Maximal Clique Computation

Robert Bakarić
rbakaric@irb.hr
bakaric@evolbio.mpg.de
23.04.2015
CliMax-1.0

Abstract

This is a C++ implementation of maximal clique computation algorithm as described by Tomita, E. et al. (2006). Given an adjacency list the program computes all maximal cliques in a given graph.

Contents

1 Installation

The simplest way to compile this program is to:

1. Unpack the CliMax package (`climax-XXX.tar.gz`):

   ```
   tar -xvzf climax-XXX.tar.gz
   ```

2. Change the current directory to `climax-XXX`:

   ```
   cd climax-XXX/
   ```

3. Configure the program for your system (–bindir is optional):

   ```
   ./configure --bindir=/absolute/directory/path/climax-xxx/bin
   ```

4. Compile the program:

   ```
   make
   ```

5. Install the program:

   ```
   make install
   ```

Your binaries should be located in your local bin directory if `--bindir` option has been set. Otherwise installation needs to be carried out with root privileges in order to be installed into `/usr/local/bin` directory.

2 Input files

CliMax takes a regular tab delimited adjacency list of vertices structured in child → parent order. The example of such file can be found in `./climax-xxx/demo` and it should look like this:

`Graph`:

```
<child>  <parent>
  2        1
  4        1
  8        2
  5        1
  67       2
  6        3
  14       4
  15       6
  68       3
  3        2
  11       67
  2        11
  11       8
  67       8
  2        67
  8        67
  8        11
  11       2
  17       67
```

3 Program options

I order to see program options type:

```
./bin/CliMax -h
```

Expected output:

```
Usage: ./program [options]

********************************************************************************
                                      CliMax
                                        by
                                   Robert Bakaric

CONTACT:
 Code written and maintained by Robert Bakaric,
 email: rbakaric@irb.hr , bakaric@evolbio.mpg.de

ACKNOWLEDGEMENT:
     Tomita, E.,Tanaka, A. and  Takahashi, H. (2006) The w0rst-case time complexity
     for generating all maximal cliques and computational experiments. Theor. Comput.
     Sci. 363:28-42

LICENSE:
 The program is distributed under the GNU General Public License. You should have
 received a copy of the licence together  with this software. If not, see
 http://www.gnu.org/licenses/
********************************************************************************


Allowed options:
  -h [ --help ]                      produce help message
  -v [ --version ]                   print version information
  -c [ --min_clique_size ] arg (=2) Minimum clique size
  -i [ --graph-file ] arg            Adjecency list (tab seperated -> child
                                     <tab> parent)
```

4 Functions and classes

CliMax() : Constructor. It takes three parameters in order to build a graph. First is a vector of parent vertices and the second is vector of child vertices. The third parameter is the cutoff value restricting the minimal size of a maximal clique identified.

`Expand` : A private function with recursive calls used to identify maximal cliques as presented in Tomita, E.,Tanaka, A. and Takahashi, H. (2006). the function takes three set objects : clique members, subgraph vertices and candidate vertices.

`ComputeMaxCliques` : A public function. The equivalent of `CLIQUES` (Tomita et al. 2006). the function operates on a pre-loaded graph.

`GetMaxCliques` : Function returns a computed set of maximal cliques. A return value is a 2D vector of vertices.

It should be noted that this program depends on a Graph class described in `GraphMan-1.0 Bakaric, R. 2015`

5 Example

5.1 CliMax.cpp

A minimal example demonstrating the usage of CliMax demo program:

```
./bin/CliMax -i demo/Graph -c 2

MaxCliqSize (Vertices)

2           (1,2)
2           (1,4)
2           (2,3)
4           (2,8,11,67)
```

5.2 Clique.hpp

Adding the `Clique.hpp` header file to your program will allow you to include all functions described in section 4. A minimal example:

```
    #include<vector>

    #include<Graph.hpp>
    #include<Clique.hpp>


/* Constructor */

/*NOTE:  parent -> vector<int|long|double>, child -> vector<int|long|double> */
    CliMax<int|long|double> clique(parent, child, 3);



/* Compute Maximal Cliques */
      clique.ComputeMaxCliques();



/* Get Maximal cliques */
      vector<vector<int|long|double>> mc = clique.GetMaxCliques();
      /*  mc contains vector of vertices of a given maximal clique */
```

6 Acknowledgement

Tomita, E.,Tanaka, A. and Takahashi, H. (2006) The worst-case time complexity for generating all maximal cliques and computational experiments. Theor. Comput. Sci. 363:28-42

7 Future work

Additional work is required in order to increase implementation efficiency. Moreover, a Maximum Clique computation algorithm should be added to CliMax class.