**Session 3: Antelope API Illustrative Examples**

*A shell script using command line filters to produce station lists of travel time residuals:*

#!/bin/csh

set dir=t.dir

set db=tonga

foreach sta (`cat ${db}\_sta.list`)

dbsubset ${db}.assoc "sta=~/$sta/" | dbselect - timeres > ${dir}/${sta}.timeres

end

*A more complicated script to build a bunch of files to be plotted with gmt:*

#!/bin/csh

set cohbase=coh

set sembase=sem

set outdir=./datafiles

foreach n (`cat gridid.list`)

dbjoin evstacks.stackstats site | \

dbsubset - gridid==$n | \

dbsubset - "fold>1" | \

dbselect - lon lat semblance fold > ${outdir}/${sembase}$n

dbjoin evstacks.stackstats site | \

dbsubset - gridid==$n | \

dbsubset - "fold>1" | \

dbselect - lon lat coherence fold > ${outdir}/${cohbase}$n

end

This produced a bunch of files in the “datafiles” directory that were used to drive this gmt script:

#!/bin/csh

set region="-R-130/-100/26/52"

set proj="-JM3.0i"

cd datafiles

foreach f (coh\*)

set outfile=../plots/$f.ps

psscale -Cgmtcoh.cpt -D1.5/0.0/2.0/0.25h -B0.2:Coherence: -K -Y2.0 > $outfile

psbasemap $region $proj -B5 -O -K -Y0.5 >> $outfile

pscoast $region $proj -Na -G200 -O -K >> $outfile

#awk '{print $1,$2,$3}' $f | \

awk '{print $1,$2,$3,0.01\*$4}' $f | \

psxy $region $proj -Cgmtcoh.cpt -St -O -K >> $outfile

pstext -Jx1.0 -R0/8.0/0/8.0 -O -K >> $outfile << EOT

4.0 4.5 24 0 0 CB $f

EOT

psscale -Cgmtcoh.cpt -D1.5/0.0/2.0/0.25h -B0.2:Coherence: -O -K -X4.5 -Y-0.5 >> $outfile

psbasemap $region $proj -B5 -Y0.5 -O -K >> $outfile

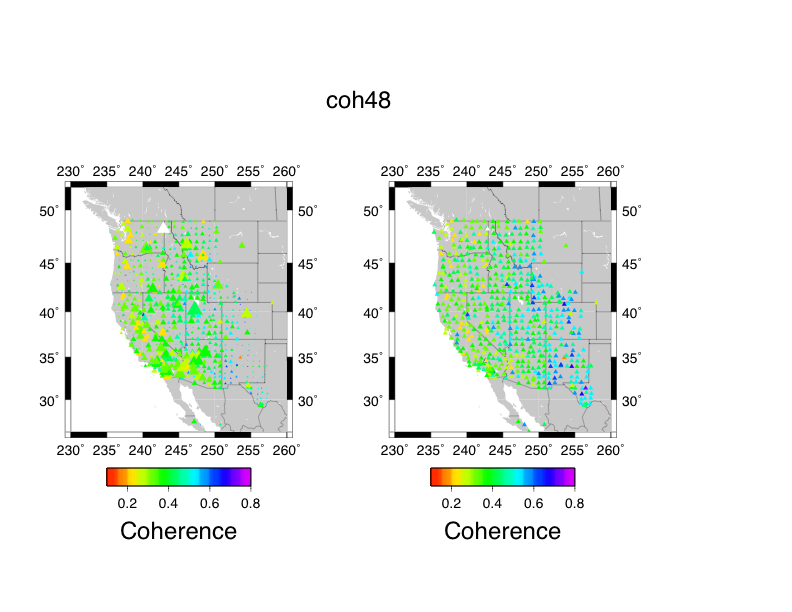
pscoast $region $proj -Na -G200 -O -K >> $outfile

awk '{print $1,$2,$3}' $f | \

psxy $region $proj -Cgmtcoh.cpt -St0.08i -O >> $outfile

end

This produced a bunch of figures like this one:



*Now a simple C program that uses Antelope API calls:*

#include <stdio.h>

#include <math.h>

#include "elog.h"

#include "coords.h"

#include "db.h"

#define RADIUS\_EARTH 6378.164

/\* For plane wave moveout computations a local cartesian coordinate

system is used wrt to a particular origin. This approximation is

reasonable until the distance of a station from the origin

becomes a significant fraction of the epicentral distance.

In pseudostation stacking we can minimize the impact of this approximation

by continually translating the origin to the pseudostation point and

computing distances to all stations wrt to that point. Since large

distance stations receive a low weight, time alignments at large

distances become unimportant. this function is used to computer

vector distances in this context.

Arguments:

nsta - number of stations to process = length of lat and lon vectors

(see below).

lat0, lon0 - origin to compute dnorth deast from

lat, lon - vectors of length nsta of station coordinates to be

converted.

dnorth, deast - vectors of length nsta to contain the results

(These are geographic dirctions +north and + east

respectively)

Author: G Pavlis

Written: June 2000

\*/

void geographic\_to\_dne(double lat0, double lon0,

double lat, double lon, double \*dnorth, double \*deast)

{

double azimuth;

double d;

dist(rad(lat0),rad(lon0),rad(lat),rad(lon),&d,&azimuth);

d \*= RADIUS\_EARTH;

\*deast = d\*sin(azimuth);

\*dnorth = d\*cos(azimuth);

}

main(int argc, char \*\*argv)

{

Dbptr db;

char \*dbname, \*refsta;

double lat0,lon0,lat,lon,dnorth,deast;

int nsta;

char sta[12];

if(argc!=3) die(0,"Usage: setdne db refsta\n");

dbname=argv[1];

refsta=argv[2];

if(dbopen(dbname,"r+",&db)==dbINVALID)

die(0,"Cannot open database %s\n",dbname);

db = dblookup(db,0,"site",0,0);

dbquery(db,dbRECORD\_COUNT,&nsta);

if(nsta<=0)die(0,"Empty site table for db %s\n",dbname);

db = dblookup(db,0,"site","sta",refsta);

if(dbgetv(db,0,"lat",&lat0,"lon",&lon0,0)==dbINVALID)

die(0,"dbgetv error reading reference station %s on row %d\n",

refsta,db.record);

db = dblookup(db,0,"site",0,0);

for(db.record=0;db.record<nsta;++db.record)

{

dbgetv(db,0,"sta",sta,"lat",&lat,"lon",&lon,0);

if(strcmp(sta,refsta))

geographic\_to\_dne(lat0,lon0,lat,lon,&dnorth,&deast);

else

{

dnorth=0.0;

deast = 0.0;

}

dbputv(db,0,"dnorth",dnorth,"deast",deast,"refsta",refsta,0);

}

return(0);

}

*I always found the Antelope API confusing so I wrote a simpler to use C++ front end. Here is an example as a section of the program dbresample in Antelope contrib:*

// This object defines mapping from external to internal namespace

// old form: AttributeMap am(pf,string("AttributeMap"));

AttributeMap am("css3.0");

// This defines the list of internal names actually extracted from db

MetadataList md\_to\_input=pfget\_mdlist(pf,

"input\_list");

// This is the list saved

MetadataList md\_to\_output=pfget\_mdlist(pf,

"output\_list");

// Input and output database handles

DatascopeHandle dbhi(dbname,false);

dbhi=DatascopeHandle(dbhi,pf,tag);

DatascopeHandle dbho(dboname,false);

// Builds the object that defines how decimation is

// and resampling is to be done.

ResamplingDefinitions rsampdef(pf);

dbhi.rewind();

for(int i=0;i<dbhi.number\_tuples();++i,++dbhi)

{

TimeSeries \*tin;

TimeSeries traceout;

string table("wfdisc");

tin = new TimeSeries(dynamic\_cast<DatabaseHandle&>(dbhi),

md\_to\_input,am);

traceout = ResampleTimeSeries(\*tin,rsampdef,dtout,trim);

// Simple method to change channel code

// only does right thing for SEED chan codes

chan=traceout.get\_string("chan");

chan[0]=chan\_code[0];

traceout.put("chan",chan);

// a crude way to alter files to preserve original structure

// append the string ".resampled"

dfile\_name = traceout.get\_string("dfile");

dfile\_name = dfile\_name + string(".resampled");

traceout.put("dfile",dfile\_name);

dbsave(traceout,dbho.db,table,md\_to\_output,am);

delete tin;

}