Database processing – Antelope based
Version 2010.237

Objective: Generating a db, visualizing the components of your db and what to do with it.

SUMMARY

a) Generate database
   seed2db –v TA_DATALESS* TA_DB
   miniseed2db TA_DATA/* TA_DB
b) Run dbdetect to execute STA/LTA detectors on the waveforms
   *can use the default dbdetect.pf file
   dbdetect –v –arrivals -pf dbdetect.pf TA_DB
c) Run ttgrid to make a grid file
   *copy the ttgrid pf file to local directory and edit
   ttgrid -pf ttgrid.pf -time all TA_grid
d) Run dbgrassoc to produce origin and event tables
   copy the the dbgrassoc.pf pf file to local directory and edit
   dbgrassoc -pf pf/dbgrassoc.pf TA_DB TA_DB TA_grid
e) Run dbloc2 on the database to modify picks and calculate earthquake locations
   dbloc2 TA_DB
f) Calculate magnitude, associate events,etc
g) Convert to other formats,
h) Have fun with it! (but don't mess it up ;)

Conventions:
Command line : > foreach file (`ls TA_DATALESS/DATAL* `)
Man page or help for command: (italic and size font 10)
Explanatory text or additional comments – regular size 12
Main Step: Antelope database generation (bold letters)
Secondary processing

Parameter file - ttgrid.pf

1) Antelope database generation:

Creation of a database with metadata and data

a) metadata – populates tables
   - calibration,
   - instrument,
   - lastid
   - network
   - schanloc

- sensor
- site
- sitechan
- snetsta

```
> foreach file (`ls TA_DATALESS/DATAL* `)
> foreach? seed2db -v $file TA_DB
> foreach? end
```

seed2db will scan a complete SEED volume (including both control blockettes and waveform data) and make or merge them into a Datascope database using the CSS3.0 schema.

b) Data (waveforms) populates table

- wfdisc

```
> miniseed2db –v TA_DATA/* TA_DB
```

c) View your database and understand it (dbe)
d) View your wfdisc table and play with it
e) Assign calibration values – what for? How?

miniseed2db reads input seed or miniseed files and creates wfdisc records which reference those files in the output database db. If an argument is a directory, then all files below that directory are examined.

```
> dbfix_calib TA_DB
```

Assign calib values from calibration table to wfdisc

Let's try to make picks and calculate earthquake locations.

2) **Location of events**

a. Use dbdetect

"dbdetect runs STA/LTA algorithm on seismic data to determine the arrival times of seismic energy. This procedure results in marking the time where the amplitude of the trace rises significantly above the background noise level. This is necessary to determine where the earthquakes are located in time in the data.
First you will need to setup a parameter file (pf file) for use as input to dbdetect. This file is located in the directory /opt/antelope/current_version/data/pf, and the file name is dbdetect.pf. Copy this file into the working directory and edit it from the working directory. Begin by running dbdetect using the default parameters in the dbdetect.pf file on three days worth of data. For the example database, use the following command"

- cp parameter file on local directory pf

```
> cp $ANTELOPE/data/pf/dbdetect.pf pf
```

- set up

> PFPATH $ANTELOPE/data/pf:. # to read default and local pf

> dbdetect –arrivals –pf pf/dbdetect.pf dbin dbout

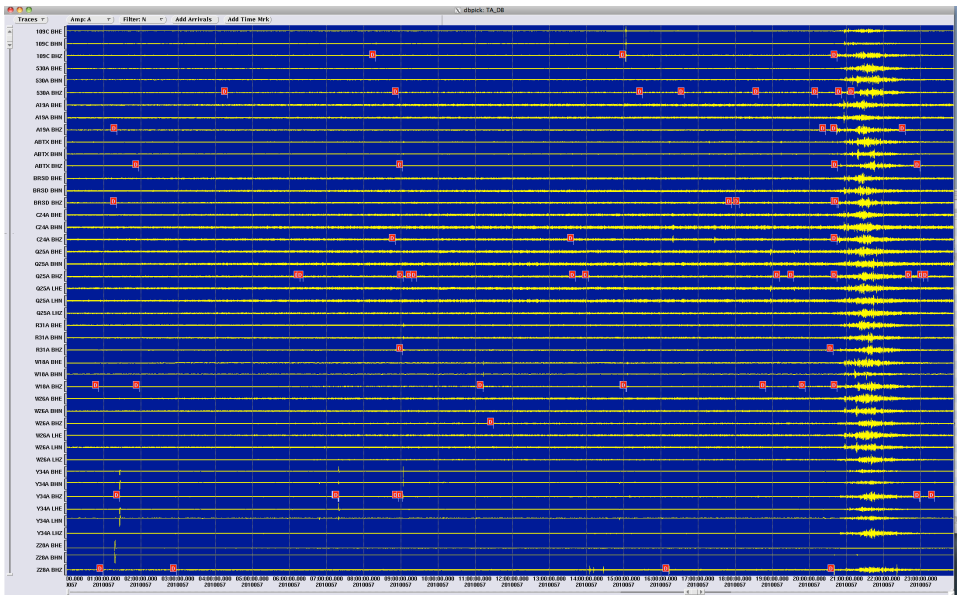
b. View your picks – dbpick

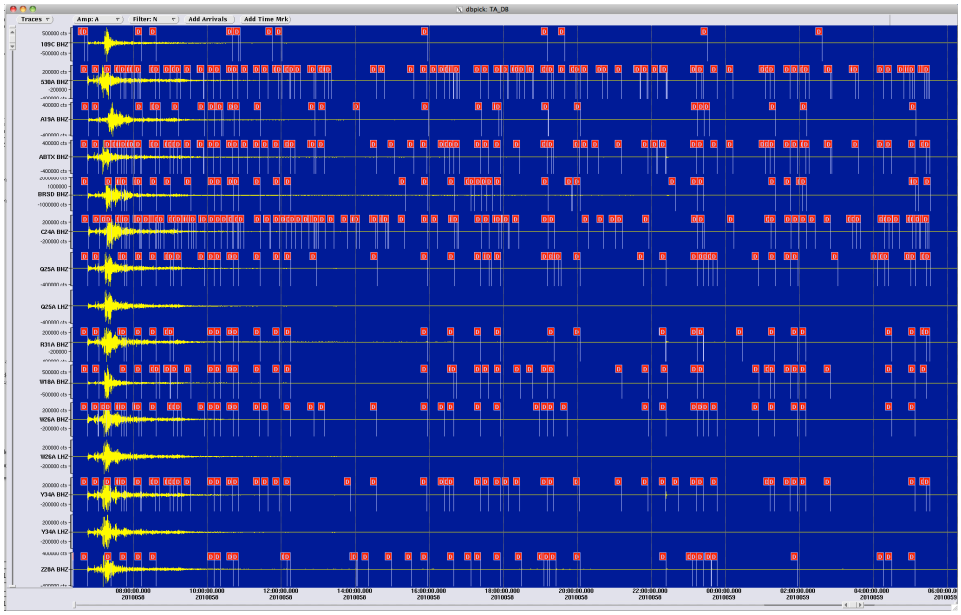
> dbpick TA_DB

say yes to display database

- fe
- sd on
- sc .*:.*Z # since in the pf file we only ask to detect Z
- ts 2010057
- tw 86000

You should see a marker labeled “D” next to each of the arrivals similar to the figure below.





Note that some arrivals have more than one detection flag. This is because the detection algorithm was run on more than one frequency band so it is possible to trigger a detection at more than one frequency band. This is OK as these extraneous arrivals will be handled by dbgrassoc. Also note that sometimes there are detection flags in undesired locations. This is also OK because they will be ignored later in dbgrassoc. The most important thing is to try and get detection flags at ALL the times of the known arrivals.

c. ttgrid - compute a travel time grid file

Once the detection times have been generated we can search for events/arrivals based on these detection times using the program dbgrassoc. First, however, one must generate a travel time grid for dbgrassoc to search over. To generate this grid use the antelope command ttgrid.

- cp parameter file on local directory pf

```
> cp $ANTELOPE/data/pf/ttgrid.pf  pf
```

- Customize ttgrid.pf
- Set path

```
>set up PFPATH $ANTELOPE/data/pf:.  # to read default and local pf files
```

Parameter file - ttgrid.pf

```
grids &Arr{
        local &Arr{
                mode            edp     # defines an equal-distance projection regular 3-D mesh
                #ndist          10000
                latr            40 # reference latitude          (origin of grid) – mid point
                lonr            -109.0  # reference longitude            (origin of grid) mid point
                nx              300     # Number of X-axis distance grid nodes
                ny              300     # Number of Y-axis distance grid nodes
                xmin            -10.0   # Minimum value of X-axis distance grid in degrees
                xmax            10.0    # Maximum value of X-axis distance grid in degrees
```

```
        ymin            10.0    # Minimum value of Y-axis distance grid in degrees
        ymax            -10.0   # Maximum value of Y-axis distance grid in degrees
        strike          90.0    # Angle from north clockwise in degrees to the X-axis
        compute_P       yes     # yes = Compute P travel times
        compute_S       yes     # yes = Compute S travel times
        method          tttaup  # method for computing travel times
        model           iasp91  # model for computing travel times
        depths &Tbl{
                0.0
                2.0
                4.0
                6.0
                8.0
                10.0
                12.0
                14.0
                16.0
                18.0
                20.0
                22.0
                24.0
                26.0
                28.0
                30.0

        }
    }

}
pf_revision_time 1214342897
```

if teleseismic events - alter the other tables

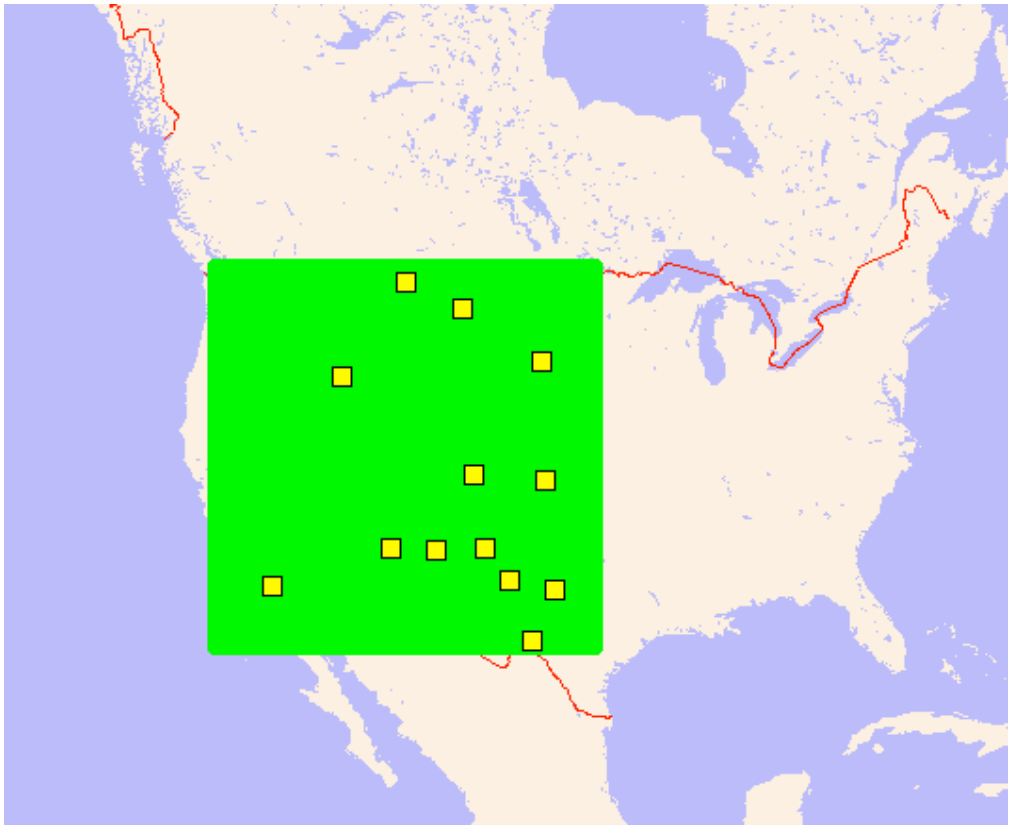
Now you can generate the grid by typing the following command:

> ttgrid -pf pf/ttgrid.pf -time all TA_DB > TA_grid

To display the grid that you just generated please type:

> displayttgrid TA_grid local

You can again check your db using dbpick
Customize your pf file as needed, considering your station's location, and if you want to calculate grids for regional and/or teleseism signals.



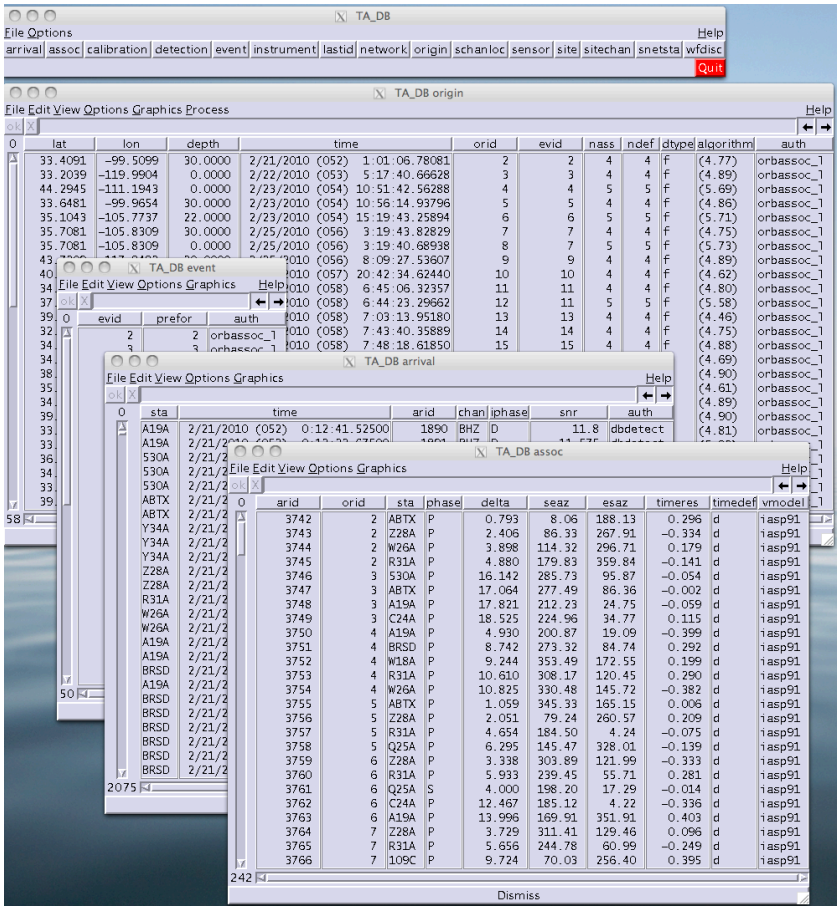
The green circles represent the locations of the travel-time grid nodes. The yellow squares represent the station locations. The grid should cover the entire region where earthquakes are likely to occur. However, there is a tradeoff. If the grid is coarse, dbgrassoc will run fast, but the preliminary eq locations will be less accurate. If the grid is fine, dbgrassoc will run slower, but the initial earthquake positions will improve.

d. dbgrassoc

spatial grid search based associator/locator dbgrassoc is the Datascope equivalent of orbassoc(1). The user should reference the orbassoc(1) man page for most program and configuration information. In fact, the dbgrassoc executable image file is the same exact file as the orbassoc executable image. Although dbgrassoc is 99% the same as orbassoc, there are a few differences which are explained here. The biggest difference is that there is no longer a dbtrigger program.

Once you have created a gridfile using ttgrid, run dbgrassoc to determine the time and preliminary locations of the earthquakes. Dbgrassoc reads the detection times from the detection table, then searches for a best fit solution to the earthquake's location problem on the travel time grid. The output of dbgrassoc is four tables:

1) origin - contains the hypocenter parameters (lat, lon, depth, origin time, etc.)
2) event - lists the preferred origin record for an event
3) arrival - lists the phase, time of the seismic arrivals
4) assoc – connects arrival records to origin records, also list travel time residuals, etc



- cp parameter file on local directory pf

> cp $ANTELOPE/data/pf/dbgrassoc pf

- Customize dbgrassoc.pf
- Set path

>set up PFPATH $ANTELOPE/data/pf:.

```
#    Parameter file for dbgrassoc

process_time_window         500.0           # Main detection processing time window
process_ncycle              20              # how often to do detection processing, in detections
process_tcycle              400.0           # how often to do detection processing, in delta time

grid_params &Arr{
        local &Arr{
                nsta_thresh         4       # Minimum allowable number of stations
                nxd                 11      # Number of east-west grid nodes for depth scans
                nyd                 11      # Number of north-south grid nodes for depth scans
                cluster_twin        1.5     # Clustering time window
                try_S               no      # yes = Try observations as both P and S
                                            # no  = Observations are P only
                associate_S         yes     # yes = Try to associate observations as both P and S
                reprocess_S         yes     # yes = Reprocess when new S-associations found
                # phase_sifter      l
                auth                orbassoc_l
                priority    5
                use_dwt             yes
                dwt_dist_near       2.0
```

```
                        dwt_wt_near         1.0
                        dwt_dist_far        6.0
                        dwt_wt_far          0.1
                }
}

#   parameters for "smart"association

assoc_method                tttaup      # method for computing predicted travel times
assoc_model                 iasp91      # velocity model for computing predicted travel times
assoc_phases                basic       # phase list for computing predicted travel times (see tt(3))
assoc_P_thresh              10.0        # P-residual threshold for associations
assoc_S_thresh              20.0        # S-residual threshold for associations
assoc_ignoreiphase  no      # should the arrival row iphase value be ignored?
assoc_firstphase    yes     # should only the earliest predicted phase be used?

assoc_screen_new    time
assoc_screen_old    (time-3600.0)::(time+3600.0)
                                        # these are screening database expressions that should
                                        #   match the existing (old) origins with the new origin for
                                        #   association processing
assoc_expression    $nass>=$nars
                                        # this is a database expression that should evaluate
                                        #   to true whenever an association is valid

author_priority &Arr{           # prefor priority as a function of assoc author
}

pf_revision_time 1214342895
```

> dbgrassoc -pf pf/dbgrassoc.pf TA_DB TA_DB TA_grid

Since this example is for a local array, ignore the regional and tele tables. Also, you may need to comment out the line that has the parameter phase_sifter as in the above example. Probably the most important parameter here is nsta_thresh. This parameter defines a cutoff for the number of arrivals at stations required to generate an earthquake location. If this number is small, dbgrassoc will find a large number of earthquake solutions (the solutions may or may not be real earthquakes). If this number is large, dbgrassoc will find a smaller number of solutions, but the results should be more robust.

Recommendations:

- *Multiple iterations*
- *Visaul check*
- *If there are more events in the origin table than found by hand, try increasing nsta_thresh in dbgrassoc.pf and/or increasing the parameter thresh in dbdetect.*
- *If there are more events found by hand than in the origin table, try decreasing nsta_thresh in dbgrassoc.pf and/or decreasing the parameter thresh in dbdetect.*

e. View your picks – dbpick

Command line

> dbpick TA_DB

say yes to display database

- sd on
- sc .*:.*Z # since in the pf file we only ask to detect Z
- ts 2010057
- tw 86000

A list of useful dbpick commands is given here.

MOUSE BUTTON	*ACTION*
Left-Click	*Time Scroll to the left*
Right-Click	*Time scroll to the right*
Middle-Drag	*Time scroll left or right*
Shift-Left-Click	*Time zoom in*
Shift-Right-Click	*Time zoom out*
Left-Click-on-Station-Label	*Select/Deselect station*
Left-Drag-across-Station-Label	*Select/Deselect station*
Left-Click-on-Arrival-Flag	*Adjust arrival time*
Shift-Left-Click-on-Arrival-Flag	*Adjust arrival time uncertainty*
Control-Shift-Left-Click-on-Arrival-Flag	*Adjust arrival amplitude-period*
Control-Left-Click-on-Arrival-Flag	*Show arrival measurements*
Middle-Click-on-Arrival-Flag	*Phase code menu*
Right-Click-on-Arrival-Flag	*Arrival menu*

COMMAND	*KEY ACTION*
fw	*F first waveform*
nw	*N next waveform*
pw	*P previous waveform*
fa	*f first arrival*
na	*n next arrival*
pa	*p previous arrival*
np phase	*next phase*
pp phase	*previous phase*
fe	*first event*
ne	*e next event*
pe	*previous event*
Fe	*find event*
dw	*delete blank traces*
swd	*show only all traces with detections*
swa	*show only all traces with arrivals*
swda	*show only all traces with detections or arrivals*
tfit	*t toggle time fit*
sfit	*s station fit*
	a Repaint arrivals
	r Repaint window
	R Redraw and repaint window
sc sta:chan	*display matching station/channels*
rec	*arrange stations by increasing distance.*
ts time	*start display at this time*
tw time	*change time window*
cw start number	*change channel zoom window*
ph phase	*change default phase*

gp {segment|zero|interp|none}
change telemetry gap processing
oa dbname *open arrival database*
oe dbname *open event database*
od dbname *open detection database*
se orid *select event from event database*
sp phase_list *select predicted arrival phases*
tc time_corr *specify overlay arrival time correction*
tse *time scroll to the current event.*
ae *associate event with arrivals in display.*
wa *write associations to input database.*
sa on/off *show/hide arrivals*
soa on/off *show/hide overlay arrivals*
sd on/off *show/hide detections*
sw on/off *show/hide waveforms*
sf on/off *show/hide first motions*
sv sta/off *show/hide velocity annotations*
cts on/off *amplitude units in couts or physical units.*
mg on/off *amplitude units in mg or nm/s**2 for acceleration.*
gr on/off *show/hide background st grids*
pal on/off *time align on P arrivals on/off*
filter index *set filter index*
sm {bilin|block} *set grid display mode*
fc hue light sat *set hue, light and sat of foreground color.*
ps plotfile *Make PostScript window dump to plotfile.*
pr *Make PostScript window dump to printer.*
exec command *execute a UNIX command*
quit *quit dbpick*
help *print this menu*

To make arrival picks:

1. Click on add arrivals button.
2. Move the cursor until it is aligned on the arrival time and left click.
3. Select the phase by middle clicking on the arrival flag.
4. Hold shift and left click an arrival flag to add error bars.

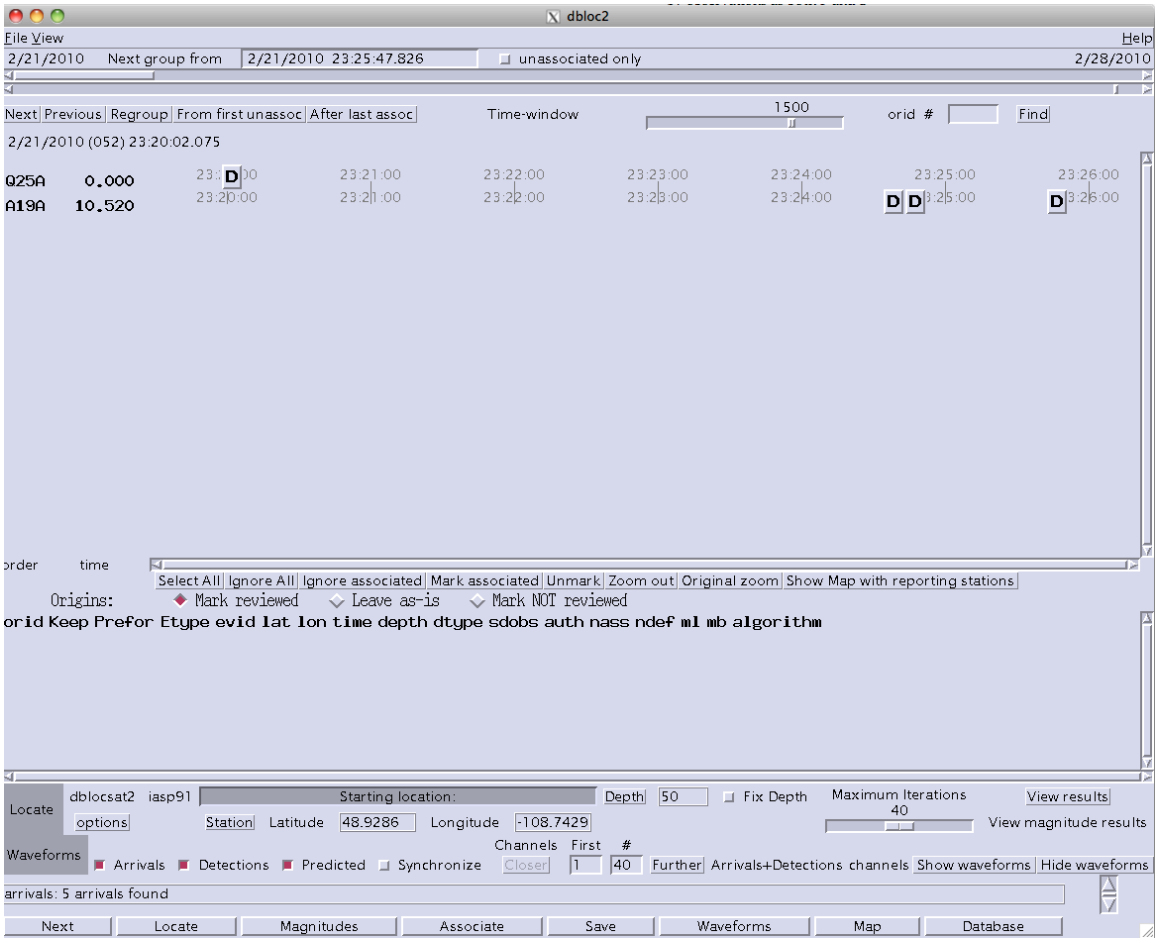
Once all of the picks have been made, use dbloc2 to locate the event.

f. dbloc2 - interactive hypocenter location

dbloc2 is a collection of several programs which run together under the control of a perl script, dbloc2. It facilitates the location of hypocenters from previously picked trace data, while allowing interaction with the location program and the ability to edit arrival picks. It can also attempt to associate groups of arrivals with other catalogs, for instance the PDE or REB catalogs

> dbloc2 TA_DB

This will bring up a window that looks something like this:



Event location

1. Select arrivals to be used in location by left clicking them in the dbloc2 window. Bold letters have been selected, gray letters have not. Initially start with all.
2. Choose location algorithm and velocity model (located just right of the word “Locate” in the lower left corner of the dbloc2 window). I recommend dbgenloc w/ the taup/iasp91 travel-time calculator/velocity model.
3. Choose location options below algorithm select. I recommend using either S-P time or a rectangular grid search for the initial location method.
4. Click the locate button.

dbloc2 Troubleshooting

1. Do not close the dbpick window when running from dbloc2. It will crash the program. Just minimize to get out of the way. The program will close when you file>quit dbloc2.
2. Do not file>quit w/out saving. This will leave a corrupt database.
3. If dbloc2 doesn’t work, try running with the –r option - dbloc2 –r ‘dbname’. This will clear the tmp/trial database.

4. If you get a message that says 'Save: arrivals have moved since location orid ##' you will have to reassociate. To do this click the button in the above figure that says 'keep' (middle left) until it says 'reassoc'. Then click the 'regroup' button near the top of the window. The regroup button is like the refresh button on your internet browser.
5. Sometimes dbloc2 will ask you to unregister buttons when trying to start. Copy this line and paste it on the command line, hit return, and try again.
6. If dbloc2 arrival flags turn red after locating, it means that your new calculated location is so far off from the previous location that the program thinks it is a new event. This may or may not be true. If it is true, use the new evid. If it is not, click on the blue button that says 226 in the above figure (middle left). Select 'set evid' and choose the previous evid.
7. Sometimes arrivals that are present in dbpick do not show up in the dbloc2 window. First try clicking the button that says predicted (middle left) and choose the option time. This will place the dbloc2 arrival flags at their absolute times. If this doesn't work, increase the time window scroll bar (near the top) and click the 'regroup' button.

3) Just for fun

a. Converting from other formats to mseed

http://www.iris.edu/software/downloads/conversion/
http://www.orfeus-eu.org/Software/conversion.html

Waveform format conversions _ FROM ORFEUS

Last update: December 12, 2008

Warning! Seismological format conversion is tricky. A format may not contain all relevant information for other formats. Consequently, the programs listed below may not solve your complete conversion problem. Please contact Torild van Eck on mistakes, suggestions, additions.

Conversion	*Program*	*Documentation*	*Programmer*	*Comments*
AH2ASCI	*ah2asc.c*	*README*	*IRIS/DMC*	-
AH2CSS	*ah2css.c*	*README*	*IRIS/DMC*	-
AH2SAC	*ah2sac.c* *PASSCAL software*	*README* *PASSCAL guide*	*IRIS/DMC* *IRIS/PASSCAL*	-
AH2SEED	*datascope*	*see program*	*IRIS/JSPC*	*CSS3.0; solaris*
AH2STD	*ah2std.c*	*README*	*IRIS/DMC*	-
AH2XDR	*ah2xdr.c*	*README*	*IRIS/DMC*	-
ASCI2AH	*asc2ah.c*	*README*	*IRIS/DMC*	-
CSS2AH	*css2ah.c*	*README*	*IRIS/DMC*	-

CSS2GSE	codeco_3.3c	Documentation	Urs Kradolfer	CSS2.8/3.0 | GSE1.0/2.x
CSS2MSEED	codeco_3.3c	Documentation	Urs Kradolfer	CSS2.8/3.0
CSS2SAC	codeco_3.3c css2sac-3.0.3 datascope	Documentation README see program	Urs Kradolfer IRIS/DMC IRIS/JSPC	- CSS3.0; solaris
CSS2SEED	datascope	see program	IRIS/JSPC	CSS3.0; solaris. Reported problems
DMUX2AH	SUDS	see program	Peter Ward, USGS	-
EQ2ASC	convseis	manual (postscript)	Oncescu & Rizescu	PCEQ (IASPEI) -> ASCII
ESSTF2SAC	sismoutil	howto.txt	Sebastien Judenherc	-
EQ2GSE	convseis	manual (postscript)	Oncescu & Rizescu	PCEQ (IASPEI) -> GSE1.0 (uncompressed)
EQ2SEGY	convseis	manual (postscript)	Oncescu & Rizescu	PCEQ (IASPEI) -> PC-SEGY
EQ2SEIS	convseis	manual (postscript)	Oncescu & Rizescu	PCEQ (IASPEI) -> SEIS89 format
EQ2SG	convseis	manual (postscript)	Oncescu & Rizescu	PCEQ (IASPEI) -> SeisGram binary
EQ2SUDS	convseis	manual (postscript)	Oncescu & Rizescu	PCEQ (IASPEI) -> PC-SUDS
GCF2AH	gcf for UNIX	User manual SAM/DM software.	ISTI	Guralp conversion software
GCF2SEGY	gcf for UNIX	User manual SAM/DM software.	ISTI	Guralp conversion software
GCF2SAC	gcf for UNIX	User manual SAM/DM software.	ISTI	Guralp conversion software
GDSN2SAC	evt2sac	README	ORFEUS	-
GDSN2AH	gdsn2ah.c	README	IRIS/DMC	-
GSE2CSS	codeco_3.3c	Documentation	Urs Kradolfer	GSE1.0/2.x | CSS2.8/3.0 UNIX HP/SUN, Linux
GSE2GSE	codeco_3.3c	Documentation	Urs Kradolfer	-
GSE2GSE	convseis	manual (postscript)	Oncescu & Rizescu	GSE1.0 <-> GSE2.0 DOS/Windows
GSE2MSEED	codeco_3.3c	Documentation	Urs Kradolfer	GSE1.0/2.x UNIX HP/SUN, Linux
GSE2MSEED	gse2mseed		IRIS, Chad Trabant	GSE 2.x/IMS 1.0, INT or CM6 Solaris, Linux, Mac OSX and Windows
GSE2SAC	codeco_3.3c gsesac	Documentation README	Urs Kradolfer/Hugues Dufumier	GSE1.0/2.x, SAC(A/B)

GSE2SEED	*gse2seed (version2.22)*	*README*	*ORFEUS, Sleeman*	*03/06/03 - version 2.22.*
GSE2SUDS	*convseis*	*manual (postscript)*	*Oncescu & Ritzescu*	*GSE1.0 -> PC-SUDS (.DMX)*
MSEED2AH	*ms2ah*	*QUG UCB README*	*IRIS/PASSCAL*	*-*
MSEED2CSS	*codeco_3.3c*	*Documentation*	*Urs Kradolfer*	*CSS2.8/3.0*
MSEED2GSE	*codeco_3.3c*	*Documentation*	*Urs Kradolfer*	*GSE1.0/2.x*
MSEED2PITSA	*HAM*	*HAM*	*Ernesto Del Prete*	*Handling miniSEED mseed to pitsa ASCII*
MSEED2SAC	*codeco_3.3c ms2sac*	*Documentation QUG UCB README*	*Urs Kradolfer Quanterra Users Group*	*-*
MSEED2SAC	*HAM*	*HAM*	*Ernesto Del Prete*	*Handling miniSEED*
MSEED2SEGY	*PASSCAL software*	*PASSCAL guide*	*IRIS/PASSCAL*	*-*
PDAS2AH	*p2ah*	*README*	*Guenther Asch*	*-*
PDAS2MSEED	*pdas2mseed*	*README*	*Paul Denton*	*-*
PDAS2SEGY	*p2segy*	*README*	*Guenther Asch*	*-*
PING2AH	*ping2ah*	*README*	*IRIS/DMC*	*-*
REF2SEGY	*PASSCAL software*	*PASSCAL guide*	*IRIS/PASSCAL*	*-*
REF2MSEED	*PASSCAL software*	*PASSCAL guide*	*IRIS/PASSCAL*	*-*
SAC2AH	*sac2ah.c*	*README*	*IRIS/DMC*	*-*
SAC2CSS	*codeco_3.3c*	*Documentation*	*Urs Kradolfer*	*CSS2.8/3.0*
SAC2ESSTF	*sismoutil-0.9b*	*README*	*Sebastien Judenherc*	*-*
SAC2GSE	*codeco_3.3c gsesac*	*Documentation README*	*Urs Kradolfer/Hugues Dufumier*	*GSE1.0/2.x*
SAC2LOCAL	*sac2local.zip*	*-*	*Meijian An*	*byte swap, sac file to local (C)*
SAC2MSEED	*codeco_3.3c sac2ms*	*Documentation QUG UTC README*	*Urs Kradolfer Quanterra Users Group users_comments*	*SAC(A/B)*
SACSUN2PC	*sac.appl*	*README*	*Tom McSweeney*	*binary byte swapping*
SACPC2SUN	*sac.appl*	*README*	*Tom McSweeney*	*binary byte swapping*
SAC2SH	*sac2sh*	*-*	*Raphael Garcia*	*-*
SAM2SUDS	*Guralp GSLConv*	*Guralp Software pages*	*Guralp*	*incl BDTS*
SAM2SAC	*Guralp GSLConv*	*Guralp Software pages*	*Guralp*	*incl BDTS*
SAM2SEGY	*Guralp GSLConv*	*Guralp Software pages*	*Guralp*	*incl BDTS*

SEED2AH	*rdseed*	-	*IRIS/DMC*	*UNIX/Linux*
SEED2CSS	*rdseed*	- *see program*	*IRIS/DMC* *IRIS/JSPC*	*limited.* *CSS3.0; UNIX/linux.*
SEED2GSE	*requested*	*README*	-	-
SEED2MSEED	*rdseed*	-	*IRIS/DMC*	*UNIX/Linux*
SEED2SAC	*rdseed*	-	*IRIS/DMC*	*UNIX/Linux*
SEGY2AH	*segy2ah.c* *PASSCAL software*	*README* *PASSCAL guide*	*IRIS/DMC* *IRIS/PASSCAL*	-
SEGY2ASC	*convseis*	*manual (postscript)*	*Oncescu & Rizescu*	*PC-SEGY -> ASCII*
SEGY2ASCI	*segy2asc.c* *PASSCAL software*	- *PASSCAL guide*	*IRIS/DMC* *IRIS/PASSCAL*	-
SEGY2ASCII	*segy2ascii*	*documentation*	*web-page information*	*includes plot fascilities.* *Mark Goldman*
SEGY2SAC	*segy2sac.c* *PASSCAL software*	- *PASSCAL guide*	*IRIS/DMC* *IRIS/PASSCAL*	-
SEGY2CSS	*PASSCAL software* *datascope*	*PASSCAL guide* *see program*	*IRIS/PASSCAL* *IRIS/JSPC*	- *CSS3.0; solaris*
SEGY2MSEED	*PASSCAL software* *segy2ms*	*PASSCAL guide* *QUG UTC README*	*IRIS/PASSCAL* *Quanterra Users Group*	-
SEGYPC2SUN	*PASSCAL software*	*PASSCAL guide*	*IRIS/PASSCAL*	*binary byte swapping*
SEGYSUN2PC	*PASSCAL software*	*PASSCAL guide*	*IRIS/PASSCAL*	*binary byte swapping*
SEISAN2MSEED	*IRISDMS software*	-	*IRIS-DMC Chad Trabant*	-
SG2EQ	*convseis*	*manual (postscript)*	*Oncescu & Rizescu*	*SeisGram binary format -> PCEQ (IASPEI)*
SGA2EQ	*convseis*	*manual (postscript)*	*Oncescu & Rizescu*	*SeisGram ASCII format -> PCEQ (IASPEI)*
SGA2ASC	*convseis*	*manual (postscript)*	*Oncescu & Rizescu*	*SeisGram ASCII format -> ASCII*
SG2SUDS	*convseis*	*manual (postscript)*	*Oncescu & Rizescu*	*SeisGram format -> PC-SUDS*
STD2AH	*std2ah.c*	*README*	*IRIS/DMC*	-
SUDS2GSE	*convseis*	*manual (postscript)*	*Oncescu & Rizescu*	*PC-SUDS (.DMX) -> GSE1.0*
VS2WGSN	*ViSeis 2 WGSN*	-	*Victor Huerfano*	*binary ViSeis to WGSN (DIMAS)*
XDR2AH	*xdr2ah.c*	*README*	*IRIS/DMC*	

b. Converting to other formats
   i. db2sac

ii. db2sd
iii. db2segy
iv. db2ah
v. db2sd
vi. db2sync

c. Moving around
   i. db2sql
   ii. db2pqc
   iii. db2neic
   iv. db2mac
   v. db2perl

d. Antelope established interfaces
   i. Command line
   ii. C
   iii. Fortran
   iv. Perl
   v. TCL/TL
   vi. PHP -contributed
   vii. Python-contributed
   viii. Matlab contributed
   ix. Java – contributed

By : Eliana Arias-Dotson at IRIS/PASSCAL (eliana@passcal.nmt.edu)