Manual for rt2detection package.

Calum Chamberlain, SGEES, Victoria University of Wellington, PO Box 600,
Wellington 6140
calum.chamberlain@vuw.ac.nz

1 Introduction

This manual is intended as a user guide for the rt2detection package, written in python, for the routine processing of seismic data collected on reftek RT130 dataloggers. This package is fairly minimal and rather than re-doing a lot of processing steps calls upon external routines that need to be installed seperately (this is covered in the installation section).

The heart of this processing flow is that we want to simply take raw data through to triggered, multiplexed miniseed files which can be read in by seisan. This processing flow also includes automatic picking capability, although this is by no means optimized and the parameters for this picker should be optimized for the network you are working with. The same can be said of the parameters used in the STA-LTA detection routine.

2 Installation

As the main linking code is written in python, there is little compliation to be done, however this package does require the following to be installed on your machine (according to each programs install instructions):

- Pascal tools (http://www.passcal.nmt.edu/content/software-resources)
- Obspy (https://github.com/obspy/obspy/wiki)

Further to this, to use the automated picker you will need to compile the C code found in the *picker* directory if this distribution, for this you will need a C compiler (gcc has been tested and works well, nothing else has been tested). Once you have installed a C compiler you should change the variables in both picker/src/Makefile and picker/libmseed/Makefile to the path of your C compiler (or, if you have installed it gcc correctly, just to gcc). You can then compile the picker by typing, when in the picker directory:

```
>make clean
>make rtquake
>cp rdtrigL ../.
```

You should also run the seisan commands:

```
>remodl
>setbrn
```

to set the local travel time tables required for the hypo71 location algorithm.

If you have not generated the seisan databases that you want the data to be stored in (in the REA and WAV directories of seisan) you should run:
>makerea
to generate them before running any of the codes.

Now you should be ready to roll.

3 Structure

This distribution contains five directories:

- doc
- picker
- RT_data
- MS_data
- Merged_date

The doc directory contains this document about the package. The picker directory contains the source code and libraries for the picking routine. The RT_data directory is where you should upload your raw reftek data to once you have run it through the pascal tools, neo software. *This must be uploaded in station directories, e.g. RT_data/WTSZ/misc.zip.* The MS_data directory will be the location of the converted, but un-merged miniseed data. Merged_data will contain all of the multiplexed miniseed files, these will all later be cleaned when running the programs.

4 Python routines

There are three python routines included in this package:

- makeSfile.py
- STALTA.py
- rt2detection.py

rt2detection.py is the overarching routine which calls all the others. This will take your data in steps from raw to triggered, picked data in the seisan database. The top of this file should be editted to include your parameters and network information. All parameters are explained in the file. Around line 71 is the network information which needs to be adjusted for your network.

rt2detection calls upon STALTA.py which currently runs the convenience STALTA methods within obspy - the parameters for this routine can be adjusted within this code, and again are commented as to what they do.

The final python routine, makeSfile.py simply builds empty nordic files which are then read in by the picking routine.

Finally all files are moved to the seisan database prescribed in the rt2detection parameters.

Parameters for the picker are set in the rtquake.par file.