rt2detection Documentation

Release 0.0.1

Calum John Chamberlain

April 22, 2015

CONTENTS

Contents:

CHAPTER

ONE

MANUAL FOR RT2DETECTION PACKAGE.

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

1.1 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.

1.2 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 emph{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.

1.3 Structure

This distribution contains five directories: doc picker RT_data MS_data Merged_data

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. emph{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.

1.4 Python routines

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.

CHAPTER

TWO

MAIN

2.1 rt2detection

Code designed to take data downloded using defaults.neo and located in the RT_data folder of this ditribution and convert this to miniseed, multiplex it and run seisan's continuous detection, sta/lta routine over this data.

ver 0.1 - Converts and defaults.merges data, but only for one day

ver 0.2 - Converts and defaults.merges data for multiple days

ver 1.1 - Converts and defaults.merges data for multiple days and can run a python detection routine (STA/LTA) over the data

ver 2.1 - Uses the filterdefaults.picker routine to automatically pick data, parameters for this are not yet optimized

07/11/14 - Calum Chamberlain - Victoria University of Wellington, NZ

CHAPTER

THREE

PRO

Codes to run main functions.

3.1 Sfile_util

Part of the rt2detection module to read nordic format s-files and write them EQcorrscan is a python module designed to run match filter routines for seismology, within it are routines for integration to seisan and obspy. With obspy integration (which is necessary) all main waveform formats can be read in and output.

Code generated by Calum John Chamberlain of Victoria University of Wellington, 2015.

class Sfile_util.**PICK**(*station*, *channel*, *impulsivity*, *phase*, *weight*, *polarity*, *time*, *coda*, *amplitude*, *peri*, *azimuth*, *velocity*, *AIN*, *SNR*, *azimuthres*, *timeres*, *finalweight*, *distance*, *CAZ*)

Pick information for seisan implimentation

Sfile_util.**readpicks**(*sfilename*)

Function to read pick informaiton from the s-file

Returns Sfile_tile.PICK

Sfile_util.**readwavename**(*sfilename*)

Convenience function to extract the waveform filename from the s-file, returns a list of waveform names found in the s-file as multiples can be present.

Sfile_util.**blanksfile**(*wavefile*, *evtype*, *userID*, *outdir*, *overwrite*)

Module to generate an empty s-file with a populated header for a given waveform.

Arguments are the path of a wavefile (multiplexed miniseed file required) # Event type (L,R,D) and user ID (four characters as used in seisan)

Example s-file format: # 2014 719 617 50.2 R 1 # ACTION:ARG 14-11-11 10:53 OP:CALU STATUS: ID:20140719061750 I # 2014/07/2014-07-19-0617-50.SAMBA_030_00 6 # STAT SP IPHASW D HRMM SECON CODA AMPLIT PERI AZIMU VELO AIN AR TRES W DIS CAZ7

Sfile_util.**populateSfile**(*sfilename*, *picks*)

Module to populate a blank nordic format S-file with pick information, arguments required are the filename of the blank s-file and the picks where picks is a dictionary of picks including station, channel, impulsivity, phase, weight, polarity, time, coda, amplitude, peri, azimuth, velocity, SNR, azimuth residual, Time-residual, final weight, epicentral distance & azimuth from event to station.

This is a full pick line information from the seisan manual, P. 341

3.2 py_picker

Python automated seismic picker implimentations for the rt2detection package.

py_picker.**picker_plot**(*stream*, *picks*, *types*, *show=True*, *savefile=''*)
Plotting fucntion for the picker routine - this will pot the waveforms and the picks to allow for pick veriication.

py_picker.**seism_picker**(*picktype*, *args*, *stream*)
A simple cover function for th eobspy picking modules

Parameters

- **picktype** (*String*) – Eother Baer or AR for Auto-regressive
- **args** (*Class*) – List of appropriate arguments for the chosen pick type - defined in picker_par.py

Returns p_pick, s_pick

3.3 STALTA

CHAPTER

FOUR

PAR

User-editable codes to input parameters for *core* files. Scripts are coded in the files with a full description of the parameters

4.1 overallpars

A python style declaration of variables used in rt2detection.py

class `overallpars.`**`STATION`**(*name*, *netcode*, *loccode*, *das*, *ch1*, *ch2*, *ch3*)
Station information for seismic station

`stacount` = **0**

4.2 picker_par

Python imput arguments for the python auto-pickers

class `picker_par.`**`AR_ARGS`**(*f1*, *f2*, *lta_p*, *sta_p*, *lta_s*, *sta_s*, *m_p*, *m_s*, *l_p*, *l_s*, *s_pick*)
Input arguments for the AR picker

class `picker_par.`**`BAER_ARGS`**(*tdownmax*, *tupevent*, *thr1*, *thr2*, *preset_len*, *p_dur*)
Input arguments for the Baer picker

4.3 trigger_par

Declaration of variables to be used as defaults in the STALTA.py routine.

A

B

O

P

R

S

T