the s-file with detection phases only
LOCATION	Try to do automatic location. 1: do automatic location. If KEEP=-1, location will not be executed. 0: no location.
AUTOMAG	Calculate Ml and Mw 1: calculate Ml and Mw 0: no magnitude calculated
DBASENAME	SEISAN database name (up to 5 letters)
WAVEDIR	SEISAN waveform directory
WAVE_DB_ACTIVE	SEISAN. Store waveform data in the specified directory in WAVEDIR or in a database structure under WAVEDIR.
ITERATION	Number of times to run hyp, remove components with bad residuals and run hyp again.
MAX RESIDUAL	Maximum residual to accept running location
MINSTALOC	Minimum number of stations with phase to accept running location.
ALLSUBNETS	All subnets as one network or separate subnetworks. If set to zero the individual subnets specified in rtquake.par will trigger individually based on the minimum number of triggers in the line NETWORK name n, for example: NETWORK Chile 6, where 6 specify the minimum number of triggers to record an event for this subnet. If set to a positive number, all subnets will be treated as on network and the minimum number of triggers to record an event will be the number specified here.
MAILn	Send mail to address.
DELAY_BUFFER	Maximum number of minutes in delay buffer.
MINUTE_NOW	Minute in delay buffer defined as current time.
DET_DELAY	Number of minutes delay before network detection.
APW	Array Propagation Window. Network detection takes place. inside this time window just after the DET_DELAY minutes.
SECONDS2SHUFLE	Number of seconds the delay buffer is shifted.
PRE_EVENT	Number of seconds to record before the trigger.
POST_EVENT	Number of seconds to record after the event has de-triggered.
HELI_DAYS	Number of days to keep helicorder plots to keep at all time.
FILTERWINDOW	FilterPicker, do not change
LTWINDOW	FilterPicker, do not change
THRESHOLD1	FilterPicker, do not change
THRESHOLD2	FilterPicker, do not change
TUPEVENT	FilterPicker, do not change
SOUND	Play sound when trigger
PRINTING	Print debug information, do not change

4.3 Station and Network configuration

Before starting RTQUAKE a configuration file must be present in a sub directory of /home/seismo/mydir/par. The user must create this sub directory. The name of the sub directory can be any legal name, but it is recommended to use a name reflecting for example the name of your network, geographical area or purpose of the configuration: NNSN, GEO1, EXAMPLE etc. This name will later be used when starting RTQUAKE.
In this sub directory the user has to create a file called rt_config (or modify an example file). The file must follow the following format described below. The lines marked with bold types are keyword lines and must be present.

rt_config parameter file:

Below is shown part of an example configuration file and after that the explanation. The lines are numbered to help the explanation, the numbers are not part of the file.

```
 1. SERVERS
 2. S01 139.17.3.177
 3. S02 rtserve.iris.washington.edu
 4. --------------------------------------
 5. ALLSUBNETS
 6. --------------------------------------
 7. NETWORK IPOC 7
 8. NW STAT  LOC CMP FL    FH     STA   LTA     T-ON  T-OFF SERVER
 9. CX PB01  ..  BHZ 2.0   8.0    2.0   100.0   3.5   1.5   S01
10. ---------------------------------------------------------
11. RECORD IPOC
12. NW STAT  LOC CMP SERVER
13. CX PB01  ..  BHZ S01
14. # This is a comment
15. CX PB01  ..  BHN S01
16. CX PB01  ..  BHE S01
17. IU LVC   00  BHZ S02
18. -----------------------------------------------------------
```

Line 1. Keyword line: must contain the word SERVERS only

Line 2. User line that contains 2 text strings:
The first string is a fixed form word naming and numbering the servers. The first letter must
be 'S' and the next 2 is the numbering 01,02……99.
The second string is the actual SeedLink address.

Line 3. User line that contains 2 text strings:
The first string is a fixed form word naming and numbering the servers. The first letter must
be 'S' and the next 2 is the numbering 01,02……99.
The second string is the actual SeedLink address.

Line 4. Keyline. Obligatory after the server definition.

Line 5. Keyword. Must be present.

Line 6. Keyline. Obligatory.

Line 7. First word is a keyword. Must be present. The second string can be used to give the network or subnet a name (not used but something must be there) Third string is the minimum number of triggers to record an event based on the stations defined in this section (a subnet).

Line 8. This line is a key line. Must be present. This is a header line to explain the input for the lines in this section.

NW	: network code
STAT	: station code
LOC	: location code (no location code must be marked with '..', two dots)
CMP	: component code
FL	: low-pass filter for detection, floating point number
FH	: high-pass filter for detection, floating point number
STA	: length of STA in seconds, floating point number
LTA	: length of LTA in seconds, floating point number
T-ON	: STA/LTA level to trigger
T_OFF	: STA/LTA level to de-trigger
SERVER	: name of server from where to get the data (S01, S02….)

Line 9. Channel definitions for the network

Line10. Keyline. Obligatory.

Line11. First word is a keyword. Must be present. The second string any name to identify network.

Line12. This line is a key line. Must be present. This is a header line to explain the input for the lines in this section.

NW	: network code
STAT	: station name
LOC	: location code (no location must be marked with '..', two dots)
CMP	: component code
SERVER	: name of server from where to get the data (S01, S02….)

Line13. Station definitions for components to be recorded.

Line14. A '#' in column one means that this line is a comment. It can hold whatever information. One use can be to exclude a component from triggering or recording due to noisy data.

Line15. Station definitions for components to be recorded.

Line16. Station definitions for components to be recorded.

Line17. Station definitions for components to be recorded.

Line18. Keyline. Obligatory.

Below some examples of different configurations with some comments:

DEMO1/rt_config

The example below is the configuration file for the test run.
Here we define one SeedLink server from where we can read all data from all stations defined. As only one network (one subnet) is defined we will treat the defined network as one network.
The trigger algorithm will use the components defined under the key line
NETWORK IPOC 7 as input, and the specified filters, STAs etc. will be used. Data will be read from S01 as specified under SERVER.
When we have 7 or more single-channel triggers on the components specified, the components specified under the keyword RECORD be retrieved and stored.

```
SERVERS
S01 139.17.3.177
--------------------------------------
ALLSUBNETS
--------------------------------------
NETWORK IPOC 7
NW STAT   LOC CMP FL    FH     STA   LTA     T-ON  T-OFF SERVER
CX PB01   ..  BHZ 2.0   8.0    2.0   100.0   3.5   1.5   S01
CX PB02   ..  BHZ 2.0   8.0    2.0   100.0   3.5   1.5   S01
CX PB03   ..  BHZ 2.0   8.0    2.0   100.0   3.5   1.5   S01
CX PB04   ..  BHZ 2.0   8.0    2.0   100.0   3.5   1.5   S01
CX PB05   ..  BHZ 2.0   8.0    2.0   100.0   3.5   1.5   S01
CX PB06   ..  BHZ 2.0   8.0    2.0   100.0   3.5   1.5   S01
CX PB07   ..  BHZ 2.0   8.0    2.0   100.0   3.5   1.5   S01
CX PB08   ..  BHZ 2.0   8.0    2.0   100.0   3.5   1.5   S01
CX PB09   ..  BHZ 2.0   8.0    2.0   100.0   3.5   1.5   S01
CX PB10   ..  BHZ 2.0   8.0    2.0   100.0   3.5   1.5   S01
CX PB11   ..  BHZ 2.0   8.0    2.0   100.0   3.5   1.5   S01
CX PB12   ..  BHZ 2.0   8.0    2.0   100.0   3.5   1.5   S01
CX PB14   ..  BHZ 2.0   8.0    2.0   100.0   3.5   1.5   S01
CX PB15   ..  BHZ 2.0   8.0    2.0   100.0   3.5   1.5   S01
CX PB16   ..  BHZ 2.0   8.0    2.0   100.0   3.5   1.5   S01
CX MNMCX ..  BHZ 2.0   8.0    2.0   100.0   3.5   1.5   S01
CX PATCX ..  BHZ 2.0   8.0    2.0   100.0   3.5   1.5   S01
CX PSGCX ..  BHZ 2.0   8.0    2.0   100.0   3.5   1.5   S01
---------------------------------------------------------
RECORD IPOC
NW STAT   LOC CMP SERVER
CX PB01   ..  BHZ S01
CX PB01   ..  BHN S01
CX PB01   ..  BHE S01
CX PB02   ..  BHZ S01
CX PB02   ..  BHN S01
CX PB02   ..  BHE S01
CX PB03   ..  BHZ S01
CX PB03   ..  BHN S01
CX PB03   ..  BHE S01
CX PB04   ..  BHZ S01
CX PB04   ..  BHN S01
CX PB04   ..  BHE S01
CX PB05   ..  BHZ S01
CX PB05   ..  BHN S01
CX PB05   ..  BHE S01
CX PB06   ..  BHZ S01
CX PB06   ..  BHN S01
CX PB06   ..  BHE S01
CX PB07   ..  BHZ S01
CX PB07   ..  BHN S01
CX PB07   ..  BHE S01
CX PB08   ..  BHZ S01
CX PB08   ..  BHN S01
CX PB08   ..  BHE S01
CX PB09   ..  BHZ S01
CX PB09   ..  BHN S01
CX PB09   ..  BHE S01
CX PB10   ..  BHZ S01
CX PB10   ..  BHN S01
CX PB10   ..  BHE S01
CX PB11   ..  BHZ S01
CX PB11   ..  BHN S01
CX PB11   ..  BHE S01
CX PB12   ..  BHZ S01
CX PB12   ..  BHN S01
CX PB12   ..  BHE S01
CX PB14   ..  BHZ S01
CX PB14   ..  BHN S01
CX PB14   ..  BHE S01
CX PB15   ..  BHZ S01
```

```
CX PB15  ..  BHN S01
CX PB15  ..  BHE S01
CX PB16  ..  BHZ S01
CX PB16  ..  BHN S01
CX PB16  ..  BHE S01
CX MNMCX ..  BHZ S01
CX MNMCX ..  BHN S01
CX MNMCX ..  BHE S01
CX PATCX ..  BHZ S01
CX PATCX ..  BHN S01
CX PATCX ..  BHE S01
CX PSGCX ..  BHZ S01
CX PSGCX ..  BHN S01
CX PSGCX ..  BHE S01
```

DEMO2/rt_config

In the example configuration below we define 2 different SeedLink servers from where we want to read data.
Recorded events will also include data from both SeedLink servers.
The use of comments is included.
For the LVC stations different filters and trigger criteria has been included to show the use of individual parameters for each component of data.

```
SERVERS
S01 139.17.3.177
S02 rtserve.iris.washington.edu
--------------------------------------
ALLSUBNETS
--------------------------------------
NETWORK CHILE1 6
NW STAT  LOC CMP FL    FH     STA   LTA     T-ON  T-OFF SERVER
CX PB01  ..  BHZ 2.0   8.0    2.0   100.0   3.5   1.5   S01
CX PB02  ..  BHZ 2.0   8.0    2.0   100.0   3.5   1.5   S01
CX PB03  ..  BHZ 2.0   8.0    2.0   100.0   3.5   1.5   S01
CX PB04  ..  BHZ 2.0   8.0    2.0   100.0   3.5   1.5   S01
CX PB05  ..  BHZ 2.0   8.0    2.0   100.0   3.5   1.5   S01
CX PB06  ..  BHZ 2.0   8.0    2.0   100.0   3.5   1.5   S01
CX PB07  ..  BHZ 2.0   8.0    2.0   100.0   3.5   1.5   S01
# Next component commented out to show use of comments in file
#CX PB08  ..  BHZ 2.0   8.0    2.0   100.0   3.5   1.5   S01
CX PB09  ..  BHZ 2.0   8.0    2.0   100.0   3.5   1.5   S01
CX PB10  ..  BHZ 2.0   8.0    2.0   100.0   3.5   1.5   S01
# The following 3 components have different filters and triggers
IU LVC   00  BHZ 2.1   8.1    2.0   100.0   2.5   1.5   S02
IU LVC   00  BH1 2.2   8.2    2.0   100.0   3.5   1.5   S02
IU LVC   00  BH2 2.3   8.3    2.0   100.0   4.5   1.5   S02
---------------------------------------------------------
RECORD CHILE1
NW STAT  LOC CMP SERVER
CX PB01  ..  BHZ S01
CX PB01  ..  BHN S01
CX PB01  ..  BHE S01
CX PB02  ..  BHZ S01
CX PB02  ..  BHN S01
CX PB02  ..  BHE S01
CX PB03  ..  BHZ S01
CX PB03  ..  BHN S01
CX PB03  ..  BHE S01
CX PB04  ..  BHZ S01
CX PB04  ..  BHN S01
CX PB04  ..  BHE S01
CX PB05  ..  BHZ S01
CX PB05  ..  BHN S01
CX PB05  ..  BHE S01
```

```
CX PB06   ..  BHZ S01
CX PB06   ..  BHN S01
CX PB06   ..  BHE S01
CX PB07   ..  BHZ S01
CX PB07   ..  BHN S01
CX PB07   ..  BHE S01
CX PB08   ..  BHZ S01
CX PB08   ..  BHN S01
CX PB08   ..  BHE S01
CX PB09   ..  BHZ S01
CX PB09   ..  BHN S01
CX PB09   ..  BHE S01
CX PB10   ..  BHZ S01
CX PB10   ..  BHN S01
CX PB10   ..  BHE S01
IU LVC    00  BHZ S02
IU LVC    00  BH1 S02
IU LVC    00  BH2 S02
------------------------------------------------------------
```

DEMO3/rt_config

In the example configuration below we define 2 different SeedLink servers from where we want to read data.
We define 2 different networks that overlaps and that will trigger and record individually.
To have the configuration to treat the two networks as one, the ALLSUBNETS 0 in rtquake.par should be changed to for example ALLSUBNETS 6 where 6 is the minimum number of triggers for the whole network. The minimum number of triggers defined for each network is overridden by this parameter.

```
SERVERS
S01 rtserve.iris.washington.edu
S02 139.17.3.177:18000
--------------------------------------
ALLSUBNETS
--------------------------------------
NETWORK CHILE1 4
NW STAT   LOC CMP FL    FH     STA   LTA     T-ON  T-OFF SERVER
IU LVC    00  BHZ 2.0   8.0    2.0   100.0   3.5   1.5   S01
IU LVC    00  BH1 2.0   8.0    2.0   100.0   3.5   1.5   S01
IU LVC    00  BH2 2.0   8.0    2.0   100.0   3.5   1.5   S01
CX PB01   ..  BHZ 2.0   8.0    2.0   100.0   3.5   1.5   S02
CX PB02   ..  BHZ 2.0   8.0    2.0   100.0   3.5   1.5   S02
CX PB03   ..  BHZ 2.0   8.0    2.0   100.0   3.5   1.5   S02
----------------------------------------------------------
RECORD CHILE1
NW STAT   LOC CMP SERVER
IU LVC    00  BHZ S01
IU LVC    00  BH1 S01
IU LVC    00  BH2 S01
--------------------------------------------------------------
NETWORK CHILE2 4
NW STAT   LOC CMP FL    FH     STA   LTA     T-ON  T-OFF SERVER
CX PB01   ..  BHZ 2.0   8.0    2.0   100.0   3.5   1.5   S02
CX PB02   ..  BHZ 2.0   8.0    2.0   100.0   3.5   1.5   S02
CX PB03   ..  BHZ 2.0   8.0    2.0   100.0   3.5   1.5   S02
CX PB04   ..  BHZ 2.0   8.0    2.0   100.0   3.5   1.5   S02
CX PB05   ..  BHZ 2.0   8.0    2.0   100.0   3.5   1.5   S02
CX PB06   ..  BHZ 2.0   8.0    2.0   100.0   3.5   1.5   S02
----------------------------------------------------------
RECORD CHILE2
NW STAT   LOC CMP SERVER
CX PB01   ..  BHZ S02
CX PB01   ..  BHN S02
```

```
CX PB01  ..  BHE S02
CX PB02  ..  BHZ S02
CX PB02  ..  BHN S02
CX PB02  ..  BHE S02
CX PB03  ..  BHZ S02
CX PB03  ..  BHN S02
CX PB03  ..  BHE S02
CX PB04  ..  BHZ S02
CX PB04  ..  BHN S02
CX PB04  ..  BHE S02
CX PB05  ..  BHZ S02
CX PB05  ..  BHN S02
CX PB05  ..  BHE S02
CX PB06  ..  BHZ S02
CX PB06  ..  BHN S02
CX PB06  ..  BHE S02
IU LVC   00  BHZ S01
IU LVC   00  BH1 S01
IU LVC   00  BH2 S01
-----------------------------------------------------------
```

4.4 Configuration files for continuous plot

The RTNET module plots selected components from seismic stations in near-real time.
RTNET needs 2 parameter files, one for defining the input streams of data and another to define the actual components to plot. The names for these files are streams_plot and stations_plot respectively and are stored in /home/seismo/mydir/DEMO1. The two files includes the same streams and stations that were configured in the /home/seismo/mydir/DEMO1/rt_config file.

streams_plot

Each line is in standard SeedLink format, but each station component and location must be included.

```
First 13 characters must be formatted as follows:
NN      Network name
-       Space
SSSSS Station name 5 characters
LL    Location 2 characters
CCC   Component 3 characters

NN-SSSSSLLCCC
CX PB01   BHZ
CX PB02   BHZ
CX PB03   BHZ
CX PB04   BHZ
CX PB05   BHZ
CX PB06   BHZ
CX PB07   BHZ
CX PB08   BHZ
CX PB09   BHZ
CX PB10   BHZ
CX PB11   BHZ
CX PB12   BHZ
CX PB14   BHZ
CX PB15   BHZ
CX PB16   BHZ
CX MNMCX  BHZ
```

```
CX PATCX  BHZ
CX PSGCX  BHZ
```

stations_plot

```
First 10 characters must be formatted as follows:
SSSSS Station name 5 characters
LL    Location 2 characters
CCC   Component 3 characters

SSSSSLLCCC

PB01   BHZ PB01 Huatacondo
PB02   BHZ PB02 Salar Grande
PB03   BHZ PB03 El Tigre
PB04   BHZ PB04 Mantos de la Luna
PB05   BHZ PB05 Michilla
PB06   BHZ PB06 Pedro de Valdivia
PB07   BHZ PB07 Cerro Tatas
PB08   BHZ PB08 Macaya
PB09   BHZ PB09 Quillagua
PB10   BHZ PB10 Juan Lopez
PB11   BHZ PB11 Quebrada Aricilda
PB12   BHZ PB12 Cerro Caramaca
PB14   BHZ PB14 Cerro Paranal
PB15   BHZ PB15 Sierra Gorda
PB16   BHZ PB16 Cerro Chaquipina
MNMCX  BHZ MNMCX Minimi
PATCX  BHZ PATCX Patache
PSGCX  BHZ PSGCX Pisagua
```

4.5 Configuration files helicorder plots

The three modules RT24, RTDR24 and RTDRUM are used to create helicorder plots of unfiltered and filtered data from streams from a SeedLink server. Two parameter files are used as input, one to define the different streams to read and another to select the actual components to plot. See chapter 16.

streams_heli This file is in standard SeedLink format and lists the stations and components that will be read from the SeedLink server for plotting. Stored in /mydir/par/DEMO1.

```
CX PB01 BHZ
CX PB02 BHZ
CX PB03 BHZ
CX PB04 BHZ
CX PB05 BHZ
CX PB06 BHZ
CX PB07 BHZ
CX PB08 BHZ
CX PB09 BHZ
```

```
CX PB10 BHZ
CX PB11 BHZ
CX PB12 BHZ
```

stations_heli

This file specifies the components that will be generated as helicorder plots. The content is station, location, component, amplification factor unfiltered data, amplification factor filtered data, low-pass frequency, high-pass frequency and the name to appear on the helicorder plot. The amplification factors can be modified dynamically when the system is running. This way the helicorder plot can be checked for reasonable amplitudes on the plot.

```
First 10 characters must be formatted as follows:
SSSSS Station name 5 characters
LL    Location 2 characters
CCC   Component 3 characters
AMP1  Amplification factor raw data
AMP2  Amplification factor filtered data
FL    Low pass filter
FH    High pass filter
```

```
SSSSSLLCCC   AMP1   AMP2   FL  FH  NAME
PB01   BHZ 0.0100 0.0300 2.0 8.0 Huatacondo
PB02   BHZ 0.0100 0.0300 2.0 8.0 Salar Grande
PB03   BHZ 0.0100 0.0300 2.0 8.0 El Tigre
PB04   BHZ 0.0100 0.0300 2.0 8.0 Mantos de la Luna
PB05   BHZ 0.0100 0.0300 2.0 8.0 Michilla
PB06   BHZ 0.0100 0.0300 2.0 8.0 Pedro de Valdivia
PB07   BHZ 0.0100 0.0300 2.0 8.0 Cerro Tatas
PB08   BHZ 0.0100 0.0300 2.0 8.0 Macaya
PB09   BHZ 0.0100 0.0300 2.0 8.0 Quillagua
PB10   BHZ 0.0100 0.0300 2.0 8.0 Juan Lopez
PB11   BHZ 0.0100 0.0300 2.0 8.0 Quebrada Aricilda
PB12   BHZ 0.0100 0.0300 2.0 8.0 Cerro-Caramaca
```

4.6 Parameterfile web-pages

Two web pages are available to monitor the events recorded and located by RTQUAKE. Both html files, LAST_TRIG.html and rt_screen1.html, are stored in /home/seismo/mydir/map and use the same parameter file that are also located in /home/seismo/mydir/map. The parameter file is called: map.par

```
NUMBER OF EVENTS           # Number of events back in time to plot
100
ZOOM FACTOR                # Google maps zoom factor
7
LATITUDE-LONGITUDE-GRID    # Add latitude/longitude grid or not
1
MAPTYPE 0-SAT,1-TER        # Maptype id Google maps: 0-SATELITE, 1-TERRAIN
0
BLINK                      # Blinking red marker, 0-blinking, 1-no blinking
0
```

4.7 Aliases and Scripts

alias rtstart='$RTQUAKE_TOP/com/rtquake_start'

The command rtstart starts the script rtquake_start. This is the start script to start RTQUAKE and the file actually installed is set up for the test run and can be used as a recipe for the user to set up the actual network. As you see in the example below, the DEMO1 subdirectory is used.

```
killall rtdet
killall rtdly
killall rtnet
$RTQUAKE_TOP/bin/rtdet -par 0 -cfg DEMO1 &
sleep 2
$RTQUAKE_TOP/bin/rtdly &
sleep 2
$RTQUAKE_TOP/bin/rtnet -x 650 -y 750 -xo 150 -yo 150 -d -m 10 -n 20 -fl 2.0
-fh 8.0 -l DEMO1/streams_plot -f DEMO1/stations_plot 139.17.3.177:18000 &
```

First any running RTQUAKE modules are stopped, then the rtdet module is started with the configuration given in DEMO1, then the rtdly module is started and finally the rtnet module is started. See module descriptions for more details on the parameters for each module.

alias rtstop='$RTQUAKE_TOP/com/rtquake_stop'

Activate the rtquake_stop script. See 4.6

The command rtstop stops all running RTQUAKE modules.

```
killall rtdet
killall rtsnr
killall rtmon
killall rtnet
killall rtdly
killall rt24
killall rtdrum
killall rtdr24
```

alias rtheli1='$RTQUAKE_TOP/com/rtquake_heli_tst1'

The command rtheli1 starts the generation of the helicorder plots, one plot per day.

rt24 –heli 1 –logol logo_left_def.gif –logor logo_right_def.jpg –col 5 –to_wi 1200 –fr_hg 600 –mt 15 –l DEMO1/streams_heli –f DEMO1/stations_heli 139.17.177

See module descriptions for more details on the parameters for each module.

alias rtheli2='$RTQUAKE_TOP/com/rtquake_heli_tst2'

The command rtheli2 starts the generation of helicorder plots always showing the last 24 hours.

rt24 –heli 0 –logol logo_left_def.gif –logor logo_right_def.jpg –col 0 –to_wi 1200 –fr_hg 600 –mt 15 –l DEMO1/streams_heli –f DEMO1/stations_heli 139.17.177

See module descriptions for more details on the parameters for each module.

5 START SCRIPT TEST RUN

When starting the main modules in RTQUAKE with the start script, several parameters are given in the script. Below is the example script rtquake_start with explanation of the parameters used.

```
$RTQUAKE_TOP/bin/rtdet -cfg DEMO1 &
sleep 2
$RTQUAKE_TOP/bin/rtdly &
sleep 2
$RQUAKE_TOP/bin/rtnet -x 650 -y 500 -xo 150 -yo 150 -d -m 10 -n 20 -l
DEMO1/streams_plot -f DEMO1/stations_plot  139.17.3.177:18000 &
```

Explanation of the parameters used in the rtquake_start:

rtdet is the acquisition and detection module. The different parameters have the following meaning:

-cfg DEMO1	Name of profile catalog under: /home/mydir/rtquake/par that contains the configuration files.

rtdly is graphically monitoring the components that are defined in the configuration file for triggers and the durations of the triggers. The module takes the following parameters:

-par 0	Instance number of the rtdet module.

rtnet plots continuous data of specified components. The module takes the following parameters:

-x 650	x size of plot frame in pixels
-y 500	y size of plot frame in pixels
-xo 150	x position of upper left corner of plot frame.
-yo 150	y position of upper left corner of plot frame.
-d	Plot geographical name instead of station name as specified in the stations_plot file, see 4.4.
-m 10	Number of minutes on screen
-fl 2.0	Specifies lower frequency in band pass filter
-fh 8.0	Specifies higher frequency in band pass filter
-l DEMO1/streams_plot	Streams or components to read from SeedLink server
-f DEMO1/stations_plot	Stations to plot
-n 20	Number of stations to plot
139.17.3.177:18000	IP address and port number for the SeedLink server.

Be aware that the IP number in the example is ONLY valid for the example configuration.

6 CONFIGURATION OF A NEW NETWORK

To configure your own network, first create a new sub-catalogue under the /mydir/par catalogue that identify your network. You can use the parameter files used in the test example as a recipe. Make a copy the files rt_config, streams_plot.demo1 and stations_plot.demo1 in the par/DEMO1 directory with a file-extension changed to reflect the name of the network or your institution. Edit the files to fit your stations, components etc. Check the rtquake.par file. As a start most default values can be used. Remember to include your station coordinates (if auto locations are to be made) in the SEISAN STATION0.HYP file if not already there in you SEISAN installation. Remember to include the response files in the SEISAN CAL catalogue if local magnitude is to be calculated. The start script should also be modified to fit your configuration.

Automatic start of RTQUAKE with a cron job

RTQUAKE can be set up to start automatically when the computer starts up. Modules may also stop due to different reasons and should then be restarted. A cron job can do this by checking that a specific module is active at regular time intervals.

```
cron_restart.csh

#!/bin/csh
#
PROCESS='rtdet'
if ps ax | grep -v grep | grep $PROCESS > /dev/null
then
exit
else
#echo "$PROCESS is not running"
#start the process
#echo "Start $PROCESS !"
#echo "put in the start command here"
#/home/seismo/rtquake/par/STARTUP-SCRIPT-FOR-RTDET > /dev/null &
/home/seismo/mydir/par/start_rtdet.csh > /dev/null &
fi
```

A crontab job to restart the rtdet module can be created by starting the script above for example every 5 minutes:

```
*/5 * * * * /home/seismo/mydir/par/cron_restart.csh
```

Both scripts cron_restart.csh and start_rtdet must be changed to “executable” to function.
A typical start_rtdet.csh would look like:

```
#!/bin/csh
source /home/seismo/mydir/com/setup_rt.csh
source /home/seismo/seismo/COM/SEISAN.csh
/home/seismo/mydir/bin/rtdet –par 0 –cfg DEMO1 &
```

NOTE: Be aware that the name of the cron script can NOT contain the name of the module you want to restart. cron_restart.csh is ok, restart_rtdet.csh is NOT ok if you want to restart the module rtdet.

For the bash shell the cron script will look like this:

```
cron_restart.bash

#!/bin/bash
#
PROCESS='rtdet'
if ps ax | grep -v grep | grep $PROCESS > /dev/null
then
exit
else
#echo "$PROCESS is not running"
#start the process
#echo "Start $PROCESS !"
#echo "put in the start command here"
#/home/seismo/rtquake/par/STARTUP-SCRIPT-FOR-RTDET > /dev/null &
/home/seismo/mydir/par/start_rtdet > /dev/null &
fi
```

```
*/5 * * * * /home/seismo/mydir/par/cron_restart.bash
```

```
#!/bin/bash
source /home/seismo/mydir/com/setup_rt.bash
source /home/seismo/seismo/COM/SEISAN.bash
/home/seismo/mydir/bin/rtdet –par 0 –cfg DEMO1&
```

7 DIRECTORY STRUCTURE

If we assume a top directory: /home/seismo, the following directory structure will be created:

/home/seismo/mydir	Main directory containing all subdirectories, include files, makefile
/home/seismo/mydir/bin	Executables
/home/seismo/mydir/com	Environment scripts and test scripts
/home/seismo/mydir/doc	All documentation in word or pdf format.
/home/seismo/mydir/heli	Example configuration, scripts etc for helicorder plots
/home/seismo/mydir/inc	Include files for RTQUAKE
/home/seismo/mydir/libslink	Libraries and include files for SeedLink library.
/home/seismo/mydir/loc	Static Google map links for plotting automatic locations
/home/seismo/mydir/map	Temporary hypocenter files for generating maps, station files
/home/seismo/mydir/par	Parameter files for the different modules
/home/seismo/mydir/par/DEMO1	Test configuration (Test run example)
/home/seismo/mydir/par/DEMO2	Demo configuration (Example)
/home/seismo/mydir/par/DEMO3	Demo configuration (Example)
/home/seismo/mydir/picker	Source files Filter-picker, include files and make file
/home/seismo/mydir/req	Request files. One file for each triggered event. The files are executable and can be run to extract the event file if it for some reason was not recorded at trigger time, for example: delayed data. Depending on the size of the segment buffer in the SeedLink server, this can be done several days after the time of the trigger.
/home/seismo/mydir/rt/latency	Catalog for latency of arriving SeedLink data from RTTIME module
/home/seismo/mydir/rt/png	Catalog for unfiltered helicorder plots
/home/seismo/mydir//rt/png_filt	Catalog for filtered helicorder plots
/home/seismo/mydir//rt/tmp	Catalog for unfiltered datafiles to make helicorder plots
/home/seismo/mydir//rt/tmp_filt	Catalog for filtered datafiles to make helicorder plots
/home/seismo/mydir/rtdet	Source files and make file for main module
/home/seismo/mydir/seedlink	Distribution and user manual SeedLink
/home/seismo/mydir/tmp	Temporary files. Removed after x days specified in the /home/seismo/mydir/com/rtquake.par file.
/home/seismo/mydir/utils1	Source files main monitoring utilities, makefile,
/home/seismo/mydir/utils2	Source files monitoring utilities, makefile.
/home/seismo/mydir/wrk	Work catalogue for testing of software

8 GENERAL DESCRIPTION AND MODULE OVERVIEW

In general the modules are dependent on data recorded by a SeedLink server. The server can be located locally or remotely as long as you as the user have access to the server through internet. Some modules are written as clients to the SeedLink server to extract data in near real time while others are used to monitor this activity. Another group of modules monitor the activity on the SeedLink server.
The modules can be categorized into 3 different groups

- **Detection, Recording and Monitoring**

RTDET Detection and recording of events. Reads data from local or remote SeedLink server.

RTPICK Automatic phase-picking and auto-location.

RTSNR Graphical monitoring of R=STA/LTA of each station, each parameter set in RTPICK.

RTDLY Monitor onsets of triggers and duration of triggers for the individual components specified in the detection parameter file.

- **Monitoring of SeedLink**

RTNET Plots selected components in "near-real-time". Reads data from a local or remote SeedLink server.

RTTIME Graphic monitoring of latency of stations transmitting to a SeedLink server.

- **Helicorder plots**

RT24 Generates temporary data files of specified station components. Files are input for RTDR24 that generates helicorder plots. Data are read from local or remote SeedLink server.

RTDRUM Creates helicorder plots of specified station components. One-day.

RTDR24 Creates helicorder plots of specified station components. Last 24 hour.

RTHPLT Creates a menu to plot individual helicorder plots. Two individual html files are generated for raw and filtered data respectively. The routine also removes files older than x days where x is given as a parameter for the routine.

9 DETECTION AND RECORDING OF EVENTS

RTDET – Detection based on input data from a SeedLink server.

RTDET is written as a SeedLink client, and executes under Linux. The program can be run on the same machine as the SeedLink server or remotely.
Up to 10 instances of the program can be executed with different parameter sets.
A SeedLink server will normally hold data from a network covering a larger geographical area. By using different parameter sets, it is possible to divide the network into several sub-nets for detection of more local events.

Each parameter set can have different parameters such as:

different and/or overlapping stations with other parameter sets
different filters
different trigger-ratio
different de-trigger-ratio
different sta & lta length
different parameter sets can record different components.

All parameters for the rtdet module are defined in the rtquake.par file and in the parameter file where stations and networks are defined.

Description of the trigger algorithm.

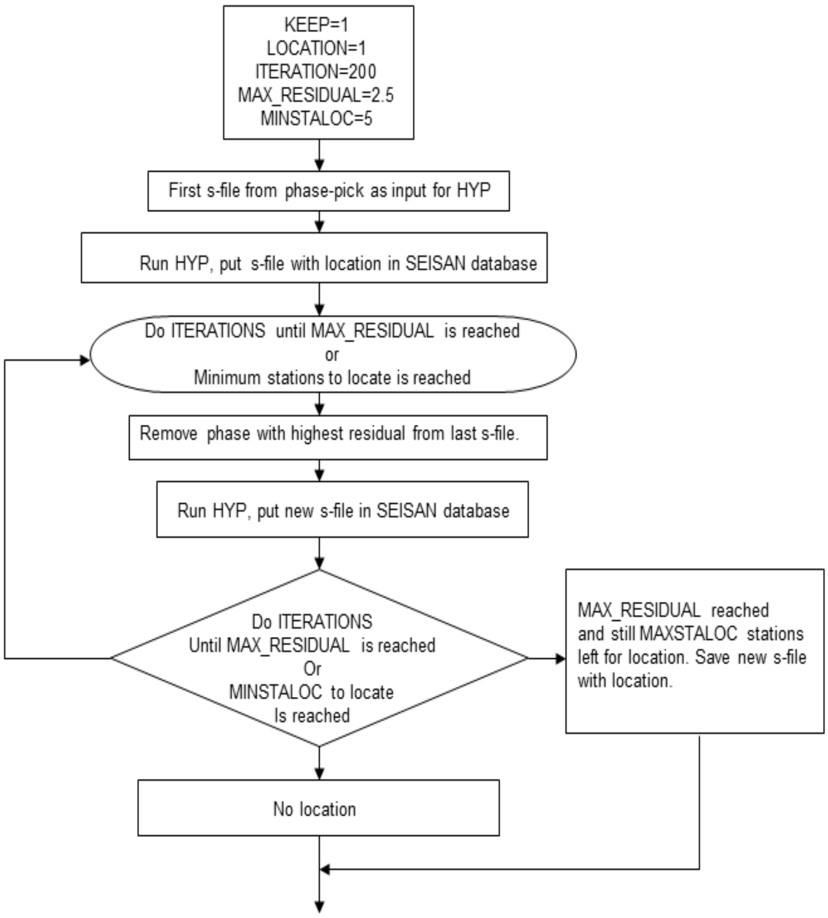
Data buffers from stations specified in the parameter files are read from the SeedLink server. Data from each component are stored in a two-dimensional array, (component, sample). Each instance of the program will keep its private array.
New data are stored in the bottom of the array, while old data are shifted out from the top. Each component is continuously checked for triggers with a standard STA/LTA computation.
A 3-dimensional array (parameterset, channel-number, length in seconds) can hold trigger-times for up to one hour. Trigger times for individual components are stored in this array in the correct place with reference to current UTC time. The array is shifted at regular intervals so that the array always keeps new data at the current UTC time. Network detection is then computed at a later time, controlled by a parameters in the rtquake.par file, see 4.2. Figure 4 above illustrates this
Triggers will exist as valid until they are shifted out of the array-propagation-window, also explained in Figure 4-5 above.
When subnets are defined, each subnet will have its own trigger-thread independent of the others.
The trigger algorithm allows triggers to be detected with a delay in time. This means that for example data from one or more stations arrives with a variable delay due to for example communication problems can be used to correctly trigger an event

10 AUTOMATIC LOCATION

To activate the automatic location in RTQUAKE, several parameters have to be set to correct values in the rtquake.par file, see 4.2. The parameters KEEP and LOCATION must both be set to 1 as explained in 4.2. This will ensure that a s-file will be created with phase-picks from the rtpick module. LOCATION=1 means that location of the event will be initiated. ITERATION, MAX_RESIDUAL and MINSTALOC are used in the iteration process to reduce the number of phases in the s-file that has high residuals. ITERATION=200 means that the program will try with up to 200 iterations to reduce the average residual to MAX_RESIDUAL=2.5. MINSTALOC=5 means that the program will need minimum 5 stations left to do location when the MAX_RESIDUAL=2.5 has been reached. See 4.2 rtquake.par.

For SEISAN the parameter "RMS residual low limit for bisquare weighting for local events" (RESET TEST(36)) should be set to e.g. 6 s i in the SEISAN0.HYP file. This means that when the RMS of travel time residuals is less than 6 s, residual weighting will start to be used and large outliers will have little or no influence in the location.

The automatic location procedure in RTQUAKE is outlined in the flowchart below:



Below is an example output from the iteration process. In the example the MAX_RESIDUAL is set to 2.5 and the MAXSTALOC to 5 stations. In bold one can see that the average residual is decreasing for each iteration until the 2.5 limit has been reached and there are still 25 stations left for location.

```
MAX_RESIDUAL 2.5


RTPICK: Path+s-filename.............: /home/seismo/snew/REA/TST__/2014/05/05-0805-20L.S201405
RTPICK: S_REC: fullpath.............: /home/seismo/snew/REA/TST__/2014/05/05-0805-20L.S201405
RTPICK: S_REC.......................: Write new s-file header to s-file.
RTPICK: S_REC: Create s-file........: /home/seismo/snew/REA/TST__/2014/05/05-0805-20L.S201405
S_REC: Write new phases to s-file.
 PB01 BZ IP    A   8 5 57.36  270
 PB02 BZ IP    A   8 5 57.31  277
 PB04 BZ IP    A   8 6  8.36  223
 PB05 BZ IP    A   8 6 14.86  263
 PB06 BZ IP    A   8 6 15.86  240
 PB07 BZ IP    A   8 6  2.46  274
 PB08 BZ IP    A   8 5 53.86  275
 PB09 BZ IP    A   8 6  8.50  269
 PB10 BZ IP    A   8 6 22.54  233
 PB11 BZ IP    A   8 5 44.85  309
 PB12 BZ IP    A   8 5 38.55  254
 PB15 BZ IP    A   8 6 22.86  129
 PB16 BZ IP    A   8 5 50.94  293
 MNMCXBZ IP    A   8 5 45.20  318
 PATCXBZ IP    A   8 5 49.60  211
 PSGCXBZ IP    A   8 5 37.20  322
 PB01 BN IS   3A   8 6 24.31
 PB08 BN IS   3A   8 6 21.46
 PB08 BE IS   3A   8 5 54.81
 PB09 BN IS   3A   8 6  9.30
 PB11 BN IS   3A   8 6  3. 0
 PB11 BE IS   3A   8 6  3. 5
 PB12 BN IS   3A   8 5 50.89
 PB12 BE IS   3A   8 5 51. 9
 PB16 BN IS   3A   8 6 13. 9
 PB16 BE IS   3A   8 6 13.79
 MNMCXBN IS   3A   8 6  5.25
 MNMCXBE IS   3A   8 6  4. 0
 PATCXBN IS   3A   8 5 50.30
 PSGCXBN IS   3A   8 5 51. 0
 PSGCXBE IS   3A   8 5 50.80
RTPICK: Create_Sfile..: Locate + new s-file.
RTPICK: comm0.....: rm hyptemp.txt
RTPICK: com10.....: cp /home/seismo/snew/REA/TST__/2014/05/05-0805-20L.S201405 s_org.out
RTPICK: comm1.....: hyp /home/seismo/snew/REA/TST__/2014/05/05-0805-20L.S201405 >> hyptemp.txt
RTPICK: comm2.....: cp hyp.out /home/seismo/snew/REA/TST__/2014/05/05-0805-20L.S201405
RTPICK: comm8..........: cp hyp.out hyp_all.out
RTPICK: Found..........: hyp.out
RTPICK: readings left..: 31 Avg.res:    34.00  phases left: 30 Avg.residual in HYP_NEW:  27.53
RTPICK: readings left..: 30 Avg.res:    27.40  phases left: 29 Avg.residual in HYP_NEW:  21.34
RTPICK: readings left..: 29 Avg.res:    21.34  phases left: 28 Avg.residual in HYP_NEW:  16.28
RTPICK: readings left..: 28 Avg.res:    16.28  phases left: 27 Avg.residual in HYP_NEW:  10.85
RTPICK: readings left..: 27 Avg.res:    10.85  phases left: 26 Avg.residual in HYP_NEW:   6.85
RTPICK: readings left..: 26 Avg.res:     6.85  phases left: 25 Avg.residual in HYP_NEW:   2.68
RTPICK: readings left..: 25 Avg.res:     2.68  phases left: 24 Avg.residual in HYP_NEW:   1.41
RTPICK: STOP iterations. Residual below..: 2.50
RTPICK: Average residual.................: 2.679000
RTPICK: No more iterations...............: Number of stations: 25 Avg: res.:    2.679
RTPICK: comm6............................: cp hyp.out hyp.tmp
RTPICK: com11............................: cp hyp.out /home/seismo/rtquake/map
RTPICK: comm2............: cp hyp.out /home/seismo/snew/REA/TST__/2014/05/05-0805-20L.S201405
RTPICK: SEISAN_TOP.......................: /home/seismo/snew
RTPICK: RTQUAKE_TOP......................: /home/seismo/rtquake
RTPICK: RSS..............................: 2679
RTPICK:..................................: Update map
RTPICK: MAG..............................: 3.4
```

```
RTPICK: mail1...........................: 0
rtn>
```

Below is the final s-file after the iteration and location process:

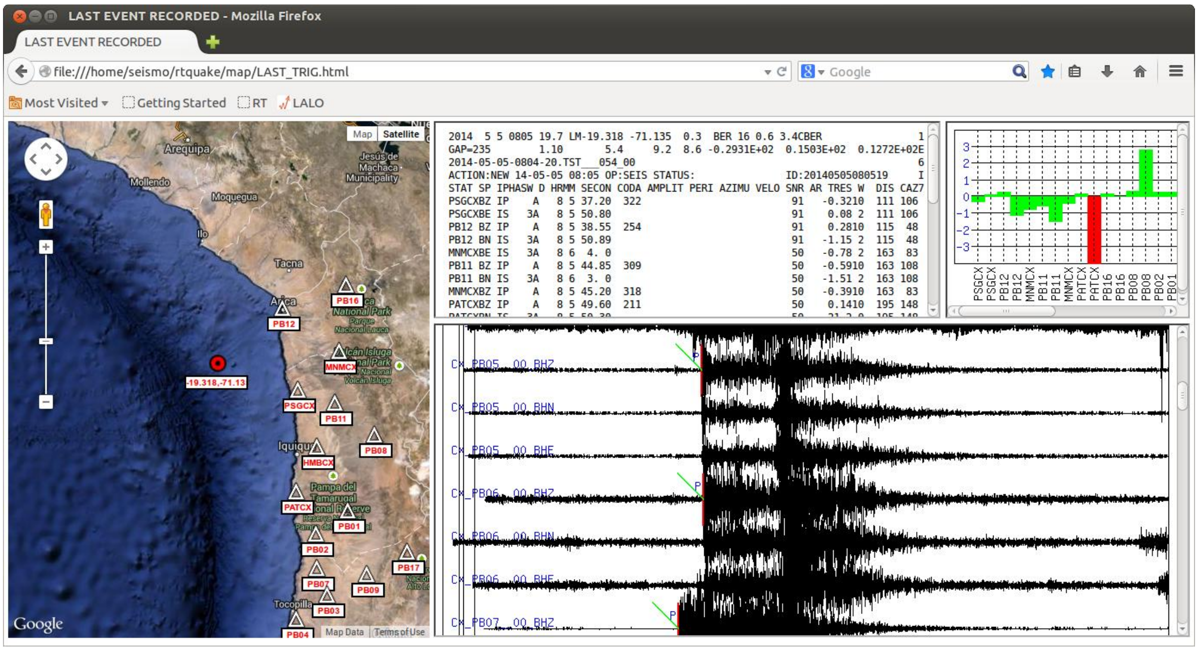
```
rtn>eev 201405050805

 2014  5 Reading events from base TST__   732
#  731   5 May 2014 08:05 19  LM-19.318 -71.135  0.3  N 0.6 3.4CBER   16  ? t

 File name: /home/seismo/snew/REA/TST__/2014/05/05-0805-20L.S201405
2014  5 5 0805 19.7 LM-19.318 -71.135  0.3  BER 16 0.6 3.4CBER                   1
GAP=235        1.10       5.4     9.2  8.6 -0.2931E+02  0.1503E+02  0.1272E+02E
2014-05-05-0804-20.TST___054_00                                                  6
ACTION:NEW 14-05-05 08:05 OP:SEIS STATUS:               ID:20140505080519     I
STAT SP IPHASW D HRMM SECON CODA AMPLIT PERI AZIMU VELO SNR AR TRES W  DIS CAZ7
PSGCXBZ IP    A   8 5 37.20  322                        91    -0.3210  111 106
PSGCXBE IS   3A   8 5 50.80                             91     0.08 2  111 106
PB12 BZ IP    A   8 5 38.55  254                        91     0.2810  115  48
PB12 BN IS   3A   8 5 50.89                             91    -1.15 2  115  48
MNMCXBE IS   3A   8 6  4. 0                             50    -0.78 2  163  83
PB11 BZ IP    A   8 5 44.85  309                        50    -0.5910  163 108
PB11 BN IS   3A   8 6  3. 0                             50    -1.51 2  163 108
MNMCXBZ IP    A   8 5 45.20  318                        50    -0.3910  163  83
PATCXBZ IP    A   8 5 49.60  211                        50     0.1410  195 148
PATCXBN IS   3A   8 5 50.30                             50    -21.2 0  195 148
PB16 BZ IP    A   8 5 50.94  293                        50     0.1610  203  58
PB16 BE IS   3A   8 6 13.79                             50    -0.01 3  203  58
PB08 BZ IP    A   8 5 53.86  275                        50     0.2910  227 114
PB08 BN IS   3A   8 6 21.46                             50     2.80 2  227 114
PB02 BZ IP    A   8 5 57.31  277                        50     0.2510  257 150
  Return to continue, q to return to EEV

PB01 BZ IP    A   8 5 57.36  270                        50     0.2410  257 138
PB01 BN IS   3A   8 6 24.31                             50    -0.52 2  257 138
PB07 BZ IP    A   8 6  2.46  274                        50     0.3510  297 154
PB09 BZ IP    A   8 6  8.50  269                        50     1.27 9  338 145
PB09 BN IS   3A   8 6  9.30                             50    -33.1 0  338 145
PB04 BZ IP    A   8 6  8.36  223                        50    -0.2710  349 163
PB05 BZ IP    A   8 6 14.86  263                        50    -0.4310  403 166
PB06 BZ IP    A   8 6 15.86  240                        50    -0.1510  409 157
PB15 BZ IP    A   8 6 22.86  129                        49     0.0410  464 158
PB10 BZ IP    A   8 6 22.54  233                        49    -0.6410  468 173
```

The web page below (see chapter 3) shows the map with the location, the s-file, a plot of the residuals each component and a plot of the waveform of the event.



11 PROCESSING DETECTIONS WITH SEISAN

SEISAN is intended to be used as the main data-inspection and processing system since the triggered events are written directly out in a SEISAN data base and, if SEISAN is mounted on the same computer as the SeisComp system, SEISAN can also read the SeisComp ring buffer system.

SEISAN trigger files

For each trigger, an S-file is created in the SEISAN data base with P-arrival times, signal duration as well as a reference to the trigger waveform file, see example below. The S-files can be used for SEISAN processing like earthquake location and plotting.

Checking triggered events
Events that have triggered the system can be found by using the command eev. E.g the command 'eev 200905' is used to inspect data for May 2009. This command can be given from any directory.

Plotting triggered events
From eev, give command po and the MULPLT program is started with the current event. The user can now do housekeeping by inspecting events, delete false triggers, and do final registration of the event into the SEISAN-database.

Plotting data with a SeisComp ringbuffer database (the archive)

In SEISAN, 'mulplt' is able to plot from the SeisComp ringbufferes (archive), so that any time-window, from any number of channels, can be seen at the same time. The ringbuffer consists of a flat file system with one channel files one day long (see SeisComp manual).
The channels to be plotted and the location of the archive must have been defined in the SEISAN.DEF file in DAT (see SEISAN manual). The procedure is then:

Start 'mulplt'.
Give option 'arc'
Select start time and interval, the plot will then come up in the usual way with all selected channels. It is now possible to move forwards and backwards in the ring buffer.

It is possible to plot and extract out data from the ringbuffer a couple of minutes after real time.

Extracting data from the archive using 'mulplt':

Use 'Out' function to extract data selected on the screen or use 'Regis' function to extract a waveform file to the WAV directory and create a corresponding new event (S-file) in the data base. This option can be used to recover data if RTDET did not trigger or trigger interval was not correct

The S-file

An example of a name is:

12-1145-22L.199911

The name consists of 'day of month' and time. The L indicates that the event is a local event by default (see SEISAN). On the left hand side is year and month. The S-files are written in ASCII and the format follows the SEISAN-definition. An example for an S-file is shown below:

```
 1999  623  7 5 30.0 L                       BER                              1
 1999-06-23-0705-30S.BERG__003                                                6
 ACTION:NEW 99-06-23 07:05 OP:SEIS STATUS:              ID:19990623070530     I
 STAT SP IPHASW D HRMM SECON CODA AMPLIT PERI AZIMU VELO SNR AR TRES W  DIS CAZ7
 BER  SZ IP    A  0705 30.10   10
 ASK  SZ IP    A  0705 30.10   10
 EGD  SZ IP       0705 30.10   10
```

A full description of the format is found in the SEISAN-manual, so only the important points will be given here. The first line is a header line giving start-time of the recording. The L stands for local event (by default). BER is the station-identifier-code. Line 2 gives the name of the corresponding waveform-file, which normally is located in the directory for event waveform-files. Line 3 is a help line for lines following, which gives the trigger-time for each channel participating in the detection. The duration of the trigger for each channel is given under 'CODA'. These times can be used by SEISAN for locating the event if more than 3 stations are present, and the magnitude is calculated from the coda.

12 MAIL

12.1 Optional : Mail

RTQUAKE has the ability to send mail when an event has been recorded and a preliminary automatic location has been calculated. This means that automatic location must be activated (see 4.2 rtquake.par) and that the location is written in the S-file. Some events may not be located due to unreliable readings and no mail is sent. This option can be activated in the rtquake.par file (see 4.2 rtquake.par). Be aware that this can cause a lot of mails to be sent if the threshold for triggering is low or if noisy signals result in false triggers. A network in a very seismic area will also cause lot of mails. For RTQUAKE to support this option the user has to install the following packages: **ssmtp** and **mutt** (text-based mail client).
SSMTP is a program to deliver an email from a local computer to a configured mailhost (mailhub). It is not a mail server and does not receive mail, expand aliases or manage a queue. One of its primary uses is for forwarding automated email (like system alerts) from your machine and to an external email address.
For the setup below the user must have access to a gmail account for this purpose. It is recommended to create a separate account for these mails. We assume an account: myaccount@gmail.com with a password: mypassword for the example configuration below.
Never use this account and password in the example above for security reasons!!!!!!
As root the user must edit the two ssmpt configuration files to contain the same information as shown below. In the rtquake.par file the user specify the real email address that will receive the mail. The gmail account will just forward the mail.

Edit **/etc/ssmpt/ssmpt.conf**:

```
#
# Config file for sSMTP sendmail
#
# The person who gets all mail for userids < 1000
# Make this empty to disable rewriting.
root=myaccount@gmail.com

# The place where the mail goes. The actual machine name is required no
# MX records are consulted. Commonly mailhosts are named mail.domain.com
mailhub=smtp.gmail.com:587

# Where will the mail seem to come from?
#rewriteDomain=

# The full hostname
hostname=smtp.gmail.com:587
UseSTARTTLS=YES
UseTLS=YES
AuthUser=myaccount
AuthPass=mypassword
AuthMethod=LOGIN

# Are users allowed to set their own From: address?
# YES - Allow the user to specify their own From: address
# NO - Use the system generated From: address
FromLineOverride=NO
```

Edit **/etc/ssmpt/revaliases:**

```
# sSMTP aliases
#
# Format:	local_account:outgoing_address:mailhub
#
# Example: root:your_login@your.domain:mailhub.your.domain[:port]
# where [:port] is an optional port number that defaults to 25.
root:myaccount@gmail.com:smtp.gmail.com:587
mainuser:myaccount@gmail.com:smtp.gmail.com:587
rtquake:myaccount@gmail.com:smtp.gmail.com:587
```

If mail has been configured and is activated in rtquake.par the email will look like this and contain the following information:

TRIGGER
Thu 17/4, 17:37
To: the-address specified in rtquake.par
2 attachments:
ALL.png
hyp.txt

http://maps.googleapis.com/maps/api/staticmap?center=-20.014000,-71.008003&zoom=7&size=900x1000&maptype=hybrid&markers=color:red%7Ccolor:red%7Clabel:Q%7C-20.014000,-71.008003&sensor=false

Clicking on the link will produce a static google map as show in Figure 8. The text "center=-20.014000,-71.008003" is the computed location for the event. The attachment ALL.png contains the plot shown in Figure 9 and hyp.txt (s-file) in Figure 10. The examples are screenshots from a tablet.

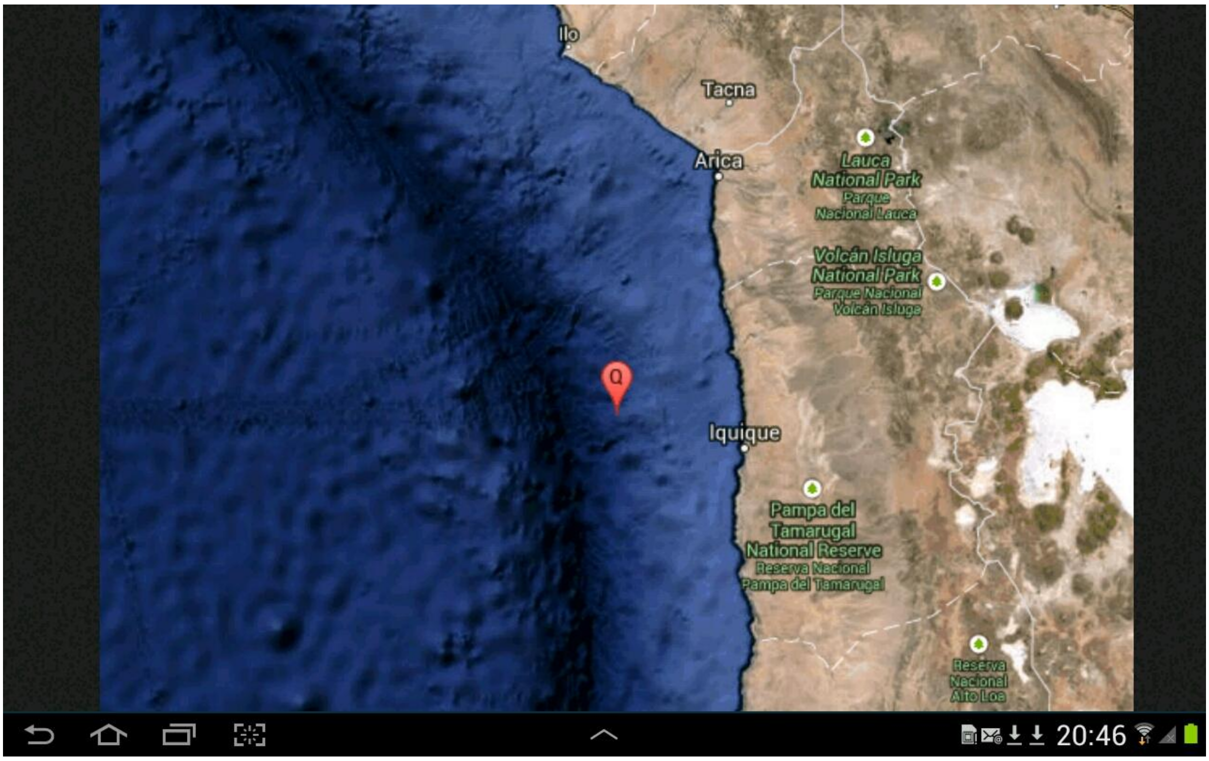


Figure 8 Static google map with suggested location marked with the red symbol.

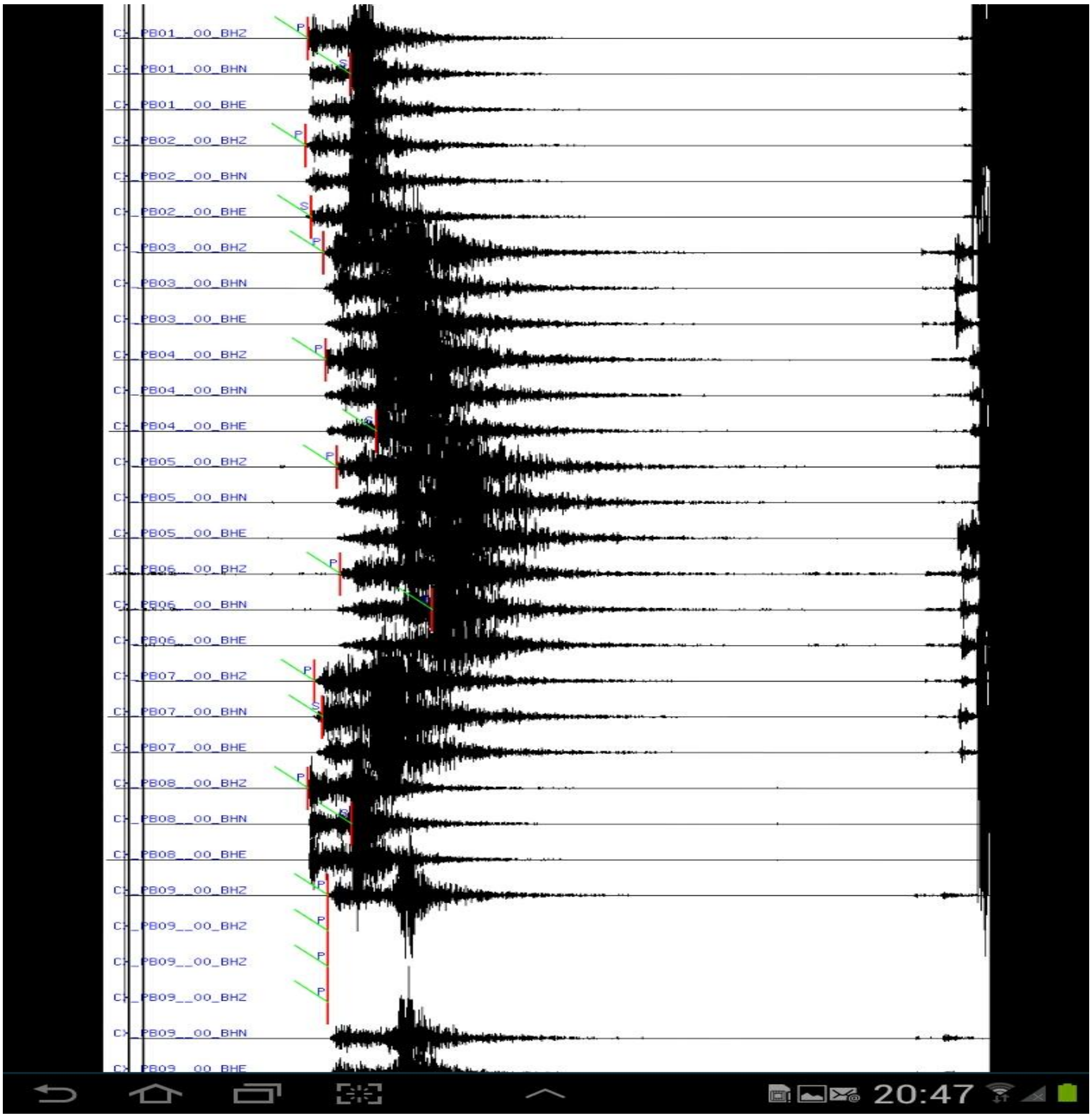


Figure 9 ALL.png attachment showing a plot of recorded stations with phases marked.

```
 2014  418 1628 31.4 LM-20.889 -69.281 15.0  BER  3 0.2 2.9CBER                1
 GAP=249        0.49      22.8     64.6  0.0 -0.1458E+04  0.3186E-01  0.1197E-01E
 2014-04-18-1627-25.TST___054_00                                               6
 ACTION:NEW 14-04-18 16:28 OP:SEIS STATUS:               ID:20140418162825     I
 STAT SP IPHASW D HRMM SECON CODA AMPLIT PERI AZIMU VELO SNR AR TRES W  DIS CAZ7
 PB01 BZ IP    A  1628 36.16  167                        115    -0.2810 27.4 231
 PB02 BZ IP    A  1628 44.46  168                         94     0.2510 79.8 233
 PB02 BN IS   3A  1628 59.26                              94     5.56 0 79.8 233
 PB02 BE IS   3A  1628 59.46                                            79.8 233
 PB11 BZ IP    A  1628 51.75  200                         55     0.0310  131 343
```

Figure 10 The hyp.txt attachment. S-file for the recorded event.

13 RTPICK

RTPICK is started by RTDET if the parameter –aut is set to 1 as described in the test run chapter. RTPICK will try to find p and s phases when possible and will update the s-file for the corresponding event in the SEISAN database. The phase picking algorithm is based on the FilterPicker algorithm (FilterPicker, Lomax et.al.,2011). RTPICK then uses the s-file as input for the hypocenter program in an iterative process to reduce the residuals to a minimum as explained chapter 10. Typical automatic readings are shown in Figure 11 and Figure 12 below.

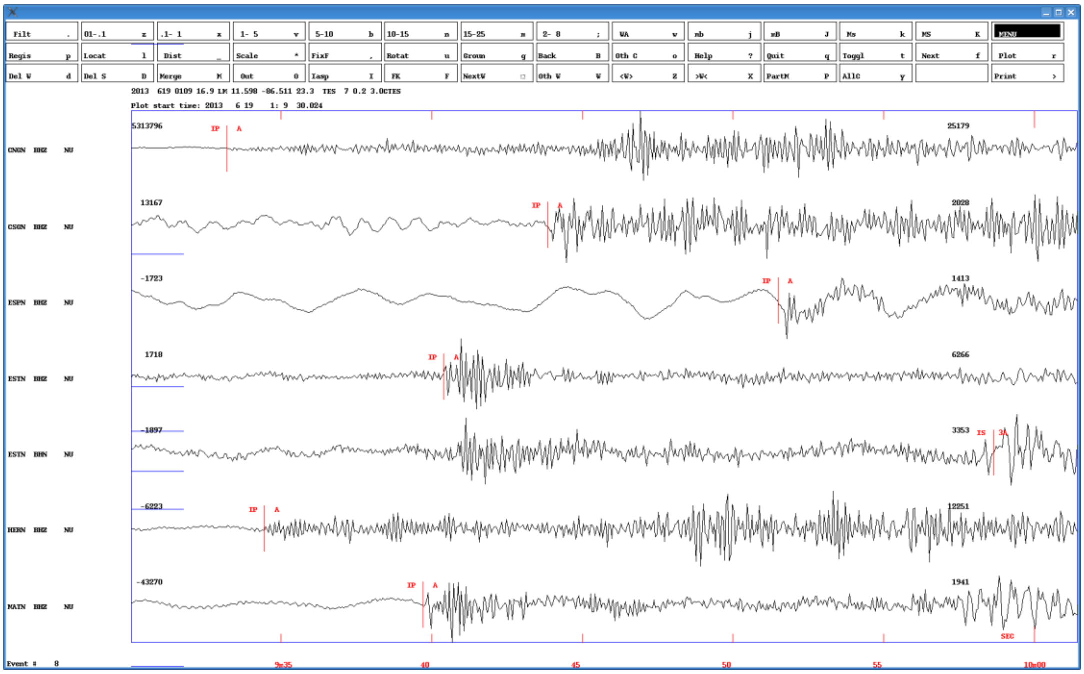


Figure 11 Automatic readings by RTPICK.

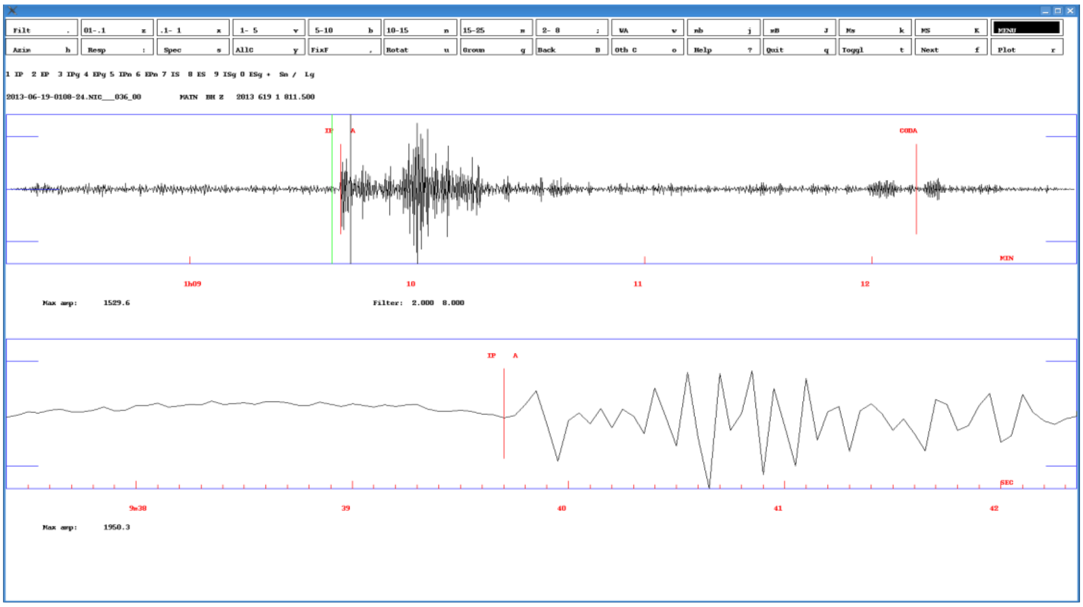


Figure 12 Automatic readings by RTPICK.

If RTPICK can produce a location for the event, two different html files are generated that will show the location on maps as shown in Figure 13 and Figure 14. A coda magnitude is computed based on the coda found by RTQUAKE. The coda is computed as the length of the event from the first pick until the signal produces a long-term-average below the de-trigger level. The coda length is thus often smaller than the coda length that would have been picked manually.

The map in Figure 13 can be shown as a normal web page by entering the following link in the browser: /home/seismo/mydir/map/LAST_TRIG.html, where /home/seismo/mydir is where RTQUAKE is installed. The page is refreshed every 5 minutes (can be changed) so new auto located events will appear when detected by RTQUAKE. A file called STATIONS must be located in the directory mydir/map. The STATIONS file must have the following format with " | " (space, vertical, space) between items:

stationname latitude longitude height areaname text1 text2

for example:

```
PB01 | -21.04 | -69.48 |  900 | Huatacondo  |  abc | def
PB02 | -21.31 | -69.89 | 1015 | SalarGrande |  aaa | bbb
```

Stations are marked on the map as triangles. Move the mouse over a triangle and press, and a plot of the signal with readings will appear in the window low-right if the signals are available.

In the window on top to the right of the map the S-file for the event is shown.

Be aware that the autolocation is far from perfect and may give completely wrong locations when the phases are picked wrong!!!! This will of course depend a lot of the quality of the data.

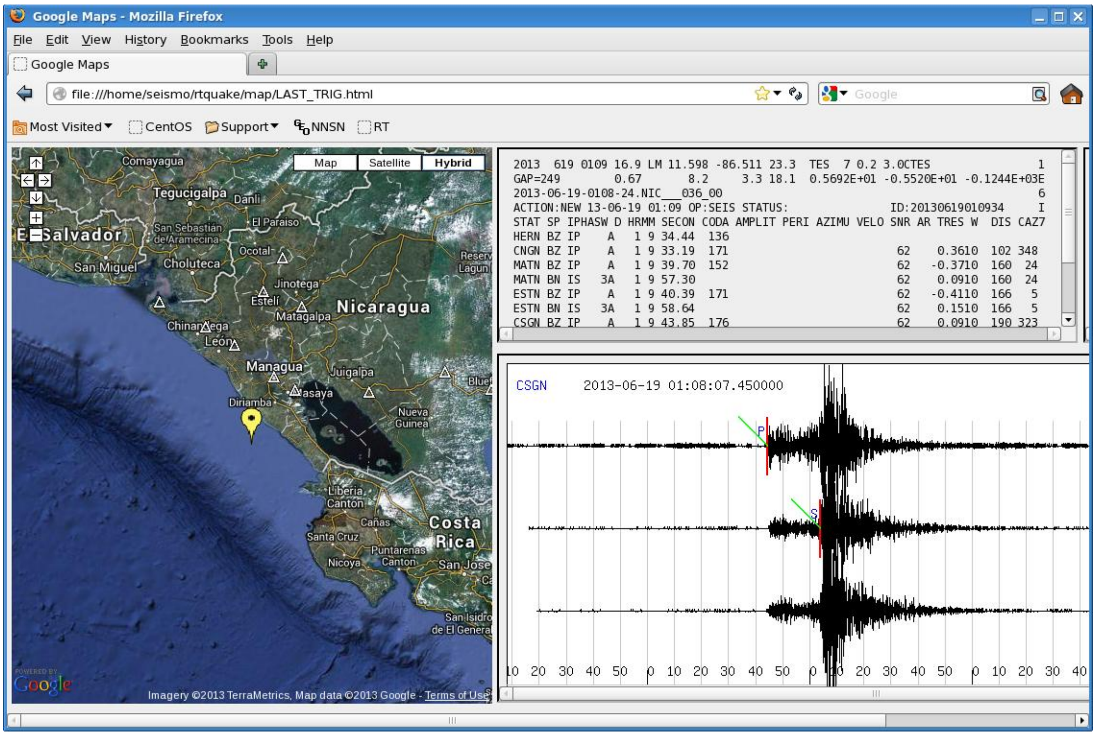


Figure 13 Web page generated by the RTPICK routine.

RTQUAKE will also generate the html code for generating a static map as in Figure 14. The html files for all automatically located events will be stored under the /home/seismo/mydir/loc catalog. The file names will have the format as for an S-file, plus the extension of html: 01-0854-34L.S201308.html

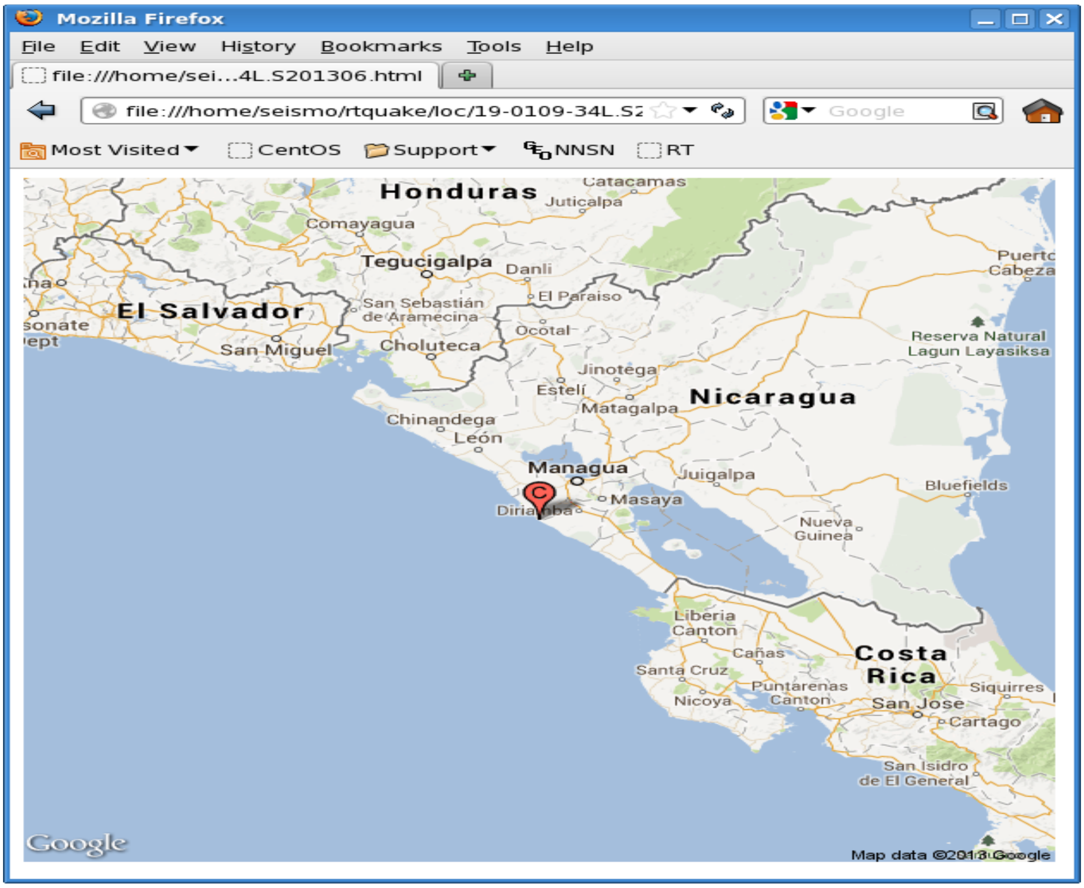


Figure 14 Web page generated by the RTPICK routine.

A location file called ALL_EPI0.txt is also generated in /home/seismo/mydir/map. New locations are added as new autolocations are computed. This can be monitored dynamically with Google Earth. See Figure 15 below.

Every time a new event is located, the yellow marker is moved to the new location and previous locations are still visible. The Google Earth map is programmed to move in what is called “fly mode”, which gives a smooth movement of the map to the new location.

How to set up:

Google Earth must be installed.

A program rtgeepi that is part of the RTQUAKE distribution must run in the background. This program continuously monitor the ALL_EPI0.txt file to check for the last location. A temporary file temp.epi.kml is generated and then copied to tu1_epi.kml. When Google Earth is started the file /home/seismo/mydir/map/rtge_refresh.kml should be opened from Google Earth. This file will read the tu1_epi.kml file every 10 seconds and update the map. When a new location is added, the centre of the map will move to this location. Old locations in ALL_EPI0.txt will also be plotted on the map. The rtgeepi must run at all the time to keep the last location up to date. The ALL_EPI0.txt will after some time contain a lot of triggers and the map may look a bit unclear. It is recommended to clean up the file by removing the oldest triggers.

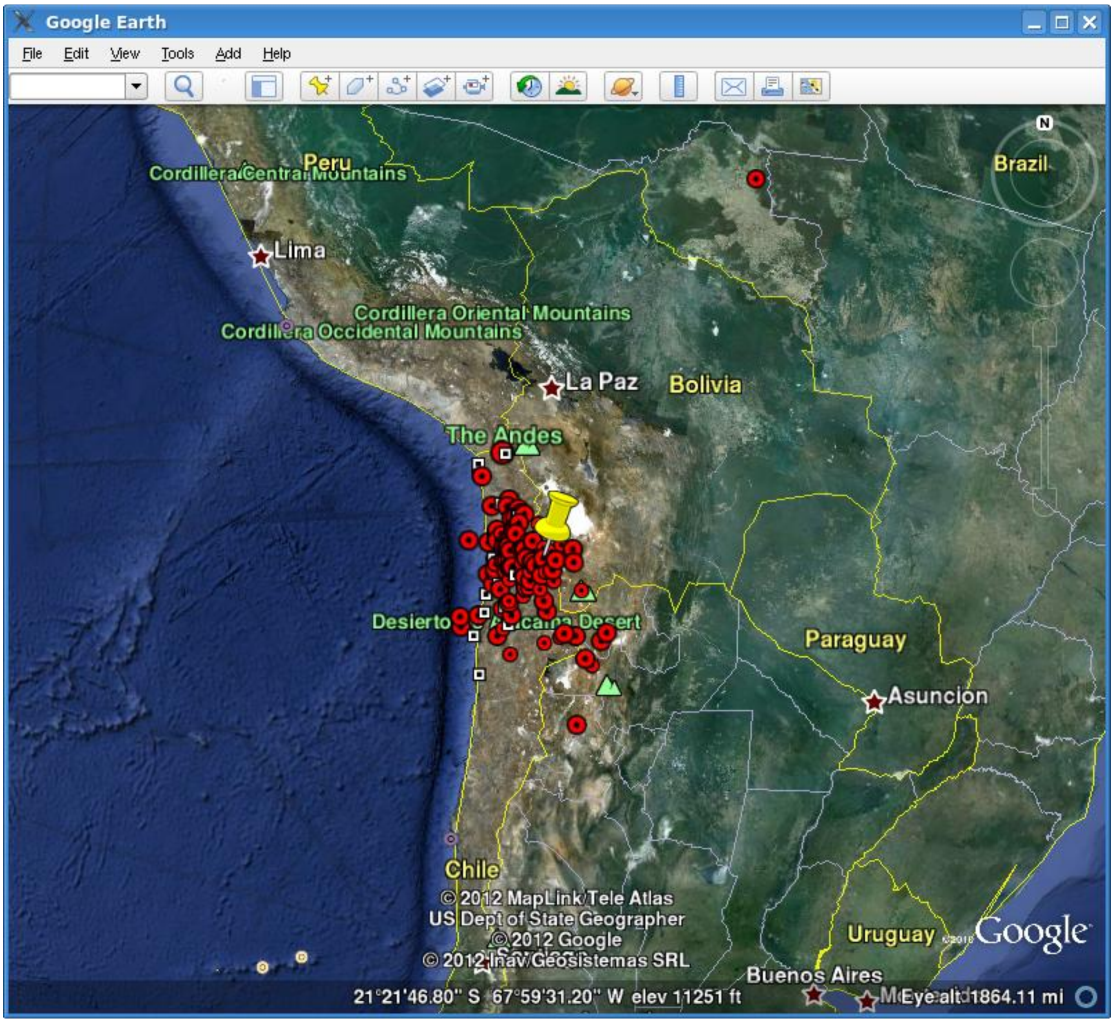


Figure 15 Auto location map using Google Earth, refresh every 10 seconds.

14 RTSNR

RTSNR monitors graphically how the current STA/LTA ratio is behaving for individual channels. Each instance of RTSNR can monitor one parameter set used by the RTDET module. Figure 16 below shows a typical output. The output shows how the ratio suddenly increase when an event occur. The different colours is just to distinguish between different channels. After a while, when the event finish, the ratio will drop back to the normal level as before the event. Before the event some sporadic noise that raises the ratio to above the trigger level, but as it occurs on one station only it is not considered a seismic event. The program can be useful to discover single channels or stations with sporadic or regular noise that causes unwanted triggers.

```
$ rtsnr -h

Command: rtsnr [options]

Options:
 -h               show this usage message

Options:
 -par     n       instance  parameter set. (default: 0)
 -xo      pixels  position of window x-direction (default: 0)
 -yo      pixels  position of window y-direction (default: 0)
```

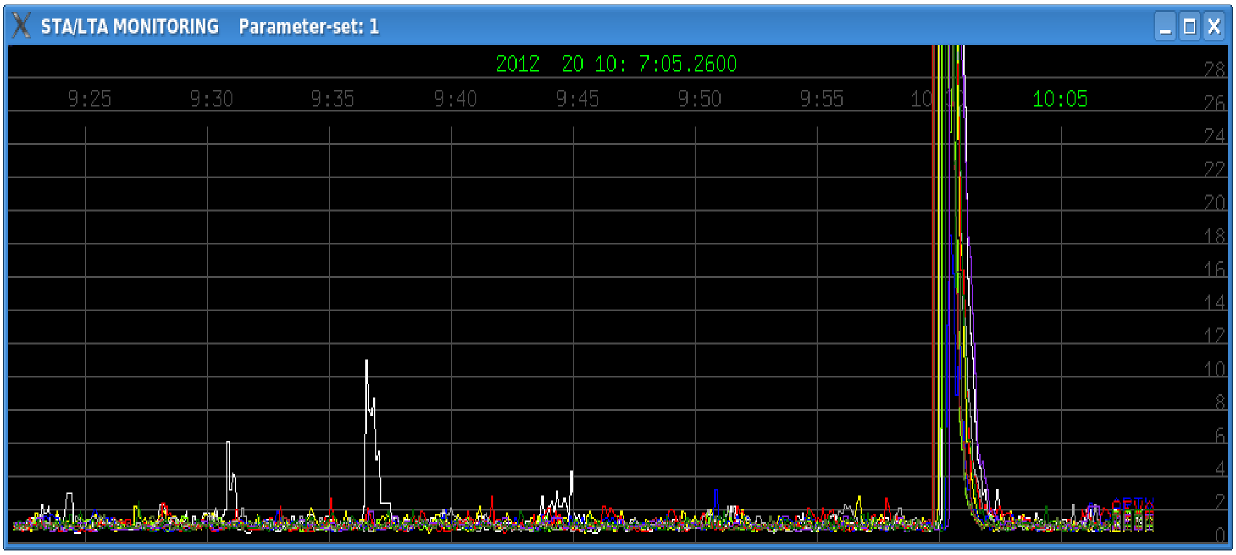


Figure 16 RTSNR

15 RTDLY

$ rtdly -h

Command: rtdly [options]

Options:

-h		show this usage message
-par	n	instance parameter set. (default: 0)
-xo	pixels	position of window x-direction (default: 0)
-yo	pixels	position of window y-direction (default: 0)

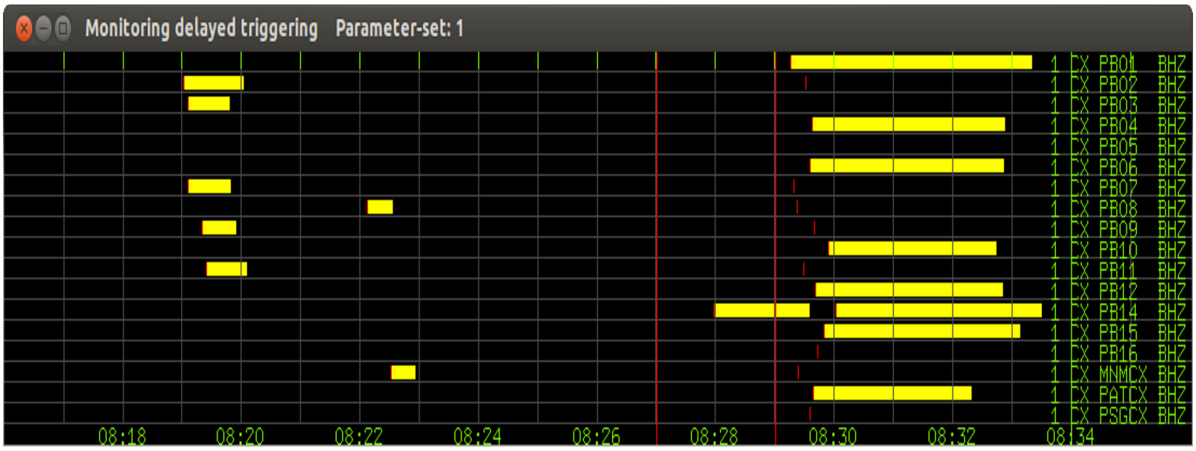


Figure 17 RTDLY shows the onset and duration of triggers (yellow lines) for individual channels and the duration of the trigger. The green vertical line to the right indicates the current time. The two vertical red lines indicate the array-propagation-window within which the detection of the event is performed. RTQUAKE can be set up to wait up to 30 minutes before checking for triggers in order to also include delayed channels. Delayed triggers will show up at correct time in the display, and the network trigger will take place within the array propagation window, in this case two minutes wide and seven minutes delayed (left red line). The red and the yellow markers seen closer to the current time-line are individual triggers for a new event. The display can be useful to optimize the delay and array propagation window parameters.

The graphics is dynamic in the sense that the user will see the onsets and duration of the triggers slowly moving to the left towards the array-propagation-window where network triggering takes place. The timelines for the APW and current time are positioned statically while the time scale at the bottom changes according to current UTC time.

Normally the trigger onsets are marked close to real time near the green line marking the current time. In cases where for example data transmission is slow, signals may be received with a significant latency. The triggers will however be marked on the plot at the correct time of occurrence when data is available. In Figure 17 we allowed for a latency of 7 minutes which is the total time from the current time to the end of the APW to the left. The APW has been set to 2 minutes. As the trigger onsets move towards and into the APW, the network trigger algorithm will decide if there are sufficient triggers to define a network trigger

This approach secures that trigger onsets arriving up to 7 minutes delayed still are contributing for the network trigger inside the APW. The allowed latency and APW are set by parameters. The display can be useful to optimize the delay and array propagation window parameters. Components that cause frequent false onsets can easily be observed on the display.

16 RTNET

The module plots selected components from seismic stations in near-real time. The module can read data from one SeedLink server only, but several instances of the module can run at the same time reading from different SeedLink servers.

```
$ rtnet -h

Command: rtnet [options] [host][:port]

 Options:
 -V              report program version
 -h              show this usage message
 -top     text   top directory (default: /home/seismo)
 -c              print stations.conf file
 -p              print details of data packets
 -d              print full station name on each seismogram
 -sc             auto-scaling each new data buffer


 ## Graphics options ##

 -x       pixels   width window in pixels (default: 1000)
 -y       pixels   height window in pixels (default: 600)
 -buf     n        0-double 1-single buffering (default: 0)
 -xo      pixels   position of window x-direction (default: 0)
 -yo      pixels   position of window y-direction (default: 0)
 -m       minutes minutes over screen: 1,2,3,4,5,10,15,30,60 (default: 15)

 ## Filter options ##

 -fl      low      lowpass frequency
 -fh      high     highpass frequency

 ## Data stream  selection and station file##

 -l       str      The routine will look for the filename you specify under the
                   /home/seismo/mydir/par/user_created_subdirectory
                   catalog. To specify a file stored in the user subdirectory, you specify the subdirectory
                   name and the filename.

                   Example:
                   A stream file stored in /home/seismo/mydir/par/DEMO1  -l DEMO1/streams_plot
-f        stat     The routine will look for the filename you specify under the
                   /home/seismo/mydir/par/user_created subdirectory
                   catalog. To specify a file stored in the user subdirectory you specify the subdirectory
                   name and the filename.

                   Example:
                   A station file stored in /home/seismo/mydir/par/DEMO1: -f DEMO1/stations_plot

 -n       no       number of stations to plot
 -a       no       which station to plot (0,1,2,3.......)

 [host][:port]   Address of the SeedLink server in host:port format
              f.ex.: 129.177.xx.yy:18000
              f.ex.: localhost:18000
```

The different seismograms are scrolled to the left on the screen when plotting reaches the right end of the defined window for the plot. Each seismogram is plotted individually in its own window. This means that each seismogram have its individual timing.
RTNET needs 2 parameter files, one for defining the input streams of data and another to define the actual components to plot. The names for these files are streams_plot and stations_plot respectively and are stored in /home/seismo/mydir/DEMO1. The two files are initially set equal to the plot files for the test run and the example plots below can be run from the command line after the installation.
The module has several input parameters. Some can also be modified interactively during execution of the program.
An option to filter the incoming data can be activated while running. This option will also start a simple detection algorithm and mark probable events on the plot.
The program will mark stations that have not received data for the last 60 seconds. Another indicator can be a red square in the upper right corner of each seismogram window that indicates that GPS timing is out of synchronization.
The different options can be controlled partly during the start command and partly while running via keyboard or menu.
All options will be described in more detail below. Some examples on how to use it will be given at the end of this documentation.
Two parameter files define the data that are available for the RTNET client:
streams_plot and stations_plot.
streams_plot contains information of which data the SeedLink server should send to the RTNET client. The format follows the standard in SeedLink for defining data streams:

```
CX PB01 BHZ    network: CX    station: PB01  component: BHZ
CX PB02 BHZ
CX PB03 BHZ
CX PB04 BHZ
CX PB05 BHZ
CX PB06 BHZ
CX PB07 BHZ
CX PB08 BHZ
CX PB09 BHZ
CX PB10 BHZ
CX PB11 BHZ
CX PB12 BHZ
```

stations_plot contains information of which data the RTNET client may use. For example a setup to only plot vertical components. The location parameter must be included. The full name of the station is used when the –d (as in the test configuration) is selected. This option is included to make the text more informative for the public.

```
PB01 BHZ PB01 Huatacondo station: PB01 component: BHZ title: PB01 Huatacondo
PB02 BHZ PB02 Salar Grande
PB03 BHZ PB03 El Tigre
PB04 BHZ PB04 Mantos de la Luna
PB05 BHZ PB05 Michilla
PB06 BHZ PB06 Pedro de Valdivia
PB07 BHZ PB07 Cerro Tatas
PB08 BHZ PB08 Macaya
PB09 BHZ PB09 Quillagua
PB10 BHZ PB10 Juan Lopez
PB11 BHZ PB11 Quebrada Aricilda
PB12 BHZ PB12 Cerro Caramaca
```

When you have generated the two parameter files streams_plot and stations_plot you can start RTNET first time like below to get all the options available:

rtnet –h

```
Usage: rtnet [options] [host][:port]

## General program options ##
-V             report program version
-h             show this usage message
-c             print stations.conf file
-p             print details of data packets
-d             print full station name on each seismogram
-sc            auto-scaling each new data buffer

## Graphics options ##
-x pixels      width window in pixels (default: 1000)
-y pixels      height window in pixels (default: 600)
-m minutes     minutes over screen:1,2,3,4,5,10,15,30,60 (default: 15)

## Filter options ##
-fl lowpass    lowpass frequency
-fh highpass   highpass frequency

## Data stream selection and station file##
-l listfile    read a stream list from this file
-f stationfile read a station list from this file
-n no_to_plot  number of stations to plot
-a station no  which station to plot (0,1,2,3.......)

[host][:port]  Address of the SeedLink server in host:port format
```

The different options are self explanatory, but a few may need some more explanation.

General program options:

-d This option can be used when the monitor is installed to give a clearer view of the station names, for example: Kongsberg instead of KONO 10BHZ, and the timing is shown as complete dates rather than day of year. Both text strings are also in bigger fonts.
The text, full station name can be added in the stations.conf file after the standard name as for example: KONO 10BHZ Kongsberg

The graphics options:

The upper left corner of the active drawing window is always placed in the upper left corner of the screen.

-x pixels This parameter sets the width of the active drawing window and must never be bigger than the total width of the screen. Default is 1000 pixels.

-y pixels This parameter sets the height of the active drawing window and must never be bigger than the total height of the screen. Default is 600 pixels.

-m minutes This parameter sets the total number of minutes across the active drawing area selected above. Options are: 1,2,3,4,5,10,15,30 and 60 minutes. Default is 15 minutes.

Filter options:
When starting the program without –fl and –fh set to any values, the data plotted on the screen are unfiltered. However, via the keyboard or the menu, a pre-set filter (2.0-8.0 Hz) can be activated. This filter can be turned on and off while running. Turning on the filter will also activate a simple detection algorithm that will mark probable events in the seismograms.

The options –fl and –fh are to be used from the command line when starting the program and can be set to the values you decide.

Data stream selection and station file options:

-l listfile	read stream list from this file
-f stationfile	read station list from this file
-n no_to_plot	number of components to plot. This parameter can be any number up to the number of components specified in the stations_plot file.
-a station	which station to plot. (0,1,2…..number of stations in the station.conf file). This option can be used to check one particular component. The default time window is 2 minutes, so that more details are visible in the seismogram.

The sequence of lines (stations and components) in both files are free. However, the sequence of the lines in the stations.conf file will decide the sequence of stations plotted. This can give a more logical sequence of stations for example from north-south, east-west etc.
The SeedLink server host:port should always be included on the command line when starting the program.

Options that can be used interactively while program is running.
They can be activated from keyboard or from a menu (right-click on mouse). To see the different options, press 'h' on the keyboard or right-click on the mouse. The different options are:

```
List of key-press functions:
----------------------------
h  : list this on screen
esc: exit
u  : increase amplitude on all channels
d  : decrease amplitude on all channels
S  : freeze graphics
s  : resume graphics
n  : next channel
+  : increase amplitude on current channel
-  : decrease amplitude on current channel
F  : turn on pre-set filter
f  : turn off filtering
1  : Set color scheme to default
2  : Color scheme 2
3  : Color scheme 3
4  : Color scheme 4
```

Note the 'S' and 's' options: The plotting can be halted with the 'S' when something interesting happens. You may take a screenshot and then resume plotting with 's'.
When the program is running, a plot of a selected channel can be plotted in a separate window to see more details in the signal. Point on the channel with the mouse (around zero-level) and left-click on the mouse.

Examples of running RTNET.

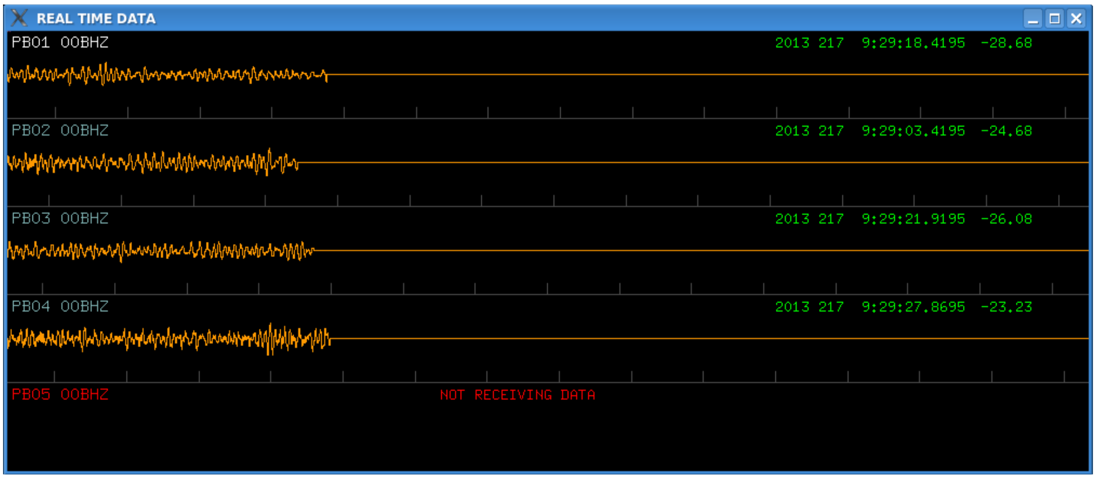


Figure 18 From command line: rtnet –n 5 –l DEMO1/streams_plot –f DEMO1/stations_plot 139.17.3.177:18000
Plot the 5 first channels in the stations_plot file. For station PB05 we see the message 'NOT RECEIVING DATA'. This means that the RTNET program has not received data from this station for at least the last 60 seconds. It may be a reason for further checking of this station.

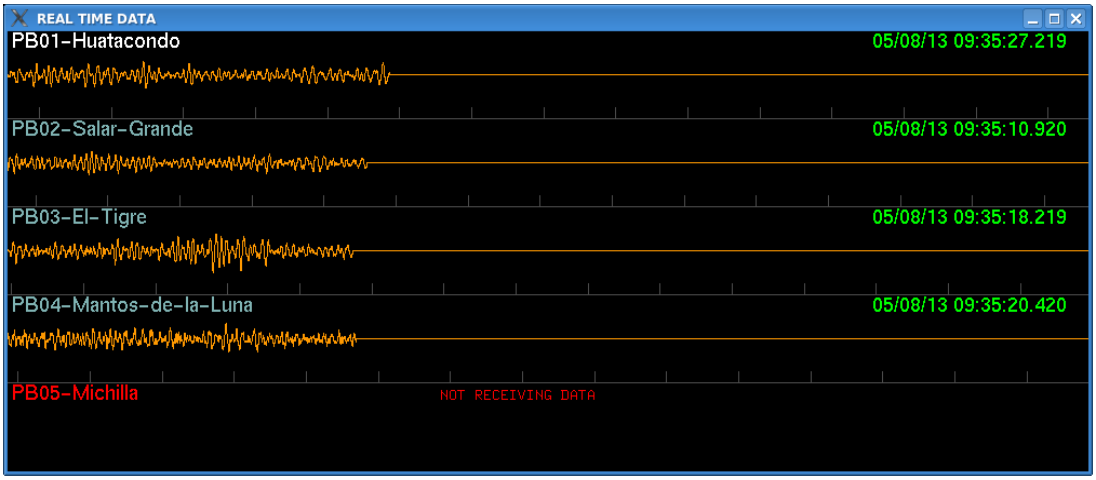


Figure 19 From command line: rtnet –n 5 –d –l DEMO1/streams_plot –f DEMO1/stations_plot 139.17.3.177:18000
Plot the 5 first channels in stations_plot file with the option –d. This will plot the text field in the stations_plot file, normally a geographical name.

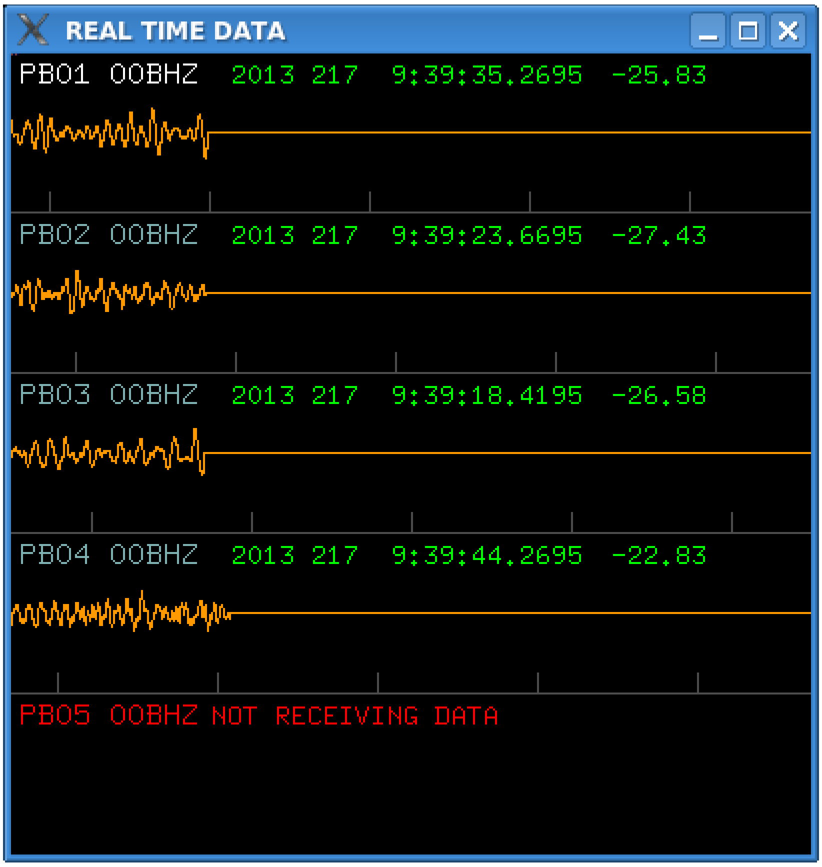


Figure 20 From command line: rtnet –n 5 –d –m 5 –x 400 –l DEMO1/streams_plot –f DEMO1/stations_plot 139.17.3.177:18000
Plot the first 5 channels in stations_plot file, textfield, 5 minutes x-axis and x-axis 400 pixels long.

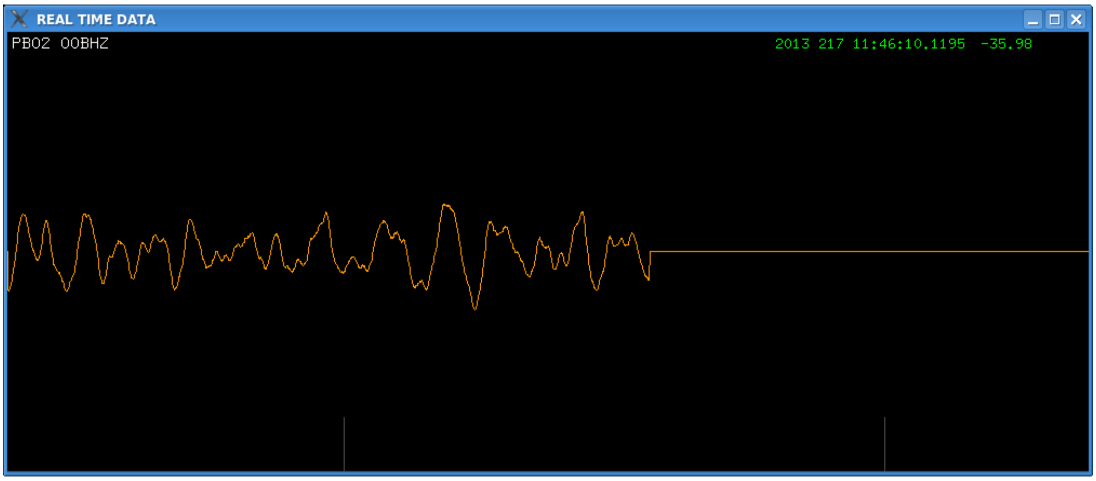


Figure 21 From command line: rtnet –a 1 –l DEMO1/streams_plot –f DEMO1/stations_plot 139.17.3.177:18000
Plot the second (1) channel in stations_plot file with default values.

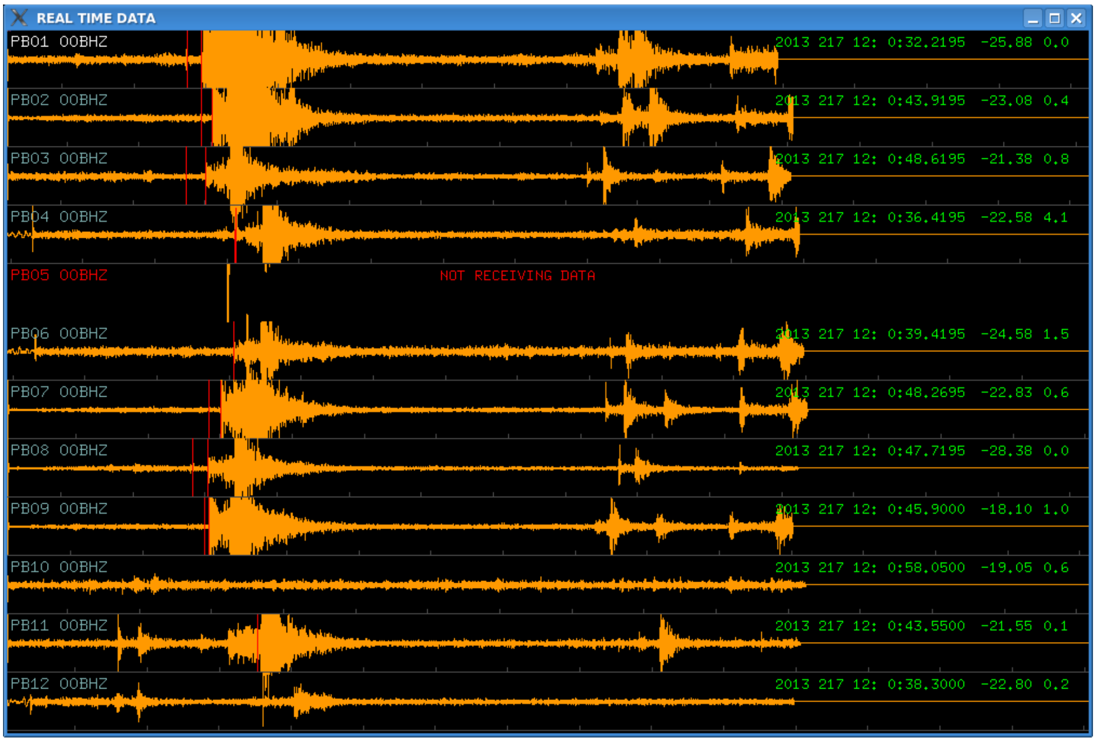


Figure 22 From command line: rtnet –n 12 –y 650 –l DEMO1/streams_plot –f DEMO1/stations_plot 139.17.3.177:18000
Plot the first 12 channels in the stations_plot file, make the drawing window 650 pixels high. Filter was turned on from keyboard 'F', and there is a detection indicated on several channels.

Several instances of RTNET can be executed at the same time reading data from the same or different SeedLink servers. Execution can also be started from script-files. This way RTNET can be started automatically at reboot of the PC.

The user can make different script-files that start RTNET with different options. In this way it is possible to for example monitor both unfiltered and filtered signals in two different windows, see Figure 23a and Figure 23b.

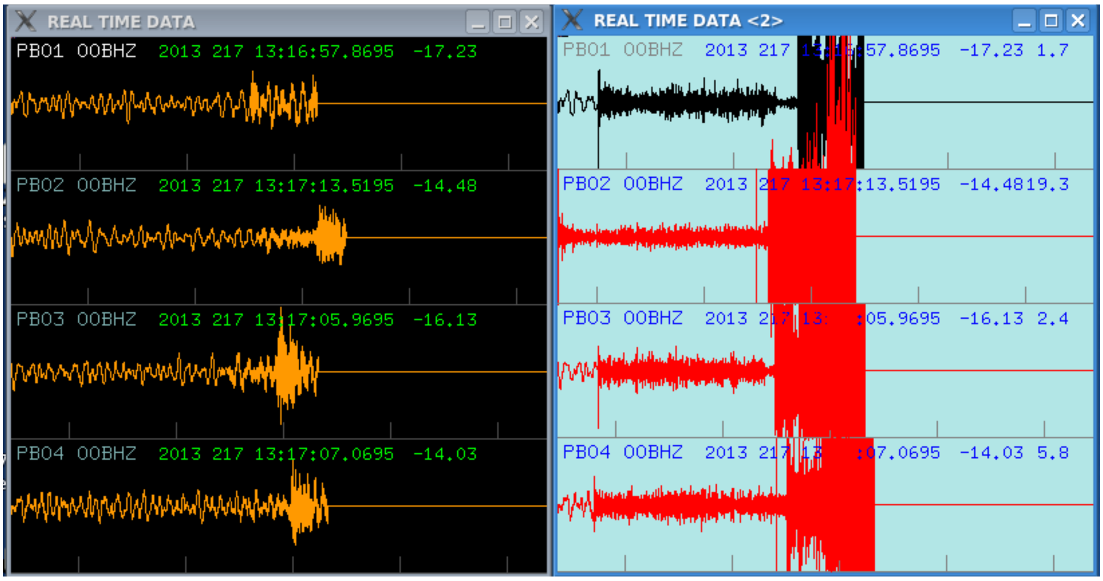


Figure 23a From command line: rtnet –n 4 –m 5 –x 400 –l DEMO1/streams_plot –f DEMO1/stations_plot 139.17.3.177:18000

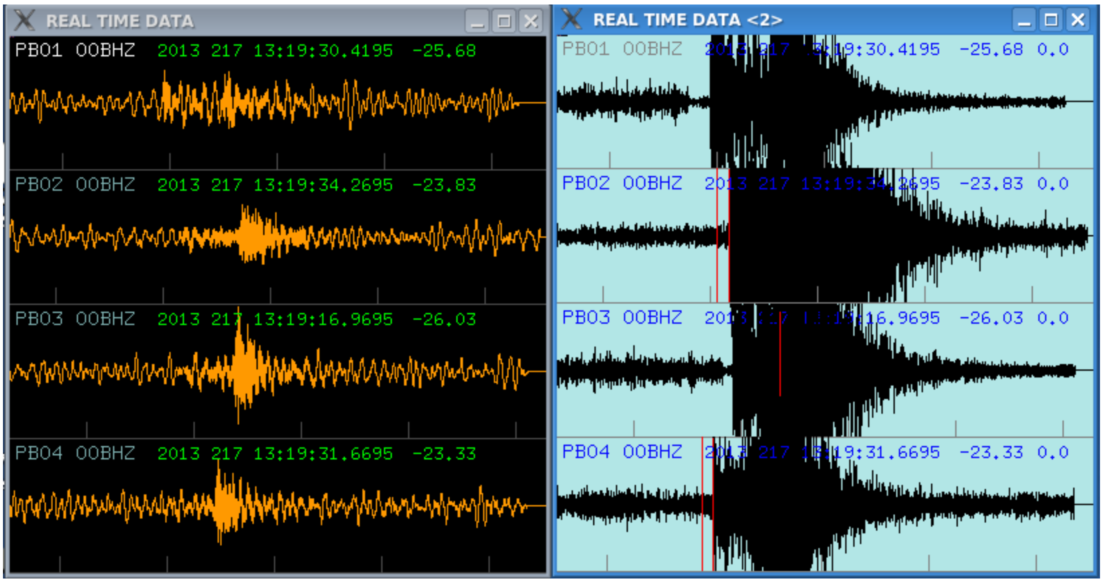


Figure 23b From command line: rtnet –n 4 –m 5 –x 400 –l DEMO1/streams_plot –f DEMO1/stations_plot **139.17.3.177:18000**

In the example above, two instances of RTNET is running side by side with the same initial command line. The colour scheme on the window on the right has been changed interactively via the right-click menu. The window on the right show filtered data (2-8 Hz). The filter was activated with the right-click menu. We also see that the detection algorithm has detected and marked a probable event. The seismogram is plotted in red as long as the detection algorithm is in trigger-mode.

17 RTTIME

This module can be used to monitor the status of the stations configured in a SeedLink server, see Figure 24a. It can be used for monitoring a number of stations that can actually fit your display. By pressing the mouse over a station that is marked green, the RTNET program will be started to give a more detailed plot of the signal from that particular station. See Figure 24b.
The module can read data from one SeedLink server only, but several instances of the module can run at the same time reading from different SeedLink servers.

$ rttime -h

Command: rtgraph [options] [host][:port]

Options:

-h	show this usage message
-top	top directory (default: /home/seismo)
-siz pixels	height of window in pixels (default: 400)
-buf n	0-double 1-single buffering (default: 0)
-xo pixels	position of window x-direction (default: 0)
-yo pixels	position of window y-direction (default: 0)
[host][:port]	address of the SeedLink server in host:port format f.ex.: 129.177.xx.xx:18000 f.ex.: localhost:18000

Figure 24a RTTIME window.

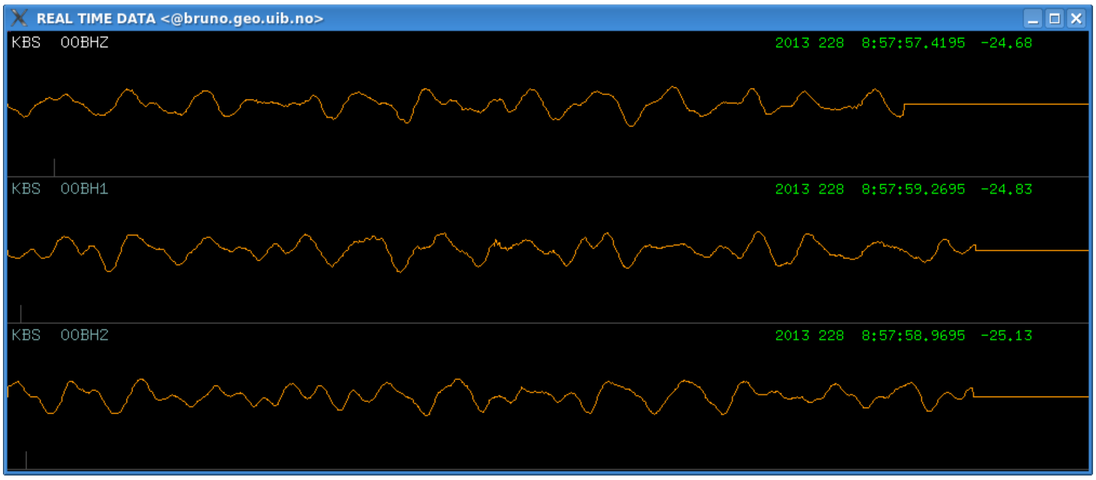


Figure 24b RTNET started from the RTTIME window.

18 RT24 + RTDR24 + RTDRUM + RTHPLT

These three modules are used to create helicorder plots of unfiltered and filtered data from streams from a SeedLink server. Two parameter files are used as input, one to define the different streams to read and another to select the actual components to plot.
RT24 generates temporary files for each component specified. Each file will contain 24 hours of data. If a filter is specified, filtered data files will be generated in addition.
The SeedLink server to use must be specified in the format ipnumber:port.
RT24 calls RTDR24 at regular intervals to make the helicorder plots based on the continuous files generated by RT24.
RTHPLT should be run to generate an index.html file in the mydir/rt/png and mydir/rt/png_filt catalogues. Loading the index.html files from a browser will give you the menu shown in Figure 25 below.

As part of the installation a directory structure is created under the RTQUAKE top directory:

/home/seismo/mydir/rt/tmp	unfiltered data
/home/seismo/mydir/rt/tmp_filt	filtered data
/home/seismo/mydir/rt/png	plot unfiltered data
/home/seismo/mydir/rt/png_filt	plot filtered data

RT24 writes 24-hour files into /home/seismo/mydir/rt/tmp and /home/seismo/mydir/rt/tmp_filt. These files are input for module RTDR24 that creates plots of the data that are in the files at the current time.
The parameter files for streams and stations must be stored under /home/seismo/mydir/rtquake/par/user_created_subdirectory.

streams_heli
The streams_heli.TST file is standard SeedLink input format where CX is the network name, PB0n is the station name and BHZ the component.

```
CX PB01 BHZ
CX PB02 BHZ
CX PB03 BHZ
CX PB04 BHZ
CX PB05 BHZ
CX PB06 BHZ
CX PB07 BHZ
CX PB08 BHZ
CX PB09 BHZ
CX PB10 BHZ
CX PB11 BHZ
CX PB12 BHZ
```

stations_heli

The format of this file:

PB01	station name
00	location
0.0100	gain for the unfiltered data, can be modified dynamically

0.0300	gain for the filtered data, can be modified dynamically
2.0	low-pass frequency
8.0	high-pass frequency
Huatacondo	name of station, geographical name

```
PB01_00BHZ 0.0100 0.0300 2.0 8.0 Huatacondo
PB02_00BHZ 0.0100 0.0300 2.0 8.0 Salar Grande
PB03_00BHZ 0.0100 0.0300 2.0 8.0 El Tigre
PB04_00BHZ 0.0100 0.0300 2.0 8.0 Mantos de la Luna
PB05_00BHZ 0.0100 0.0300 2.0 8.0 Michilla
PB06_00BHZ 0.0100 0.0300 2.0 8.0 Pedro de Valdivia
PB07_00BHZ 0.0100 0.0300 2.0 8.0 Cerro Tatas
PB08_00BHZ 0.0100 0.0300 2.0 8.0 Macaya
PB09_00BHZ 0.0100 0.0300 2.0 8.0 Quillagua
PB10_00BHZ 0.0100 0.0300 2.0 8.0 Juan Lopez
PB11_00BHZ 0.0100 0.0300 2.0 8.0 Quebrada Aricilda
PB12_00BHZ 0.0100 0.0300 2.0 8.0 Cerro Caramaca
```

RTDR24 reads the component files specified by -comp every -upd second and generates a helicorder plot in directory specified by -www and also in directory -www_filt if the -flt option is specified.

A script in /home/seismo/mydir/com, rtquake_heli will start an example run of the program.

rtquake_heli

```
rt24 -to_wi 1200 -fr_hg 600 -mt 15 -l streams_heli -f stations_heli
139.17.3.177
```

$ rt24 -h

Usage: rt24 [options] [host][:port]

General program options

-V		report program version
-h		show this usage message
-top	text	top directory (default: /home/seismo)
-www	text	directory helicorder plots, unfiltered (/home/seismo/mydir/rt/png)
-www_filt	text	directory helicorder plots, filtered (/home/seismo/mydir/png_filt)
-to_wi	pixels	total width in pixels
-fr_hg	pixels	total frame height in pixels
-mt	min	minutes across frame
-col	n	color scheme

Data stream selection and station file##

-l listfile read a stream list from this file (streams_heli)
-f stationfile read a station list from this file (stations_heli)

[host][:port] Address of the SeedLink server in host:port format
f.ex.: 129.177.xx.yy:18000
f.ex.: localhost:18000

```
$ rtdr24 -h

Usage: rtdrum [options]

General program options:
 -h              show this usage message
 -top            top directory (default: /home/seismo)
 -www            directory helicorder plots (/home/seismo/mydir/rt/png)
 -www_filt       directory helicorder plots, filtered (/home/seismo/mydir/rt/png_filt)
 -logo1  txt     name of left side logo (gif file)
 -logo2  txt     name of right-side logo (jpg file)
 -gain           gain factor signals. For example: 0.003
 -flt            will generate filtered helicorder plots
 -comp           filename component to plot, 10 char.(ex: ASK__00EHZ)
 -upd  n         update plot every n seconds (default: 120)
```

rtheli1

Helicorder plot. This command will generate one-per-day helicorder plots for as many days you decide using the configuration files found in /home/seismo/mydir/par/DEMO1 and /home/seismo/mydir/com/rtquake.par. The user can use these files as recipes for different configurations.
Helicorder plots are generated in /home/seismo/mydir/rt/png and /home/seismo/mydir/rt/png_filt for unfiltered and filtered data respectively. The helicorder plots are updated with some minutes delay.
Plots can be shown with a standard browser. To generate a menu of all plots the command rthplt is run automatically to create index.html files.

rthplt

Creates the html files the user can use to look at the helicorder plots from the different stations. Enter the address /home/seismo/mydir/rt/png/index.html or /home/seismo/mydir/rt/png_filt/index_filt.html and click on the station you want to check.
Another function of this routine is that files older than n days (specified in /home/seismo/mydir/com/rtquake.par) are removed.

rtheli2

Helicorder plot. This command will generate one helicorder plot per station, but only one per station for the last 24 hours. The plots can be shown as a “slide-show”, switching station automatically every 10-15 seconds. Enter the following address in your browser: /mydir/heli/slide_tst.html

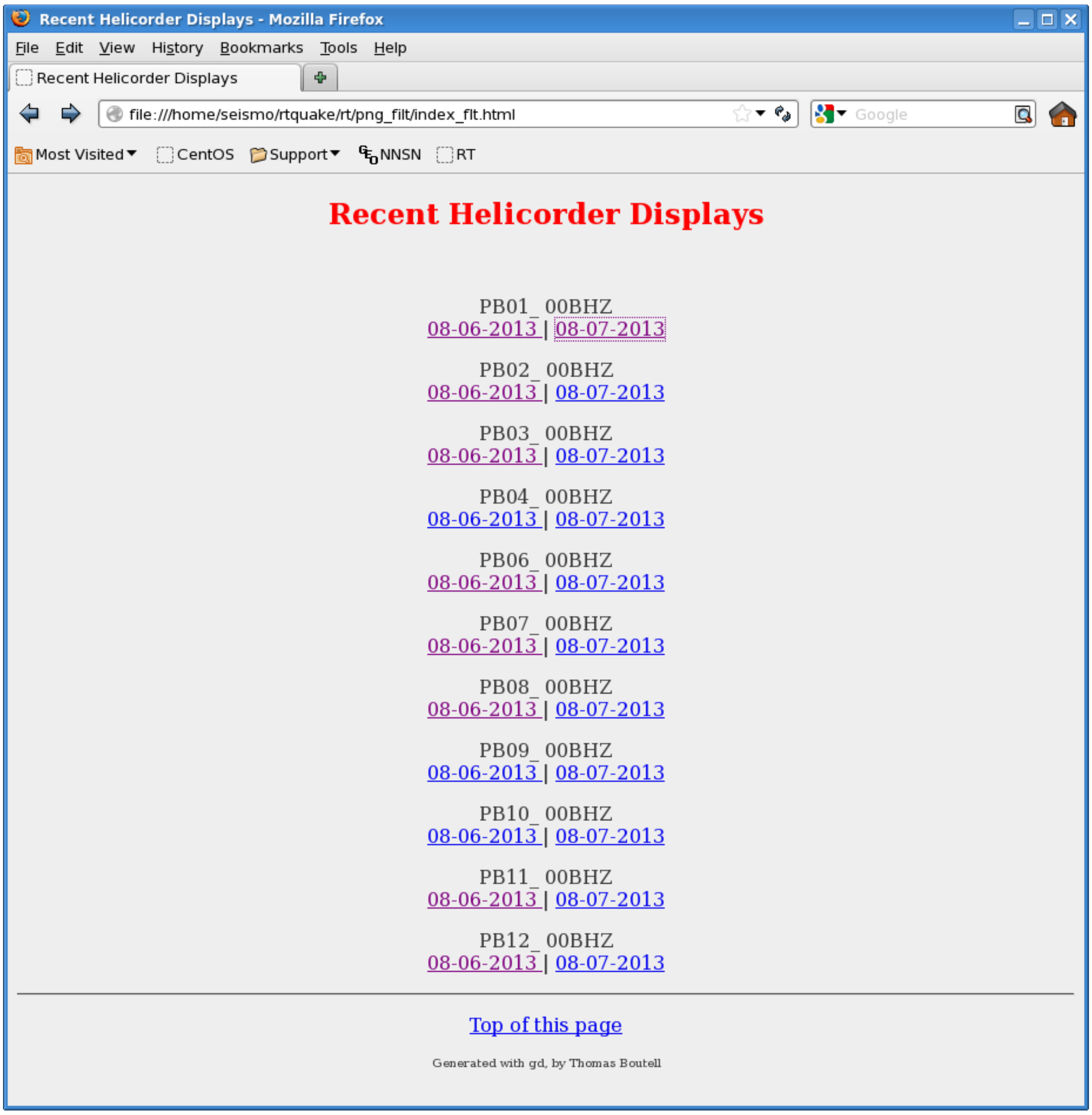


Figure 25 Menu helicorder plots

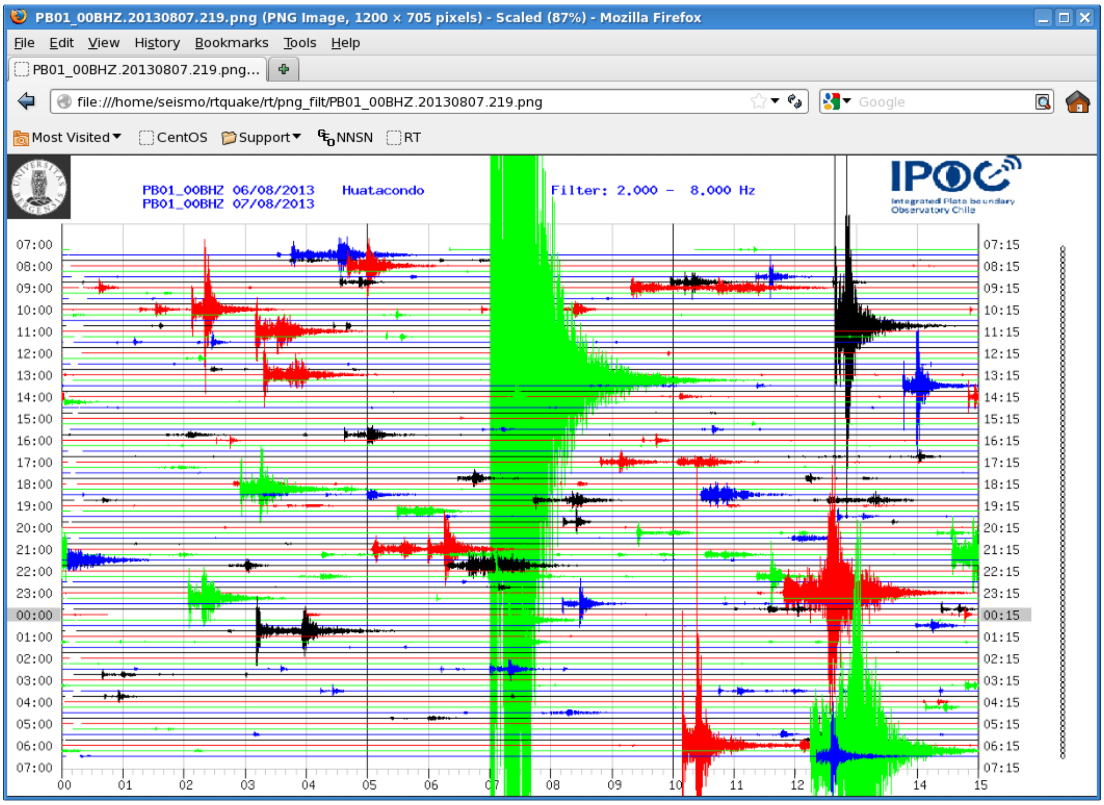


Figure 26 Helicorder plot

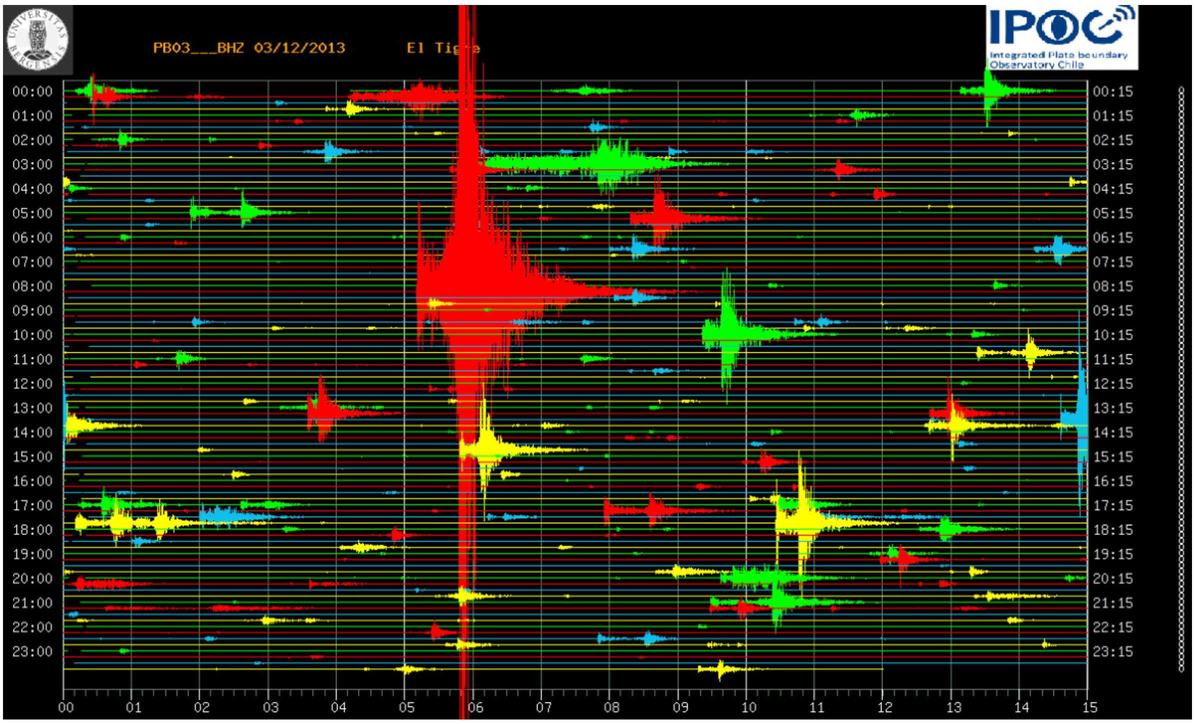


Figure 27 Helicorder plot

19 DIRECTORY OVERVIEW AFTER INSTALLATION

mydir
    Makefile
mydir/com
    rtquake.par
    rt_IPCH
    rt_STOP
    setup_rt.bash
    setup_rt.csh
    STATION0.HYP
mydir/doc
mydir/inc
    libslink.h
    sh_mem_rt.h
    slplatform.p
mydir/libslink
    the SeedLink client library, Chad Trabant
mydir/loc
    empty
mydir/map
    ALL_EPI0.txt
    emns_epi.kml
    emns_refresh.kml
    glasses.wav
    icon49.png
    icon56.png
    LAST_LOC.txt
    LAST_TRIG.html
    rtge_refresh.kml
    STATIONS
    triangle.png
    tu1_refresh.kml
    yellow-dot.png
mydir/par
    brygge2.jpg
    brygge2.jpg.ok
    brygge2.white.jpg
    detect.TST
    IPOC.TST
    Logo70X70.gif
    record.TST
    stations.conf
    stations_heli.TST
    stations_plot.TST
    streams.conf
    streams_heli.TST
    streams_plot.TST
    streams.TST

mydir/picker
- miniseed library, Chad Trabant
- modified FilterPicker, A.Lomax

mydir/req
- empty

mydir/rt
- empty catalog structure for helicorder plots

mydir/rtdet
- getwindow
- Makefile
- msi
- rtdet.c
- slinktool

mydir/seedlink
- seedlink-2.5 distro

mydir/tmp
- empty

mydir/utils1
- Makefile
- rtdly.c
- rtmon.c
- rtnet.c
- rtsnr.c

mydir/utils2
- Makefile
- rt24.c
- rtdr24.c
- rthplt.c
- rttime.c

mydir/wrk
- rt_lPCH
- rtquake_heli
- rtquake_start
- rtquake_stop
- rt_STOP
- STATION0.HYP

20 REFERENCES

Havskov, J and L. Ottemøller (1999). SEISAN earthquake analysis software. Seismological Research letters, 70, 532-534

Lomax, A:, C. Satriano and M. Vassallo (2012). Automatic Picker Developments and Otimization: FilterPicker – a Robust,Broadband Picker for Real-Time Seismic Monitoring and Earthquake Early Warning. Seismological Research Letters, 83.

Utheim, T. Havskov, J. Ozyazicioglu, M. Rodriguez, J. Talavera, E. (2014).
RTQUAKE, A Real-Time Earthquake Detection System Integrated with SEISAN. Seismological Research Letters 85, 735-742