Finding Coincidences in the Database

Peter Shawhan
LIGO / Caltech

LIGO Scientific Collaboration Meeting
August 14, 2001

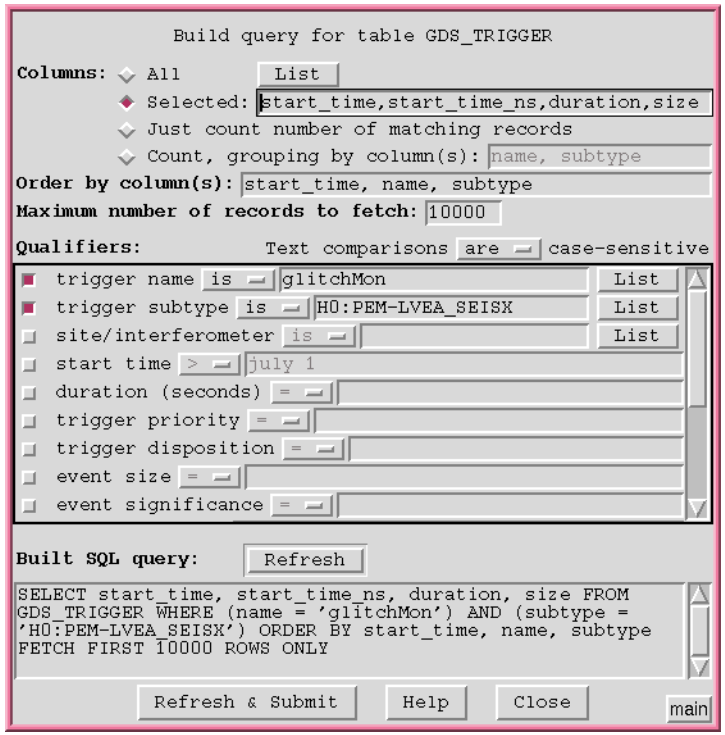
Basic approach:

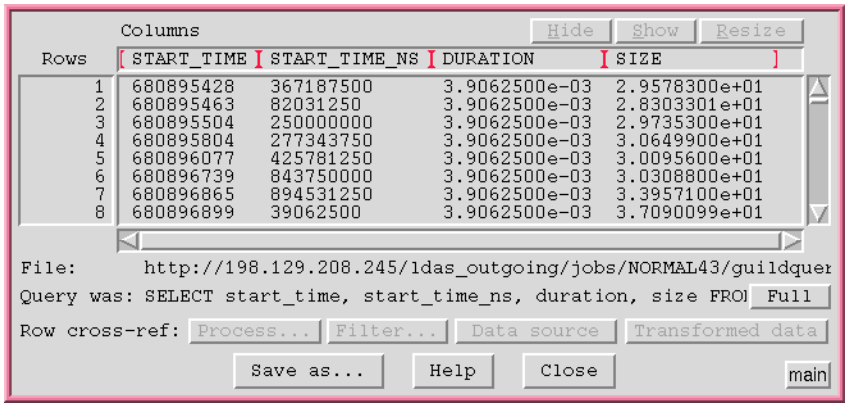
1. Use guild to select event lists, and save them to files in LIGO_LW format

2. Write a C program which reads the files, finds coincidences, and does what you want with them

LIGO-G010283-00-E

How to Select Events





After table of events appears, use the “Save as...” button to save the file in its original format (LIGO_LW)

Alternatively, use the getMeta command-line interface to submit an SQL query and put the output directly into a file

A Sample C Program to Find Coincidences

```
/*============================================================
find_coinc - A simple utility to find coincidences between
event lists
Written Aug 2001 by Peter Shawhan
Uses the "Metaio" parsing code by Philip Charlton
============================================================*/

#include <stdio.h>
#include <string.h>
#include "metaio.h"

#define MAXSTACK 100

/*============================================================*/
int main( int argc, char **argv )
{
    char file1[256], file2[256];
    double window;
    int status, status2, file2eof=0, i, ncoinc=0;
    int isec1, ins1, isec2, ins2;
    int sec1, ns1;
    double time1, time2, delta;

    struct MetaioParseEnvironment parseEnv1, parseEnv2;
    const MetaioParseEnv env1 = &parseEnv1;
    const MetaioParseEnv env2 = &parseEnv2;

    /* Stack of rows from the second file */
    int nstack = 0;
    int ssec[MAXSTACK], sns[MAXSTACK];
    int newlow;

    /*------ Beginning of code ------*/

    /*-- Check number of command-line arguments --*/
    if ( argc != 4 ) {
      printf( "Look for coincidences in two event lists\n" );
     printf( "Usage:  find_coinc <file1> <file2> <window>\n"
);
      return 3;
    }

    /*-- Parse command-line arguments --*/
    strncpy( file1, argv[1], sizeof(file1) );
    file1[sizeof(file1)-1] = 0;
    strncpy( file2, argv[2], sizeof(file2) );
    file2[sizeof(file2)-1] = 0;
    if ( sscanf( argv[3], "%lf", &window ) < 1 ) {
      printf( "Error parsing window size %s\n", argv[3] );
      return 1;
    }

    /*-- Open the files and read to beginning of Stream --*/
    status = MetaioOpen( env1, file1 );
    if ( status != 0 ) {
      printf( "Error opening file %s\n", file1 );
      MetaioClose( env1 );
      return 2;
    }

    status = MetaioOpen( env2, file2 );
    if ( status != 0 ) {
      printf( "Error opening file %s\n", file2 );
      MetaioClose( env2 );
      return 2;
    }

    /*-- Locate start_time and start_time_ns columns --*/
    isec1 = MetaioFindColumn( env1, "start_time" );
    ins1 = MetaioFindColumn( env1, "start_time_ns" );
    if ( isec1 < 0 || ins1 < 0 ) {
      printf( "File %s does not contain start_time and/or
start_time_ns\n",
              file1 );
      MetaioClose( env1 ); MetaioClose( env2 ); return 4;
    }

    isec2 = MetaioFindColumn( env2, "start_time" );
    ins2 = MetaioFindColumn( env2, "start_time_ns" );
    if ( isec2 < 0 || ins2 < 0 ) {
      printf( "File %s does not contain start_time and/or
start_time_ns\n",
              file2 );
      MetaioClose( env1 ); MetaioClose( env2 ); return 4;
    }

    /*---------------------------------------*/
    /*-- Loop over rows in the first file --*/
    while ( (status=MetaioGetRow(env1)) == 1 ) {

      /*-- Get the time of this event --*/
      sec1 = env1->ligo_lw.table.elt[isec1].data.int_4s;
      ns1 = env1->ligo_lw.table.elt[ins1].data.int_4s;

      /*-- Check for coincidences (within the window
interval) using events from second file --*/
      /*-- We check the stack, enlarging it as needed --*/
      newlow = 0;
      for ( i=0; i<nstack+1; i++ ) {

        /*-- If i==nstack, we need to enlarge the stack --*/
        if ( i == nstack ) {
          /*-- If we hit the end of file2, break --*/
          if ( file2eof ) { break; }

          /*-- Make sure we have more room in the stack --*/
          if ( nstack == MAXSTACK ) {
            printf( "Ran out of space in the event stack
(size=%d)\n",
                    MAXSTACK );
            MetaioClose( env1 ); MetaioClose( env2 );
            return 5;
          }

          /*-- Read a row --*/
          status2 = MetaioGetRow(env2);
          if ( status2 == -1 ) {
           printf( "Error while getting row from file %s\n",
file2 );
            MetaioClose( env1 ); MetaioClose( env2 );
            return 6;
          } else if ( status2 == 0 ) {
            /*-- Reached end of file --*/
            file2eof = 1;
            break;
          }

          /*-- Add this event time to the stack --*/
          ssec[nstack] =
              env2->ligo_lw.table.elt[isec2].data.int_4s;
          sns[nstack] =
              env2->ligo_lw.table.elt[ins2].data.int_4s;
          nstack++;
        }

        delta = (double) (ssec[i]-sec1)
            + 1.0e-9 * (double) (sns[i]-ns1);

        /*-- If stack row is earlier by more than the size of
             the window, get ready to delete it --*/
        if ( delta < -window ) {
          newlow++;
        } else if ( delta <= window ) {
          /*-- Found a coincidence! --*/
          ncoinc++;
          printf( "Coinc: time1=%d.%09d time2=%d.%09d
delta=%.9f\n",
                  sec1, ns1, ssec[i], sns[i], delta );
        } else {
          /*-- Stack row is too far in the future --*/
          break;
        }
      }
      /*-- If necessry, compress stack to get rid of
           far-past rows --*/
      if ( newlow > 0 ) {
        for ( i=newlow; i<nstack; i++ ) {
          ssec[i-newlow] = ssec[i];
          sns[i-newlow] = sns[i];
        }
        nstack -= newlow;
      }

    }

    /*-- Close files --*/
    MetaioClose(env1);
    MetaioClose(env2);

    printf( "Total coincidences found: %d\n", ncoinc );

    return 0;
}
```

A Sample C Program to Find Coincidences

Notes about the program:

- Uses the “Metaio” parsing code, which is included in LIGOtools
- Treats any pair of events within the window as a “coincidence”; thus, an event may participate in more than one coincidence

To compile on Solaris:

```
gcc find_coinc.c -I$(LIGOTOOLS)/include -L$(LIGOTOOLS)/lib -ldataflow \
-lsocket -lnsl -o find_coinc
```

Sample output:

```
pshawhan@sheratan> find_coinc seisx.xml seisz.xml 0.5
Coinc: time1=680896932.015625000 time2=680896932.019531250 delta=0.003906250
Coinc: time1=680897023.441406250 time2=680897023.421875000 delta=-0.019531250
Coinc: time1=680898473.765625000 time2=680898473.800781250 delta=0.035156250
Coinc: time1=680898629.824218750 time2=680898629.835937500 delta=0.011718750
Coinc: time1=680908541.023437500 time2=680908541.328125000 delta=0.304687500
Coinc: time1=680912773.871093750 time2=680912773.855468750 delta=-0.015625000
Coinc: time1=680924229.460937500 time2=680924229.171875000 delta=-0.289062500
Coinc: time1=680924267.058593750 time2=680924267.046875000 delta=-0.011718750
Coinc: time1=680961281.890625000 time2=680961281.910156250 delta=0.019531250
Coinc: time1=681065215.941406250 time2=681065215.941406250 delta=0.000000000
Coinc: time1=681076976.367187500 time2=681076976.523437500 delta=0.156250000
Coinc: time1=681089782.566406250 time2=681089782.570312500 delta=0.003906250
Coinc: time1=681123259.546875000 time2=681123259.175781250 delta=-0.371093750
Coinc: time1=681126999.468750000 time2=681126999.816406250 delta=0.347656250
Coinc: time1=681140723.082031250 time2=681140723.175781250 delta=0.093750000
Coinc: time1=681140776.160156250 time2=681140776.136718750 delta=-0.023437500
Coinc: time1=681140808.699218750 time2=681140808.855468750 delta=0.156250000
Coinc: time1=681157665.875000000 time2=681157665.796875000 delta=-0.078125000
Total coincidences found: 18
```