

The SeisComP AutoLoc(1) package

- is the preliminary SeisComP automatic location package to be replaced by AutoLoc2,
- is based on a mixture of FORTRAN code for picking, pick association, magnitude determination, C code (LocSAT) for final location and perl scripts for process interaction, supervision and post event processing.

Picking

read_seed

multi-purpose FORTRAN program (part of SeedStuff) acting on “active” SeisComP files in continuation mode or individual files only,

- using standard STA/LTA detection with station and target dependent recursive pre-filtering,
- pick processing on vertical trace only,
- one read_seed task per channel (station) => possible performance problems with very large numbers of channel (at GFZ: ~120),
- automatic read_seed restart in case of crashing (re-processing of current day) by **start_detec** cronjob (perl script).

get_amp

- detached post-detection sub process for amplitude determination (for mb and MI),
- applied without prior knowledge on the location on pre-defined time windows.

Pick association, Pre-Location, Location Iteration, Magnitude Determination

make_list

- main cronjob (perl script) started every n minutes (typically $2 \leq n < 10$ min),
- invoking all following processes in timely sequential order.

sort_list

- script for the creation of timely sorted picklist for the current day (or any previous day),
- derived from the individual read_seed log files.

autoloc

- automatic location management program (FORTRAN),
- re-processes the whole current day every time called by make_list, but can also operate in single event mode,
- pick selection within pre-defined time and spatial windows, preferably for a network of neighboring stations,
- single stations outside of the defined spatial window are neglected,
- test, if a plane wave can be fitted to the pick distribution:
- if test positive, plane wave slowness and azimuth is used for the determination of a preliminary start solution for iterative location process,
- if test negative, it is assumed that the location lies within the network => no start location,

autoloc (continuation)

- iterative spawning of **LocSAT** standard location program with decreasing number of picks until the pre-defined residual criteria are fulfilled, depth is fixed,
- magnitude determination for each station, either mb or ML, depending on epicentral distance,
- spawning of **proc_alert** script for post-processing of event.

Post Event Location Processing

proc_alert

perl script, which decides on the alert level and invokes the alert dissemination tools.

mail_alert, mail_big, mail_xxl

perl scripts for the dissemination of E-mail and SMS alert messages for the different alert levels.

make_alert_html

perl script for the generation of HTML code for the “Automatic Event Bulletin” web page.

Further add-ons (not part of standard distribution):

make_alert_text: generation of plain text event list,

make_alert_rss: RSS feed generation,

plot_all.sh: GMT based map generation.

The *new* SeisComP AutoLoc2 package

is the future SeisComP automatic location package to replace AutoLoc1, mainly because of

- too many slarchive/read_seed processes (one per station!), which produce too high system load
- limitations of the association algorithm, which was originally designed to be used with only european stations, but global station distribution requires more sophisticated association
- need for additional criteria (polarization, dominant frequency, ...)

Picking

detector.py

Simple STA/LTA detection program written in Python and C. Advantages:

- only one process for all stations
- determines additional parameters like various amplitudes, polarization characteristics, etc.
- easy to configure
- easy to modify and extend

Pick association and event location

autoloc2.py

- global grid search technique for association, LocSAT for location
- no restrictions w.r.t. station location
- immediate response as soon as a new detection is produced by detector.py
- less sensitive to wrong triggers -> possible to increase detector sensitivity
- only locates events and writes results to files; further tasks to be handled by other scripts



