

NETDET, event detection and location in real time or offline.

Introduction

NETDET is a program for event detection and location of local events. It can process data in near realtime from archived data streams (few minutes delay) or on time intervals given by the user. The program CONDET can detect and write out events from a SEISAN continuous data base but not in real time. Program NETDET does a similar job, but in near real time. While CONDET can use both archives and SEISAN style continuous data bases, NETDET can only use the SDS (SeisComp) or BUD archives and it can only use one archive. The only trigger option is STA/LTA. The program is intended to run on a Linux system and the included scripts for operation are for Linux but it can also work on Windows, however it has not been tested for realtime operation. The program can write out S-files and waveform files for triggered events, optionally repick phases with AUTOPIC and calculate magnitudes with AUTOMAG. The automatically located events can be display automatically on a map.

Requirements:

Linux computer

SEISAN

Data in SDS or BUD archives

Linux clock is synchronized using e.g. NTP

Optimal: Google Earth

The Generic Mapping Tool (GMT)

How it works

Real time operation: The programs starts with getting the current GMT time from the operating computer and will then start reading data some minutes after real time as specified in the parameter file. Data availability in the archive is normally delayed a few minutes relative to real time, typically 2-5 minutes depending on where the data comes from. In addition, when an event is detected, a waveform file can be written so the longer that is, the longer delay is needed. If an ARC reference is used in the S-file, no waveform file is written out, so a shorter delay can be used.

One minute of data is read from all channels and used for detection. For each channel, triggers are found and stored in a ring buffer. The triggering is done using STA/LTA on filtered data. The trigger parameters are then minimum STA/LTA ratio, minimum duration of trigger and minimum time since previous trigger.

Optionally 3 different channel trigger sets can be used so the same data can be tested with e.g. 3 different filters and trigger ratios. All triggers for the different trigger sets are store in the same trigger ring buffer. For more details, see section on parameter file. In most operations only one trigger set would be used.

Once all channels have been processed for a one minute time window, the triggers for all trigger sets are checked for network detection (number of triggers in the array propagation window). If there are multiple triggers for the same station within the array propagation window, only the first one is used. The triggers do not have to be within the same minute just read in, since an event cannot be declared before the array propagation time has passed after the first trigger. If an event is declared, a corresponding S-file can be written out, if specified in parameter file. The S-file can be written in a specified data base. The S-file will by default have the ARC reference for the trigger time of the earliest trigger. However, a waveform file can also be written out (locally, in WAV or in WAV data base) and referenced in the S-file, in which case the ARC reference is deleted. The network code is given by parameter MERGE_WAVEFORM in SEISAN.DEF. The trigger times are written in the S-file and in this way one can see which channels are triggering and where. However, if the option to use AUTOPIC is used, no trigger times are written out since now the automatic picks from AUTOMAG are shown. Optionally the event can be located (using the program HYP in reject mode), and optionally the magnitudes can be calculated with AUTOMAG program (Ml and Mw). Optionally, the phases can be repicked with AUTOPIC which often will give more phases and will in addition pick S-phases, but then the trigger times are not given. Finally there is an option for sending out an email in case an event triggers with either a minimum number of stations, a location and/or a given minimum magnitude.

Real time operation can work in the background (cron job) or be started from the prompt. In that case the window cannot be closed. Several instances of NETDET can be started as cron jobs, each with a different parameter file.

After the network detection, the next minute is read and the process continues.

Offline operation: In this case the user must give start time and end time and the program goes through the data base in given interval in a similar way as CONDET. There is then no option for sending email but S-files will always be output in one file, netdet.out, in addition to optional output in data base of S-files and WAV-files. The location and magnitude is only calculated for events going to the data base. The offline operation is useful for testing trigger parameters or for processing data which is not available in real time like aftershock data from a field experiment.

Installation

NETDET and corresponding scripts are part of SEISAN. The only additional requirement is that the directory SEISAN_TOP/CRON is created. The data bases for writing S-files and WAV files

must be created manually. The script for starting NETDET has two versions, netdet_start.csh and netdet_start.bash and the relevant one must be edited to fit your environment. It can be given any name.

Data source

The data can come from a local SeisComp system or the data can be pulled from an available SeedLink server into a local archive using slarchive.

<https://ds.iris.edu/ds/nodes/dmc/software/downloads/slarchive/>

How to check trigger operation

When starting NETDET in real time mode from a terminal window, there will be a running output showing the readings each minute. If there is a crash or other malfunction, it will be shown. There is also a parameter in the parameter file that turns on debug details of the triggering process. The files are netdet_debug_trig.out and netdet_debug_sta_lta.out. The latter will only be written if a specific station has been defined in the parameter file. Debug output is turned off in realtime mode to avoid filling up the disk. So to test for the best trigger parameters, the program can be operated in offline mode, as described above, and it can then be verified which parameter best triggers on known events in the given data interval.

Examples of netdet_debug_trig.out

In this file, a line is written each time one minute of data is read for one channel. When reading starts for the next minute, the message 'Read next minute' is given. If there is a trigger, a message 'Trigger now' is given. The trigger is then checked for duration, and if ok, the message 'Trigger ok with ratio and duration' is given and the trigger is accepted and put into trigger ring buffer. The trigger time is given as well as which parameter set is used (e.g. filt=2). When the dettrigger takes place, a message is also given 'Dettrigger now.' which can be due to ratio or duration. This message might come in the following minute.

Example with 3 filter sets and no events declared

```
DOMB HHZ NS00 2022 01-01 12:59:40.0 100.0 10001
*****
HYA HHZ NS00 2022 01-01 12:59:40.0 100.0 10001
*****
KMY HHZ NS00 2022 01-01 12:59:40.0 100.0 10001
*****
SUE HHZ NS00 2022 01-01 12:59:40.0 100.0 10001
*****
ODD1 HHZ NS00 2022 01-01 12:59:40.0 100.0 10001
*****
```

```

FOO  HHZ NS00 2022 01-01 12:59:40.0 100.0 10001
*****
HOMB HHZ NS00 2022 01-01 12:59:40.0 100.0 10001
READ next minute *****
*****
ASK  HHZ NS00 2022 01-01 13:00:40.0 100.0 10001
Trigger now ***** 01 13:01: 9.3 filt=2
Trigger ok with ratio and duration *****
TRIGGER ACCEPTED *****
ASK  HHZ 2022 01-01 13:01: 9.3

Detrigger now due to ratio 01 13:01:29.4
Trigger now ***** 01 13:01:59.4 filt=2
Trigger now ***** 01 13:01: 7.1 filt=3
Trigger ok with ratio and duration *****
TRIGGER ACCEPTED *****
ASK  HHZ 2022 01-01 13:01: 7.1

```

When an even is detected, the message ‘NEW EVENT’ is printed out together with the number of stations triggering. After that follows the content of the entire trigger buffer at that time where the trigger time, station, trigger status and filter set used are shown. Trigger status 0 means that the trigger has not yet been associated with a trigger from another station, 1 means that it has been associated but maybe not used and 2 means it is a duplicate and not used. In the example below, 3 stations triggered occurred within the array propagation window starting at (hour:minute:second) 13:04:19.7, on 1 January 2022. The first station is HYA, the next is ASK with trigger set 3, the next 2 from ASK are duplicates and not used and the third one used is DOMB.

```

SUE  HHZ NS00 2022 01-01 13:05:40.0 100.0 10001
*****
ODD1 HHZ NS00 2022 01-01 13:05:40.0 100.0 10001
*****
FOO  HHZ NS00 2022 01-01 13:05:40.0 100.0 10001
*****
HOMB HHZ NS00 2022 01-01 13:05:40.0 100.0 10001

NEW EVENT
2022 01-01 13:04:19.7 ndet= 3
2022 01-01 13:01:57.0 ASK  trig=2 filt=3
2022 01-01 13:01:58.5 FOO  trig=1 filt=3
2022 01-01 13:01:59.1 FOO  trig=2 filt=2
2022 01-01 13:01:59.3 FOO  trig=2 filt=1
2022 01-01 13:01:59.4 ASK  trig=1 filt=2
2022 01-01 13:02:30.8 HYA  trig=1 filt=3
2022 01-01 13:03: 4.0 ASK  trig=1 filt=3
2022 01-01 13:03:23.1 FOO  trig=1 filt=3
2022 01-01 13:03:31.6 ODD1 trig=1 filt=1
2022 01-01 13:03:31.7 ODD1 trig=2 filt=2
2022 01-01 13:03:43.3 ASK  trig=2 filt=3
2022 01-01 13:03:45.3 ASK  trig=2 filt=2
2022 01-01 13:04:19.7 HYA  trig=1 filt=3
2022 01-01 13:04:22.7 ASK  trig=1 filt=3
2022 01-01 13:04:22.9 ASK  trig=2 filt=2

```

```

2022 01-01 13:04:31.7 ASK   trig=2 filt=1
2022 01-01 13:04:56.8 DOMB trig=1 filt=3
2022 01-01 13:06:46.4 ASK   trig=0 filt=3
2022 01-01 13:06:47.1 ASK   trig=0 filt=2
2022 01-01 13:06:48.8 DOMB trig=0 filt=3
READ next minute *****
*****

```

The detailed triggering can also be checked for one station and output is in file `netdet_debug_sta_lta.out`. For this to be written out, a station code must be given in the parameter file in addition to setting the debug flag. An example is shown below.

```

BLS5 HHZ 01 13:05:50.0 STA=      1.2 LTA=      1.2 R=      1.0 Trig=      F Filt= 1
BLS5 HHZ 01 13:05:51.0 STA=      1.3 LTA=      1.2 R=      1.0 Trig=      F Filt= 1
BLS5 HHZ 01 13:05:52.0 STA=      1.4 LTA=      1.2 R=      1.1 Trig=      F Filt= 1
BLS5 HHZ 01 13:05:53.0 STA=      1.5 LTA=      1.2 R=      1.2 Trig=      F Filt= 1
BLS5 HHZ 01 13:05:54.0 STA=      2.0 LTA=      1.2 R=      1.6 Trig=      F Filt= 1
BLS5 HHZ 01 13:05:55.0 STA=      2.1 LTA=      1.3 R=      1.7 Trig=      F Filt= 1
BLS5 HHZ 01 13:05:56.0 STA=      1.6 LTA=      1.2 R=      1.3 Trig=      F Filt= 1
BLS5 HHZ 01 13:05:57.0 STA=      1.5 LTA=      1.2 R=      1.2 Trig=      F Filt= 1
BLS5 HHZ 01 13:05:58.0 STA=      1.3 LTA=      1.2 R=      1.0 Trig=      F Filt= 1
BLS5 HHZ 01 13:05:59.0 STA=      1.1 LTA=      1.2 R=      0.9 Trig=      F Filt= 1
BLS5 HHZ 01 13:06: 0.0 STA=      0.9 LTA=      1.2 R=      0.7 Trig=      F Filt= 1
BLS5 HHZ 01 13:05: 0.0 STA=      0.5 LTA=      0.5 R=      0.9 Trig=      F Filt= 2
BLS5 HHZ 01 13:05: 1.0 STA=      0.4 LTA=      0.5 R=      0.8 Trig=      F Filt= 2
BLS5 HHZ 01 13:05: 2.0 STA=      0.5 LTA=      0.5 R=      0.9 Trig=      F Filt= 2
BLS5 HHZ 01 13:05: 3.0 STA=      0.5 LTA=      0.5 R=      0.9 Trig=      F Filt= 2
BLS5 HHZ 01 13:05: 4.0 STA=      0.5 LTA=      0.5 R=      0.9 Trig=      F Filt= 2
BLS5 HHZ 01 13:05: 5.0 STA=      0.4 LTA=      0.5 R=      0.8 Trig=      F Filt= 2
BLS5 HHZ 01 13:05: 6.0 STA=      0.5 LTA=      0.5 R=      0.9 Trig=      F Filt= 2
BLS5 HHZ 01 13:05: 7.0 STA=      0.5 LTA=      0.5 R=      0.9 Trig=      F Filt= 2
BLS5 HHZ 01 13:05: 8.0 STA=      0.5 LTA=      0.5 R=      0.9 Trig=      F Filt= 2

```

For this particular station, for each second STA, LTA, trigger ratio (R), trigger status and parameter set is shown.

Sending email

The conditions for sending an email are minimum number of stations and/or minimum magnitude but there is no requirement for a location being present. However if magnitude is present, it is assumed that there is a location. If minimum magnitude is set to -1000.0, then only the requirement of minimum number of stations is used. The command for sending email as well as the email addresses to send to is defined in `SEISAN.DEF`. The command is the same as the one used by `EEV` while the email addresses are specified separately with the keyword `ALERT_EMAIL_NETDET`.

Output of S-file

In offline mode, all trigger S-files are written to one CAT file `netdet.out`. This is also the case if individual S-files are written. However, in real time operation, nothing is written to `netdet.out` since that file could be accumulating for a long time and eventually become very big. So only S-

files are written to the data base which can be a local data base or a normal data base. The location and operator of readings will have the agency of the institution as given in netdet.par and the operator is net. In the S-file, the trigger times are labelled IP and can be used to make an initial location.

Running netdet

Offline mode: Just start NETDET with command netdet and you will be asked for a time interval. NETDET will then start with the default parameter file netdet.par. Alternatively you can start with another parameter file with command netdet -parfile xx.par where xx.par is the parameter file. The parameter file must end with '.par'. The parameter file must be in local directory or DAT.

Real time interactive: Start by typing 'netdet realtime' and the program starts. There will be output for every minute on the screen. Here also an alternative parameter file can be chosen.

Cron job in the background: A script e.g. netdet_start located in COM is used to start NETDET. Use either the csh or bash version to make this file. It checks if NETDET is running and if not, it starts NETDET. For a cron job, the parameter file must be in DAT.

The parameter file name must be given, even if the default name, in order to distinguish different NETDET cron jobs. Note that the netdet_start has been renamed to the name without extension.

The csh script starting NETDET is

```
#
#!/bin/csh
#
# directory where netdet is running, also set in netdet.par
#
cd /home/seismo/netdet
#
# start netdet if it is not running
# the file must be edited for your environment
#
#
# check if netdet is running
#
set check=`ps ax | grep -v grep | grep "netdet realtime -parfile netdet.par" |
wc -l`

if ($check > 0) then
/usr/bin/echo "netdet with parameter file netdet.par is running"
else
/usr/bin/echo "netdet is not running, start again"
```

```

#
#   define env variables
#
source /home/s2000/seismo/COM/SEISAN.csh
#
#   start netdet
#
/home/seismo/PRO/netdet realtime -parfile netdet.par >/dev/null
#
#   send mail that it has been started
#
#/usr/bin/echo "netdet has been started" | /usr/bin/mailx -s "netdet startet"
mail.adr@gmail.com
#
/usr/bin/date >> /home/seismo/CRON/netdet.log
/usr/bin/echo "netdet started:" >> /home/seismo/CRON/netdet.log
/usr/bin/echo "netdet.par" >>/home/seismo/CRON/netdet.log
/usr/bin/echo " "
endif

```

In this script, a mail is sent to a user to inform that NETDET has started. Both the mail command and the mail address are hardwired and must be edited for your own system. If NETDET crashes, the user will then be informed within 5 min (example above) when NETDET starts again.

If you want to run 2 jobs at the same time, make another netdet_start with a different name and hardwire the new name of the parameter file like e.g. n.par. An example crontab file is then

```

#
*5/ * * * * /home/seismo/COM/netdet_start2
*5/ * * * * /home/seismo/COM/netdet_start1

```

The netdet.log file only contains the start time of the corn jobs for all jobs running like

```

Fri Jan 21 18:55:46 CET 2022
netdet started:
netdet.par
Sat Jan 22 08:45:42 CET 2022
netdet started:
ne.par
Mon Jan 24 10:17:09 CET 2022
netdet started:
netdet.par

```

Example of the bash file for starting netdet

```

#!/bin/bash
#
# directory where netdet is running, also set in netdet.par
#
cd /home/seismo/netdet
#
# start netdet if it is not running
#
#
# check if netdet is running
#
if ps ax | grep -v grep | grep "netdet realtime -parfile netdet.par"
then
/usr/bin/echo "netdet is running"
else
/usr/bin/echo "netdet is not running, start again"

#
# define env variables
#
source /home/seismo/COM/SEISAN.bash
#
# start netdet
#
/home/seismo/PRO/netdet realtime -parfile netdet.par >/dev/null
#
# send mail that it has been started
#
/usr/bin/echo "netdet has been started" | /usr/bin/mailx -s "netdet startet"
mail.adr@gmail.com
#
/usr/bin/echo "netdet started:" >> /home/seismo/CRON/netdet.log
/usr/bin/echo "netdet.par" >>/home/seismo/CRON/netdet.log
/usr/bin/date >> /home/seismo/CRON/netdet.log
fi
#

```

Monitoring realtime netdet

A script will start a small monitoring window (requires X-windows). Type `netdet_monitor` and you get:



```

NETDET MONITOR (rick.geo.uib.no)
Parameter file used: netdet.par
Processing now starting with: 2022 02-08 09:10:01
Netdet has run 0 days 18 hours 42 minutes
Number of defined and active channels: 10 10
Time of last detection : 2022 02-07 23:09:50 nstat= 8
Time of last location : 2022 02-07 23:08:44
61.672 11.958 12.0F BER 5 .90 2.9LBER 2.5WBER
Current GMT time : 2022 02-08 09:20:46

```

For netdet with parameter file n.par write netdet n

The window will stay up as long as the screen is on. The lines show the parameter file used, the time of the current data being processed, how long NETDET has been running, the number of channels defined for triggering, the number of channels actually received, the time of the last detection with number of stations triggering, the time of the last location, the last location and magnitude and the current GMT time. The number of active channels could be less than the number of defined channels if a station is out or the time delay is larger than given in parameter file. If NETDET stops running, the message NETDET IS NOT RUNNING will appear on the top. When a new detection occurs, a beep will sound. By default, the window corresponding to netdet.par is shown. If another process using parameter file n.par is used, type netdet_monitor n to get up a window for that process. It is also possible to get up the same data in a non graphics window. Then use command netdet_display for netdet.par or netdet_display n for n.par. The same data is also visible with a browser The html file netdet used for displaying the status in a browser is described in the section below named 'NETDET in a browser' .

If data is written out to a data base, and EEV is active, one can check for new events in the data base with EEV command up.

In case we see that the process starts frequently, then something is wrong. To find the problem, either run NETDET offline for the same time period or use interactive NETDET.

NETDET cronjobs running can be found with command netdet_check, which also can be used to stop a job.

```
*****netdet jobs running*****
```

```
1 netdet realtime -parfile netdet.par
2 netdet realtime -parfile n.par
```

Delete job, enter which one, enter to stop

Changing the parameters

If the parameters have to be changed while in real time operation, first change the parameter file and the stop NETDET with `netdet_check`. NETDET will then start up again at the time given in the crontab file.

Parameter file

The parameter file can be located in local directory or in DAT. The file name must end with `.par` and the default name is `netdet.par`. If the parameter file is in the local directory (where NETDET is started), that file will be used instead of file in DAT. The parameters will be written out on the screen when NETDET is started so that is a way of checking that they have been entered correctly.

The channels of the network must be defined in SEISAN.DEF and it is important that both location and network codes are correct and identical in SEISAN.DEF and `netdet.par`.

Channels to be written out can be given separately (CHAN OUT parameter). If none are given, all channels defined in SEISAN.DEF are written out.

The channel detection can be made with up to 3 different sets of trigger parameters, namely filter, STA, LTA, minimum trigger duration, minimum trigger interval, trigger ratio and dettrigger ratio. The number of trigger sets are defined by how many filter sets are given since the most common parameter to set differently is the filter. So if e.g. 2 sets will be used with only the filter as a difference, only the two filters are given and other parameters will be given identical to the parameters for the first filter. In general, one trigger set is used.

The parameter file is intended to be self-explanatory, an example is shown below.

```
#
# netdet.par: netdet input parameter file, comments can be added to this file,
#           but the comments must not match a KEYWORD
#
#
#
#
KEYWORD.....Comments.....Par 1.....Par 2
#
#   the network agency and operator, 3 chars
#
#.....1.....2.....
AGENCY                TES        jh
#.....xxxxxxxxxxxxxxxxxxxxxx
#
#   directory for netdet temporary files,
#   must be given also in netdet_start
```

```

# script, 40 chars
# if not given, WOR will be used
#
#.....1.....
NETDET DIR /home/seismo/netdet
#.....xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
#
# Debug and station to debug. Set
# debug to 1.0 if debugging. Station
# can be blank if no STA/LTA output
# for that station. Debug will be
# turned off in real time mode.
#
#.....1.....2.....
DEBUG 0.0 FOO
#.....xxxxxxxxxxxxxxxxxxxxxxxxxxxxx

#
# Channels used for triggering
#
# station, component, network, location
#
#.....1.....2.....
CHANNEL TRIG ASK HHZNS00
CHANNEL TRIG BER HHZNS00
CHANNEL TRIG BLS5 HHZNS00
CHANNEL TRIG DOMB HHZNS00
CHANNEL TRIG HYA HHZNS00
CHANNEL TRIG KMY HHZNS00
CHANNEL TRIG SUE HHZNS00
CHANNEL TRIG ODD1 HHZNS00
CHANNEL TRIG FOO HHZNS00
CHANNEL TRIG HOMB HHZNS00
#
# Channels used for output, if none, all
# in achive will be used
#
# station, component, network, location
#
#.....1.....2.....
#CHANNEL OUT ASK HHZNS00
#CHANNEL OUT BER HHZNS00
#CHANNEL OUT BLS5 HHZNS00
#.....xxxxxxxxxxxxxxxxxxxxxxxxxxxxx

#
#####
# delay to start data collection
# relative to GMT, usually minimum
# 300 s
#####
#
#.....1.....

```

```

REAL TIME DELAY                600.0
#.....XXXXXXXXXX
#
#####
# channel trigger parameters
#####
#
# length of short term average
#
#.....1.....2.....3.....
STA LENGTH          in seconds      03.
#.....XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
#
# length of long term average
#
#.....1.....2.....3.....
LTA LENGTH          in seconds      300.
#.....XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
#
# minimum trigger duration
#
#.....1.....2.....3.....
MIN TRIG DURATION  in seconds        5.
#.....XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
#
# time to next trigger must be least
# this number
#
#.....1.....
MIN TRIG INTERVAL          30.
#.....XXXXXXXXXX
#
# frequency limits of filter, 0 for
# no filter
#
#.....1.....2.....3.....
FILTER LOW                2.0
FILTER HIGH                7.0
#.....XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
#
# trigger and dettrigger ratios
#
#.....1.....
TRIGGER RATIO              3.
DETRIGGER RATIO            2.
#.....XXXXXXXXXX
#
# keep LTA during trigger flag up,
# currently not implemented.
# 1.0 keep LTA, blank or 0.0 do not use
#
#.....1.....
FREEZE LTA                 0.
#.....XXXXXXXXXX
#

```

```

#####
# network detection
#####
#
# minimum number of detections in array
# propagation window
#
#.....1.....
NET MIN DET                4.
#.....xxxxxxxxx
#
# array propagation window
#
#.....1.....
NET WINDOW SEC            50.
#.....xxxxxxxxx
#
#####
# waveform data extraction
#####
#
# extract waveform blank or 0.0: no, 1: yes
# if blank or 0.0, an ARC reference is made,
# the ARC reference is removed if 1.0.
# WAV files can be made even if S-files
# are not made. waveform files are made
# in current directory. can be moved
# or copied elsewhere, see COPY WAV.
#
#.....1.....
WAV EXTRACT                1.0
#.....xxxxxxxxx
#
# pre event time when extracting or
# setting ARC reference
#
#.....1.....
PRE EVENT TIME            60.
#.....xxxxxxxxx
#
# duration if extraction, also duration
# for ARC reference
#
#.....1.....
EXTRACT DURATION          300.
#.....xxxxxxxxx
#
# if a wav file extract, where to copy
# or move the file as given by COPY WAV
# blank: working directory
# m: move file, c: copy file
#
#.....1.....
COPY WAV                    m
#.....xxxxxxxxx

```

```

#
# where to write WAV file if move or copy
# blank: WAV directory
# 1-5 letter: WAV data base directory
#
#.....1.....
WAV BASE TEST
#.....XXXXXXXXXX
#
#####
# making S-files
#####
#
# parameter needed if s-files are made
# in addition to a cat file with all
# events. the cat file is not made in
# real time mode so in that case
# the S-file flag must be set to
# get S-files.
# 0.0 or blank: no S-files
# 1.0 : make S-files
#
#.....1.....
MAKE SFILE 1.0
#.....XXXXXXXXXX
#
# base for S-files.
# blank : def base
# 1-5 letter: data base
# ,, : local data base
#
#.....1.....
BASE SFILE TEST
#.....XXXXXXXXXX
#
#####
# Real time location and email
#####
#
# pick phases with AUTOPICT (1.0), if
# not netdet triggers are used.
#
#.....1.....
AUTOPICT 1.0
#.....XXXXXXXXXX
#
# min. number of stats. for location
#
#.....1.....
MIN STAT LOCATION 4.00
#.....XXXXXXXXXX
#
# fix location depth:
# blank or < 0: do not fix
# if ge 0.0 : fix to that depth

```

```

#
#.....1.....
FIX DEPTH                12.0
#.....xxxxxxx
#
# min. number of station for email
#
#.....1.....
MIN STAT EMAIL          99.0
#.....xxxxxxxxx
#
# excute a command after location,
# like making a map
#
#.....1.....
COMMAND
#.....xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
#
# min. magnitude for email in addition
# to minimum number of stations. If
# magnitude not used, set to -1000,
# then only number of stations is
# used. More mail parameters in
# SEISAN.DEF
#
#.....1.....
MIN MAG EMAIL           2.0
#.....xxxxxxx
#
#####
# the end
#####
#

```

SLARCHIVE example

In this part an example is given on how to run the SLARCHIVE program as a cron job on a linux server. SLARCHIVA can e.g. be downloaded from

<https://ds.iris.edu/ds/nodes/dmc/software/downloads/slarchive/>

One can as an alternative to the example below e.g. use a seiscamp server.

The example below shown how to stream data from IRIS and GEOFON. Here SLARCHIVE from IRIS is installed in the folder `/home/seismo/PRO`.

Setup the cron job to make sure SLARCHIVE is running and check every 5 minute, by adding this line to the users crontab:

```
*/5 * * * * /home/seismo/COM/autorun-slarchive.sh 2>&1
```

This is the script autorun-slarchive.sh used to run SLARCHIVE, it will restart SLARCHIVE if it has stopped. The script is here given using the SH shell, other shells will work too:

```
#!/usr/bin/sh
#
echo This script will dump mseed data from several seedlink servers to a SDS
archive, in real time.
echo The script makes sure that all subscripts are running at all times. See
crontab.
echo ----
date
echo $0
uname -a
echo ----

#
SLA="Slarchive IRIS"
SSH="slarchive -k 704"
SSH_PID=""ps -efa | grep "$SSH" | grep -v grep | tr -s ' ' | cut -d ' ' -f
3`"
if [ $SSH_PID ]; then
echo $SLA is running
else
date
echo $SLA is NOT running, starting $SLA ...
/home/seismo/PRO/slarchive -k 704 -l /home/seismo/DAT/streamlist.iris -SDS
/home/seismo/archive rtserve.iris.washington.edu:18000
fi
#

SLA="Slarchive GEOFON"
SSH="slarchive -k 705"
SSH_PID=""ps -efa | grep "$SSH" | grep -v grep | tr -s ' ' | cut -d ' ' -f
3`"
if [ $SSH_PID ]; then
echo $SLA is running
else
date
echo $SLA is NOT running, starting $SLA ...
/home/seismo/PRO/slarchive -k 705 -l /home/seismo/DAT/streamlist.geofon -SDS
/home/seismo/archive 139.17.3.177:18000
fi

##
echo -----
echo REMEMBER TO remove old archive files, if needed
#/usr/bin/find /home/seismo/SDS/2022 -type f -mtime +26 -exec rm {} \;
#echo remove old archive folders
#/usr/bin/find /home/seismo/SDS/2022 -type d -empty -delete
```

```
#echo Done $0
echo -----

exit 0

#EOF
```

The two input files giving the network and station codes requested from the IRIS and GEOFON seedlink servers:

```
NU BILN
NU CARN
NU RIBN
TC DRK0
TC PTJ1
TC TCS1
TC TRT2
SV CEDA
SV CNCH
```

```
NU ACON
NU BLUN
NU BOAB
NU ESPN
```

Add other stations or seedlings servers to improve the monitoring in your area.

Fit the paths given above so that they fit your system.

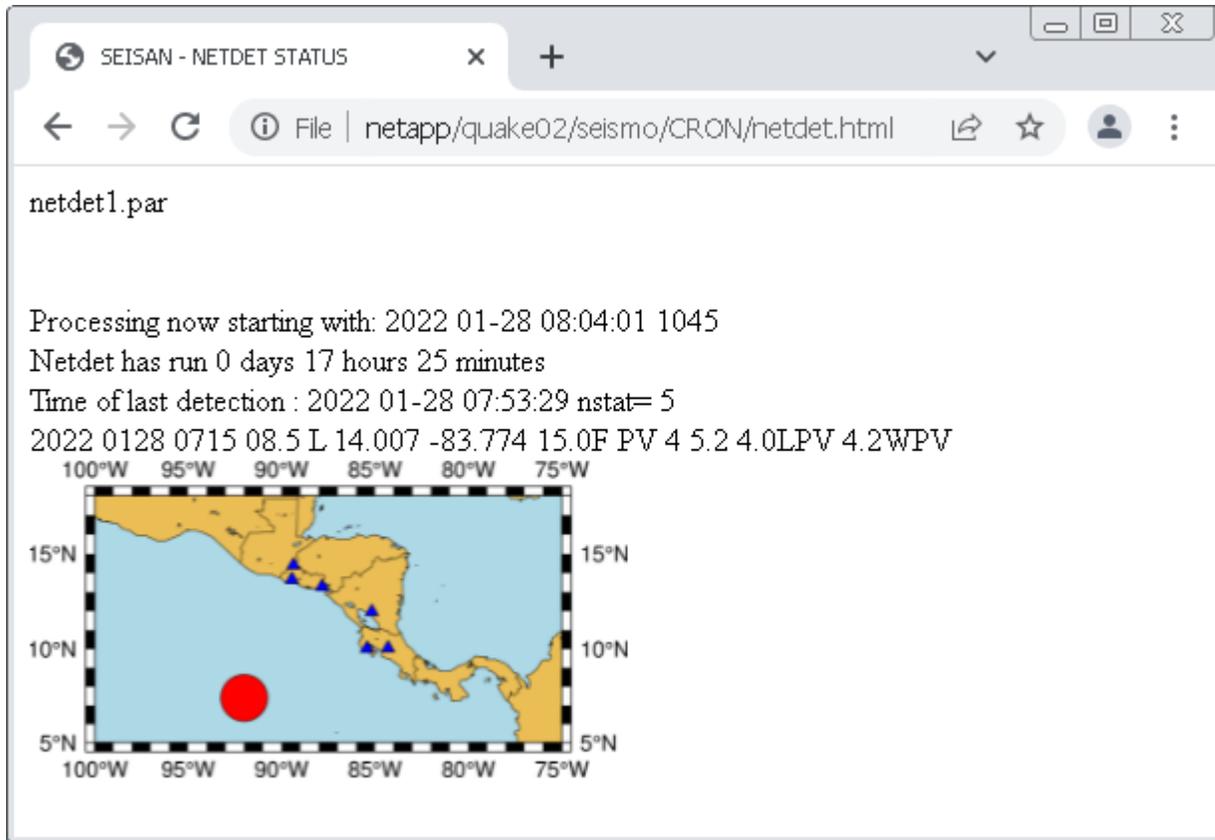
Once the cron job is started, data will be found in the /home/seismo/SDS folder.

Find more information on SLARCHIVE at:

<https://ds.iris.edu/ds/nodes/dmc/software/downloads/slarchive/2-2/manual/>

NETDET in a browser

NETDET will create a netdet.html file in the working folder. The html file is updated every minute with new information on the last detection.



An example on the NETDET html file is seen in a browser above. In addition the html file can display the epicentre on a map, but this requires that a command program or script is executed by NETDET (see COMMAND in the parameter file). In the example below, found in COM, is shown an example of a Linux script using SEISAN and GMT to create the map file.

```
COMMAND /home/seismo/COM/plot-netdet.gmt
#.....xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
```

The script plot-netdet.gmt is found in COM and is as follows:

```
#!/bin/ksh
#
# script to map epicenters from the SEISAN netdet program
# works with GMT - The Generic Mapping Tools, Version 6.1.1
# (c) 2022 peter voss
#

cd /home/seismo/netdet
echo $0 > plot-netdet.log
date >> plot-netdet.log

echo run seigmt
printf "hyp.out\n\n0.1\n" | seigmt
echo done seigmt
```

```
AP="-R-100.0/-75.0/5.0/18.0 -JM9c"

gmt begin netdet_map png
  gmt coast -Wthinest -Dl -N1 $AP -B -Slightblue -Glightbrown --
FORMAT_GEO_MAP=dddF
  gmt plot gmtstxy.out -St0.3 -Wthinest -Gblue
  gmt plot gmtxy.out -Sc -Wthinest -Gred
gmt end

echo ----- >> plot-netdet.log

#EOF
```

Execute the code and inspect the file netdet_map.png with a image viewer. If the map does not show the area of your network, change the area in the AP line, where

```
AP="-Rminlon/maxlon/minlat/maxlat -JM9c"
```

Further details on the GMT commands are given in the GMT documentation, see <https://docs.generic-mapping-tools.org>